

Handling Complex Information Environments: A Multi-Agent Framework

Travis Bauer and David B. Leake

Computer Science Department
Lindley Hall, Indiana University
Bloomington, IN 47405, U.S.A.
{trbauer, leake}@cs.indiana.edu

Abstract. The enormous amount of information available electronically, combined with the wide range of methods for information retrieval and manipulation, produces a complex electronic information environment that may be difficult for users to exploit. This paper summarizes how motivations and issues for a multi-agent approach to information retrieval are addressed in our multi-agent system called Calvin. Calvin observes users while they are accessing documents, proactively finds related documents, and provides a unified interface to the information environment. Calvin's agents individually implement high level abstractions of parts of the information retrieval task, integrating information from disparate sources. Calvin is built on top of a new Java middle-ware system called Geneva, a general multi-agent system framework which provides the resources necessary for supporting personal information retrieval. This paper gives an overview of how Geneva and Calvin address key issues for AOIS and shows how personal information agents can benefit from the open, flexible environment of a multi-agent system approach.

1 Introduction

The proliferation of information sources on the Internet and the many ways in which information can be analyzed and accessed has produced a complex information environment. The complexity of this environment arises from:

1. *Multiple sources of information for the user:* Information may be drawn from WWW search engines or company knowledge bases, all with different types and quality of information, and ways of being queried.
2. *Multiple sources of information about the user:* Users may implicitly provide multiple types of information about themselves during document access. For example the user may use multiple applications at once, such as a web browser, word processor, and email application. Each application is a different source of data.
3. *Multiple ways of interacting with user:* Multiple types of assistance can be given to a user, and these can be presented in multiple ways.
4. *Multiple analysis techniques:* There are multiple techniques for analyzing the user's document access behavior and for building user profiles.

Many systems are being developed to address specific personal information retrieval needs (e.g., [Rhodes, 2000]). These agents typically are constructed as individual programs, not part of a larger multi-agent system. While they give good performance at their specific tasks, no single specialized agent can provide the flexibility needed to deal with the previously-mentioned facets of the information environment. In particular, a multi-agent architecture offers four primary advantages over the traditional approach for information systems: *flexibility, robustness, distributed computing, and facilitating development.*

However, building a personal information agent out of a multi-agent community also presents new kinds of problems compared to self-contained personal information agents, including issues in security, coordination, potential additional complexity from the user's perspective, and information integration. We are working on methods to deal with these issues and are implementing them in the multi-agent system described in the following section.

2 System Overview

2.1 Geneva

Our multi-agent system is composed of two parts: Geneva, the middle-ware layer providing basic MAS services, and Calvin, a set of information-retrieval-specific agents. This architecture is depicted in figure 1. Geneva is the general framework underlying the agent communication system. It is built to be general enough for future expansion and change, while still providing the basic resources an AOIS needs. Geneva provides basic agent communication, authentication, and encryption functionality, and uses an open XML specification for communication among agents. Geneva is designed to allow agent developers to think in terms of the information being analyzed and retrieved rather than in terms of encryption and basic communication protocols.

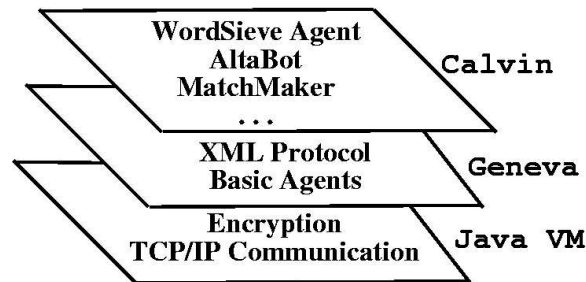


Fig. 1. Overview of Architecture

2.2 Calvin: An MAS for Personal Information Retrieval

Calvin is a set of agents build on top of the Geneva architecture. Each user has a personal profile which is persistent across sessions, developed and recorded individually by

analysis agents. The analysis agents generate and broadcast descriptions of the user's current context. The retrieval agents use these context descriptions to query standard search engines. The results are sent back to the user interface agent to be suggested to the user. Figure 2 illustrates the agent interactions, with each box representing a separate agent.

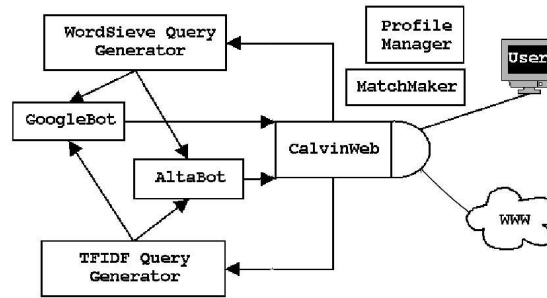


Fig. 2. Calvin's Agent Architecture

The user needs a way to interact with the agent community. Given the already complex nature of information retrieval tasks, it is undesirable to force the user to create individual mental models of all the agents. Thus the user needs an interpreter to translate information about him to the system and interpret the answers the community. In Calvin, this is accomplished by the user only interacting with a single agent, CalvinWeb.

CalvinWeb serves not only as an interface for the user to get information from the MAS, but also as an interface to enable the MAS to gather information about the user. CalvinWeb gathers the web pages the user accesses and passes them on to the user analysis agents, which in turn pass their analysis to the data retrieval agents. The data retrieval agents use this information to send queries to specific information sources, and return the results to CalvinWeb. Currently Calvin has two data retrieval agents, GoogleBot and AltaBot. These agents issue queries to the Google and Alta Vista search engines, respectively. Each takes the raw query from the User Analysis agents and changes it into a query which has the proper syntax for its search engine and is adapted to search-engine characteristics (e.g., by limiting the number of terms of the query for Google) and has the proper syntax.

3 Comparison to other work

Personal information agents are an area of active research. Usually, these programs are designed to assist the user via unobtrusive watching and assisting. Some agents, like Watson [Budzik *et al.*, 2001] are designed to do background web searches on behalf of the user based on the content of the current page. Other agents [Rhodes, 2000] are designed to index and search the files on a user's hard drive, to suggest existing files

at the appropriate times. Unlike the agents of Calvin/Geneva, these are by their nature independent programs, not designed to work as part of a multi-agent system.

Calvin/Geneva has the most in common with the multi-agent information system Profile [Simons *et al.*, 2000]. Both systems use an MAS approach to providing a flexible interface to the electronic information environment. The primary difference is in how the user interacts with the system. Profile has a strong focus on explicit user interaction. The user gives queries to the system, and, as the authors note, this requires some new users to develop new concepts to understand what the system is doing. Calvin, on the other hand, monitors user's behavior in the background.

OySTER, another multi-agent information system, takes a more information retrieval oriented approach and includes user profiles and query refinement [Muller, 1999]. It differs in the a greater attention to security, and to a slightly different style of openness. OySTER is open and extendable, but uses a fixed language for communication. In Calvin, the documents are broadcast into the environment and agents can do whatever they want with them. There is no blackboard intermediate step for representing a user's status.

4 Conclusion

Using an MAS approach to construct personal information retrieval agents provides flexibility and distributed computing capabilities along with facilitating development and research. It also requires addressing certain challenges such as security, coordination, and an inter-agent communication. This paper has presented a multi-agent infrastructure called Geneva, and the information retrieval agent built on it, called Calvin. Calvin, as a whole, observes, analyzes, and assists users while accessing documents on their computer. By building Calvin as a multi-agent system on top of Geneva, we are able to take advantage of the flexibility and distributed computing that an MAS offers while dealing with the issues of security, coordination, and complexity. This research suggests one way that advantages can be realized and challenges met by a multi-agent system approach to information retrieval.

References

- [Budzik *et al.*, 2001] J. Budzik, K. Hammond, and L. Birnbaum. Information access in context. In *Knowledge based systems*, 2001.
- [Muller, 1999] Martin E. Muller. An intelligent multi-agent architecture for information retrieval from the internet. Technical report, University of Osnabruk, 1999.
- [Rhodes, 2000] Bradley J. Rhodes. Margin notes: Building a contextually aware associative memory. In *Proceedings of the 2000 international conference on Intelligent user interfaces*, pages 219–224, Jan 2000.
- [Simons *et al.*, 2000] J. Simons, A. T. Arampatzis, B. C. Wondergem, L. R. Schomaker, P. van Bommel, Th.P. van der Weide, and C. H. A. Koster. Profile - a multi-disciplinary approach to information discovery. In *Second International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2000)*, Jun 2000.