A different approach to information provision and retrieval

Stephen Marsh
National Research Council Canada
Institute for Information Technology, Interactive Information Group
<steve@ai.iit.nrc.ca>

Information retrieval finds itself at an interesting juncture, where the amount of information that is available to people increases every day from its already bewildering limit. The problem is how to get the information we need in a timely and efficient fashion, without delivering useless or unwanted information. Current Web-Based IR systems do their best, but they will find it increasingly difficult to provide the services people want. In this paper we present a system which is based on the use of intelligent agents to search for information and also to distribute information. Information agents distribute the information their owners want others to see, whilst search agents interact with these information agents to find information their owners are looking for. Since we expect the agents to communicate with each other, the end result of the system is a community of agents that are able to deliver information to users when they want it, sharing their knowledge about who wants what, and adapting to the changing needs of the architecture's users. This paper presents the architecture and provide thoughts for its possible future development.

1. Introduction

Today's networks carry a large amount of information for a large number of people. Finding the information you want in such systems is no trivial matter, and can only get harder. The problem is that it is difficult to know where to look for the information, and when you've looked, it is hard to know if what information you have is timely, accurate, or 'what is out there.' Additionally, when something has been written, it's just as hard to make sure the people who would find it interesting or useful even get to see it, even if the author knows who they are. In short, networks of information,. such as the World Wide Web or corporate intranets pose interesting problems not only of timely retrieval, but also of accurate provision, of information.

There is an upside to this chaos, however. The Web, or indeed any network of machines and people, provides a low- or zero-cost addition to our individual

knowledge in that it embodies the knowledge of the community that uses it. If we can use this 'knowledge of others about others' in our search and distribution tasks, we can remove many of the problems associated with such a large body of information. As human beings, if we need to know something, we can ask people. If the people we ask don't know, they may know someone who does, or at least, someone else to ask (Hill et al, 1995). This network of community information is an invaluable tool in our search to make systems such as the Web more useful and usable.

This paper presents an overview of ACORN (Agent-based Community Oriented Retrieval Network), a system under development at the National Research Council which makes use of community-based knowledge, placing it in artificial agents which aim to distribute and search for information in networks. Each of these agents is capable of sharing its own community knowledge with others, forming an artificial community which facilitates the movement and routing of information. We discuss the main components of the architecture, present its present state and future development, and compare it with similar systems in existence today.

2. The ACORN architecture

ACORN is a prototype architecture for information provision and retrieval across networks. It makes use of the concept of 'document as agent,' in which individual pieces of information (at present, textual documents) are embodied as software agents. The agents themselves are capable of reasoning in a limited fashion about who may be interested in the document they embody, and are capable of moving through the ACORN network (which sits on top of the Internet) in order to distribute this information. The community of such agents produces useful behaviour via the interactions of agents with each other, in which they share knowledge about their own documents, others they have 'seen,' people they have visited and other agents they have interacted with. In this way, the agents are able to glean extra information from the community to augment what they already know about it, using this information to better guide their information distribution tasks (and search tasks).

Whilst the architecture is composed of many agents, the majority of them are present for infrastructure purposes. All of them add their own value to the system, but conceptually it is possible to think of the architecture in terms of the two mobile agents in it. These agents represent the documents (InfoAgents) and the queries (SearchAgents) that users of the system generate. Accordingly, this paper concentrates on giving an overview of the Info- and SearchAgents in the system, showing how they share information, how they embody it, and how they use their

mobility to facilitate information search and distribution. A lengthier, more detailed view of the system is available from the author (Marsh, 1997).

2.1 Smart Documents — Information Agents

In many ways, it is the InfoAgent, or Smart Document, that distinguishes our architecture from similar architectures. In most similar systems, it is the search for information that has been automated in some way via a process of 'matchmaking', see for example Foner and Crabtree (1996). In ACORN, this process is extended to the provision of information: producer push.

The InfoAgent is the embodiment of a piece of information in the ACORN system. It makes use of the community of agents and their knowledge to navigate around the system and present itself to those it knows or believes will be interested in the information it carries. Once a piece of information is created that an author wishes to distribute, a InfoAgent is created to store that information.

When first created, the InfoAgent is ignorant as to the contents of the document it represents. In order for it to behave usefully in the system, it needs to get what information it can about the document. There are several steps in this process, and all are guided by the author's user interface (itself an agent), the GateKeeper. The first step involves obtaining information from the author (via a simple dialog), the information being, for example, a title, the type of the information (an article, a picture, etc.), how the information is structured, and so on. Much of this information can in fact be obtained automatically, and this is something we are working towards. The next step is to automatically process the document to obtain more information, such as its size, a summary (if possible), and other automatically available information. All of this information is stored in the InfoAgent as a set of Dublin Core elements (Weibel et al, 1995), with some ACORN-specific extensions. For more details, see the appendix to this paper. Other information, such as the location of the document, is also obtained in this step and included in the InfoAgent's Dublin Core. Note that the majority of this information is meant to be human readable (it is the human, after all, who will eventually read the document and its summary as carried by the InfoAgent). As such, the agent need know little or nothing about how to process this information other than how to carry it and exchange it with others (which is discussed further below).

The third step in the seeding process involves the InfoAgent obtaining information about people or places who might be interested in the document it represents. These recommendations relate to static agents in the ACORN architecture, what we call 'Infrastructure Agents'. These may be specific ACORN users, or their GateKeepers. They may also be Archive Agents (sites which store only information and have no users), or MeetingPlace NameServers (which are

discussed further below). This data can come from the author of the information (since we hope the author will have some idea of their intended audience), although it need not. In addition, the GateKeeper can, from previously returned Info- and SearchAgents and what they learned on their travels, possibly suggest sites of potential worth. The InfoAgent stores these in a 'Recommended' list, tagged with the name of the recommending agent (the author, in this case).

Once initialised in this fashion, the InfoAgent is given an ACORN ID. This ID is unique in the ACORN architecture, and consists of the address of the creating site, the time the agent was created, and the partial ID of the GateKeeper who created it. Once this is set up, the agent is ready to carry out its task of information dissemination. Before proceeding to a discussion of migration and comparison, we introduce the other principal actor in this architecture, the mobile query, or SearchAgent.

2.2 Mobile Queries — SearchAgents

The InfoAgent in ACORN represents information. Each InfoAgent represents a single piece of information, from something as small as a paragraph or a single picture to larger pieces of information such as articles of text and pictures, video clips, and so on. InfoAgents are capable of sharing information about places they have been, users and agents they have interacted with, and so forth. Whilst this is an innovation in itself, it does not necessarily ensure that information is routed to those who seek it or would find it of interest.

In order for new users to become part of the loop, or for already present users to perhaps indicate new interests, or passing interests, in specific topics, we embody their interests or queries in Mobile Queries, the ACORN SearchAgents.

The SearchAgent is to all intents an purposes an InfoAgent whose piece of information is a query, or an expression of interest. To that end, the creation and seeding of a SearchAgent is much the same as has been described above. In the first instance, the user gives a set of search terms and any other information which may be of use. For example, the user may be looking for satellite images of the United Kingdom dated between December 1996 and January 1997. In this case, the Dublin Core elements instantiated include a descriptive field, a geographical descriptor, a temporal descriptor, and a request for images. For a query looking for single articles related to autonomous agents, in English, written after August 1995, the process is similar. Another possible query is for a person who is interested in encryption techniques and who has published papers in the area in the last three months. The queries that can be expressed are consequently detailed yet straightforward to set up. Their structure, along with other ACORN details, are presented in Marsh (1997).

The user, and the GateKeeper (user interface for a specific user) can further seed the SearchAgent in terms of Infrastructure Agents who may be able to supply relevant documents, or who may know places to look. These recommendations may be related to the query, or they may simply be agents who in the past have been helpful. These are put into the SearchAgent's 'Recommended' lists of sites and users.

The SearchAgent is now ready to begin its task, which is to find documents or users who can satisfy the query it has been given.

Despite the fact the authors of information may know some of their audience, or that the searchers may know of some places to look (much as today people know of Web search engines they prefer), data will be incomplete, users may not be know, and so on. In order to facilitate searches and distribution of information, more is needed to augment the system in order to make it useful in conditions of incomplete agent information. The process we use for this is information sharing between the agents in the system, and this is the community aspect of ACORN. information sharing is accomplished via agents coming together and interacting, and the next section discusses this.

2.3 Coming together

When an agent arrives at an ACORN site (the migration process is described below) they may have a specific person (their GateKeeper) to approach at that site. This they can do, and information can be shared between user and mobile agent. The mobile agent may also be able to register at specific 'MeetingPlaces' in order to 'meet with' other like minded agents. In addition, there is the possibility of approaching or being approached by other mobile agents at a site at random. In all such cases, an interaction occurs (possibly among several agents at once), and in all such cases, several Infrastructure Agents come into play. There are, however, subtle differences between these interactions, and we cover them, and their similarities, here.

2.3.1 Mobile Agent interacts with GateKeeper

When approaching a user, a mobile agent must perforce approach the user's GateKeeper. There is, in ACORN, no means of approaching a user directly: all agents are fielded by the GateKeeper, an Infrastructure Agent which also operates as the user's interface to the system. The GateKeeper can be set up as needed by the user to adapt its behaviour to the user's needs, even at specific times. For example, the GateKeeper can be instructed not to disturb the user between specific times, or not to forward any agents other than the responses to a specific query, and so on. Ultimately, of course, all agents (at least, their documents or queries) that approach

the GateKeeper can be examined by the user, but it is clear that constant disturbance is unpleasant and unnecessary.

In any case, when the GateKeeper is approached by any mobile agent, an interaction is started. In this, another Infrastructure Agent, the MeetServer, is called into play. The mobile agent is passed to the MeetServer, which compares it with what data the GateKeeper holds about its user, and a result is returned to both mobile agent and GateKeeper to inform them of their relatedness (the details of the result differ from site to site. As will be discussed below, this is one of ACORN's strengths). Based on this result, the GateKeeper can recommend other agents to approach (all Infrastructure Agents, and therefore static, or documents, and likewise static), and can prioritise the mobile agent's Dublin Core data for presentation to the user. The mobile agent places the recommendations on its 'Recommended' list, and puts this interaction on its 'Visited' lists, and is then free to go on its way. At this point, the mobile agent knows little of what the user thought of its document, if it is an InfoAgent. To handle the topic of relevance feedback in this instance, we use a mechanism of feedback to the author of the information: the reader may, eventually, look at the data stored by the GateKeeper and may fetch a document and/or recommend sites to visit (whether the agent was a Search- or InfoAgent). They may also provide relevance feedback on a document. This information is forwarded to the author of the information or query, and it is they who inform the mobile agent at migration time. This mechanism is described below.

2.3.2 Mobile agent interacts with mobile agent

The second form of interaction concerns ether chance meetings at a site or meetings at a specified 'MeetingPlace' at a site. Before discussing how such interactions are carried through, we present the idea of a MeetingPlace and how they work.

2.3.2.1 MeetingPlaces —- increasing interactions between mobile agents

As so far presented, an agent goes from place to place, presents its information or query to users, gets suggestions of where else to go, and carries on. While this may prove useful, it is limited, and can just as easily be achieved through, for example, an e-mail forwarding system (In fact, this is not strictly true, since the feedback from readers to authors is novel, and can be used to update or reroute agents. In addition, the author can know at any time where the document is and who has seen it. This is not always possible with e-mail forwarded from reader to reader(s)). To further extend the notion of information sharing, we propose Infrastructure Agents which support and facilitate interactions between agents at a site, These static agents may also provide some enhancements to the data mobile agents carry, allowing them to carry these enhancements with them in future, providing added

value in more ways than simple meeting facilitation. These Infrastructure Agents are MeetingPlaces, NameServers, and MeetServers.

In essence, a MeetingPlace is simply a virtual space in which mobile agents coexist. MeetingPlaces can be devoted to specific subjects, such as AI, or they can be more general. The emphasis of a MeetingPlace is up to the person who sets it up. A MeetingPlace is not in itself an agent, but it is useful to think of it as such conceptually. Agents interact with MeetingPlaces via NameServers and MeetServers. A site may have many such MeetingPlaces, or none. Each MeetingPlace is registered at a site, and agents are allowed to register with more than one MeetingPlace at a time. Registration at a MeetingPlace is handled by its NameServer, an Infrastructure Agent which maintains a list of agents registered at a MeetingPlace and can supply individual agents with lists of the same. The NameServer, on registration, can pass the mobile agent to an 'augmenter' which can scan the agent and augment its metadata. For example, augmenters could supply Library of Congress data for a document and store this in the agent. This information can now be used at this and subsequent MeetingPlaces who know how to use it. In this fashion, agents can be continually augmented throughout their lifecycle. In future, individual sites may be specifically noted for useful augmentation of agents, and may be visited by mobile agents specifically for augmentation.

In order to interact with others, an agent need know only their IDs. These are given, individually or in multiple, to the MeetServer for this MeetingPlace. MeetServers are similar in all MeetingPlaces (or at site level): they take the agents' data and compare it, sharing the 'Recommended' and 'Visited' lists in each agent between those who may find them useful: each new place to visit is put in the agents' Recommended lists, along with the ID of the agent which it came from. The interaction is then over: the agents themselves need know nothing of themselves or the other agents. All that is done, in their view, is an augmenting and possible reordering of their 'Recommended' lists, and thus of the places they are to visit in their lifetime.

Data is shared during the interactions mobile agents have. The primary purpose of this information sharing is the extension and possible reordering of their 'Recommended' list. It is this list that determines the path an agent takes around a network. It is the 'Visited' lists which contains the history of the agent's travels, along with any results of interactions at specific places. The structure of the Recommended and Visited lists, and the process of migration itself, is discussed in the next section.

2.4 Going places

Migration in ACORN is vital to the system. Without migration, agents cannot reach new agents to interact with, information cannot be shared, and ultimately the system stagnates. In fact, in the system, the agents themselves do not migrate. Rather, the information or query they embody does. In addition, the path they take around a network is dictated solely by the 'Recommended' list they carry, and the path they have taken is held in their 'Visited' list.

The structure of the two lists is important and useful to the mobile agent. The Visited list contains details not only of what sites were visited, but any augmentations added at that site (such as Library of Congress classification, or advanced summarisation). It will also contain relevance feedback from that site if applicable (which is given by the agent's owner at migration time: see later in this section). The Recommended list contains IDs and addresses of Infrastructure agents to visit (be they GateKeepers, MeetingPlaces, or others), their type, and who recommended them (this is useful since, if the recommender was the agent's creator, the agent can know that these sites should definitely be visited, while less faith may be placed in other recommendations). The Recommended list can be reordered from site to site, and indeed one of the value added benefits of specific sites may be the reordering that is done there, resulting in a more efficient traversal of the network. Once a site reaches the top of the Recommended list, the agent migrates there. To do that, however, certain protocols have to be maintained.

2.4.1 Migration

Once an agent decides to migrate, the first thing it does it to inform its creator. This is done via a message to its creator's GateKeeper (ACORN's messaging system is based on simple TCP/IP protocols, and is discussed in detail in Marsh (1997)). It is at this point that the creator of the document can forward any recent relevance feedback to the agent to be incorporated in its Visited list data. In addition, the creator can forward new sites to visit, to be placed in the Recommended list. Finally, the creator can ultimately deny migration to the intended site (for security reasons, perhaps) and can also, if necessary, request the agent's immediate return with no further migrations (perhaps if the document requires to be changed, or if a specific search has been completed to the user's satisfaction).

If the migration is allowed, the agent next contacts the 'SiteMaster' for this site. SiteMasters are the controllers of all agents at a site (in the present implementation, all agents a Java threads under the control of the SiteMaster). The SiteMaster at this site then decides itself whether or not the migration can take place. If not, it informs the mobile agent, which must rethink its strategy (perhaps attempting migration to

the next site in its Recommended list). The denial is accompanied by a reason, so that the mobile agent can make such a rethink in a more informed manner.

If migration is allowed from the local to the remote site, the remote SiteMaster is contacted and the mobile agent's metadata is passed to it. This metadata is the complete details of the agent, from Dublin Core to Visited and Recommended lists. The remote SiteMaster can scan this information and decide to allow or deny the agent to run at this site. If the agent is denied, it remains in operation at the original site. If it is allowed, a new agent is created at the remote site with the data passed over, and the agent at this site is destroyed (although, for tracking reasons, its data is held until it moves to the next site along, and then we can safely destroy it).

Once at the remote site, the agent removes its address from the Recommended list, and puts it into the Visited list. This entry in the Visited list is augmented as necessary to reflect the agent's progress at this site.

2.5 The spread of information, and its containment

As can be seen from the brief description above, the information held in ACORN's agents is mobile. It is the intended result of this mobility that the community of agents can spread their information to users who did not know of its existence beforehand, and of whose existence the author of the information or the originator of the query was equally ignorant.

The strength of the architecture is simply that such ignorance on either or both sides is not an impediment to the furthering of each user's knowledge. In addition, since SearchAgents can exist as long as their creator deems, they can act as 'browsers for information', finding useful information and forwarding it to their creator with little or not intervention on the creator's behalf. New information, which previously may have been missed by a searcher because it came out days, or even hours, after a previous search, can be caught and forwarded in a timely and efficient fashion.

Meanwhile, InfoAgents need only consume resources as long as they are deemed fit to by those who read them. The feedback they give to the writer can help the writer to refine and retarget the information with minimal effort. In order to stop such misuses as 'spamming', all readers have to do is inform their GateKeepers, and indeed other potential readers, to ignore similar messages, or messages from the same author, and so on. It should not take long before such unwanted information finds it has no places left to go, if the system works correctly.

3. ACORN: summary and current status

The ACORN architecture is still in its infancy. Development is under way using Sun's Java programming language, and we are extending the architecture to use

PDAs. Notably we are using Apple's Newton PDA, although any TCP/IP connected machine would suffice.

The current implementation is expected to be complete within the next few months, and will be ready for live testing shortly afterwards. A prototype is up and running on a single machine with simulated users, and results from this are being used to refine the architecture. Initial results are, however, promising, but show the need for careful consideration of the mechanisms by which the recommended list in agents are manipulated. Too little information, and agents simply stagnate, too much, and inevitably users can get visited more than once by the same information, which is unsatisfactory. We cannot hope that the authors of the information watch their mobile agents religiously to stop this, so mechanisms are being developed to limit it.

A beta testing phase is expected to be initiated in the near future, and interested potential testers are invited to contact the author directly. Further information of the current status of the project can be found in Marsh (1997), available from the author.

4. Related work

Autonomous agents are not new, and the idea of using them to 'connect' us with other people, or to search for information for us, is not new either. The Web is the ideal place for such agents, and many search agents exists to help us make use of it, for example CIG's Searchbots (Oates, 1994).

ACORN, however, goes further than simple searching into the realm of what has become known as 'matchmaking'. In fact, multi-agent systems are a useful tool in this approach, as is evidenced by the number of systems, both prototype and conceptual, that exist. The closest system to ACORN is Foner's Yenta (Foner and Crabtree, 1996). In the Yenta paradigm, agents communicate with each other on a network to discover like minded agents and introduce each other to other such agents. Clusters of agents result which can share relevant information with each other. In addition, when an agent is looking for someone (an expert) the clusters can use 'word of mouth' to direct the searchers to the proper agent to talk to. Whilst this is close to ACORN, the finer granularity of document as agent, or single piece of information as agent, is not yet achieved.

Yenta is not the only matchmaking system, either. Kuokka and Harada's (1995) matchmaking system is a centralised system which stores information, and can forward it to those seeking it. Being centralised, we feel that this system lacks some of the robustness and graceful degradation that ACORN achieves through distributing its information agents. In addition, for it to work, everyone has to know

of the existence of such a system. This is not a large barrier, but barrier it is, and it would be better for barriers to be as few and as low as possible.

Maltz and Ehrlich's (1995) 'active collaboration filter' provides a system where people can send pointers to interesting information to those they know are interested This system uses Lotus Notes rather than the Web, although it is not automated. Again, it is necessary for people to know of the site's existence, but the concept is useful and closely related to much of what ACORN attempts to automate.

Davies et al (Davies, Weeks and Revett, 1996) are developing the Jasper system of intelligent agents for the Web. The Jasper system can summarise Web pages, making recommendations to users with similar interests automatically.

Kautz, Milewski and Selman's (1995) 'Agent Amplified Communication' is another means of aiding "person to person communication in information finding tasks." Here, the informal person to person links within an organisation are used to form 'referral chains' using autonomous agents to hold user profiles and help to organise the process. This is an information finding mechanism using a community-based approach within an organisation, and as such may not be scalable, although it is possible to envision research networks working in such a fashion. In addition, the information is passive rather than active as in ACORN. We believe that the active information can disseminate faster to interested parties.

5. Conclusions and further development

Whilst still in a prototype phase, the ACORN architecture promises a great deal. In none of the similar systems described above is the entirety of what ACORN can achieve reached, yet all of them provide in some way a part of ACORN's functionality. The concept of producer push coupled with active consumer pull of information is a powerful one, and one which ACORN makes full use of. It is a robust system which we envisage will be able to adapt to the changing needs of its users, and which will provide utility with minimal user effort.

A beta version of the architecture will be available for live testing in the near future.

References

Davies, N.J., R. Weeks, and M. C. Revett. 1996. Information agents for the World Wide Web. *BT Technology Journal* 14(4):105-114.

Foner, L. and I. B. Crabtree. 1996. Multi-agent matchmaking. *BT Technology Journal* 14(4):115-123.

Hill, W., L. Stead, M. Rosenstein, and G. Furnas. 1995. Recommending and evaluating choices in a virtual community of use. *Proceedings CHI'95.* Available at: http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/wch_bdy.htm.

Kautz, H., A. Milewski, B. and Selman. 1995. Agent amplified communication. Paper read at *AAAI-95 Spring symposium on information gathering from heterogeneous, distributed environments*, Stanford, CA.

Kuokka, D., and L. Harada. 1995. Matchmaking for information agents. *Proceedings of the International Joint Conference on artificial intelligence, Montreal, Canada*, 672-678. Morgan Kaufmann, CA.

Maltz, D., and K. Ehrlich. 1995. Pointing the way: Active collaboration filtering. *Proceedings CHI'95.* Available at: http://www.acm.org/sigchi/chi95/Electronic/documnts/papers/ke_bdy.htm.

Marsh, S. 1997. Agent based community aided information dissemination and search: A detailed examination of the acorn architecture. Interactive Information Group, Institute for Information Technology, NRC Canada. In press.

Oates, T., M. V. NagendraPrasad, V. R. and Lesser. 1994. Cooperative information gathering: A distributed problem solving approach. University of Massachusetts Technical Report 94-66.

Weibel, S., J., Godby, E. Miller, and R. Daniel. 1995. OCLC/NCSA metadata workshop report. Available at:
http://www.oclc.org:5046/oclc/research/conferences/metadata/dublin_core_report.html.

Whitehead, S. D. 1994. Auto-FAQ: an experiment in cyberspace learning. 2nd International Conference on Mosaic and the World Wide Web. Available at:
http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Agents/Whitehead/Whitehead.html.

Appendix: The Dublin Core

The following are the metadata elements for a document in the Dublin Core set, taken from Wiebel et al (1995).

Subject: The topic addressed by the work
Title: The name of the object
Author: The person(s) primarily responsible for the intellectual content of the object
Publisher: The agent or agency responsible for making the object available
OtherAgent: The person(s), such as editors and transcribers, who have made other significant intellectual contributions to the work
Date: The date of publication
ObjectType: The genre of the object, such as novel, poem, or dictionary
Form: The data representation of the object, such as Postscript file or Windows executable file
Identifier: String or number used to uniquely identify the object
Relation: Relationship to other objects
Source: Objects, either print or electronic, from which this object is derived, if applicable
Language: Language of the intellectual content
Coverage: The spatial locations and temporal durations characteristic of the object

In addition, ACORN agents carry a unique ID, a summary of the information if possible (for example an abstract), and other data related to finding the author of the information.