A Re-usable Broker Agent Architecture with Dynamic Maintenance Capabilities

Article · June 1999

DOI: 10.1145/301136.301242 · Source: CiteSeer

2 authors:

Catholijn M. Jonker
Delft University of Technology
543 PUBLICATIONS **6,371** CITATIONS
SEE PROFILE

Jan Treur
Vrije Universiteit Amsterdam
778 PUBLICATIONS **7,853** CITATIONS
SEE PROFILE

A Re-usable Broker Agent Architecture with Dynamic Maintenance Capabilities

Catholijn M.Jonker, Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
Email: {jonker, treur}@cs.vu.nl
URL: http://www.cs.vu.nl/~{jonker, treur}

Keywords: broker agent, dynamic maintenance, reusable

1. INTRODUCTION

To support users on the World Wide Web, various types of agents can be, and actually have been, developed. For example, to support broker processes in electronic commerce, personal assistant agents can be developed that support a user offering products (or services) at the Web, or agents that support search for information on products within a user's scope of interest, or agents that combine both functionalities. In general, applications in this area are implemented in an ad hoc fashion without an explicit design at a conceptual level, and without taking into account the dynamic requirements imposed by the domain of application and the maintenance problem (including desired extension or modification of functionality) implied by this dynamic character.

The agent architecture described in this paper can be instantiated by adding specific types of knowledge to support functionalities and behaviour required. Depending on the choice of these requirements, an agent is created for a specific context by including the appropriate types of knowledge. For example, a search agent with functionality restricted to (incidental) search for information upon a user's request can be built by adding only knowledge needed for this task. Such an agent, for example, is not able to store and maintain the user's query or information that has been found, nor is it able to provide information to other agents. If these functionalities are required as well, the necessary types of knowledge have to be added. The agent architecture introduced here supports its own modification due to the fact that basic functionalities are specified in an explicit declarative manner, in the form of knowledge. It is possible to dynamically modify the agent by adding or deleting some of its knowledge. Since this declaratively represented knowledge can be the subject of communication between agents, modification can be performed at a distance: another agent (e.g., a dedicated maintenance agent) communicates the knowledge needed for the modification to the agent that is to be modified.

Autonomous Agents'99 Seattle WA USA

Thus a flexible agent implementation is obtained that can be maintained and evolve over time on the basis of communication with other agents only.

2. DESIGN OF THE GENERIC BROKER AGENT

For the design of the generic broker agent a compositional generic agent model (introduced in [1]), supporting the weak agency notion (cf. [3]) is used. Within the agent, a number of processes can be distinguished. First, a process that manages communication with other agents, modelled by the component agent interaction management. This component analyses incoming information and determines which other processes within the agent need the communicated information. Moreover, outgoing communication is prepared. Next, the agent needs to maintain information on the other agents with which it co-operates (e.g., their scopes of interests): maintenance of agent information. The component maintenance of world information is included to store the information on world information (e.g., information on attributes of products). The process own process control defines different characteristics of the agent and determines foci for behaviour. The component world interaction management is included to model interaction with the world (e.g., with the World Wide Web world): initiating observations and receiving observation results. The agent processes discussed are generic agent processes. Many agents perform these processes. In addition, often agent-specific processes are needed: to perform tasks specific to one agent, for example directly related to a specific domain of application. In the current example the agent has to determine proposals for other agents. In this process information on available products (communicated by information providing agents and kept in the component maintenance of world information), and about the scopes of interests of agents (kept in the component maintenance of agent information), is combined to determine which agents might be interested in which products. For the broker agent this agent-specific task is called determine proposals.

For reasons of space limitation the generic and domain-specific information types within the agent model are not presented; for more details; see [2].

3. GENERIC AND DOMAIN SPECIFIC KNOWLEDGE

For each of the components of the agent architecture, knowledge bases have been developed:

- strict match kb
- agent interest identification kb
- agent interest maintenance identification kb
- subscription identification kb
- provider info identification kb
- provider info maintenance identification kb
- provider scope maintenance identification kb
- proposal communication kb
- info provider request kb
- focus kb
- observation info identification kb
- observation maintenance identification kb
- observation initiative kb.
- attribute and scope kb
- provider scope abstraction kb
- product scope abstraction kb

4. THE BEHAVIOUR

The following basic functionalities (depending on the required types of behaviour) have been identified:

1. Observation of information available within a certain part of the world; e.g., a specified area within the Web, such as a specific Web-site or set of Web-sites.
2. Communication with agents asking for information on products.
3. Communication with agents providing information on products.
4. Maintenance of acquired information on products.
5. Maintenance of scopes of interest of (other) agents.
6. Maintenance of scopes of products (other) agents can provide.
7. Own control
8. Determining matches between products and scopes of interests.

These functionalities have been specified by knowledge bases that can be used within the agent components. Combinations of these functionalities define specific types of agents. For example, if a provider agent is designed, functionalities 2., 4., 5., 7. may be desired, whereas functionalities 1., 3., 6. could be left out of consideration. If an agent is designed to support a user in finding information on products within a certain scope, functionalities 1., 3., 6., 7. (and perhaps 4.) may be desired, whereas 2. and 5. may be less relevant. For a mediating agent, or for an agent that has to play different roles, almost all functionalities (i.e., 2. to 7.) may be desired.

The generic agent architecture can be instantiated in different manners to obtain, among others, the types of agents mentioned. The relation between the agent's basic functionalities, its knowledge, and where the knowledge is used is summarized in the table below.

basic functionality	*knowledge specifying functionality*	*in comp*
1. observation	observation initiative kb	WIM
	observation info identification kb	WIM
2. communication with agents asking for information	agent interest identification kb	AIM
	subscription identification kb	AIM
	proposal communication kb	AIM
3. communication with agents providing information	provider info identification kb	AIM
	provider scope identification kb	AIM
	provider request kb	AIM
4. maintenance of product information	observation info maintenance identification kb	WIM
	provider info maintenance identification kb	AIM
5. maintenance of scopes of interest	agent interest maintenance identification kb	AIM
6. maintenance of scopes of products agents can provide	provider scope maintenance identification kb	AIM
	provider scope abstraction kb	MAI
	attribute and scope kb	MAI
7. own control	focus kb	OPC
8. matching between product characteristics and scopes of interests	attribute and scope kb	DP
	product scope abstraction kb	DP
	strict match kb	DP

5. MAINTENANCE BY COMMUNICATION

A sketch of the process of manitenance by communication is as follows:

- A user determines that a broker agent needs more functionality, e.g., maintenance of product information.
- This user communicates this need to broker agent .
- The broker agent communicates this need to the maintenance agent.
- The maintenance agent determines which knowledge is needed for the required functionality (in the example case: observation info maintenance identification kb and provider info maintenance identification kb), and in which components this knowledge should be placec (in the example case: WIM and AIM respectively).
- The maintenance agent communicates to the broker agent: the needed knowledge bases, and in which components the knowledge bases have to be placed, with respect to the required functionality.
- The broker agent places the communicated knowledge bases in its respective components and communicates to the user that the required functionality is now available.

For more details and further references, see [2].

REFERENCES

[1] BRAZIER, F.M.T., JONKER, C.M., TREUR, J., Formalisation of a cooperation model based on joint intentions. In: J.P. Müller, M.J. Wooldridge, N.R. Jennings (eds.), Intelligent Agents III (Proc. of the Third International Workshop on Agent Theories, Architectures and Languages, ATAL'96), Lecture Notes in AI, volume 1193, Springer Verlag, 1997, pp. 141-155.

[2] JONKER, C.M., TREUR, J., Compositional Design and Maintenance of Broker Agents. In: J. Cuena (ed.) *IT & KNOWS, Information Technologies and Knowledge Systems, Proceedings of the XVth IFIP World Computer Congress*, 1998, pp. 319-332. http://www.cs.vu.nl/~wai/Papers/ITK98.broker24.ps

[3] WOOLDRIDGE, M., JENNINGS, N.R., Agent theories, architectures, and languages: a survey. In: WOOLDRIDGE, M., JENNINGS, N.R. (eds.), Intelligent Agents, (Proc. of the First Int. Workshop on Agent Theories, Architectures and Languages, ATAL'94), Lecture Notes in AI, vol. 890, Springer Verlag, 1995, pp. 1-39.