

A PRIVACY-BASED BROKERING ARCHITECTURE FOR COLLABORATION IN VIRTUAL ENVIRONMENTS

AbdulMusalib Masaud-Wahaishi¹, Hamada Ghenniwa¹ and Weiming Shen²

¹*Department of Electrical and Computer Engineering, University of Western Ontario*
amasaud@uwo.ca, hghenniwa@eng.uwo.ca

²*Integrated Manufacturing Institute, National research Council, CANADA*
weiming.shen@nrc.gc.ca

With the rapidly growing development of applications in open virtual enterprises, privacy is becoming a critical issue. This paper presents an agent-based architecture that provides coordination services, with special focus on capability-based integration as brokering services. These services take into consideration any privacy desires that may be required from various participants in an open, dynamic, and heterogeneous environment. A proof-of-concept prototype system has been implemented to support and provide information-gathering services in healthcare environments.

1 INTRODUCTION

A Virtual Enterprise (VE) is an organization that consists of multiple co-operating autonomous entities (enterprises) that are jointly act in a specified limited domain to fulfill a common enterprise mission. VEs are supported by geographical distribution and heterogeneous entities with no central control. Therefore, building VEs involve dealing with challenges that go beyond traditional integration approaches and design paradigms. The future success of building systems in terms of more sophisticated components, often-entire systems themselves, and integrating them requires an engineering and scientific basis that support high-level of abstraction for connection and interaction in VEs. Although these systems are independently created and administered, they need to collaborate and cooperate to fulfill a desired functionality or service.

A cooperative distributed systems (CDS) approach is a promising design paradigm that is suitable for many virtual enterprises applications; however, coordination is a major challenge in developing cooperative distributed systems in open environments. In a Cooperative Distributed Virtual Environments (CDVEs), entities usually need to work together to accomplish individual or social tasks. However, in open environments, this becomes a challenge where it is no longer feasible to expect designers or users to hardcode, to determine or to keep track of the entities and their capabilities. Brokering is a coordination and cooperation activity among heterogeneous entities in cooperative distributed systems environment that can be used effectively to support integration in VEs.

With the rapidly growing development of applications in open distributed environments, such as e-Business, privacy is becoming a critical issue. Consequently, distributed systems architects, developers and administrators are faced with the challenge of securing the requester's privacy as well as the provider's. In general, requesters and service providers are concerned about their privacy from different perspectives. For example, they may wish to protect their

identities from being used, or decide by whom it will be revealed, and for what purposes, or retain the choice about whether or not to reveal their personal interests or capabilities. This paper presents a novel privacy-based brokering architecture as a capability-based aspect of coordination in CDVEs. The architecture supports ad hoc configurations among distributed, possibly autonomous and heterogeneous entities with various degrees of privacy requirements in terms of three attributes: entity's identity, capability and preferences.

2 AGENT-BASED BROKERING ARCHITECTURE: PRIVACY MODEL

In addition to the dynamic nature of the environment, VEs usually involve complex and nondeterministic interactions, often producing results that are ambiguous and incomplete which requires that the components of these VEs be able to change their configuration to participate in different, often simultaneous roles.

These requirements could not be accomplished using traditional ways of manually configuring software. We strongly believe that agent technology is a very promising design for brokering in cooperative distributed VEs. Here we view agent-orientation as a metaphorical conceptualization tool at a high level of abstraction (knowledge level) that captures, supports and implements features that are useful for distributed computation in open environments. These features include cooperation, coordination, interaction, as well as intelligence, adaptability, economic and logical rationality. An agent is an individual collection of primitive components that provide a focused and cohesive set of capabilities⁷ [7].

Within this context, we model CDVEs as a distributed group of intelligent, goal oriented, autonomous, and rational agents that act in a cooperative and interactive manner to enable collaboration between various entities. Architecturally, the brokering is viewed as a layer of services where a brokering service is modeled as an agent with a specific architecture and interaction protocol that are appropriate to serve various requests. The architecture allows various entities in the domain to join and benefit from the functionalities provided by the brokering service through a registration and naming service that permits the addition or removal of any agents, service and sources at runtime. The brokering service uses the registration and naming service to build up a knowledge base of the environment in order to facilitate locating and identifying the relevant existing providers that are capable of serving a specific request.

2.1 The Privacy Model

CDVEs users and service providers are concerned about their personal privacy from different perspectives. For example, they may wish to protect their identities from being used, or decide by whom it will be revealed, and for what purposes, or retain the choice about whether or not to reveal their personal interests or capabilities. The objective of the work presented here is correlated mainly with tackling the privacy concerns from the perspective of preserving the identities and capabilities of the various entities that constitute CDVE. Here, we define the degree of privacy in terms of three attributes: the entity's identity, capability and goals. Therefore, an agent can categorize its role under several privacy degrees. Formally, an agent can be represented as a 2-tuple

$$Ag \equiv \langle (RA : Id, G); (PA : Id, Cap) \rangle$$

Where RA and PA refer to the agent role as requester, and provider respectively, Id, G, and Cap refer to the agent’s identity, goals and capabilities, which might have a null value. For example an agent might participate with a privacy degree that enables the hiding of its identity as a requester by setting the value of (Id) to null. The challenge in this context is how to architect the brokering layer with the appropriate set of services that enable cooperation across the different degrees of privacy. The interaction protocols represent both message communication and the corresponding constraints on the content of messages. Table 1 summarizes the different brokering scenarios categorized by the possible combination of privacy attributes of the Requester and Provider agents.

Table 1 Brokering Layer Interaction Patterns with Service Requestors

Privacy Attributes					Interaction Patterns	
Case	g(RA)	Id(RA)	Id(PA)	Cap(PA)	With Requesters	With Providers
1	Revealed	Revealed	Revealed	Revealed	Receive request Deliver result	Search for relevant agents, Negotiate and Obtain result
2	Revealed	Revealed	Revealed	Hidden	Receive request Deliver result	Advertise request to known PA
3	Revealed	Revealed	Revealed	Hidden	Receive request Deliver result	PA check for requests PA to reply with results
4	Revealed	Hidden	Provider’s privacy attributes are either one of the status shown in (1,2, or 3)		Receive request RA retrieve result	same interaction protocol depicted in any selected case shown in (1,2, or 3)
5	Hidden	Known			Advertise services to RA	
6	Hidden	Hidden			RA to check for services.	

2.2 The Brokering Interaction Patterns

Requesters Revealing identities and goals:

Agents playing the role of requestors are required to reveal their identities, and goals to the relevant broker within the layer. The interaction pattern is as follow: Formulating service’s requests, Contacting a set of provider-side brokers, forwards, and controls appropriate transaction to achieve the requester’s goal; and receive and returns the result of the services to the requester agent as shown in Figure 1.

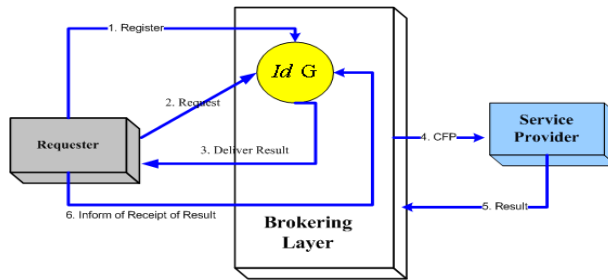


Figure 1 Interaction Pattern for Requesters Revealing Privacy Attributes

Requesters hiding identities

Requesters may wish to access services or seek further assistance without revealing their identities. As shown in Figure 2, requesters are responsible of checking the availability of the service’s result. The interaction imposes a significant effort on the performance and efficiency.

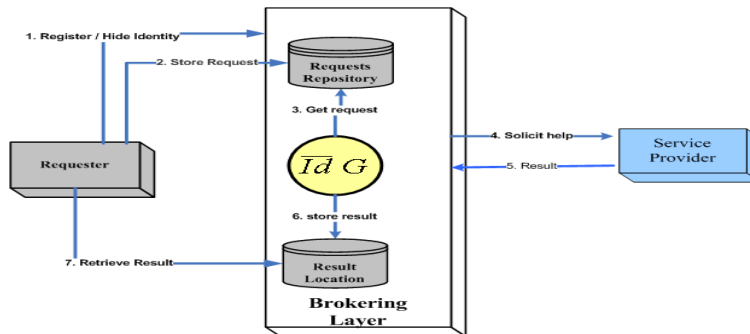


Figure 2 Interaction Pattern for Requester Hiding Identity

Requesters hiding goals:

The layer permits those requestors to check a service’s repository for further information or to browse service offerings that have been previously posted as shown Figure 3. The interaction is not restricted only to those service providers who had initiated such advertisements, but it extends to those providers with unknown capabilities, since there might be the case where the same advertised service can be accomplished and achieved as well (for example with better quality or cost).

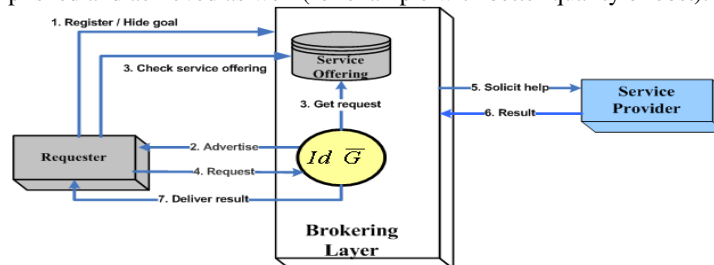


Figure 3 Interaction Pattern for Requestor Hiding Goals

Requesters hiding identities and goals

Requesters would have the possibility to hide their identities and goals from the entire environment; as shown in Figure 4, they have the option either to post their want-ads to the layer’s service repository directly, or might check services that would be of an interest. Requesters are responsible to check for the availability of the service’s result and hence retrieve it. It is to be noted that a requester with this degree of privacy will have no further dynamic notification in case the service directory becomes unavailable.

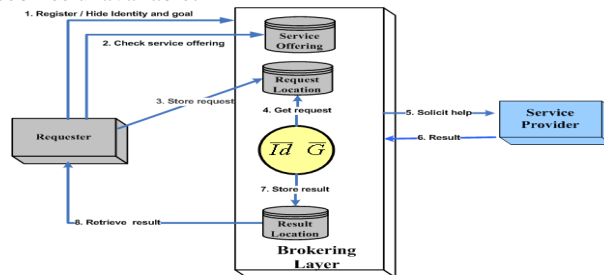


Figure 4 Interaction Pattern for Requestor Hiding Privacy Attributes

Providers revealing identities and capabilities

The brokering layer issues a call-for-proposals (CFP) to available providers informing them of the problem’s specifications. Each potential contractor (provider) determines the evaluation parameters (such as goal quality, goal expiration time, and cost) and accordingly submits a bid to the brokering agent, or might reject the proposal. After receiving the bids, the layer selects the most appropriate bid that satisfies with the request’s parameters.

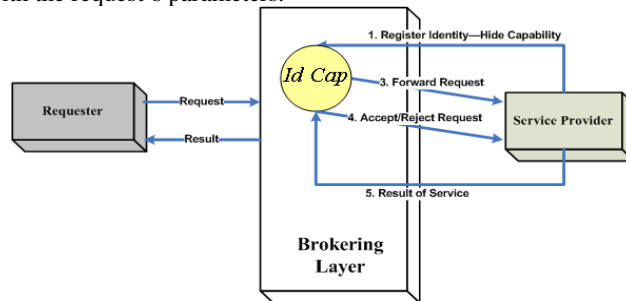


Figure 5 Interaction Pattern for Provider Revealing Privacy Attributes

Providers hiding capabilities

After receiving a request, the layer’s interaction protocol with providers will forms requests out to every registered provider with unknown capability.

It is noteworthy that, for every advertised request, providers need to determine whether the request is within its capabilities and/or of an interest, which implies that a considerable elapsed time will be spent on evaluating every single request. Therefore, providers would be deluged by a variety of service requests, which significantly impact performance and efficiency. Figure 6 shows the corresponding interaction pattern.

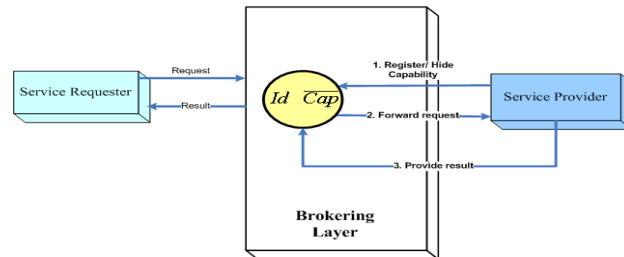


Figure 6 Interaction Pattern for Provider Hiding Capability

Providers hiding identities and capabilities

As shown in Figure 7, the brokering layer's functionality is mainly seen as a directory service, in which a brokering agent maintains a repository of service's requests along with any required preferences. Providers will have the ability to browse this repository and accordingly determine relevant requests that might be of an interest and within their capabilities.

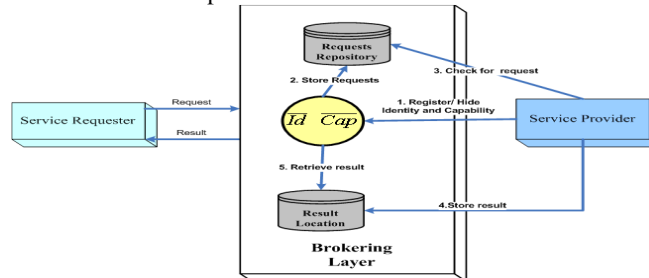


Figure 7 Interaction Pattern for Provider Hiding Privacy Attributes

3 IMPLEMENTATION GUIDELINES

Due to page limitation, we describe the implementation guidelines of one pattern associated with a requestor and provider; both are hiding their identities. The protocol is represented using the FIPA-ACL performatives. All interactions utilize the Contract Net Protocol as a negotiation mechanism. The brokering agent (acts as a manager) issues a call-for-proposals (CFP) to those all available registered providers (under different privacy degrees) in the environment informing them of the problem specifications. The protocol is represented as follow:

- Requester to store: **(Request)** – a service request is stores in the service location (repository)
- Broker to **Query-If** for posted services' requests
- Broker to store the **(CFP)** in a special repository to be browse by the Provider
- Provider to store **(Propose)**, a service proposal for a particular request or **(Refuse)** – provider might decline the service request
- Broker to **Query-If** – for possible service proposals
- Broker to store (for potential providers)
- **(Accept-Proposal)** – an acceptance notification or **(Reject-Proposal)** – or a rejection message is stored in case of not winning
- Provider to **Query-If** for acceptance/rejection of the propose service offerings, **Accept-Proposal** or for **(Reject-Proposal)**

- Store(**Confirm**) – Notification of service offering confirmation by the provider
- Store(**Disconfirm**) or Store(**Cancel**) – The Provider stores a cancellation message
- Provider to store (**Inform**) indicating the availability of a service's result
- Broker to **Query-If** for the availability of the service's result
- Broker to retrieve the service' result and to store (**Inform-Done**) indicating the receipt of the result
- Provider to store (**Inform**) (for requester) indicating the availability of a service's result
- Requester to **Query-If** for the received result, retrieve it and to store (**Inform-Done**)
- Broker to **Query-If** for (**Inform-Done**)→ end of the protocol

A web-based prototype of the proposed system has been implemented using Jade [8] to support and provide information-gathering capabilities to different participants in healthcare environments. Healthcare services are modeled and implemented as CDVEs that consist of a collection of autonomous units that act independently and collaborate in providing services and synergize medical data according to mutual privacy interests.

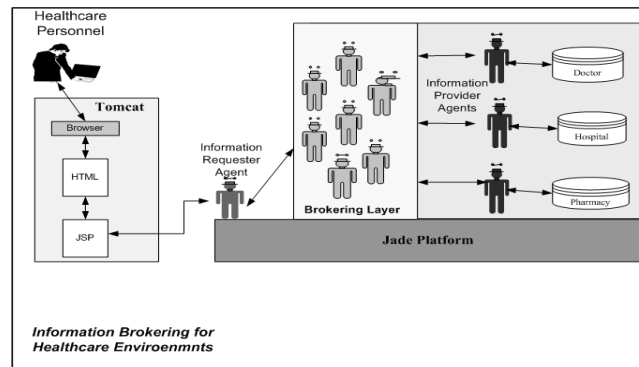


Figure 8 Prototype Architecture

As shown in Figure 8, three relational databases represent various medical data for three distributed locations (hospital, doctors' records, and a pharmacy), each being managed by a dedicated agent (provider). Different brokering agents with different roles are available to domain-specific agent that might play the role of a service provider as well as the role of a requester. Domain agent (requesters and providers) need to register with the brokering layer for which the layer identifies the relevant brokering agent that is appropriate to support the required privacy degree.

4 RELATED WORK AND DISCUSSION

Different techniques were used to enable collaboration in virtual environments including brokering via middle agents [4]. The work in [5] has proposed agent-based mediation approach, in which privacy has been treated as a base for classifying the various mediation architectures only for the initial state of the system. In another approach, agents' capabilities and preferences are assumed to be common knowledge, which might violate the privacy requirements of the participants. Other approaches such as in [10][11][9] have proposed framework structures to facilitate coordination between web services by providing semantic based discovery and mediation services that utilize semantic description languages. The work in [6] reviewed and classified the functional requirements that are needed for collaboration

in networked virtual organizations. Another recent approach distinguishes a resource brokering architecture that manages and schedules different tasks on various distributed resources on the large-scale Grid [2]. Other initiatives proposed the use of privacy policies along with physical access means (such as smartcards), in which the access of private information is granted through the presence of another trusted authority that mediate between information requesters and information providers in virtual organizations [1]. However, none of the above mentioned approaches has treated privacy as an architectural element that facilitates the collaboration between various entities of cooperative distributed virtual enterprises.

5 CONCLUSION

In this paper we have developed an agent-based brokering architecture that provides seamlessly integrated collaboration and cooperation for different entities within distributed virtual environments. Unlike the traditional brokering, the brokering role is further classified into several sub-roles categorized by the desired privacy attributes of service requesters and providers. Each brokering role is modeled as an agent with a specific architecture and interaction protocol that is appropriate to fulfill a required privacy degree.

The proposed approach is innovative, in the sense that it treats the privacy as a design issue for brokering systems. Different application domains can benefit from the proposed model, such as intelligent cooperative information systems, and agent-based electronic business. The opportunities for exploiting the proposed architecture in building collaborative virtual enterprise are enormous. Based on the level and the amount of information that can be released, virtual enterprise members can securely collaborate while translating their privacy concerns to an applicable privacy degree that suits their desires and objectives.

6 REFERENCES

1. Amin T., Keng P. H.: "Inter-organizational Workflow Management System for Virtual Healthcare Enterprise ", 3rd IFIP Working Conference on Infrastructures for Virtual Enterprises PRO-VE'02, Portugal, May 1-3, 2002.
2. Brook, J. and Fellows, D. "An Architecture for Distributed Brokering on the Grid", 11th International Euro-Par Parallel Processing - Portugal, 2005.
3. Clarke R., "Identification, Anonymity and Pseudonymity in Consumer Transactions: A Vital System Design and Public Policy Issue": at: <http://www.anu.edu.au/people/Roger.Clarke/DV/AnonPsPol.html> , available as of Feb, 2007
4. Decker K, Sycara K. , and M. Williamson , " Middle-agents for the internet" In IJCAI97 International Joint Conference on Artificial Intelligence, Nagoya, Japan, 1999
5. Decker K., et al., "MACRON: An Architecture for Multi-agent Cooperative Information Gathering", Proceedings of the CIKM '95 Workshop on Intelligent Information Agents, 1998
6. Ellman S. and Eschenbaecher J. "Collaborative Network Models: Overview and Functional Requirements" Virtual Enterprise Integration: Technological and Organizational Perspectives, IDEA Group, Inc, imprints 2005.
7. Ghenniwa, H. and Huhns, M., "Intelligent Enterprise Integration: eMarketplace Model", Creating Knowledge Based Organizations, Idea Group Publishing, Pennsylvania, USA, pp. 46-79, 2004.
8. Java Agent Development Framework: Jade, Home Page: <http://jade.cse.ust.hk/>
9. Li, L. and Horrocks I., "A Software Framework for Matchmaking Based on Semantic Web Technology", In Proceedings of the 12th International Conference on WWW, Hungary 2003.
10. Motta, E, Domingue, J, Cabral, L, and Gaspari, M. "IRS-II: A Framework and Infrastructure for Semantic Web Services" In Proc. of the International Semantic Web Conference, SA, 2003.
11. Paolucci, M. et al. "Semantic Web Services", AAAI Spring Symposium Series - March, 2004