The Distributed Agent Architecture of the University of Michigan Digital Library (Extended Abstract)

William P. Birmingham, Edmund H. Durfee, Tracy Mullen, and Michael P. Wellman*

Artificial Intelligence Laboratory
University of Michigan
1101 Beal Avenue
Ann Arbor, MI 48105-2110
{wpb,durfee,mullen,wellman}@eecs.umich.edu

Abstract

We provide an overview of a distributed-agent architecture design for the large-scale digital library environment currently under development at the University of Michigan. This paper presents some of the design principles and techniques underlying our approach, as well as some preliminary observations from our first steps in developing the system.

Overview: Diversity in the UMDL

One of the most interesting and technically challenging features of digital libraries is *diversity*—in users, in information sources, and in many other features. For example, user skills, information demands, and level of professional competence will vary greatly among users of digital libraries. Similarly, the collections will vary over many parameters, including breadth and depth of subject, medium, and format. We expect that diversity will continue to increase and even accelerate, especially as digital libraries facilitate the spread of the publishing function from traditional publishing organizations to individuals.

Diversity in large-scale information systems is also manifest across time. The types of tasks users perform and the tools available for performing these tasks are constantly evolving. Because the technology is changing at such a rapid pace, we believe that it is inadvisable to provide an all-encompassing set of standards for both user interfaces and collections. Thus, we do not advocate standards for describing and performing tasks, such as standard query languages or search engines. Instead, we have chosen to concentrate on defining an architecture that performs task-management tasks (meta-tasks), such as allocating resources to user-requested tasks and brokering connections among collaborating system modules. For example, we define a language and protocol for communicating about informational or processing capabilities and interests, so that interfaces and collections can be brought together appropriately. At that point, however, what they do with that connection (*how* they interact to accomplish their task) is not a concern at the level of our architecture.

The advantage of avoiding general standards regarding information storage, retrieval, representation, or processing is that it makes incremental growth of our system both possible and appealing to developers. It certainly does not preclude the creation of partial standards, which is in itself a worthy goal, but it does not rely on their universal adoption. In fact, a goal of our endeavor is to provide an infrastructure where new capabilities—abilities to provide a better user interface, search more efficiently, translate among languages or representations, and so on—can be incorporated easily and can participate in tasks for which they are especially well suited without having to overhaul the existing components. The challenge that we face, therefore, is to provide an infrastucture that supports interoperability among potentially widely diverse collections, user interfaces, and other information providing and processing components.

In building the University of Michigan Digital Library (UMDL) (Birmingham *et al.* 1994), our approach is to distribute information-retrieval tasks to highly specialized agents. Large numbers of fine-grained agents promote modularity, flexibility, and incrementality of the system, and provide a framework for tackling the interoperability problems. As new services become technologically feasible (e.g., information filters), agents can be added to the architecture; similarly, as new bodies of information become available we can likewise create new agents to manage these collections.

*Authors alphabetical. We gratefully acknowledge the assistance of many others participating the UMDL project, especially Gene Alloway. This work is supported in part by the National Science Foundation Digital Library Initiative.

UMDL Agents

We distinguish three broad classes of agents populating the UMDL:

- UIAs (User Interface Agents) provide a communication wrapper around a user interface. This wrapper performs two functions. First, it *encapsulates* user queries in the proper form for the UMDL protocols (see below). Second, it *publishes* a profile of the user, which is used by mediator agents to guide the search process.

- Mediator agents (Wiederhold 1992), of which there are many types, perform a variety of functions: essentially, all tasks that are required to refer a query from a UIA to a collection, monitor the progress of the query, transmit the results of a query, and perform all manner of translation and bookkeeping. Initially, two types of mediator populate the UMDL. *Registry* agents capture the address and contents of each collection. *Query planning* agents receive queries and route them to collections, and then collect the results.

- CIAs (Collection Interface Agents) provide a communication wrapper for a collection. While performing translation tasks similar to those performed by the UIA for a user interface, the CIA also publishes the contents and capabilities of a collection in the *conspectus* (described below).

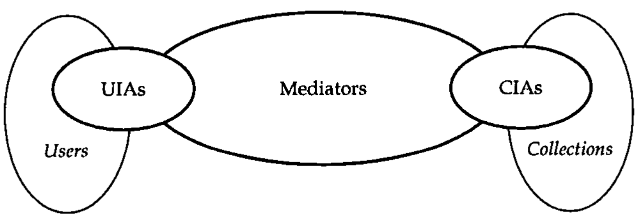

Figure 1: UMDL agent types.

The use of a heterogeneous agent architecture enables us to develop specialized capabilities, and add these capabilities to the UMDL architecture as needed. For example, we can create user interfaces that are customized to a particular class of users, rather than to a particular collection or access mechanism (e.g., boolean search over controlled vocabulary). As the architecture is developed, the broad classes of agents depicted in Figure 1 will be continually refined.

Agents in the UMDL accomplish particular tasks through various special-purpose languages and protocols, which are outside the scope of the UMDL architecture. Which tasks to perform and how is determined through inter-agent communications, via a set of negotiation and service protocols, which in turn build on languages describing such properties as collection content (conspectus), agent capabilities (agent ontology), and other task-related items (task ontology). Figure 2 depicts the three levels of communication protocols employed by UMDL agents.

UMDL Protocols

- Generic agent communication language (KQML)
- Descriptions of collection contents
- Descriptions of agent capabilities

Task-Specific Protocols

(e.g., query languages, application-driven interactivity)

Network Protocols

(e.g., TCP/IP, HTTP)

Figure 2: UMDL agents communicate using protocols defined at three distinct levels of abstraction. The UMDL architecture defines only the highest level.

Achieving Interoperability

Interoperability issues arise at two levels: individual communications among mediator agents, and patterns of search over heterogeneous collections.

Partitioning the Information Space

The space of information in the UMDL is potentially enormous, as is the possibility of bringing the system to its knees with rogue query processes. To limit queries to potentially applicable CIAs, we reason about the contents of each collection to derive an estimate of its likely usefulness. This leads us to a two-level partition of the information space:

- Conspectus: a set of descriptions of collections. For each collection, the conspectus describes, among other things, the content of the collection, the capabilities of search engine(s) associated with the collection, and the structure of the material (documents) in the collection.

- Collection: the set of actual documents in a collection. These documents are in native formats, and the search engines are engaged through native query languages.

The conspectus comprises abstracted descriptions of all collections in the UMDL. Each CIA publishes to a special mediator (the registry agent) the conspectus information for its collection. These conspectus entries contain a *normalized* description of the collection's content. Normalization avoids some of the problems arising from the various ways each collection has of describing the same thing (e.g., title as TI or TL). To help normalize terms, we are using a variety of thesauri developed by various researchers around the world.

The conspectus information is encoded in a special UMDL Conspectus Language (UCL). Although the UCL itself is defined by the UMDL design group, the actual conspectus information expressed in UCL is specified by those managing the separate collections. Our aim is that UCL (and its associated resources, such as various thesauri and cataloging systems) provide sufficient structure for developing compatible representations of collections. Thus, the conspectus facilitates interoperability for some search and retrieval methods through a common representation over collections.

To illustrate the sort of information that is contained in the conspectus, Table 1 presents a sample conspectus entry for the *Blue-Skies* collection, a weather information resource incorporated in the UMDL. As described in the entry, Blue-Skies is a service developed at the University of Michigan (Samson, Hay, & Ferguson 1994) to provide a variety of basic weather information for K-12 education. The Blue-Skies service includes real-time interactive weather images, as well as various forms of archival data, images, and documents. Thus, we see that the notion of "collection" in the UMDL is a rather broad one.

Since the conspectus will be enormous both in scope and in size, it will be distributed and hierarchically organized. As it becomes necessary, we intend to create special mediator agents whose sole responsibility is to maintain the efficient organization and integrity of the conspectus.

The UCL and associated maintenance agents support interoperability or information resources at the conspectus level. We deliberately do not, however, directly support interoperability at the collection level. Given the wide diversity of the collections, it is not possible to structure them completely and uniformly. Therefore, the UIAs will create specific user queries in any number of query languages that will be delivered by mediators to the CIAs. The mediators will determine the subset of appropriate CIAs capable of handling the query, using information gleaned from the conspectus. The ramification of our decision is that it is not possible for every UIA to access every CIA, thereby potentially missing an important source of information. The existence of specialized *translating agents* can ameliorate the problem somewhat, by increasing CIA interoperability at the query-language level. However, we do not believe that completeness guarantees are practically attainable with current or foreseeable technology. Indeed, given that the *substantive* completeness that the user would *really* like—"Does this library contain resources that will answer my question?"—clearly cannot be reasonably promised in a large-scale digital library, completeness guarantees of other sorts are of limited actual value.[1]

Agent Interoperability

As the tasks of delivering queries and documents is a complex one, particularly in a highly distributed, heterogeneous environment, mediation agents of various capabilities must be able to communicate to form teams. To facilitate, indeed enable, communication we are developing a language to describe agent capabilities and a set of protocols for specific tasks. The work on describing agent capabilities is just beginning, and will not be discussed further.

Our protocols for agent interaction and negotiation call for agents to deal with each other as far as possible at the knowledge level—specifying goals to be achieved, for example, not the method to achieve them. In our initial specification and implementation, we are using KQML (Finin *et al.* 1994) with several extensions to the set of performatives. For example, to register a new collection, messages implementing the following are exchanged:

1. Each agent registers its IP address.
2. Advertise
   - CIAs advertise their conspectus.
   - The query planner advertises its capability.

In order to participate in UMDL, an agent must use our protocols. Since these protocols are minimally restrictive in how a task is accomplished, we believe they are not a significant impediment to the development of agents by third parties. Standardizing the protocols but not the lower-level languages and procedures strikes a balance between flexibility and ease of integration into the UMDL environment. Participation by third-party agents is to be encouraged, provided that the protocols impose sufficient restrictions on agents to prevent them from manipulating the environment in detrimental ways (Rosenschein & Zlotkin 1994).

[1]This is not to deny that reasoning about *local* completeness can be very useful for reasoning about the scope of queries. For an interesting treatment of this problem, see (Levy, Sagiv, & Srivastava 1994).

Collection Title	Blue Skies
Description	Recipient of one of Apple Computer's "Cool Tools Award for Internet Programming Efforts," this Macintosh and Windows application offers a graphical interface to weather and environmental information featuring interactive graphics and user input features.
Content Provider	Weather Underground, AOSS Department, University of Michigan
Rights Owner	Weather Underground, AOSS Department, University of Michigan
Conspectus Contact	James Alloway, cerebus@engin.umich.edu
Broad Topic(s)	Earth sciences
Specific Topic(s)	Atmospheric sciences
Searchable Form(s)	
Deliverable Form(s)	Text: HTML (Level 3), Software: Macintosh/Windows
Audience	Students: K-12, Teachers: K-12, Adult: General, Juvenile: General
Treatment	Introductory,Guide
Language	English
Chronological coverage	1995
Update Frequency	Daily
URL	`http://cirrus.sprl.umich.edu/Weather_Underground.html`
Native Search Engine	

Table 1: Sample conspectus entry for the Blue-Skies collection.

Mediator Agents

Mediator agents collectively provide a wide variety of capabilities to the UMDL. A mediator might have capabilities specific to query processing, such an ability to decompose a particular class of queries into subqueries, to fuse results into a coherent picture, or to translate among query formats. Other mediator capabilities might be in the areas of resource identification, allocation, and coordination. Such mediators must be able to find useful resources (collections and/or services) based on registration information or based on ongoing probing within the network. They must be able to dynamically construct agent teams for a task, based on the capabilities and costs of the team members. And they must be able to control the performance of those teams to avoid redundant and conflicting activities, to resolve conflicts when they arise, and to manage the resource budgets that may be in force for some tasks.

In our initial development of mediators, we are focusing on agents that can identify, allocate, and coordinate capabilities and collections. Our approach is to use the UMDL protocols to exchange information about objectives (such as "answer a query" or "update the list of agents providing service X") and constraints on how to achieve those objectives (such as "do not probe more than N collections"). A recipient mediator must then use its local knowledge to determine how (if at all) it could best meet the objective within the constraints. To carry out its procedure, it might in turn generate objectives (such as "determine the availability of agent Y") to be accomplished by other mediators. Through the selective exchange of messages, and pattern-directed invocation of alternative procedures at each mediator (we are beginning by considering a version of the PRS plan language (Ingrand, Georgeff, & Rao 1992; Lee *et al.* 1994)), the system can construct situation-dependent teams of agents (resources, capabilities), which then can internally coordinate to decompose tasks further and search for more efficient assignments of resources to their subtasks.

Resource Allocation in the UMDL

Given the size and scope of the digital library, there is potentially unbounded demand for computational resources requested by information agents. For example, any amount of pre-processing of data in the collections-such as indexing, meta-data gathering, or caching might improve the response of the system to subsequent user requests. In order to achieve an appropriate level of the various preprocessing options, we seek principled methods for expressing the operating tradeoffs and allocating the computational resources toward their maximal expected benefit. Moreover, we require that these methods be sufficiently flexible to adapt to patterns of usage that evolve during the operation of the digital library.

In approaching this resource allocation problem, we treat the alternative information services as competing economic activities. Given a measure of priorities over the end-user services provided, the various agents effectively compete to provide the highest level of service using the minimal computational resources. If the competition operates smoothly, the result can be an efficient overall allocation of resources towards the optimal provision of services to users.

To organize the processing activities within an economic framework, we view the interactions between agents as supplier-producer relationships, where each agent produces value-added information products from the input products provided by others. Agents dynamically connect with each other as opportunities arise for mutually beneficial exchanges. The collections provide the ultimate "raw materials" in this process, whereas the end users are the ultimate consumers of the "finished goods". The mediators ("middlemen") bridge the gap by bringing to bear knowledge, processing, storage, or other computational resources to improve in some way the expected value of the information as it passes along the chain from agent to agent.

To begin exploring some of the economic issues arising in the UMDL, we have developed a simple computational market model of a generic network-service distribution problem (Mullen & Wellman 1995). The problem we posed involves a LAN (or WAN) which uses several network information products, such as weather or news services, that could be encapsulated as UMDL collections. One central resource allocation question involves locating and caching these service providers on the network. In particular, under what conditions is it worthwhile to establish one or more mirror sites for these products on the LAN? Where should they be?

To answer this question, we elaborate on our previous work on computational economies for transportation services (Wellman 1993). Our initial model represents a simple network composed of one internet site and two local sites. We used *Blue-Skies* (see Table 1) as our information service product. In this example, Blue-Skies is produced on the internet, and local users access it frequently. Accessing it from the internet site may mean longer delays, but setting up a mirror site means using up local resources like disk space and machine access time, as well as incurring an initial overhead cost of transporting the entire product line to the local mirror site. Clearly current and anticipated usage patterns by the consumer should be the determining factor in which setup makes more sense.

In order to design a computational economy, we need to specify the goods and the agents that produce and consume these goods. In our model, each end user, or *consumer*, is endowed with basic network resources, which they can "spend" buying Blue-Skies. On the other hand, in our model the consumers want to purchase Blue-Skies, not at the internet site, but on their local site. In other words, Blue-Skies@internet is a different good than Blue-Skies@local-site. The units in which these services are described combine measures of quantity and quality (delays, fidelity) of service provided.

Producers are determined by their *technology*, which describes the resources required to produce varying qualities of service. In our model, we assume that the cost or resource required goes up quadratically with the level of service. We use another type of mediator producer to bundle complementary goods into one good. So Blue-Skies@local-site is produced by a Delivery mediator which bundles buying and transporting Blue-Skies into one good.

Initial runs of the model show that the caching choices of the service providers are sensitive to the relative prices (or loads) on the network. Depending on the initial setup and the parameters of cost and demand, the system produces a range of qualitatively distinct behaviors. Moreover, these behaviors are predictable and designable according to principles of economic theory.

Future development of this computational economy include adding more products and thus allowing tradeoffs between them. This includes adding more information services and also representing qualities such as timeliness, delay, and fidelity to the current goods. For example, there could be mirror sites which do not update as often as the internet source but are cheaper, or mirror sites which carry a higher-quality version of Blue-Skies. This also ties into issues about how to setup licensing and distribution rights which could require setting up some kind of futures market in the UMDL. Several other extensions are planned to test the scope and scalability of this approach.

The Prototype System

We have developed a small, working prototype of the UMDL architecture. This system contains the following agents:

- <u>UIA</u>: A prototypical UIA that profiles a user and can register itself is operational. This agent uses the NetScape^{TM} World-Wide Web (WWW) browser to display HTML documents, and SoftQuad's Panorama^{TM} to display SGML documents.
- <u>CIA</u>: We have developed CIAs to represent a collection of materials-science journals, several WWW sites containing weather and earth-science information, and a science and technology encylopedia. In

total, we have five collections, and CIAs, running. The CIAs are capable of publishing to a registry agent their conspectus, and supporting a variety of search engines, each engine specialized to a collection.

- Mediators: We have developed a registry agent that maintains a list of all active agents and a database of conspectus entries. We have also developed a task-planning agent that interacts with both the registry angent and a UIA to determine which CIA(s) should receive a query.

While this is a simple system, and it does not support any sophisticated method of resource allocation, we can deliver information to users. We intend to develop the full-scale version of the UMDL architecture by incremental extension of this working system.

Conclusions

Work on the UMDL is too preliminary to offer any serious conclusions. It is clear already, however, that the scale and diversity of the digital library conceived here will put to the test our technical ideas—on distributed agents, interoperability, mediation, and economic allocation of resources.

References

Birmingham, W. P.; Drabenstott, K. M.; Frost, C. O.; Warner, A. J.; and Willis, K. 1994. The University of Michigan Digital Library: This is not your father's library. In *Proceedings of Digital Libraries '94*, 53–60.

Finin, T.; Fritzon, R.; McKay, D.; and McEntire, R. 1994. KQML as an agent communication language. In *Proceedings of the Third International Conference on Information and Knowledge Management.*

Ingrand, F. F.; Georgeff, M. P.; and Rao, A. S. 1992. An architecture for real-time reasoning and system control. *IEEE Expert* 7(6):34–44.

Lee, J.; Huber, M. J.; Durfee, E. H.; and Kenny, P. G. 1994. UM-PRS: An implementation of the procedural reasoning system for multirobot applications. In *Proceedings of the AIAA/NASA Conference on Intelligent Robotics in Field, Factory, Service, and Space*, 842–849.

Levy, A. Y.; Sagiv, Y.; and Srivastava, D. 1994. Towards efficient information gathering agents. In *AAAI Spring Symposium on Software Agents.*

Mullen, T., and Wellman, M. P. 1995. A simple computational market for network information services. Submitted for publication.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers.* MIT Press.

Samson, P. J.; Hay, K.; and Ferguson, J. 1994. Blue-Skies: Curriculum development for K-12 education. In *American Meterological Society Conference on Interactive Information and Processing Systems.*

Wellman, M. P. 1993. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research* 1:1–22.

Wiederhold, G. 1992. Mediators in the architecture of future information systems. *IEEE Computer* 25(3):38–49.