Applied Intelligence 15, 7–24, 2001

A Reusable Multi-Agent Architecture for Active Intelligent Websites

CATHOLIJN M. JONKER*, REMCO A. LAM AND JAN TREUR†
Department of Artificial Intelligence, Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
jonker@cs.vu.nl
treur@cs.vu.nl

Abstract. In this paper a reusable multi-agent architecture for intelligent Websites is presented and illustrated for an electronic department store. The architecture has been designed and implemented using the compositional design method for multi-agent systems DESIRE. The agents within this architecture are based on a generic information broker agent model. It is shown how the architecture can be exploited to design an intelligent Website for insurance, developed in co-operation with the software company Ordina Utopics and an insurance company.

Keywords: intelligent website, information agent

1. Introduction

Most current business Websites are mainly based on navigation across hyperlinks. A closer analysis of such conventional Websites reveals some of their shortcomings. For example, customer relationships experts may be disappointed about the unpersonal treatment of customers at the Website; customers are wandering around anonymously in an unpersonal virtual environment and do not feel supported by anyone. It is as if customers are visiting the physical environment of a shop (that has been virtualised), without any serving personnel.

Marketing experts may also not be satisfied by the Website; they may be disappointed in the lack of facilities to support *one-to-one marketing*. In a conventional Website only a limited number of possibilities are provided to announce new products and special offers in such a manner that all (and only) relevant customers learn about them. Moreover, often Websites do not acquire information on the amounts of articles sold (sales statistics). It is possible to build in monitoring facilities with respect to the amount of products sold over time, but also the number of times a request is put forward on a product (demand statistics). If for some articles a decreasing trend is observed, then the Website could even advice employees to take these trends into account in the marketing strategy. If on these aspects a more active role would be taken by the Website, the marketing qualities could be improved.

The analysis from the two perspectives (marketing and customer relationships) suggests that Websites should become more active and personalised, just as in the traditional case where contacts were based on humans. Intelligent agents provide the possibility to reflect at least a number of aspects of the traditional situation in a simulated form, and, in addition, enables to use new opportunities for, e.g., one-to-one marketing, integrated in the Website.

The generic agent-based architecture presented in this paper offers these possibilities. This generic architecture for active intelligent Websites was first introduced for the application domain of a department store, which has been analysed in co-operation with the software company CMG (cf. [22]). It reuses the generic architecture of information broker agents developed earlier (cf. [21]), which in turn was designed as a specialisation of the generic agent model GAM introduced in [8]. As a second step the reusability of the

*URL: http://www.cs.vu.nl/∼jonker
†URL: http://www.cs.vu.nl/∼treur

generic multi-agent architecture for active intelligent Websites has been tested by applying it in a project on an intelligent Website for insurance in co-operation with the software company Ordina Utopics and an insurance company (cf. [20]). The testbed chosen for this application involves information and documents that need to be exchanged between insurance agents and the insurance company main office. The goal of the intelligent Website is to provide insurance agents with an accurate account of all relevant available documents and information. The supporting software agents are able to provide a match (either strict or soft) between demand and available information. They support proactive information provision, based on profiles of the insurance agents that are dynamically constructed. A prototype system for this application is described in more detail in the second part of the paper.

In this paper in Section 2 the global design of a multi-agent architecture for an intelligent Website is presented; the different types of agents participating in the Website are distinguished. In Section 3 their characteristics and required properties are discussed. In Section 4 the compositional generic information broker agent architecture is described and applied to obtain the internal structure of the agents involved in the multi-agent architecture. In Section 5 the insurance application domain is introduced. In Section 6 the application of the architecture to insurance is discussed in more detail and illustrated by some example behaviour patterns. Section 7 concludes the paper by a discussion.

2. A Generic Multi-Agent Architecture for Intelligent Websites

In this section a global multi-agent architecture, that can be used as a basis for an intelligent Website, is introduced. Although the architecture is generic, for reasons of presentation some of its aspects will be illustrated in the context of the applications.

The domain has been identified as a multi-agent domain. Therefore, it makes sense to start with the agents as the highest process abstraction level within the system. Four classes of agents are distinguished at the level of the multi-agent system (see Fig. 1):

- *customers* (human agents),
- *Personal Assistant agents* (software agents, denoted by PA),
- *Website Agents* (software agents, denoted by WA),
- *employees* (human agents).

In Fig. 1, the shaded area at the right hand side shows the agents related to the Website; the shaded area at the left hand side shows the two agents at one of the customer sites. In this figure, for shortness only two Website Agents, one employee, one Personal Assistant agent and one customer (user of the Personal Assistant) are depicted. Moreover, for the sake of simplicity, the Website itself is left out of the picture. The Website has the role of the external world for the agents; note that is not considered an agent itself. All agents can have interaction with this external world to perform

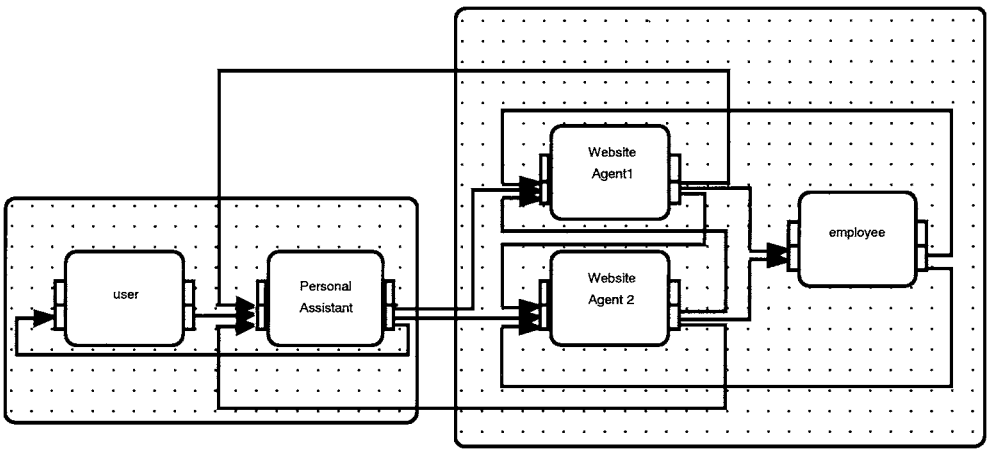

Figure 1. The overall multi-agent architecture.

observations. The Website agents and employees can also perform actions in this world, e.g., to change the information on one of the Webpages.

Note that the Personal Assistant is involved as a mediating agent in all communication between its own user and all Website Agents. From the user it can receive information about his or her interests and profile, and it can provide him or her with information assumed interesting. Moreover, it can receive information from any of the Website Agents, and it can ask them for specific information. The Website Agents communicate not only with all Personal Assistants, but also with each other and with employees. The customer only communicates with his or her own Personal Assistant. This agent serves as an interface agent for the customer. If a customer visits the Website for the first time this Personal Assistant agent is instantiated and offered to the customer (during all visits).

The first application domain to illustrate the architecture addresses the design of an active, intelligent Website for a chain of department stores. The system should support customers that order articles via the Internet. Each of the department stores sells articles according to departments such as car accessories, audio and video, computer hardware and software, food, clothing, books and magazines, music, household goods, and so on. Each of these departments has autonomy to a large extent; the departments consider themselves small shops (as part of a larger market). This suggests a multi-agent perspective based on the separate departments and the customers. For each department in the department store a Website Agent can be designed, and for each customer a Personal Assistant agent serves as an interface agent.

3. Requirements for the Software Agents

The departments should relate to customers like small shops with personal relationships to customers. The idea is that customers know at least somebody (a Website Agent) related to a department, as a representative of the department and, moreover, this agent knows specific information on the customer.

3.1. Characteristics and Requirements for the Website Agents

Viewed from outside the basic agent behaviours *autonomy*, *responsiveness*, *pro-activeness* and *social behaviour* such as discussed, for example, in [38] provide a means to characterise the agents (see Table 3). Moreover, the following external agent concepts to define interaction characteristics are used:

Table 1. World interaction characteristics for a Website Agent.

Website Agent—Interaction with the world		
observation	passive	– its own part of the Website
		– product information
		– presence of customers/Personal Assistants visiting the Website
observation	active	– economic information
		– products and prices of competitors
		– focusing on what a specific customer or Personal Assistant does
		– search for new products on the market
performing actions		– making modifications in the Website (e.g., change prices)
		– showing Web-pages to a customer and Personal Assistant
		– creating (personal or general) special offers
		– modification of assortment

- *interaction with the world* (observation, action performance)
- *communication with other agents*

In Tables 1 and 2 the interaction characteristics for the Website Agents have been specified and illustrated for the case of the department store.

The following requirements have been imposed on the Website Agents:

- *personal approach; informed behaviour with respect to customer*
  In the Website each department shall be represented by an agent with a name and face. Furthermore, some of these agents (those who have been in contact with the customer) know the customer and his or her characteristics, and remember what this customer bought previous times.
- *being helpful*
  Customers entering some area of the Website shall be contacted by the agent of the department related to this area, and asked whether he or she wants some help. If the customer explicitly indicates that he or she only wants to look around without getting help, the customer shall be left alone. Otherwise, the agent

Table 2. Communication characteristics for a Website Agent.

Website Agent—Communication	
incoming	*from Personal Assistant*:
	– request for information
	– request to buy an article
	– paying information
	– customer profile information
	– customer privacy constraints
	from employee:
	– requests for information on figures of sold articles
	– new product information
	– proposals for special offers and price changes
	– confirmation of proposed marketing actions
	– confirmation of proposed assortment modifications
	– proposals for marketing actions
	– proposals for assortment modifications
	from other Website Agent:
	– info on assortment scopes
	– customer info
outgoing	*to Personal Assistant*:
	– asking whether Website Agent can help
	– providing information on products
	– providing information on special offers
	– special (personal or general) offers
	to employee:
	– figures of articles sold (sales statistics)
	– analyses of sales statistics
	– numbers of requests for articles (demand statistics)
	– proposals for special offers
	– proposals for assortment modifications
	to other Website Agent:
	– info on assortment scopes
	– customer info

Table 3. Basic types of behaviour of a Website Agent.

Website Agent—Basic types of behaviour	
Autonomy	– functions autonomously, especially when no employees are available (e.g., at night)
Responsiveness	– responds to requests from Personal Assistants
	– responds to input from employees
	– triggers on decreasing trends in selling and demands
Pro-activeness	– takes initiative to contact Personal Assistants
	– takes initiative to propose special offers to customers
	– creates and initiates proposals for marketing actions and assortment modifications
Social behaviour	– co-operation with employees, Personal Assistants, and other Website Agents

takes responsibility to serve this customer until the customer has no wishes anymore that relate to the agent's department. The conventional Website can be used by the Website Agents to point at some of the articles that are relevant (according to their dialogue) to the customer.

- *refer customers to appropriate colleague Website Agents*
  A customer which is served at a department and was finished at that department can only be left alone if he or she has explicitly indicated to have no further wishes within the context of the entire department store. Otherwise the agent shall find out in which other department the customer may have an interest and the customer shall be referred to the agent representing this other department.
- *be able to provide product and special offer information*
  For example, if a client communicates a need, then a product is offered fulfilling this need (strictly or approximately), and, if available, a special offer.
- *dedicated announcement*
  As soon as available new products and special offers shall be announced to all relevant (on the basis of their profiles) customers, (they shall be contacted by the store in case they do not frequently contact the store).
- *analyses for marketing*
  The Website Agents shall monitor the amounts of articles sold (sales statistics), communicate them to employees (e.g., every week) and warn if substantially decreasing trends are observed. For example, if the figures of an article sold decrease during a period of 3 weeks, then marketing actions or assortment modifications shall be proposed.
- *actions for marketing*
  Each Website Agent shall maintain the history of the transactions of each of the customers within its department, and shall perform one to one marketing to customers, if requested. The employees shall be able to communicate to the relevant Website Agents that they have to perform a marketing campaign. The agent shall propose marketing actions to employees.

Table 4. Interaction characteristics for the Personal Assistant.

Personal Assistant—Interaction characteristics		
A. Interaction with the world		
observation	passive	– observe changes and special offers at the Website
observation	active	– observe the Website for articles within the customer needs
performing actions		
B. Communication with other agents		
incoming		*from Website Agent*:
		– product info
		– special (personal and general) offers
		from customer:
		– customer needs and preferences
		– agreement to buy
		– privacy constraints
outgoing		*to Website Agent*:
		– customer needs
		– payment information
		– profile information
		to customer:
		– product information
		– special offers

- *privacy*
  No profile is maintained without explicit agreement with the customer. The customer has access to the maintained profile.

3.2. *Characteristics and Requirements for the Personal Assistants*

For the Personal Assistants the interaction characteristics are given in Table 4, and their basic types of behaviour in Table 5. The following requirements can be imposed on the Personal Assistants:

- *support communication on behalf of the customer*
  Each customer shall be supported by his or her own Personal Assistant agent, who serves as an interface for the communication with the Website Agents.
- *only provide information within scope of interest of customer*
  A customer shall not be bothered by information that is not within his or her scope of interest. A special offer that has been communicated by a Website Agent leads to a proposal to the customer, if it fits in the profile, and at the moment when the customer wants such information
- *sensitive profiling*
  Customers are relevant for a special offer if they have bought a related article in the past, or if the offer fits in their profile as known to the Personal Assistant.
- *providing customer information for Website Agents*
  Every week the relevant parts of the profile of the customer is communicated to the Website Agent, if the customer agrees.
- *privacy*
  The Personal Assistant shall protect and respect the desired privacy of the customer. Only parts of the profile information agreed upon are communicated.

Table 5. Basic types of behaviour for the Personal Assistant.

Personal Assistant—Basic types of behaviour	
Autonomy	autonomous in dealing with Website Agents on behalf of customer
Responsiveness	responsive on needs communicated by customer
Pro-activeness	initiative to find and present special offers to customer
Social behaviour	with customer and Website Agents

4. The Internal Design of the Information Broker Agents

The agents in the multi-agent architecture for intelligent Websites presented in the previous sections have been designed on the basis of a generic model for a broker agent. The process of brokering as it often occurs as a mediating process in electronic commerce involves a number of activities. For example, responding to customer requests for products with certain properties, maintaining information on customers, building customer profiles on the basis of such customer information, maintaining information on products, maintaining provider profiles, matching customer requests and product information (in a strict or soft manner), searching for information on the WWW, and responding to new offers of products by informing customers for whom these offers fit their profile. In this Section a generic broker agent architecture is presented that supports such activities. This generic information broker model has been used as a basis for both the Website Agents and the Personal Assistant agents. As

these architectures have been designed using the compositional design method for multi-agent systems DESIRE, first a brief overview of DESIRE is presented (Section 4.1), next the generic broker agent model is briefly discussed (Section 4.2), and finally the two types of information broker agents that are used in the generic multi-agent architecture for intelligent Websites are discussed: Website Agent (Section 4.3) and Personal Assistant (Section 4.4).

4.1. Compositional Design of Multi-Agent Systems

The emphasis in DESIRE is on the conceptual and detailed design. The design of a multi-agent system in DESIRE is supported by graphical design tools within the DESIRE software environment. The software environment includes implementation generators with which (formal) design specifications can be translated into executable code of a prototype system. In DESIRE, a design consists of knowledge of the following three types: *process composition*, *knowledge composition*, and the *relation* between process composition and knowledge composition. These three types of knowledge are discussed in more detail below.

4.1.1. Process Composition. Process composition identifies the relevant processes at different levels of (process) abstraction, and describes how a process can be defined in terms of (is composed of) lower level processes.

Identification of Processes at Different Levels of Abstraction. Processes can be described at different levels of abstraction; for example, the process of the multi-agent system as a whole, processes defined by individual agents and the external world, and processes defined by task-related components of individual agents. The identified processes are modelled as *components*. For each process the *input and output information types* are modelled. The identified levels of process abstraction are modelled as *abstraction/specialisation relations* between components: components may be *composed* of other components or they may be *primitive*. Primitive components may be either reasoning components (i.e., based on a knowledge base), or, components capable of performing tasks such as calculation, information retrieval, optimisation. These levels of process abstraction provide process hiding at each level.

Composition of Processes. The way in which processes at one level of abstraction are composed of processes at the adjacent lower abstraction level is called *composition*. This composition of processes is described by a specification of the possibilities for *information exchange* between processes (*static view* on the composition), and a specification of *task control knowledge* used to control processes and information exchange (*dynamic view* on the composition).

4.1.2. Knowledge Composition. Knowledge composition identifies the knowledge structures at different levels of (knowledge) abstraction, and describes how a knowledge structure can be defined in terms of lower level knowledge structures. The knowledge abstraction levels may correspond to the process abstraction levels, but this is often not the case.

Identification of Knowledge Structures at Different Abstraction Levels. The two main structures used as building blocks to model knowledge are: *information types* and *knowledge bases*. Knowledge structures can be identified and described at different levels of abstraction. At higher levels details can be hidden. An *information type* defines an ontology (lexicon, vocabulary) to describe objects or terms, their sorts, and the relations or functions that can be defined on these objects. Information types can logically be represented in order-sorted predicate logic. A *knowledge base* defines a part of the knowledge that is used in one or more of the processes. Knowledge is represented by formulae in order-sorted predicate logic, which can be normalised by a standard transformation into rules.

Composition of Knowledge Structures. Information types can be composed of more specific information types, following the principle of compositionality discussed above. Similarly, knowledge bases can be composed of more specific knowledge bases. The compositional structure is based on the different levels of knowledge abstraction distinguished, and results in information and knowledge hiding.

4.1.3. Relation between Process Composition and Knowledge Composition. Each process in a process composition uses knowledge structures. Which knowledge structures are used for which processes is defined by the relation between process composition and knowledge composition.

4.2. *A Generic Information Broker Agent Architecture*

The generic information broker agent architecture was designed as a refinement of the generic agent model GAM (cf. [8]), supporting the weak agency notion (cf. [38]). First we will briefly describe the generic model GAM and next we discuss how this model was refined to the generic information broker model.

4.2.1. The Generic Agent Model GAM. At the highest process abstraction level within the compositional generic agent model GAM introduced in [8], a number of processes are distinguished that support interaction with the other agents. First, a process that manages communication with other agents, modelled by the component agent interaction management in Fig. 2. This component analyses incoming information and determines which other processes within the agent need the communicated information. Moreover, outgoing communication is prepared. Communication is modelled in a first-order logic approach, comparable, for example, to KIF. Communication from agent A to B takes place in the following manner:

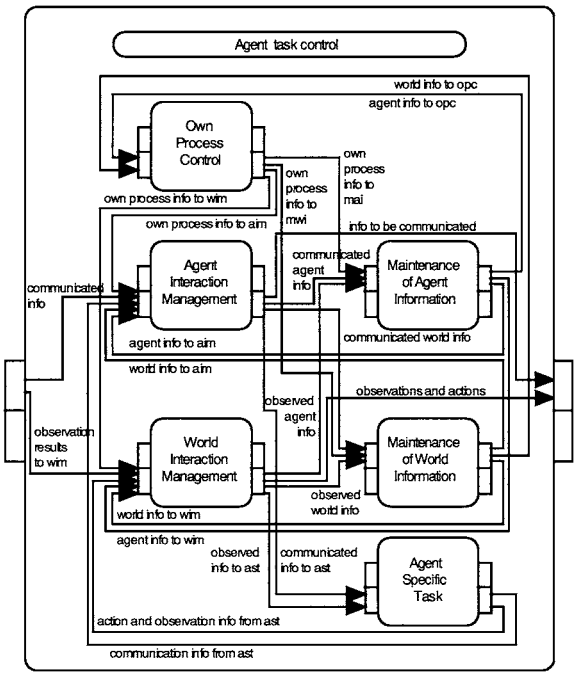

Figure 2. Composition within the generic information broker agent model.

- the agent A generates at its output interface a statement of the form:

  to_be_communicated_to (<*atom*>, <*illocution*>, B)

- the information is transferred to B; thereby it is translated into

  communicated_by (<*atom*>, <*illocution*>, A)

If needed, it is not difficult to replace this format by more extensive formats used in KQML or FIPA-ACL.

Next, the agent needs to maintain information on the other agents with which it co-operates: maintenance of agent information. The component maintenance of world information is included to store the 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 (with the World Wide Web world, in the example application): initiating observations and receiving observation results.

The agent processes discussed above 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. This is the purpose of the component Agent Specific Task. Figure 2 depicts how the generic agent is composed of its components.

4.2.2. Refinement of GAM to the Generic Information Broker Agent Model. The refinement of a generic model may involve both specialisation of the process composition and instantiation of the knowledge composition. The specific refinement discussed here only involves instantiation of the knowledge composition. Part of the exchange of information within the generic broker agent model can be described as follows. The broker agent needs input about scopes of interests put forward by agents and information about attributes of available products that are communicated by information providing agents. It produces output for other agents about proposed products and the attributes of these products. Moreover, it produces output for information providers about interests. In the information types that express communication information, the subject information of the communication and the agent from or to whom the communication is directed are expressed. This means that communication information

consists of statements about the subject statements that are communicated.

Within the broker agent, the component own process control uses as input belief info, i.e., information on the world and other agents, and generates focus information: to focus on a scope of interest to be given a preferential treatment, i.e., pro-active behaviour will be shown with respect to this focus. The component agent interaction management has the same input information as the agent (incoming communication), extended with belief info and focus info. The output generated includes part of the output for the agent as a whole (outgoing communication), extended with maintenance info (information on the world and other agents that is to be stored within the agent), which is used to prepare the storage of communicated world and agent information.

Information on attributes of products is stored in the component maintenance of world information. In the same manner, the beliefs of the agent with respect to other agents' profiles (provider attribute info and interests) are stored in maintenance of agent information. The component agent specific task uses information on product attributes and agent interests as input to generate proposals as output. For reasons of space limitation the generic and domain-specific information types within the agent model are not presented; for more details; see [21]. The information broker agent may have 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.

4.3. The Website Agent: Internal Design

The broker agent architecture provides an appropriate means to establish the internal design of the two types of agents involved.

For the Website Agent, the internal storage and updating of information on the world and on other agents (the beliefs of the agent) is performed by the two components maintenance of world information and maintenance of agent information. In Table 6 it is specified which types of information are used in these components. Profile information on customers is obtained from Personal Assistants, and maintained with the customer's permission. Also identified behaviour

Table 6. Maintenance information for the Website Agent.

Website Agent—Maintenance of information	
world information	– info on products within the Website Agent's assortment
	– info on special offers
agent information	– info on customer profiles
	– info on customer privacy constraints
	– info on customer preferences in communication
	– info on which products belong to which other Website Agent's assortments
	– info on providers of products

instances of the Personal Assistants can give input to the profile. Profile information can be abstracted from specific demands; how this is performed may depend on the application that is made.

The component agent interaction management identifies the information in incoming communication and generates outgoing communication on the basis of internal information. For example, if a Personal Assistant agent communicates its interests, then this information is identified as new agent interest information that is believed and has to be stored, so that it can be recalled later.

In the component agent specific task specific knowledge is used such as, for example:

- if the selling numbers for an article decrease for 3 weeks, then make a special offer with lower price, taking into account the right season
- if a customer asks for a particular cheap product, and there is a special offer, then this is proposed
- if an article is not sold enough over a longer period, then take it out of the assortment

Within this component non-strict (or soft) matching techniques can be employed to relate demands and offers.

4.4. The Personal Assistant: Internal Design

In this Section some of the components of the Personal Assistant are briefly discussed. For the Personal Assistant, as for the Website Agent, the internal storage and updating of information on the world and on other agents is performed by the two components maintenance of world information and maintenance of agent

Table 7. Maintenance information for a Personal Assistant.

Personal Assistant—Maintenance of information	
world information	– product information
	– special offers
agent information	– customer needs and profile
	– customer privacy constraints
	– offers personal to the customer
	– Website Agents assortment scopes

information. In Table 7 it is specified which types of information are used in these components.

As in the previous section, the component agent interaction management identifies the information in incoming communication and generates outgoing communication on the basis of internal information. For example, if a Website Agent communicates a special offer, then this information is identified as new agent information that is believed and has to be stored, so that it can be recalled later. Moreover, in the same communication process, information about the product to which the special offer refers can be included; this is identified and stored as world information.

4.5. Profile Modelling Approaches that can be Used Within the Agents

Within the generic architecture for Website Agents and Personal Assistants no commitment has been made to specific approaches to user profiling. In this section a number of these approaches are briefly discussed (for a more detailed treatment, see [11]). The profile of a user can be used to determine how interesting an information item is to that user. It can be used to select and prioritise information items in a personalised manner. The structure and properties of profiling approaches may vary with the application area in which they are used. For example, in multi-attribute decision systems (see [3, 23, 37]) the user profile or preference for an item is defined in terms of values of various attributes of the item and the preferences of the user towards those attributes (i.e., the importance of those attributes). On the other hand, in the area of recommendation systems the profile may as well be defined in terms of statistical correlation between users and their rated items.

The preferences of a user towards a set of items can be defined in terms of the content of the items (*content information*) or the preference of the items by a society of users (*collaborative or social information*). In the content-based approach a user is defined to have preference for an item if the item is similar in attribute values to other items that are preferred by the user. Also ratings for the (relative) relevance of attributes for a user are often included.

In the collaborative-based approach a user is defined to have preference for an item if the user is similar (in preferences of other items) to other users who have preference for the item. Both the content information as well as the collaborative information can be used to construct user profiles.

The construction of a profile can be a time consuming matter. For example, in the content-based approach the user may have to express his or her preferences towards various (combinations of) attributes and attribute values in extensive forms. Some systems (e.g., see [12]) derive the preferences of a user by suggesting an item to the user and ask her to correct this suggestion. The user corrects the system's suggestion by indicating why the suggested item does not match his or her needs. Based on these corrections, profiles of users are constructed or updated. In other, collaborative-based applications such as recommendation systems, a user may be asked to rate several, sometimes hundreds, of (other) items before an item can be recommended.

A number of systems employ methods to induce the user profile by observing the behaviour of that user over time (e.g., see [16, 26, 29, 31]). These methods are usually not intended to fully model user profiles, but to model the more frequent and predictable user preferences. Applications that require huge efforts from their users may become ineffective (e.g., see [27, 28]). To model user profiles in an application, a balance is to be found between the amount of interaction with the user and the effectiveness of the constructed user profile.

Modelling user profiles on the basis of content or collaborative information can be considered as a learning problem where the aim is to learn the so-called preference function for a certain user. The preference function for a user maps items from a certain domain to some values that express the importance of those items for that user. Various types of preference functions may exist. The type of a preference function characterises the structure of profile (e.g., see [23, 37]). Another profile learning approach, based on Inductive Logic Programming, can be found in [5, 6, 11].

Several collaborative-based recommendation systems have been introduced in which the preferences of

users are modelled automatically. Examples of online recommendation systems that employ a collaborative approach are MovieFinder [39] and FireFly [13]. The preferences of a user are modelled automatically by observing the behaviour of that user and applying statistical methods to the observed behaviour (e.g., see [4, 16, 17, 33]).

In contrast to the collaborative-based approach, the content-based approach can be applied only when items are described in terms of properties and attribute values. The content-based profiling approaches have been used in online recommendation systems such as BargainFinder [1] and Jango [18]. Unlike collaborative-based preference models, the content-based preference models are also used in applications such as integrative negotiation where the utility function is defined in terms of user preferences towards various attribute values (e.g., see [3, 15, 23, 29, 37]). The collaborative-based and content-based approaches do not exclude each other; in fact they can be combined into an integrated approach to model user profiles (see [2]).

The effectivity of collaborative-based and content-based approaches to profiling may depend on the application. For example, collaborative-based profiling approaches may be more effective in applications where it is unrealistic to collect a large amount of information about the preferences of an individual user, or where the number of users is too large. Using collaborative-based profiling models is also effective for applications where the content of the items neither is available nor can be analysed automatically by a machine (e.g. items like a picture, video, sound). However, the collaborative-based profiling approaches are less effective for applications like integrative negotiation (e.g., see [3, 14]) in retail Electronic Commerce where negotiation is considered to be a decision making process over items that are described as multiple interdependent attributes.

5. Reuse of the Generic Architecture in the Insurance Domain

The reusability of the designed generic multi-agent architecture was tested in a new domain: insurance. In this section this domain is briefly introduced. One of the largest insurance companies in the Netherlands is organised on the basis of (human) mediating insurance agents. To support these agents a Website was created with information about products offered by the company, forms to support administrative actions, and other related information. The Website is structured around four main sections: Store, Desk, Newsstand and Office.

The *store* provides information about the insurance products offered. The various insurance policies can be found here, as well as request forms for more information, brochures, and personalised proposals. From the store a couple of useful programs can be downloaded as well: spreadsheets, an anti-virus toolkit, and an insurance dictionary.

When the insurance agent is faced with a problem, he or she can turn to the *desk*. Apart from a Frequently Asked Questions page also a form is available for specific questions. The desk further contains the editorials that address certain problems in depth. Finally, an address book is available, in which the various departments and teams operating within the company can be found.

At the *news-stand* the visitors of the site can find the most recent information. Newsletters can be found, and a calendar can be checked for upcoming events. Furthermore, various links to other interesting sites and assorted articles are offered here. Whenever new interesting sites or articles are added, the visitor can be notified of this by email.

At the *office*, the sale of insurance products is supported. Here resources to improve the insurance agent's job can be found: telemarketing scripts, newsletter articles, advertisements that only need further filling out and sales letters. Furthermore, the agent can find his or her personal production figures for the company's products.

The Website consists of a collection of variable information sources: images, programs, documents, addresses, phonebooks and personal data. New information is added daily. Keeping up to date with the most recent relevant information, is time-consuming. The multi-agent system has been developed to support the human agent in this task.

The aim of the multi-agent system integrated in the Website is to improve the use of the resources offered by the Website. From the visitors point of view, more interesting information can be obtained. The agent, with its knowledge of the user improves the customer experience. Application forms can be offered, already (partially) filled out by the software agent. The employees maintaining the Website can use information collected by the multi-agent system to improve marketing. The appropriate visitor can be contacted about new (possibly personalised) products or offers that are relevant to him or her.

6. Instantiation of the Generic Architecture

The generic multi-agent architecture has been instantiated for the new domain of insurance described in Section 5. Application-specific information types and knowledge bases were specified and included in the model. The system is explained for two cases: behaviour initiated by an information request of a user (user initiated), and behaviour initiated by update or addition of information to the Website (Website initiated). In both cases, after initiation a reactivity chain is triggered. In the first case the main reactivity chain follows the path

user-PA-WA-PA-user

The first half of this path deals with the queries, and the second half (back) with answers on these queries. In the second case the main reactivity chain follows the path

WA-PA-user-PA-WA

The first half deals with voluntarily offered information (one-to-one marketing), and the second half (back) with feedback on usefulness of the offered information (in order to update profile information). In the explanation of these behavioural traces, it is shown which knowledge bases were used to instantiate the generic architecture.

6.1. Information used in the System

This system is only a prototype; as such it does not work with the actual information on the Website. Instead a sample of the information objects on the Website was selected and a description of each of these was made.

In cooperation with employees from the insurance company the following attributes were selected to describe the information:

- **Title:** The title of the information object.
- **Author:** The department or person that created the information object.
- **Subject:** Subject of the information object.
- **First Relation:** The first related subject.
- **Second Relation:** The second related subject.
- **Date:** Date of creation/availability.
- **Language:** The language used in the information.
- **Persistency:** An indication of how soon the information will be outdated.
- **Kind:** The form of the information object (mailform, text, audio, etc.).
- **Type:** The type of information in the information object (e.g., FAQ, newsletter, personal information).
- **URL:** The hyperlink to the actual information object

6.2. Behaviour Initiated by a User

When a user asks a question, the Personal Assistant agent performs a number of actions. The question is analysed to find similarities to previous questions and if these exist, new interests are created within the user profile. Furthermore, the agent attempts to respond to the information request using information available within the Personal Assistant itself and by contacting the appropriate Website Agents.

First it is described how an answer to a question is found. Next, the process of updating the user profile is discussed.

Handling a Question. The behaviour of the system is first described from the user's point of view. Subsequently, the processes that are invisible for the user are described in more details.

The User Interaction. A trace is described in which a user needs information about car insurance. As a first step the user communicates this question to the Personal Assistant using the interface (Fig. 3): the user selects the subject 'car insurance' in the scrollable list under the heading 'Subject'. The Personal Assistant will start to acquire useful information on behalf of its user.

The Personal Assistant inspects all information it has in store and it contacts appropriate Website Agents for more information. All the gathered relevant information is communicated to the user, using the display in Fig. 4; each title is a link to a description of the information. The user can indicate whether or not he or she evaluates the information as interesting.

The Processes within the Multi-Agent System. When the Personal Assistant agent receives a question from the user, it identifies the communication as a question in the component agent interaction management. The question is further processed in the task specific

Table 8. Knowledge involved in user-initiated behaviour.

r1	
if	query(Q:QUERY_ID, scope(subject, S:SUBJECT))
and	object_scope(O:OBJECT_ID, scope(related_subject, S:SUBJECT))
then	possible_answer_to_query(O:OBJECT_ID, Q:QUERY_ID);
r2	
if	query(Q:QUERY_ID, scope(A:ATTRIBUTE, V1:VALUE))
and	object_scope(O:OBJECT_ID, scope(A:ATTRIBUTE, V2:VALUE))
and	**not** A:ATTRIBUTE = subject
and	**not** V1:VALUE = V2:VALUE
then	rejected_answer_for_query(O:OBJECT_ID, Q:QUERY_ID);
r3	
if	possible_answer_to_query(O:OBJECT_ID, Q:QUERY_ID)
and	**not** rejected_answer_for_query(O:OBJECT_ID, Q:QUERY_ID)
then	selected_answer_to_query(O:OBJECT_ID, Q:QUERY_ID);

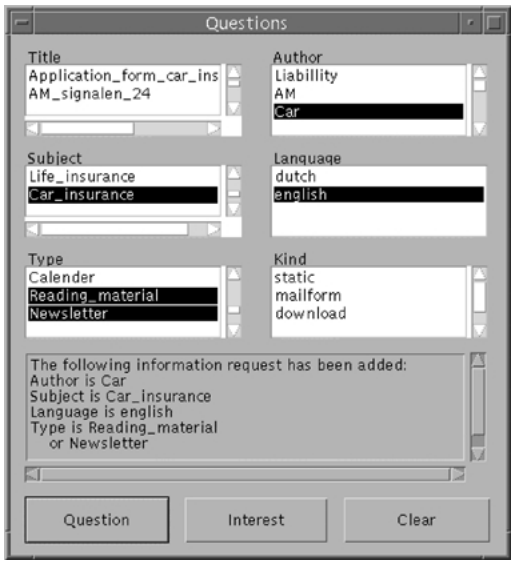

Figure 3. User interface for asking questions and stating user interest.

Figure 4. Display for the answers to questions and offers made (by a Personal Assistant).

component determine proposals of the Personal Assistant. That component matches the request to the information objects available in the memory of the agent (component maintenance of world information). Two types of matching are covered: *strict matching* and *soft matching*. For strict matching, attributes need to have exactly the same value, or an overlapping value range. For soft matching, it can be specified when values of attributes are considered close (but not necessarily equal) to each other. This closeness relation may be based on various techniques. In the current prototype the closeness relation for the subject attribute is taken as a point of departure, abstracting from the manner in which it is determined. One of the matching rules is rule r1 in Table 8.

The subject of the query is matched with the related subject of the object under consideration. If the rule succeeds, the object is selected as a possible answer. A criterion for this possible answer to become a definite answer is that the object does not differ on other attributes (see rule r2). Rule r3 is used to derive the final answer to the question.

Simultaneously, in the same component determine proposals, the relevant Website Agents are selected.

Table 9. Knowledge involved in selection of Website Agents.

```
r4
if      query(Q:QUERY_ID, scope(subject, S:SUBJECT))
  and   webagent_subject(W:WA, S:SUBJECT)
then    main_wa_for_answer(W:WA, Q:QUERY_ID)
  and   found_wa_for_query(Q:QUERY_ID);
r5
if      query(Q:QUERY_ID, S:SCOPE)
  and   can_provide_scope(W:WA, S:SCOPE)
then    secondary_wa_for_answer(W:WA, Q:QUERY_ID)
  and   found_wa_for_query(Q:QUERY_ID);
r6
default_wa(wa1)
r7
if      main_wa_for_answer(W:WA, Q:QUERY_ID)
then    selected_wa_for_answer(W:WA, Q:QUERY_ID);
r8
if      secondary_wa_for_answer(W:WA, Q:QUERY_ID)
then    selected_wa_for_answer(W:WA, Q:QUERY_ID);
r9
if      not found_wa_for_query(Q:QUERY_ID)
  and   default_wa(W:WA)
then    selected_wa_for_answer(W:WA, Q:QUERY_ID);
```

This is done in three steps. First, the Personal Assistant agent looks for a Website Agent that is known to provide information about the subject occurring in the query; see rule r4 in Table 9. Rule 4 makes use of the agent model for the Website Agent that is stored by the Personal Assistant within component maintenance of agent information. Information about the subjects that a Website Agent can provide is expressed by the statement webagent_subject(W:WA, S:SUBJECT).

Rule 4 will not succeed, however, when the question does not contain a subject term or when the Personal Agent does not know a relevant Website Agent. In this case the Personal Assistant agent uses a second method to determine an appropriate Website Agent, by considering another part of the agent models it maintains of Website Agents; see rule r5. Finally as a fail-safe, each Personal Assistant has a default Website Agent it can contact. The name of this default Website Agent is stored in the component own process control and is also available in the component determine proposals. The final selection of the Website Agent is performed by the knowledge specified in rules r6 to r9.

Next the selected_wa_for_answer and selected_object_for_query information is transferred to the component agent interaction management where communication to the selected Website Agent(s) is actually initiated.

Website Agents handle questions in the same way as the Personal Assistant. The component determine proposals of a Website Agent attempts to find a match with the known information objects. The matches are communicated back to the Personal Assistant. The component agent interaction management of the Personal Assistant passes the received answers on to its user. The information contained in received answers is also stored by the Personal Assistant: in the future it can supply this information by itself.

Update of User Profile. The focus of the current prototype lies on the agent interaction and document selection. Profile management had a lower priority. Therefore the mechanisms for profile management used are simple. As stated earlier, the Personal Assistant compares questions to each other. When similarities are found in three questions, these similarities are added to the user profile. This is performed by the (composed) component interest creator.

A new question is first compared to all previous questions. A simple method has been chosen to create these candidates: whenever three different questions match on one or more attribute values, these attribute-value pairs are selected as a candidate interest specification; see rule r11 in Table 10. The three query id's are combined to create a temporary candidate id. The created candidate is compared to the already existing interests; see rule r10. Because this component is reasoning about changes in interests, it is at a meta-level compared to the component maintenance of agent information, in which the interests are maintained. So reasoning about interests is done by encapsulating them within the belief statement (rule r11). If a candidate is not a duplicate of an already existing interest it is added to the user profile. First a unique interest identifier is created (see rule r12), and next the new interest is created in the component maintain agent information using an information link. Extra constraints could be added to the creation of these candidates. For example, the questions must be asked within a certain (temporal) distance of each other.

6.3. Behaviour Initiated by the Website

The second type of behaviour discussed here is initiated by the Website. First the behaviour to directly serve the

Table 10. Knowledge involved in profile update: user-initiated case.

r10	
if	asked(query(Q1:QUERY_ID, scope(A:ATTRIBUTE, V:VALUE)))
and	asked(query(Q2:QUERY_ID, scope(A:ATTRIBUTE, V:VALUE)))
and	asked(query(Q3:QUERY_ID, scope(A:ATTRIBUTE, V:VALUE)))
and	**not** Q1:QUERY_ID = Q2:QUERY_ID
and	**not** Q1:QUERY_ID = Q3:QUERY_ID
and	**not** Q2:QUERY_ID = Q3:QUERY_ID
then	candidate_for_interest(candidate_id(Q1:QUERY_ID, Q2:QUERY_ID, Q3:QUERY_ID), scope(A:ATTRIBUTE, V:VALUE));
r11	
if	candidate_for_interest(C:CANDIDATE_ID, scope(A:ATTRIBUTE, V1:VALUE))
and	belief(interest(I:INTEREST_ID, scope(A:ATTRIBUTE, V2:VALUE))
and	**not** V1:VALUE = V2:VALUE
then	different(C:CANDIDATE_ID, I:INTEREST_ID);
r12	
if	new_interest_id(I:INTEREST_ID)
and	approved_candidate(C:CANDIDATE_ID, S:SCOPE)
then	to_be_created(interest(I:INTEREST_ID, S:SCOPE));

user is discussed, and subsequently the behaviour to update the user profile is described in more detail.

Offering the User New Information. First the behaviour shown to the user is described. Next a more detailed description is given of the processes within the multi-agent system itself.

The User Interaction. The Personal Assistant takes the initiative to notify its user when relevant information has been found, using the display depicted in Fig. 4. Again, the user can click on a title to get more information about the proposal. Furthermore, the user can choose to accept the proposed information or to reject it.

The Processes within the Multi-Agent System. When new information becomes available at a Website, the Website Agent identifies possible interested parties. The Website Agent has built a profile of the Personal Assistants it has been in contact with. In the component determine proposals the Website Agent uses this information to match the new information to the Personal Assistants interests; see rule r13 in Table 11.

For each scope in the new object a comparison to the existing interests in the profile is made. When they match, the object is partly selected. However, on another scope, the interest and the new object may differ. Only if all of the scopes of the object match is the object selected. The offer is made by the component agent interaction management. The Personal Assistant receives this offer and compares it to the interests in its user profile. This is performed in the Personal Assistant's determine proposals, as it is done in the Website Agent; see rule r14. Again, when no conflicting scopes can be found between the interest and the offered object, it is selected using rule r15. The selected offer is communicated to the user, who can reply to the offer.

Update of User Profile. After the user has communicated to the Personal Assistant whether he or she rates the offer interesting or not, a profile update process is initiated, if necessary, by removing those interests repeatedly receiving negative feedback. This feedback is used in the component interest remover to select interests for removal. Similar to the creation of new interests, a simple mechanism is used to select interests for removal. A circular list is kept of the last three responses to offers based on an interest. This list has three objects; when all three objects show a negative response, the interest is marked for removal; see rule r16 in Table 12.

An interest marked for removal is not automatically removed. Before actual removal, the user has to give his or her approval. When the user disapproves of the

Table 11. Knowledge involved in Website-initiated behaviour.

r13	
if	new_object_scope(O:OBJECT_ID, S:SCOPE)
and	interest(P:PA, I:INTEREST_ID, S:SCOPE)
then	partly_matched_new_object(O:OBJECT_ID, P:PA, I:INTEREST_ID);
r14	
if	offered_object_scope(O:OBJECT_ID, S:SCOPE)
and	interest(I:INTEREST_ID, S:SCOPE)
then	partly_matched_offer(O:OBJECT_ID, I:INTEREST_ID);
r15	
if	partly_matched_offer(O:OBJECT_ID, I:INTEREST_ID)
and	**not** rejected_offer(O:OBJECT_ID, I:INTEREST_ID)
then	accepted_offer(O:OBJECT_ID, I:INTEREST_ID);

Table 12. Knowledge involved in profile update: removal in Website-initiated case.

r16	
if	last3_suggestions_response(last_id1, rejected, I:INTEREST_ID)
and	last3_suggestions_response(last_id2, rejected, I:INTEREST_ID)
and	last3_suggestions_response(last_id3, rejected, I:INTEREST_ID)
then	to_be_confirmed(remove(I:INTEREST_ID))
r17	
if	removal_response(I:INTEREST_ID, confirmed)
and	belief(interest(I:INTEREST_ID, S:SCOPE))
then	to_be_removed(interest(I:INTEREST_ID, S:SCOPE));

removal, the three last responses to that interest are reset; thus again three rejections in a row must be received before the agent considers the interest for removal. When the user approves, the removal is performed; see rule r17. As for the interest creator, this component reasons about changes in interests and is therefore at a meta-level compared to the component maintenance of agent information. The interest is actually removed by an information link, similar to how interests are created.

7. Discussion

In this paper a generic, reusable multi-agent architecture for active intelligent Websites is presented. This generic architecture for active intelligent Websites was first designed for one application domain: a department store (cf. [22]). This application reuses the generic architecture of information broker agents developed earlier (cf. [21]), which in turn was designed as a specialisation of the generic agent model GAM introduced in [8]. The model has been designed in such a way that the generic, reusable structures are separated from the application-specific aspects in a transparent manner.

The reusability of the generic multi-agent architecture for active intelligent Websites has been tested in a second application: a project on an intelligent Website for insurance in co-operation with the software company Ordina Utopics and an insurance company (cf. [20]). The outcome of this test was clearly positive. With not much effort (an investment of only a few person months) a prototype multi-agent system for an intelligent Website in insurance has been designed and implemented, based on the generic architecture. The actual work concentrated mainly on the specification of the domain concepts and application-specific knowlege bases.

A Website, supported by the architecture introduced has a more personal look and feel than the usual Websites. Within the architecture, also negotiation facilities can be incorporated.

In the agent literature, a number of architectures for (information) broker agents can be found; e.g., [9, 10, 24, 25, 30, 32, 34, 36],. The design of most of these architectures is not formally specified in detail; usually they are only available in the form of an implementation, and at the conceptual level some informal pictures and natural language explanations. In general, the aim for the development of these architectures in the first place is to have a working piece of software for a specific type of application. The design of the generic architecture for intelligent Websites introduced in this paper has a different aim. The generic model was meant as a unified design model, formally specified in an implementation- and domain-independent manner at a high level of abstraction. The (multi) agent architecture described here was designed and implemented in a principled manner, using the compositional design method for multi-agent systems DESIRE [7]. Due to its compositional structure it supports reuse and maintenance; a flexible, easily adaptable architecture results. A success criterion for this aim is the possibility to specialise and instantiate the model to obtain conceptual, formal specifications of design models for different applications. The positive experience in the insurance domain, discussed above, shows that the aim was achieved.

Applications of broker agents (addressed in, e.g., [9, 10, 24, 25, 30, 32, 34, 36]), often are not implemented in such a principled manner: without an explicit design at a conceptual level. Compared to, for example, systems designed using CORBA, or other object-based methods, a main difference is that in our approach functionality can be specified at the level of design in an explicit declarative manner (in the form of ontologies and knowledge bases). Especially for applications in knowledge-intensive domains this provides appropriate means to specify a design. The RETSINA approach (cf., [34, 35]) is more comparable to the design method DESIRE as such, and not to the generic architecture for the specific application type of intelligent Websites proposed here. A difference is that DESIRE is based on a formal specification language for design models. The same difference applies to the work on SIMS, described in [24]. However, in [24], also the problem of information integration is addressed (i.e., integration of information expressed in different ontologies), which has not (yet) been addressed in the architecture proposed here. A next step is to refine our model with possibilities for information integration, for example adopted from SIMS.

The question whether the approach scales up has not been explicitly investigated in the research reported, by performing experiments. Since an essentially distributed approach has been chosen, the Personal Assistant agents can all be implemented on an own server. Also it is possible to implement different Website Agents on different servers, thus avoiding too much interaction overload of one server.

For the particular application in insurance the generic broker agent model has been instantiated with domain ontologies and domain knowledge. In the prototype some of these instantiations have been done in an ad hoc manner, without the intention to propose these instantiations as a generic approach for more domains. Current research addresses more principled manners to use dynamic taxonomies in profile creation and techniques from inductive logic programming to induce profiles from examples. In [11] (see also Section 4.5 above) an overview is given of a number of these profiling approaches and it is shown how they can be incorparated). As an example, in further research a component-based generic agent architecture for multi-attribute (integrative) brokering and negotiation has been developed in co-operation with, among others, Dutch Telecom KPN. The agent architecture was designed as a refinement of the compositional generic agent model GAM. Within the component Maintenance of Agent Information (MAI) within this agent architecture, a profile of the human user of the agent is maintained, which includes

- evaluation functions per attribute assigning to each attribute value an evaluation value between 0 and 1,
- importance factors (between 0 and 1) for the different attributes.

Within such a more sophisticated content-based profile model (which, for example, is also used in [3], it can be expressed, for example, that a car with colour blue is evaluated as 0.9, whereas a yellow colour is evaluated as 0.1, and a CD player of high quality is rated 0.8 whereas a CD player with low quality as 0.2. Moreover, the attribute 'colour' can be assigned, e.g., importance 0.6, whereas the attribute 'CD player' can be assigned importance 0.8.

Acknowledgments

Pascal van Eck (Vrije Universiteit) supported other experiments with some variants of the broker agent model. Working on a design for the department store application with various employees of the software company CMG provided constructive feedback on the architecture introduced. The authors are also grateful for discussions with employees of Ordina Utopics (in particular, Richard Schut) and an (anonymous) insurance company.

References

1. BargainFinder URL:<http://bf.cstar.ac.com>
2. C. Basu, H. Hirsh, and W. Cohen, "Recommendation as classification: Using social and content-based information in recommendation," in *Proc. Fifteenth National Conference on Artificial Intelligence*, 1998, pp. 714–720.
3. W. Benn, O. Görlitz, and R. Neubert, "Enabling integrative negotiations by adaptive software agents," in *Cooperative Information Agents III, Proc. Third International Workshop on Cooperative Information Agents, CIA'99*, edited by M. Klusch, O. Shehory, and G. Weiss, Lecture Notes in Artificial Intelligence, vol. 1652, Springer Verlag, 1999, pp. 335–346.
4. D. Billsus and M. Pazzani, "Learning collaborative information filters," in *Proc. Fifteenth International Conference on Machine Learning*, Morgan Kaufmann: Madison, WI, 1998, pp. 46–54.
5. H. Blockeel and L. De Raedt, "Lookahead and discretization in ILP," in *Proc. 7th International Workshop on Inductive Logic Programming*, Lecture Notes in Artificial Intelligence, vol. 1297, Springer-Verlag, 1997, pp. 77–85.
6. H. Blockeel, L. De Raedt, N. Jacobs, and B. Demoen, "Scaling up inductive logic programming by learning from interpretations," *Data Mining and Knowledge Discovery*, vol. 3, no. 1, pp. 59–93, 1999.
7. F.M.T. Brazier, B. Dunin-Keplicz, N.R. Jennings, and J. Treur, "Formal specification of Multi-Agent Systems: A real-world case," in *Proc. First International Conference on multi-agent systems, ICMAS'95*, edited by V. Lesser, Cambridge, MA: MIT Press, pp. 25–32. 1995, Extended version in *International Journal of Co-operative Information Systems*, special issue on Formal Methods in Co-operative Information Systems: Multi-Agent Systems, edited by M. Huhns, M. Singh, vol. 6, pp. 67–94, 1997.
8. F.M.T. Brazier, C.M. Jonker, and J. Treur, "Compositional design and reuse of a generic agent model," *Applied Artificial Intelligence Journal*, vol. 14, pp. 491–538, 2000. Shorter version in *Proc. of the Knowledge Acquisition Workshop*, *KAW'99*, Banff, 1999.
9. A. Chavez and P. Maes, "Kasbah: An agent marketplace for buying and selling goods," in *Proc. First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'96*, The Practical Application Company Ltd.: Blackpool, 1996, pp. 75–90.
10. A. Chavez, D. Dreilinger, R. Gutman, and P. Maes, "A real-life experiment in creating an agent market place," in *Proc. Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'97*, The Practical Application Company Ltd.: Blackpool, 1997, pp. 159–178.
11. M. Dastani, N. Jacobs, C.M. Jonker, and J. Treur, "Modeling user preferences and mediating agents in electronic commerce," in *Agent-Mediated Electronic Commerce; a European Perspective*, edited by F. Dignum and C. Sierra, Lecture Notes in AI, vol. 1991, Springer Verlag, 2001, pp. 164–196.
12. E.C. Freuder and Richard J. Wallace, "Suggestion strategies for constraint-based matchmaker agents," in *Principles and Practice of Constraint Programming – CP'98*, Pisa, October 1998.
13. FireFly URL:<http://www.firefly.com
14. R.H. Guttman and P. Maes, "Agent-mediated integrative negotiation for retail electronic commerce," in *Agent Mediated Electronic Commerce*, *Proc. Workshop on Agent Mediated Electronic Trading (AMET'98)*. Lecture Notes in AI, vol. 1571, Springer Verlag, 1999, pp. 70–90.
15. R.H. Guttman, A.G. Moukas, and P. Maes, "Agent-mediated electronic commerce: A survey," *Knowledge Engineering Review*, vol. 13, pp. 147–161, June 1998.
16. W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in *Proc. Conference on Human Factors in Computing Systems (CHI95)*, ACM Press: Denver, CO, 1995, pp 194–201.
17. T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *Proc. Sixteenth International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, July 31–August 6, 1999.
18. Jango URL:<http://www.jango.com>
19. C.M. Jonker, M. Klusch, and J. Treur, "Design of collaborative information agents," in *Cooperative Information Agents IV, Proceedings of the Fourth International Workshop on Cooperative Information Agents, CIA 2000*, edited by M. Klusch and L. Kerschberg, Lecture Notes in Artificial Intelligence, vol. 1860, Springer Verlag, 2000, pp. 262–283.
20. C.M. Jonker, R.A. Lam, and J. Treur, "A multi-agent architecture for an intelligent website in insurance," in *Cooperative Information Agents III, Proc. Third International Workshop on Cooperative Information Agents, CIA'99*, edited by M. Klusch, O. Shehory, and G. Weiss, Lecture Notes in Artificial Intelligence, vol. 1652, Springer Verlag, 1999, pp. 86–100.
21. C.M. Jonker and J. Treur, "Compositional design and maintenance of broker agents," in *Proc. 15th IFIP World Computer Congress, WCC'98, Conference on Information Technology and Knowledge Systems, IT&KNOWS'98*, edited by J. Cuena, pp. 319–332, 1998.
22. C.M. Jonker and J. Treur, "Information broker agents in intelligent websites," in *Multiple Approaches to Intelligent Systems (Proc. of the 12th International Conference on Industrial and Engineering Applications of AI and Expert Systems, IEA/AIE'99)*, edited by I. Imam, Y. Kodratoff, A. El-Dessouki, and M. Ali, Lecture Notes in AI, vol. 1611, Springer Verlag, 1999, pp. 430–439.
23. R. Keeney and H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Trade-offs*, John Wiley and Sons, 1976.
24. C.A. Knoblock and J.L. Ambite, "Agents for information gathering," in *Software Agents*, edited by J. Bradshaw, AAAI/MIT Press, 1997.

25. D. Kuokka and L. Harada, "On using KQML for matchmaking," in *Proc. First International Conference on Multi-Agent Systems, ICMAS'95*, edited by V. Lesser, MIT Press: Cambridge, MA, 1995, pp. 239–245.
26. K. Lang, "Learning to filter news," in *Proc. Twelfth International Conference on Machine Learning*, Lake Tahoe, CA, 1995, pp. 331–339.
27. P. Langley, "Machine learning for adaptive user interfaces," in *Proc. 21st German Annual Conference on Artificial Intelligence*, Freiburg, Germany: Springer, 1997, pp. 53–62.
28. P. Langley, "User modeling in adaptive interfaces," in *Proc. Seventh International Conference on User Modeling*, 1999.
29. H. Lieberman, "Letizia: An agent that assists web browsing," in *Proc. 14th International Joint Conference on Artificial Intelligence (IJCAI'95)*, Montreal, Canada, 1995.
30. D. Martin, D. Moran, H. Oohama, and A. Cheyer, "Information brokering in an agent architecture," in *Proc. Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'97*, The Practical Application Company Ltd.: Blackpool, 1997, pp. 467–486.
31. M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine Learning*, vol. 27, pp. 313–331, 1997.
32. T. Sandholm and V. Lesser, "Issues in automated negotiation and electronic commerce: Extending the contract network," in *Proc. First International Conference on Multi-Agent Systems, ICMAS'95*, edited by V. Lesser, MIT Press: Cambridge, MA, pp. 328–335, 1995.
33. U. Sharadanand and P. Maes, "Social information filtering: Algorithms for automating 'Word of Mouth'," *Proc. CHI-95, Conference*, 1995.
34. K. Sycara, K. Decker, A. Pannu, and M. Williamson, "Designing behaviors for information agents," *Proc. Agents 97*, pp. 404–412.
35. K. Sycara, K. Decker, A. Pannu, M. Williamson, and D. Zeng, "Distributed intelligent agents," *IEEE Expert*, vol. 11, pp. 36–45, 1996.
36. M. Tsvetovatyy and M. Gini, "Toward a virtual marketplace: Architectures and strategies," in *Proc. First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, PAAM'96*, The Practical Application Company Ltd.: Blackpool, 1996, pp. 597–613.
37. P. Vincke, *Multicriteria Decision-Aid*, John Wiley and Sons, 1992.
38. M. Wooldridge and N.R. Jennings, "Agent theories, architectures, and languages: A survey," in *Intelligent Agents, Proceedings of the First International Workshop on Agent Theories, Architectures and Languages, ATAL'94*, edited by M. Wooldridge and N.R. Jennings, Lecture Notes in AI, vol. 890, Springer Verlag, 1995, pp. 1–39.
39. MovieFinder URL: <http://www.moviefinder.com