Abstract
Like middle-men in physical world, middle-agents assist in locating and connecting the ultimate (information or services) provider with the ultimate requester in the multi-agent systems [3]. In this report, the basic concept of middle agents is introduced, a list of taxonomy dimensions along which middle agents can differ and possible values for these dimensions are presented, then some typical middle agents types and their brief descriptions are discussed. Middle agents are now widely used in many multi-agent systems, some applications and general descriptions of how middle agents work in such systems are listed. I concluded this report with some open issues in this research area.

Introduction
A distributed, dynamic, open multi-agent system usually involves a large amount of active agents, include information providers, information requesters, and users that both provide and consume information. In such systems, agents involved may appear and disappear unpredictably, therefore, effective, automated search and selection mechanisms of relevant services and resources become more and more essential for multi-agent systems. Middle agents appear to address this issue, the notion of middle agents is proposed by [1] – “Agents that deal with preference or capability information that are neither requesters nor providers (from the standpoint of the translation under consideration) we call middle-agents”. Which means that middle agents are entities to which the providers advertise their capabilities and from which the requesters may find and use the services and information. This make middle agents differ from other two types of agents in open multi-agent systems: provider agents which offer services,
and requester agents which need provider agents to perform some service to them. In the following sessions, I will introduce the taxonomy and several different types of middle agents, explore their performance, applications and open issues.

Middle Agents Taxonomy

To find agent with desired capabilities, end-agents can send (‘push’) information about themselves to the middle agents, or ask (‘pull’) the middle agent about what is available in the system [3]. Based on the finding procedures, the following six dimensions and their respective answers identified in [3] that could characterize and make the taxonomy of middle agents,

1. Who sends information to middle agent? The senders could be either providers or requesters.
2. How much information is sent to a middle agent? The answer could be either capabilities/requests or parameters/preferences.
3. What happens to the information sent? It could be broadcast or kept in a local database.
4. How is the content of the database used? It can be browsed or queried. If it is browsed, then the ‘puller’ receives the content of the whole database. Otherwise, the ‘puller’ gets only a subset of the information in the database, determined by what is specified in the query.
5. How much inform is specified in a query to a middle agent? The information could be capabilities/requests or service parameters/preferences.
6. Does the middle agent intermediate transactions between end-agents? Middle agents may intermediate transactions to implement anonymity of the end-users, or to guarantee fairness [3].

Typical Middle Agents

There are several types of agents that lies in the definition of middle agents. [1] The three most commonly used types are discussed below,

- Blackboard
  A blackboard is a middle agent that keeps track of requests to which requesters post their problems and providers can query for the events that they have the capability to handle. It actually acts as an repository that receives and holds requests for other agents to process. The use of blackboard could protect the privacy of the providers’ capabilities while requesters’ preferences are public to both requesters and providers.

- Broker
  A broker is a middle agent that receives requests and performs actions using services from other agents in conjunction with their own resources. It understands both the requesters’ preferences and providers’ capabilities, but neither the requesters nor providers know directly about others in a transaction. Therefore, it could protect the privacy of both the requesters and providers. (Fig. 1)
Matchmakers and yellow pages
A matchmaker or yellow page is a middle agent that assists service requesters to find service provider agents based on advertised capabilities [3]. In this organization, providers advertise their capabilities with a matchmaker. If those capabilities change, or these agents exit the open system, the providers unadvertise. A matchmaker stores these advertisements in a local database. A requester wishing to ask a query first formulates a meta-query asking for advertisements from matchmaker, the matchmaker answers this meta-query with a set of matching advertisements. A requester can then use its full preferences to choose a provider, and make its request directly. In addition, the requester could keep local cache of the current advertisement, if the same type of its query happens very frequently. This local cache could make the whole matchmaking process more robust[1]. (Fig.2)

Facilitators
A facilitator is a middle agent to which other agents surrender their autonomy in exchange for the facilitator’s services [4]. Facilitators can coordinate agents’ activities and can satisfy request on behalf of their subordinated agents.

The following table list some middle agents with respect to privacy considerations of service providers’ capabilities and requesters’ preferences.
Each multi-agent system relies on three basic roles: requesters, middle agents, and providers. Any single agent in a domain system might take multiple roles, an agent that requests basic information from several providers, does some complex integration, and then serves the integrated information to other requesters [1]. Additionally, in an open, dynamic system, hybrid middle agent type could exist in the same organization, for example, an provider may query a matchmaker to find an appropriate broker, and then advertises with one broker. Brokers advertise summary capabilities built from all the providers that have advertised with them, these capabilities are in turn to the matchmaker [1].

### Performance

Efficiency, robustness and adaptiveness are the metrics adopted in [1] to analyze the performance of three basic types of middle agents, the blackboard, the broker, and the matchmaker. The results show that different types of middle agents exhibit different characteristics: The broker agent is more efficient than the other two if the elapsed time taken by a requester to satisfy a service objective is used as the efficiency performance attribute [1]. The matchmaker and blackboard with caching are significantly more robust than broker. The adaptiveness means that how well these three types of middle agents handle the effect of dynamically changing preferences and capabilities. An example of capabilities changes is the entry or exit of a new provider. The ability of an middle agent organization to quickly adapt to new preferences or capabilities is a function of the distance that the information has to travel, and the costs of keeping that information up-to-date [1]. The broker agent organization has better adaptiveness performance than the others.

### Application

Typically there are two different ways to adopt middle agents in a multi-agent system.

One is using a single agent and every user has a connection to this middle agent. Because the middle agent stores information about the capabilities of all the resources, it can easily provide appropriate services in response to user requirements [10]. This method may be efficient and economic in small multi-agent systems, like the auction agency system we designed as this course project, but it will lead to an inevitable bottleneck problem once the
number of resources and requirements has become too large, because the workload of this middle agent can easily be overloaded. Additionally, when the middle agent fails to work properly the whole multi-agent system will come to a halt. The sample applications of this method is as follows, SHADE [6], COINS [5], InfoSleuth [7], WARREN [8] and a self-building agent system in [9].

Another middle agent adoption method is to adopt several middle agents to mediate between providers and requesters, such as the Open Agent Middleware (OAM) of IDIoMS [11] and Abrose [12]. In such system, several middle agents is adopted and each of them takes charge of part of the resources. The middle agents cooperate with one another when the services required go beyond a middle agent’s capability [10]. By allocating system responsibility onto several agents, distributed middle agent can undertake services many times more substantial than one middle agent and be robust to certain failures [1]. However, in these multi-agent systems, the relationship between users and middle agents is always pre-defined by the human designer. In Abrose, there is certain flexibility between users, as users can maintain a list describing their preferred neighborhood, by enhancing or decreasing their confidence in other users [10]. However, this networking only happens in a local area controlled by the same middle agent. The allocation of a user to a middle agent is still a priori in Abrose. These middle agent applications show that there is often quite a considerable delay between finding a resource and supplying it to a requester. Consequently these distributed systems may become very unwieldy if the information held by users or other resources is widely distributed[10] [12].

The facilitator, which is one of agent variants known in the literature is adopted in SRI’s Open Agent Architecture [14]. The OAA is an architecture that makes it possible for software services to be provided through the cooperative efforts of distributed collections of autonomous agents. Facilitator in OAA is responsible for mediate the communication and cooperation between agents, such as matching requests, from users and agents, with descriptions of the capabilities of other agents. Thus, it is not generally required that a user or agent knows the identities, locations, or number of other agents involved in satisfying a request [14]. It is also good for protecting privacy of each individual agent.

**Open Issues**

A key issue concerning multi-agent system based on middle agents is how to organize users and middle agents so that the users can receive appropriate services quickly and efficiently [10]. The organization provides the fundamental basis for middle agents to supply and deliver suitable services targeting fitting users. So I think a good organization method is very important especially for large-scale, distributed, and dynamically changing environments so that sufficient services are provided even in such complicated environments. Therefore, the design and theoretical research of flexible organizations that reflect actual, real-time user preferences and capabilities is desired.

Matchmaking in middle agents is another crucial issue in multi-agent systems, especially those used in open environments [13]. In open environments, heterogeneous agents can appear and disappear dynamically, therefore, the quality and efficiency of matchmaking becomes challenging. It is paramount important to improve the matchmaking quality of middle agents with satisfying efficiency.
The Foundations for Intelligent Physical Agents (FIPA) made a lot of efforts for producing standards for the interoperation of heterogeneous software agents. I think middle agents design and research should not ignore FIPA specifications. For example, a middle agent could store the user agents descriptions and services based on the FIPA specification.

From my viewpoint, the last open issue exists in the taxonomy of middle agent. The taxonomy discussed in [3] is not sufficient to facilitate the selection of middle agents, some other characteristics should also be considered, such as interoperability, privacy, robustness, etc. The reason is as follows, when a multi-agent system adopts a middle agent, its interoperability may affect the integration process directly, and its privacy and robustness are the designers main concern when he/she needs to choose one type of middle agents for a multi-agent system.

Conclusion

A question is frequently asked in multi-agent systems concerning the efficient search for suitable agents to solve a specific problem, different types of middle agents are employed to answer this question. In this report, some commonly used middle agents are discussed and their process procedures are illustrated. Middle agent taxonomy and performance metrics are important aspects and still at their development stages, I think some more characteristics should be adopted, such as interoperability, privacy, robustness, and load-balancing. Two different methods are used to adopt middle agents in a multi-agent system, one is to use one single middle agent to connect with all other user agents, another is to use several middle agents that could work together to perform the matchmaking tasks. The applications of both methods are represented. Finally I summarized some open issues in this middle agent research area.

Reference:


[10]. F. Wang, Self-organizing Communities Formed by Middle Agents, AAMAS’02, July, 2002


