THOMAS: A Service-Oriented Framework For Virtual Organizations

E. del Val, N. Criado, C. Carrascosa, V. Julián, M. Rebollo, E. Argente, V. Botti

ABSTRACT

This paper describes the THOMAS framework, a useful framework for the development of virtual organizations, on the basis of a service-based approach.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multi-agent Systems

Keywords

Service-Oriented Architecture, Open Multi-Agent Systems, Virtual Organizations

1. SERVICE ORIENTED VIRTUAL ORGANIZATIONS

Virtual Organizations (VOs) are a set of individuals and institutions that need to coordinate resources and services across institutional boundaries [1]. Thus, they are open systems formed by the grouping and collaboration of heterogeneous entities and there is a separation between form and function that requires defining how a behaviour will take place. As a consequence of the heterogeneity feature, organizational concepts are needed for achieving the coordination. In addition, system functionalities should be modelled as services in order to allow heterogeneous agents or other entities to interact in a standardized way. The integration of MAS and Service Technologies has been proposed as the basis for these new and complex systems [3].

The present demo represents a step forward on this direction, providing a service-oriented framework specifically designed for the execution of VOs and based on the THOMAS architecture [2].

2. THOMAS FRAMEWORK

The THOMAS framework allows any agent to create a VO with the structure and norms needed, along with the demanding and offering services required. Its purpose is to obtain a product wholly independent of any internal agent platform, thus fully addressed for open systems. The framework is in charge of the management of the organization structure, norms and life cycle, as well as controlling the visibility of the offered and demanded services and the fulfilment of the conditions to use them. All the functionalities of the framework are offered as semantic web services which are classified into two different entities: the *Organization Management System* (OMS) and the *Service Facilitator* SF.

Next, a general description of how services have been implemented in the THOMAS framework is presented. The specific implementation of both OMS and SF services and how these OMS and SF entities are encapsulated into agents to be accessible by other agents are also described.

THOMAS Services are implemented as semantic web services, using Apache Axis2/Java© as core engine for web services. More over, the use of RT-Java for providing time-bounded web services has been tested [4]. Each service has a semantic description in OWL-S© which details the properties and capabilities of a web service in unambiguous, computer-interpretable form. It facilitates the automation of web service tasks, including automated service discovery, execution, composition and interoperability.

3. AGENT IMPLEMENTATION

Both the OMS and SF functionalities have been implemented as semantic services and can be accessed using web service standards. In addition, OMS and SF agents, which are mainly gateways to the OMS and SF services, have been implemented in order to allow JADE™ agents to access the framework's services. The logic of these agents has been implemented using the OWL-S API© provided by Mindswap. This OWL-S API employs a Java API for programming access to read, execute and write OWL-S service descriptions. When the SF agent (or the OMS agent) receives a FIPA request message from a client, it employs the OWL-S API to access the service description in OWL-S and execute the correspondent web service.

SF Implementation. The SF has to be able to register and manage the services provided by external agents. In order to make these services machine-understandable, semantic information is added as OWL-S descriptions for services and ontology specification in OWL© for agent beliefs interchange and the specification of service parameters. Thus, tools to handle semantic information have been added to the SF. JENA© and SPARQL© have been employed to manage all these semantic data in OWL. JENA provides a programmatic environment for RDF, RDFS and OWL, as well as an implementation of the RDF model interface that stores the triples persistently in a database. SPARQL has been used as language that queries the information held in the models

Cite as: THOMAS: A Service-Oriented Framework For Virtual Organizations, E. del Val, N. Criado, C. Carrascosa, V. Julián, M. Rebollo, E. Argente and V. Botti, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1631-1632

stored in the database.

OMS Implementation. The set of services provided by the OMS are related to the organization life-cycle management as well as the organizational dynamics. In addition, informative services for allowing agents to obtain the organizational description have been defined. The implementation of these services should consider the maintenance of the organizational state as well as the existence of norms that regulate their access. In this sense, before providing any service, the current normative context should be checked in order to determine whether this service provision is allowed or not. On the other hand, after the provision of a service, the OMS should update the organizational state. These two functionalities have been implemented as internal services of the OMS, by means of a rule-based system in Jess® that maintains a fact base representing the organizational state and it also detects norm activation. Thus, the *update* service is in charge of adding and deleting facts into the rule system in order to register the organizational current state. The determination of the allowed actions is made by means of the analysis of the normative context. It has been performed as an internal *query* service offered by the normative manager to the OMS services.

4. APPLICATION EXAMPLE

In order to illustrate the performance of the THOMAS framework with greater detail, a case-study example for making flight and hotel arrangements has been implemented. The *Travel Agency* example is an application that facilitates the interconnection between clients (individuals, companies, travel agencies) and providers (hotel chains, airlines); delimiting services that each one can offer and/or request.

This case study is modelled as an organization, in which two kinds of roles can interact: customer and provider. The *Customer* role requests travel services of the system. The *Provider* role is in charge of performing these travel services. Thus, a *provider* agent offers hotel or flight search services and can also offer booking hotel rooms or flight seats.

This example illustrates how the employment of the Web Service standards for both providing and publishing services allows external agents to discover and make use of both agent and THOMAS functionalities. The system dynamics is shown through the specification of a scenario in which an external agent joins the THOMAS framework, registers itself as a client and carries out a service request (Figure 1).

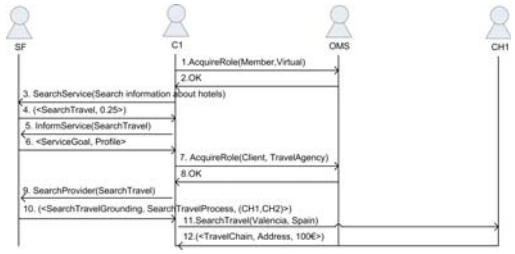

Figure 1: Client Registering

C1 is an agent that represents a client interested in hotel information. External agents join the THOMAS framework by requesting OMS to be a *Member* of the *Virtual* organization using the *AcquireRole* service (Figure 1, message 1). Then C1 requests the *SearchService* to the SF component so as to find services of its interest (message 3). The result of this service is shown in message 4. Then, C1 employs *InformService* in order to know inside which organizational unit this *SearchTravel* service is provided (messages 5 and 6). As indicated in the service profile, C1 must acquire the *Customer* role to demand this service (messages 7 and 8). Then, it employs the *SearchProvider* service to know who the service providers are and how this service can be requested (messages 9 and 10). As shown in message 10, the *SearchTravel* service has one implementation, with two different providers (CH1 and CH2). C1 chooses to make a service petition to CH1 agent, so then according to this service process, C1 sends a message for requesting this *SearchTravel* service to CH1 agent (messages 11 and 12).

5. CONCLUSIONS

This paper contains a general overview of the THOMAS framework and illustrates how an external agent is capable of discovering and employing functionalities in THOMAS. The main contributions of the THOMAS framework are: i) it gives support to VOs as a social abstraction for coordinating the autonomous behaviours of agents; ii) system functionalities are described and provided as *services* in order to make possible the interaction among heterogeneous entities in a standardised way; iii) it provides *discovering* and *composition* services for allowing external agents, which have been designed independently of the THOMAS framework, to participate inside it; iv) it provides mechanisms, i.e. services, for the adaptation and modification of both the structure and functionality of the organization in response to the environmental changes. In this sense, it allows agents to reorganize the system dynamically.

A detailed description of the THOMAS architecture and the first version of this framework are available in the THOMAS homepage[1].

Acknowledgment

This work is supported by TIN2009-13839-C03-01 project, CONSOLIDER-INGENIO 2010 under grant CSD2007-00022, FPU grants AP-2007-01256 awarded to N.Criado and AP-2008-00601 awarded to E. del Val.

6. REFERENCES

[1] G. Boella, J. Hulstijn, and L. W. N. van der Torre. Virtual organizations as normative multiagent systems. In *HICSS*, 2005.

[2] C. Carrascosa, A. Giret, V. Julian, M. Rebollo, E. Argente, and V. Botti. Service oriented mas: An open architecture. In *AAMAS*, pages 1291–1292, 2009.

[3] M. Luck and P. McBurney. Computing as interaction: Agent and agreement technologies. In *IEEE SMC Conf. Distributed Human-Machine Systems*, pages 1–6, 2008.

[4] E. D. Val, M. Navarro, V. Julian, and M. Rebollo. A framework to guarantee time-bounded composed services. In *The 2009 IEEE / WIC / ACM International Conferences on Web Intelligence*, pages 434–437. IEEE Computer Society, 2009.

[1] http://www.dsic.upv.es/users/ia/sma/tools/Thomas