

Capability-based Agent Matchmaking*

Anthony Cassandra
arc@cassandra.org

Damith Chandrasekara
MCC
3500 W. Balcones Ctr. Dr.
Austin, TX 78759
damith@mcc.com

Marian Nodine
MCC
3500 W. Balcones Ctr. Dr.
Austin, TX 78759
nodine@mcc.com

ABSTRACT

In an agent-based system, where different agents form dynamic associations for the purposes of collaborative processing, agents must be able to locate other agents that can provide them with needed capabilities to accomplish specific tasks at a given time. This implies a need for agents to be able to describe themselves to other agents, perhaps in terms of its interfaces, the services it can provide, the data it can provide, etc. To be practical, these descriptions cannot be as complex as a formal specification; neither can they be as simplistic as the distributed object systems use. This paper presents an example of the approach to agent capability description and matching that we have implemented in the InfoSleuth system. This approach is expressive enough to capture complicated agent functionality, yet simple enough to be scalable to large and diverse agent systems. It relies on shared, focused ontologies that provide a common vocabulary for describing information and services. An agent can then advertise itself in terms of the focused ontologies, and query about other agents using these same ontologies.

1. APPROACH

Our experience with the InfoSleuth¹ [1; 3; 2] agent system led us to the conclusion that we needed to represent agent capabilities at multiple levels, including (1) the agent conversations that are used to communicate about the service, (2) the interface to the service, (3) the semantics of what the service does, and (4) the information a service operates over; each in terms of its own focused ontology. This representation relies on the availability of individual ontologies that codify specific conversations, languages, services and domains into a set of agreed-upon terms that can be shared among agents and users. An important property of our framework is that, because it is independent of the ontologies being reasoned over, it is extensible and flexible.

*Extended version is in MCC Technical Report SRI-072-99.

¹InfoSleuthTM is a trademark of Microelectronics and Computer Technology Corporation.

2. ADVERTISING

This paper presents an extended example of advertising and matching in the InfoSleuth system. In this example, we have a large collection of environmental resources, each containing information about various contaminants and sites. Each resource has its own local resource agent that makes the information available in terms of the environment ontology. The resource agent named tx-env-resource provides environmental contamination information related to contaminated sites in Texas. Its advertisement is shown in Table 1. This advertisement is interpreted to mean that the agent tx-env-resource has several capabilities, any one of which can match a given query.

advertisement

```
capability monitor-subscription-capability
...
capability env-subscription-capability
ontology conversation
class conversation
slot conversation-type in {subscribe}
slot language in {kqml}
ontology sql
ontology environment
class site
slot contaminant
slot remedial-response
slot city
slot state in {Texas}
class contaminant
...
ontology services
class data-response
slot language in {tuple-format}
slot delivery in {http, inline}
class subscription
slot computation in {direct}
capability env-query-capability
...
```

Table 1: Advertisement for agent tx-env-resource.

This agent advertises generic capabilities, such as the ability to be monitored (the monitor-subscription-capability). It also advertises two capabilities specific to itself – its ability to be subscribed to (the env-subscription-capability) and its ability to be queried (the env-query-capability). Only the env-

```

capability subscription-capability
  ontology conversation
    class conversation
      slot conversation-type in {subscribe}
      slot language in {kqml}
  ontology sql
  ontology services
    class data-response
      slot language in {tuple-format}
      slot delivery in {http, inline}
    class subscription
      slot computation in {indirect}
      slot adaptability in {dynamic}
  ontology data-independent

```

Table 2: Subscribe capability for agent my-subagent.

subscription-capability is described in detail. This capability consists of a set of related ontology fragments. The fragment of the conversation ontology specifies that the agent accepts conversations of the form used by subscriptions, using the language KQML. The sql ontology fragment specifies that the query should be specified in SQL syntax. The environment ontology fragment specifies the classes and slots that have data in the agent. The service ontology fragment specifies that the service is direct (accessed locally within the agent), as well as other service properties.

The system also may incorporate agents that merge subscriptions from these multiple resource agents into a unified view. The advertised subscription capability for one such agent is shown in Table 2. The subscription-capability states that the subscription agent can respond to subscribe conversations in KQML where the subscription query is specified in SQL over any ontology, and the response is delivered either inline in the KQML message or off-line using http. The indirect specification in the services indicates that the actual information is retrieved from other agents, and the dynamic adaptability indicates that its answers can take into account new information from new resource agents. The data-independent ontology fragment indicates that this agent is a general subscription agent that can provide subscriptions over any data domain.

3. QUERYING

Within the context of an agent system, a user agent supports the users in inserting subscriptions into the agent system and processing the results. Suppose that the user presents a subscription request “SELECT contaminant FROM site WHERE site is in Austin, TX”.

The user agent queries the matchmaker is shown in Table 3. This query matches resource agent tx-env-resource, but not subscription agent my-subagent. It also matches any other direct information resource agent that has some compatible capability advertised, so the user agent may receive a list of several agents that match. In this case, the user agent may decide to use a subscription agent to return an amalgam of information from multiple resources as a single subscrip-

```

query
  capability subscribe-to-capability
  ontology conversation
    class conversation
      slot conversation-type in {subscribe}
      slot language in {kqml}
  ontology sql
  ontology environment
    class site
      slot contaminant
      slot city in {Austin}
      slot state in {Texas}
  ontology services
    class data-response
      slot language in {tuple-format}
      slot delivery in {inline}
    class subscription
      slot computation in {direct}

```

Table 3: Subscription query for agents that have data on new contaminants at sites in Austin, TX.

tion. The query for a subscription agent is nearly the same as that for a specific resource, except the computation mode is indirect and the environment ontology is replaced with the data-independent ontology. With this new query, the matchmaker would match the request to my-subagent.

When the agent my-subagent receives the subscription request, it in turn asks the matchmaker for all direct suppliers of the requested information. It subscribes to all agents that are returned from the matchmaker, assembling the results together into a single result stream. When new resources come online that have relevant information, my-subagent subscribes to those agents as well.

4. CONCLUSIONS

This paper has provided an example of a methodology for general-purpose matching in a dynamic agent system. This approach has proved to be both adaptable and efficient in a wide variety of real applications. This research has been implemented within the context of the InfoSleuth agent system, and would not have been possible without the insightfulness and hard work of the MCC InfoSleuth group members.

5. REFERENCES

- [1] R. Bayardo et al. InfoSleuth: An agent-based semantic integration of information in open and dynamic environments. *Proc. SIGMOD Int'l Conference on Management of Data*, 1997.
- [2] M. Nodine et al. Active information gathering in InfoSleuth. *Int'l Journal of Cooperative Information Systems*, 9(1/2), 2000.
- [3] M. H. Nodine and A. Unruh. Facilitating open communication in agent systems: the InfoSleuth infrastructure. In *Proc. 4th Int'l Workshop on Agent Theories, Architectures, and Languages*, 1997.