

Distributed Scheduling to Support a Call Centre: a Co-operative Multi-Agent Approach*

Frances M.T. Brazier, Catholijn M. Jonker, Frederik Jan Jüngen¹, Jan Treur

Vrije Universiteit Amsterdam

Department of Mathematics and Computer Science, Artificial Intelligence Group
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands

URL: <http://www.cs.vu.nl>, Email: {frances,jonker,treur}@cs.vu.nl

Abstract

This paper describes a multi-agent system architecture to increase the value of 24 hour a day call centre service. This system supports call centres in making appointments with clients on the basis of knowledge of employees and their schedules. Relevant activities of employees are scheduled for employees in preparation of such appointments. The multi-agent system architecture is based on principled design, using the compositional development method for multi-agent systems DESIRE. To schedule procedures in which more than one employee is involved, each employee is represented by its own personal assistant agent, and a work manager agent co-ordinates the schedules of the personal assistant agents, and clients through the call centre. The multi-agent system architecture has been applied to the banking domain, in co-operation with and partially funded by the Rabobank.

1 Introduction

Over the past few years, more and more companies and organisations have become aware of the potential of a 24 hour market. An increasing number of call centres are now providing 24 hour service to their customers. One of the areas of industry in which this phenomenon has become manifest is finance. To increase service level, the Rabobank, one of the largest banks in the Netherlands, for example, now provides its bank relations 24 hour a day telephone service. Relatively straightforward client requests can be answered directly by the call centre. Other, more complex requests, are forwarded to the most appropriate local bank. Outside working hours this means the request is received and processed the next morning. Within working hours the request is received and processed immediately. Processing a request entails deciding on a procedure to follow within the organisation and scheduling this procedure and the necessary resources (mainly employees' time). To increase the level of service, the Rabobank is currently exploring the option to partially automate this process.

* To appear in: Applied AI Journal, 1998

¹ Current affiliation: Cambridge Technology Partners, Amsterdam, the Netherlands.
Current email address: fjunge@ctp.com

This paper describes a prototype system, developed in close co-operation with the Rabobank, that automatically schedules procedures on the basis of client requests forwarded by a call centre. This system, a multi-agent system, uses a generic model of co-operation based on joint intentions to model the two types of automated agents involved: the work manager and personal assistants. Interaction with the other agents involved: the client, the call centre employee and all other employees, is also explicitly modelled.

More detail on the Rabobank itself with respect to this application is provided in Section 2. Section 3 describes the multi-agent approach to this problem. Section 4 describes the work manager. The conceptual model of the personal assistant is described in Section 5. The role of the employee is discussed in Section 6. An example of a design process is given in Section 7. Section 8 discusses the results.

2 Problem Description

The Rabobank is one of the largest banks in the Netherlands with a co-operative structure with autonomous branch offices, each responsible for specific geographical areas. These local organisations (local banks) all service both the consumer market and industry.

2.1 The Problem

The Rabobank's aim is to achieve a stronger position in the financial market by using its resources more efficiently and effectively, and binding potential clients directly to the bank.

In the past, the Rabobank's client advisors were responsible for most client interaction: most client requests and questions were addressed by client advisors. Teller personnel only dealt with relatively simple requests. The result of this strategy was that client advisors were spending a large percentage of their time on relatively simple requests.

As, in today's society, clients and potential clients are more inclined to switch between service providers than in the past, depending on the service level provided and the cost involved, this strategy is not sufficiently effective. The task of client advisors has become more proactive: to focus both on finding new clients and satisfying existing client needs.

2.2 The Organisational Solution

Part of the solution the Rabobank has adopted is 24 hour a day availability together with new procedures aimed at binding clients directly to the bank. The 24 hour a day service is provided

by a call centre that (automatically) takes over the operation of the local banks after normal business hours.

Clients' requests and questions can be divided into three categories:

- simple questions that can be answered directly by teller personnel, e.g. a question about the current advertised interest rates for the different types of savings accounts the bank offers.
- simple questions and requests that can be handled right away without any further contact with the client but require further processing, e.g. a request for new cheques.
- complex questions and requests which need the attention of a client advisor, e.g. an inquiry about a mortgage.

The assumption behind 24 hour a day service is that clients will be less likely to shop around and take their business elsewhere, if their requests are taken seriously. Operators of the call centre have been trained to deal with the relatively simple client requests. These simple requests (the first two types) amount to about 70 percent of the calls. This approach reduces the number of simple requests client advisors need to address, leaving more time for other activities such as, for example, more complex client requests or client acquisition. Operators schedule appointments for clients with complex requests with a qualified client advisor. It is important that the appointment takes place as soon as possible.

As local banks are autonomous, the agendas of client advisors of the local banks are not directly available to the call centre. The overall procedure employed is as follows:

1. A client or a potential client calls the local bank and the call is redirected to the call centre.
2. If the request of the client is relatively simple, the operator deals with it right away.
3. If the request of the client is more complex and needs to be serviced by a client advisor, the computer system of the operator contacts a computer system at the local bank with a request for service.
4. The computer system at the local bank determines if this request can be serviced and suggests a number of possible appointments with the client. The client can choose one of these appointments. The computer system at the local bank can do this by selecting an appropriate procedure to service the request and schedule the activities of that procedure in the agendas of the employees of the bank

Within the Rabobank procedures have been defined for most types of client requests. These procedures are all specified in a process definition language, defining the workflow within the organisation. In this paper a simplified version of the procedure for dealing with requests

related to financing consumer expenditure is used to illustrate the types of activities (and the relations between activities) to be scheduled. The procedure for dealing with requests related to financing consumer expenditure, as specified by the Rabobank, consists of 22 activities of which 8 are completely automated (including, for example, information retrieval, calculations and provision of standard contract conditions). In short three types of loans can be provided for consumer expenditure: personal loans, revolving credit, and student loans. This paper focuses on three (groups of) activities within this procedure:

- Client advise (requires activities such as the acquisition of information on credit rating and financial status (including current income and expenditure), analysis of available information, decision with respect to maximum loan, overview of possible options)
- Written agreement (requires additional information from the client and possibly other sources, possibly requiring approval by authorised persons depending on factors such as amount and risk involved)
- Administrative transaction.

Employees are fully responsible for their own agendas. They can refuse or change appointments in their agendas. Changes to an employee’s agenda should take the profile and wishes of an employee into account, e.g., the employee’s preferences for specific types of activities, the employee’s capabilities and authorisation with respect to specific activities, the employee’s preferences for allocation of specific activities to specific times of day, or the employee’s availability (e.g., due to holidays or illness).

3 A Multi-Agent System Approach to the Design of a Call Centre Support System

The problem description clearly defines the problem as a distributed problem: one call centre services several local banks as depicted below in Figure 1.

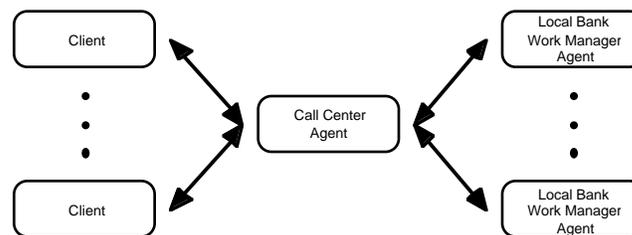


Figure 1 One call centre for several local banks

As described above in Section 2, the clients, the local banks and the call-centre are autonomous, distributed entities: entities responsible for their own internal processes in interaction with (and in response to) other entities. As described the entities involved fulfil the characteristics of weak agency proposed by (Wooldridge and Jennings, 1995): autonomy (all agents are in full control of their own processes), social ability (all agents are able to communicate and co-operate with other agents), pro-activeness (all agents are able to initiate processes independently and take the initiative to initiate new processes when necessary) and reactiveness (all agents are able to respond to new incoming information. In fact, to model the activities involved, a more detailed analysis of the actors involved is required: in this example the local banks not only have a Work Manager but also Employees. A multi-agent system has been designed to provide this functionality.

3.1 System Design: Additional Requirements

To perform the tasks distinguished above in Section 2 appropriately, the system as a whole, needs to satisfy the following requirements:

1. The system needs to be able to cope with changes made by employees. An employee can change his/her agenda, s/he can:
 - a) refuse to perform a specific activity.
 - b) refuse to perform it within a specific period of time.
 - c) reschedule his/her agenda.
 - d) delay some of the activities.
2. The system needs to know :
 - a) the capabilities of the employee.
 - b) the preferences of the employee (e.g., time periods during which the employee does or does not want to perform specific kinds of activities).
 - c) the availability of the employee (e.g., holidays).
3. The system may only reschedule the agenda of an employee in a way that respects the profile of the employee.
4. The system needs to be able to interact with the each and every employee's agenda.

3.2 The Agents and their Interaction

To acquire the functionality required (24 hour availability), namely to effectively and efficiently be able to schedule activities within a procedure, with the agents distinguished above, employees would need to be available 24 hours a day. As this, in practice, is clearly undesirable, automated Personal Assistant agents have been introduced, to support the Employees. As a result, five types of agents are distinguished: the Clients, the Call Centre Agent, Work Manager agents (one for each local bank), Personal Assistants (one for each Employee) and Employees.

Client

Clients (or potential clients) identify a need to consult their bank and call their local bank with a request. Their calls are automatically redirected to a Call Centre. A client's request is either answered directly by the Call Centre Agent or, if a client's request can not be answered directly, the Call Centre Agent proposes a set of possible appointments from which the client can choose. The client's choice is the agreed appointment.

Call Centre Agent

Relatively simple requests are handled by the Call Centre Agent directly. Other, more complex, requests are forwarded to the Work Manager of an appropriate local bank by the Call Centre. The Work Manager, in turn, informs the Call Centre Agent of a number of possible appointments for the client with a client advisor. In this case the Call Centre Agent proposes these options to the client.

Work Manager

After interpretation of a request, Work Manager selects an appropriate procedure for the request and creates a schedule for the activities in that procedure. The Work Manager communicates requests for commitment to the Personal Assistants of selected Employees. The Work Manager may have to adapt its schedule (or even choose another procedure) depending on the information received from the Personal Assistants. If the proposed schedule is accepted by all relevant Personal Assistants, the Work Manager determines when appointments can be made and proposed this set of possible appointments

Personal Assistant

Given a request for commitment a Personal Assistant checks the agenda and profile of the Employee it represents and (possibly) reschedules the Employee's agenda. The Personal Assistant informs the Work Manager of the degree to which it is possible to reschedule the Employee's agenda to meet the schedule the Work Manager has proposed. As a result of a request from the Work Manager the Personal Assistant informs its Employee of the contents of his/her agenda and of changes made in the agenda. An Employee can also decide to change the contents of his/her agenda and/or profile for other (self-imposed) reasons. Based on the information the Personal Assistant receives from the Employee, the Personal Assistant updates the Employee profile and the agenda. If previous commitments made by the Personal Assistant cannot be met due to new appointments or modifications made by the Employee the Personal Assistant informs the Work Manager.

Employee

Employees check their agenda regularly, make changes to it according to their own needs and preferences, and execute the activities as scheduled in their agenda.

The initial, prototype multi-agent system designed to support the call centre approach is depicted in Figure 2. Note that, in addition to the seven agents (one Client, one Call Centre Agent, one Work Manager, two Personal Assistants and two Employees), this model also includes a global clock to ensure that the schedules of the Personal Assistants are synchronised with the schedules of the Work Manager.

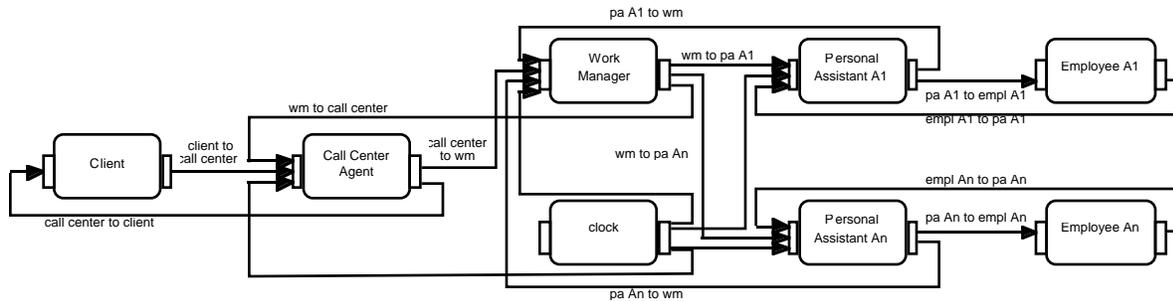


Figure 2 A multi-agent system for a local bank

Not only does Figure 2 depict these agents, it also depicts information exchange between the individual agents, as described below. More detailed descriptions of a Work Manager, a Personal Assistant and (an interface for) an Employee, are provided in Sections 4, 5 and 6.

Communication between the agents is modelled explicitly on the basis of the information links (the arrows in Figure 2) between the agents. The language employed in the interfaces of the agents (the small boxes on the left and right side of the agents) is the basis for the communication language. It is based on order-sorted predicate logic, similar to standard knowledge interchange formats such as KIF (Genesereth and Fikes, 1992). The ontology used for communication can make use of predefined generic communication predicates, but also can be defined depending on the application. In Section 4 more details are given on this communication language.

4 Work Manager

To successfully develop a support system for co-operation in a complex, dynamic and not always predictable environment, a well-defined and transparent model of co-operation is required: a model that is robust and flexible enough to cope with unexpected events. In

Jennings' model of co-operation (Jennings, 1995), agents are capable of organising and monitoring projects to reach given goals. To this purpose, one agent first determines a set of activities to reach a given goal and the temporal dependencies between these activities. The organising agent then identifies which other agents are capable of performing which activities. In interaction with these agents, the organising agent determines which agents are willing and able to participate in the project. On the basis of this information (the activities to be performed, the order in which the activities are to be performed, and the deadline), the organising agent tries to put together a project team and a project schedule (called a *recipe*). The creation of this recipe is an iterative process requiring interaction with the other agents on their own schedules (including, for example, information on commitments to other projects). When completed, the recipe is sent to all participants, and the project commences.

Once committed, each participating agent (including the organiser) receives the final recipe, and is committed to the relevant time interval in the recipe. Each agent has the same obligation towards the project: each member monitors the progress of the project and is equally responsible for its success. If a team member discovers a problem that endangers the project, s/he informs all relevant participants. One of the agents (e.g., the project manager) can then take the initiative to modify the project plan, to create a new project for the same goal, or to inform all relevant participants that the goal is unattainable or that the goal is no longer valid.

How the informally described co-operation model of (Jennings, 1995) was formally specified within DESIRE is shown in (Brazier, Jonker and Treur, 1997); an application of this formalized co-operation model to project co-ordination was presented in (Brazier, Jonker, and Treur, 1996). The multi-agent system to support a Call Centre presented in this paper is based on this generic co-operation model. At the top level of the Work Manager agent only four components of the generic agent model from (Brazier et al., 1997) are used; see Figure 3: agent interaction management, own process control, maintenance of agent information, and cooperation management. Each of these components is discussed in more detail below.

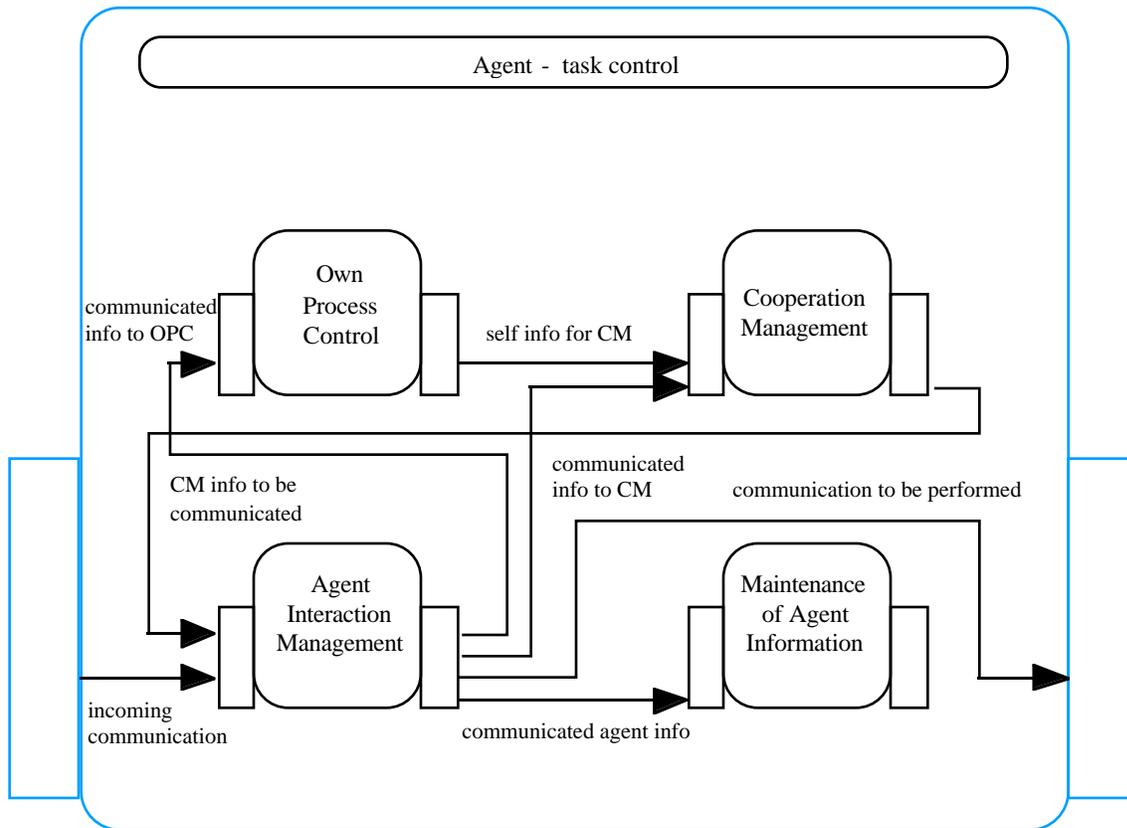


Figure 3 The part of the generic agent model reused

As described in Sections 2 and 3, a Work Manager is free to decide whether or not to accept a request communicated by a Call Centre Agent. If, after interpretation of a request, a Work Manager decides to accept the request, this request is translated into a goal for the Work Manager to adopt. This part of the process of the Work Manager is modelled within the Work Manager's component own process control. To achieve the adopted goal, co-operation with Personal Assistants is required. Within the component cooperation management the Work Manager selects an appropriate plan (the procedure to which the Bank's description of the application domain referred) for this goal and determines a schedule for the activities in the procedure. The Work Manager asks Personal Assistants of those Employees selected to execute the schedule (this communication is managed by its component agent interaction management) whether or not they can commit to specific activities. If a proposed schedule is accepted by all relevant Personal Assistants, the Work Manager selects a set of possible appointments with the client and communicates this set to the Call Centre Agent. If not, the Work Manager needs to either adapt its schedule or choose another procedure, depending on the information communicated by the Personal Assistants. This process is discussed in more detail in this section.

4.1 Agent Interaction Management

The component agent interaction management is composed of two components: incoming communication management and outgoing communication management. A Work Manager can communicate with either the Call Centre Agent or a Personal Assistant.

4.1.1 Management of incoming communication

Within the component incoming communication management incoming communication is analysed and communicated information identified. For example, the following types of information can be identified:

- a new request (from a Call Centre Agent)
- commitment: a Personal Assistant commits to activity A, with deadline D, priority P, earliest starting time E, and latest starting time L.
- conditional commitment: a Personal Assistant commits to the request under the condition that the Work Manager relieves it of the commitment to activity A' that has a lower priority P' than activity A.
- refusal: a Personal Assistant cannot commit to activity A, with deadline D, priority P, earliest starting time E, and latest starting time L.
- a progress report on already scheduled activities (from a Personal Assistant); e.g., a reported delay.

Monitoring information communicated by a Personal Assistant in the form of a progress report specifies whether the Personal Assistant expects its employee to be able to perform a specific activity A (within a given time slot). A Personal Assistant can report, e.g., that a commitment A cannot be kept because no start has been/will be made at time L, the deadline D will not be met, or the necessary information/material regarding A is not available.

Depending on the type of information received, the implications for information to be provided to appropriate component(s) within the Work Manager are identified. For example, communicated information on a new request from a Call Centre Agent is needed by the Work Manager's component own process control, whereas the other two types of communicated information listed above are needed by the component cooperation management. Conclusions of the form `new_own_process_info(I:INFO_ELEMENT, S:SIGN)` are transferred to the component own process control by the information link `communicated info to OPC`, and conclusions of the form `new_cooperation_info(I:INFO_ELEMENT, S:SIGN)` are transferred to the component cooperation management by the information link `communicated info to CM`.

4.1.2 Management of outgoing communication

The component outgoing communication management prepares the following types of outgoing information:

- appointment proposals (to a Call Centre Agent)
- commitment requests (to a Personal Assistant)
- commitment confirmations (to a Personal Assistant)

The information to be communicated is provided by the component cooperation management, and transferred to agent interaction management through the information link cooperation info to AIM. Preparation of communication includes, for example, labelling outgoing communication so that the agents that receive the information can refer to this information in their reply. Conclusions of the form `to_be_communicated_to(I:INFO_ELEMENT, S:SIGN, PA:PERSONAL_ASSISTANT, I:CID)` drawn by agent interaction management are transferred to the output interface of the Work Manager through the information link `info to be communicated`, and from the agent's output interface to the input interfaces of the agents to which the communication is directed (as discussed in Section 3).

4.2 Own Process Control

The component own process control has two sub-components: determine goals and commitments and maintenance of own characteristics. Within the component determine goals and commitments requests communicated by the Call Centre Agent are analysed and the decision whether or not to accept a request (and as a consequence adopt a goal to respond to the request as one of the Work Manager's own goals) is made. The agent's own characteristics are explicitly represented within the component maintenance of own characteristics. Examples of specific agent characteristic are: that a request for a certain type of client (e.g., for a known client) is to be given priority, or that requests for credit card services are processed with higher priority than student loan services.

4.3 Maintenance of Agent Information

Within the component maintenance of agent information the Work Manager maintains information on the capabilities and preferences of the other agents. One example of information maintained by the Work Manager is the information a Work Manager has on the activities for which each Personal Assistant can be approached.

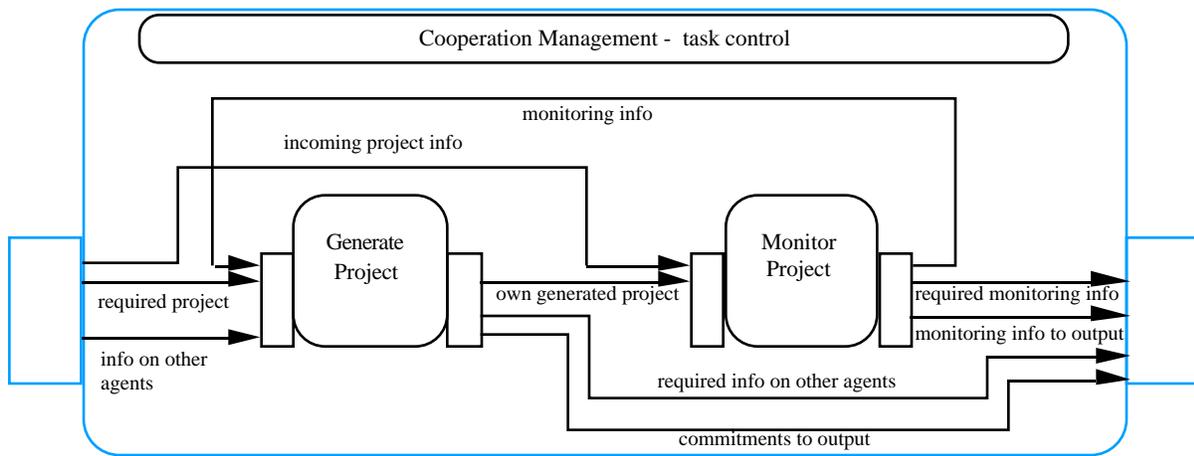


Figure 4 Cooperation Management

4.4 Co-operation Management

The component cooperation management is a composed component responsible for all processes concerning projects, project commitments and co-operation. It consists of two components; one for the generation of projects and one for the monitoring of existing projects (see Figure 4). The interaction between the components of cooperation management and cooperation management's environment, and between the components of cooperation management is defined by the links (the arrows) depicted in Figure 4.

cooperation management needs the following types of information:

- goal, deadline, and necessary activities: to create a new project
- capabilities of other agents: to find participants for a project
- commitments of other agents: to build a joint recipe
- observation information: to monitor existing projects
- communicated project information: to monitor existing projects

cooperation management provides the following types of information:

- recipe elements relevant for possible participants
- joint recipe relevant for all participants
- monitoring information relevant for all participants

Each of its components is discussed in more detail in this section.

4.4.1 Generate Project

The component generate project is composed of two components: prepare project commitments, and generate and modify project recipes. Links are defined to regulate the interaction between generate project's components and its environment, see Figure 5.

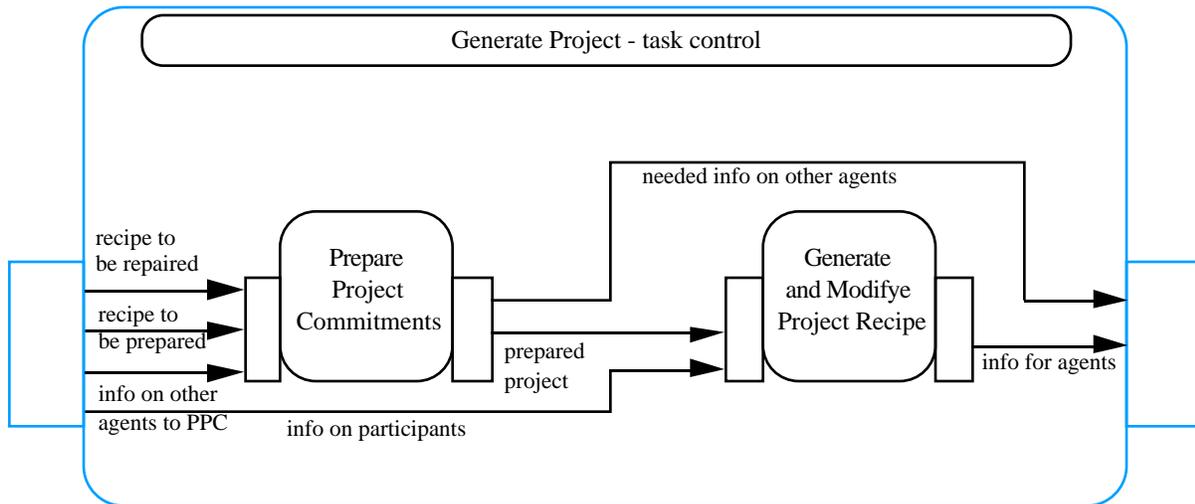


Figure 5 Generate Project

Within the component generate project the component prepare project commitments receives the Work Manager's own goals. The component prepare project commitments's aim is to determine procedures that can be followed to achieve a given goal. This is performed using knowledge relating requests to procedures, for example of the following form:

```

if          own_goal(appointment_request(service(credit_card)))
then       selected_procedure(procedure(cp2))

```

To determine which activities are required to execute the procedure, knowledge is used that relates procedures to activities, and knowledge that defines duration of activities, and temporal relations between them; for example knowledge of the form:

```

if          selected_procedure(procedure(cp2))
then       selected_activity(activity(a1), duration(5))
and        selected_activity(activity(a2), duration(3))
and        precedence(activity(a1), activity(a2))

```

Within the component generate project the component generate and modify project recipe receives the selected activities, their duration and temporal relations between them. It determines which Personal Assistant is capable of taking responsibility for a given activity (using agent

information maintained in maintenance of agent information) and proposes a schedule. This process involves intensive interaction with the Personal Assistants and may iterate a number of times, until the proposed schedule is accepted by all participating Personal Assistants.

4.4.2 Monitor Project

Progress of the procedure is monitored within the component monitor project. The component monitor project is composed of two components: assess viability and determine consequences. These components and the links for interaction within monitor project are depicted in Figure 6. If substantial deviations from the committed schedule are identified, this component determines which actions are to be undertaken. The component assess viability monitors the viability and validity of the procedure. To monitor the process assess viability uses information on the progress of the activities of the participants; it can also actively formulate requests for information. The component determine consequences interprets assess viability's monitoring results. The component determine consequences issues requests to find new recipes or to adapt existing recipes, to the component project generation of cooperation management. The component determine consequences also determines when a goal should be withdrawn (for example, because the goal is unattainable, or the goal has been reached) and prepares and issues communication to that effect to each participating Personal Assistant. The process of monitoring is described in more detail in Section 7.

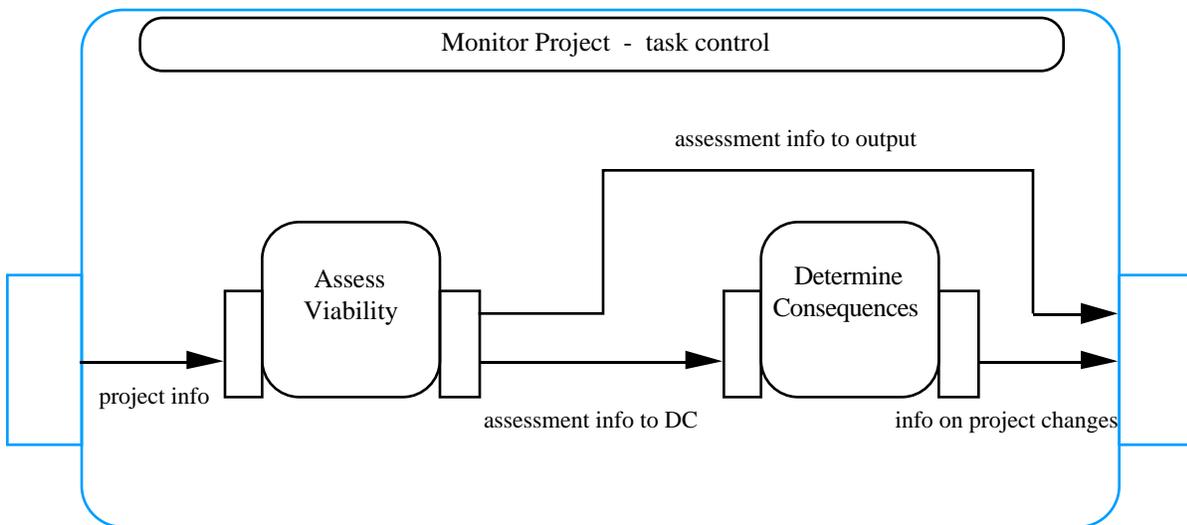


Figure 6 Monitor Project

5 Personal Assistant

All Personal Assistants are also modelled as a refinement of (the part of) the generic model of a co-operative agent depicted in Figure 3, consisting of the four components: own process control, agent interaction management, cooperation management, and maintenance of agent information. A Personal Assistant communicates with both the Work Manager and the Employee.

5.1 Interaction with the Work Manager

A Personal Assistant (PA) receives requests from a Work Manager for:

1. a commitment to a specific activity A before a certain deadline
2. (possibly with) additional information on the importance of the activity (priority P), the earliest starting time (E) and the latest starting time (L).
3. cancellation of a commitment
4. monitoring information on a specific activity A.

5.1.1 Requests for commitment and cancellation

Incoming communication in the form of requests for commitment (from the Work Manager), is analyzed and the relevant communicated information is identified and classified (comparable to the process described in Section 4.1). An identified request for commitment is transferred from the component agent interaction management to the component own process control. The component own process control decides whether or not to accept a request as a goal for the Personal Assistant (see Section 4 for further explanation). If a request is accepted, the request information is transferred to component cooperation management. The component cooperation management is composed as depicted in Figures 4, 5, and 6. The component prepare project commitments determines whether or not the Employee the Personal Assistant represents is capable of performing the activity, and the component generate and modify schedule determines whether a new schedule can be generated in which the requested commitment can be awarded. If a new schedule can be generated, this schedule is forwarded to the component monitor project. The component monitor project uses information about the schedule, commitments to identify contradictions and to take appropriate action.

If a new schedule cannot be generated by the component generate project without changing existing commitments (with information acquired by the component prepare project commitments), information about the nature of the conflict is transferred to the component agent interaction management: commitment can only be acquired if a commitment with lower priority is cancelled, otherwise given the current priorities and schedule, commitment is not possible. The component agent interaction management manages the communication on the issue with the Work Manager.

5.1.2 Requests for monitoring information

A Personal Assistant also receives requests for monitoring information. These requests are identified by the component agent interaction management and transferred to the component cooperation management. Within the component cooperation management, the component monitor project is responsible for monitoring a project, and providing the necessary information (through the output interface of the component cooperation management) to the component agent interaction management to communicate to the Work Manager. To monitor a project the component monitor project requires information on the current status of a schedule. This is transferred to the output interface of the component cooperation management, and from there to agent interaction management, which manages the communication with the Employee.

5.2 Interaction with the Employee

The Personal Assistant requests information about current commitments and schedules from the Employee. These requests are devised by the component monitor project within the component cooperation management. In addition, the component receives information communicated by its Employee without having initiated interaction. An Employee may provide information on changes in his/her schedule without having been explicitly requested to do so. This information is identified in agent interaction management and transferred to the component cooperation management. Within the component cooperation management the component monitor project detects possible new conflicts.

6 The Employee

Within the multi-agent system the Employee only interacts with his/her Personal Assistant, as shown in Figure 2. Interaction with the Employee is modelled by the interface in the prototype system. The interface, see Figure 7, presents the contents of the agenda as received from the Personal Assistant. The interface allows for the Employee to make changes to the agenda. These changes are communicated to the Personal Assistant. In addition, the Employee can make changes to the Employee profile maintained by the Personal Assistant.

Agenda		
<p>Monday, November 3</p> <p>09.00 Preparing mortgage contract for Mrs. Robinson</p> <p>13.00 Lunch</p> <p>14.00 Check Mr. Pepper</p> <p>18.00</p>	<p>Tuesday, November 4</p> <p>09.00 Check Mr. Pepper</p> <p>13.00 Lunch</p> <p>14.00 Acquire information on Mrs. Martelli</p> <p>18.00</p>	<p>Wednesday, November 5</p> <p>09.00 Acquire information on Mrs. Martelli</p> <p>13.00 Lunch</p> <p>14.00 Check Mrs. Martelli</p> <p>18.00</p>
<p>Thursday, November 6</p> <p>09.00 Check Mrs. Martelli</p> <p>13.00 Lunch</p> <p>14.00 Authorize student loan for Mrs. Martelli</p> <p>18.00</p>	<p>Friday, November 7</p> <p>09.00 Authorize student loan for Mrs. Martelli</p> <p>13.00 Lunch</p> <p>14.00 Appointment with Mrs. Robinson for her mortgage</p> <p>15.00</p> <p>18.00</p>	<ul style="list-style-type: none"> ▶ your profile ▶ next week ▶ previous week ▶ month overview ▶ help

Figure 7 Interface to Employee

By double clicking on an agenda item more information becomes available. For example, by double clicking on the item Monday, November 3, a new window pops up displaying the deadline for the activity, the priority, the latest possible starting time, which other Employee will provide the necessary information, and the client’s identification (e.g., bank account number).

The item in the bottom right rectangle can also be opened. For example, by clicking on the item “your profile” the window in Figure 8 pops up.

Employee profile		
<p>Capabilities:</p> <p>business loan contract</p> <p>morgage contract</p> <p>client appointments</p>	<p>Preferences</p> <p>Between 09.00 and 11.00: only activities of type S</p> <p>Rather not activities of type X</p> <p>No appointments with Mr. Brute</p>	<p>Authorizations</p> <p>Signing of contracts up to Level 2</p> <p>Access to information up to level 2</p>

Figure 8 Employee profile

The Employee profile can be modified by pull-down menus. For example, the Employee can choose from a standard list of capabilities of bank employees.

7 Description of an example process

An appointment for a client is scheduled in two phases. In the first phase, the Work Manager decides upon an appropriate procedure for the preparation of an appointment. In the second phase, Personal Assistants are asked to perform activities of the selected procedure. Section 7.1 discusses the 2 phases. Section 7.2 discusses monitoring and control, i.e., detecting potential and actual risks to the schedule and eliminate these risks.

7.1 The Selection of a Procedure

The Work Manager receives a request from the Call Centre in the following form:

```
appointment_request(c1, service(student_loan), urgency(high), priority(low))
```

The Call Centre Agent requests an appointment for client c1 concerning a student loan, which is highly urgent for the client has a low priority for the bank. The Work Manager first determines its intentions with respect to this request. If it decides to honour the request, the Work Manager creates a goal for itself that represents the request. The following goal of the Work Manager is an example:

```
goal(arrange_appointment(c1, service(student_loan), urgency(high), priority(low)))
```

Reasons to honour the request could, for example, be:

- To service the client (which is the main reason to be co-operative towards requests of the Call Centre Agent)
- Low workload
- Increase profit for the part of the organisation for which the Work Manager is responsible

Reasons not to honour the request could, for example, be:

- Very high workload and low priority request
- Lack of expertise in the request area (another Work Manager would perhaps be more qualified to deal with the request)

As soon as the goal has been set, an appropriate procedure needs to be selected for the goal. To service a request, the bank can use different procedures. These procedures vary in, for example, costs, required resources, and duration. Which procedure is chosen will depend on a number of factors, including workload, cost/benefit, urgency of the client. The example procedure described in Section 2 is used in this trace. The procedure selected is the consumer

finance procedure, instantiated to service a student loan. Once the component own process control has derived the goal for the Work Manager, the component cooperation management selects and instantiates the proper procedure.

The goal is accepted by the Work Manager and transferred to the component cooperation management. The component prepare project commitments first selects a procedure that achieves the goal. The following rule is used to this purpose.

```
if      own_goal(choose_appointment(C: Clients, service(student_loan), U: Urgencies, priority(low)))
then    selected_procedure_instance(consumer_finance, student_proc)
```

The selection of the procedure implies a number of activities that need to be performed:

```
if      selected_procedure_instance(consumer_finance, student_proc)
then    selected_activity(activity(advise_client), duration(30), role(client_service_emp))
        and selected_activity(activity(close_deal), duration(30), role([client_advisor, client]))
        and selected_activity(activity(administrative_closure), duration(30), role(client_service_emp))

        and precedence(activity(advise_client), activity(close_deal), finish_start)
        and precedence(activity(close_deal), activity(administrative_closure), finish_start)
        and ultimate_deadline(deadline(1440))
```

Note that this procedure only includes the activities to be executed by the relevant local bank. It consists of three half-hour activities that need to be finished within 3 days. The roles specify the capabilities and authorisation that people need to have to perform the activities. Note that for the activity `close_deal`, the client needs to be present. The component cooperation management needs to find employees whom can fulfil the roles required by the procedure and queries its agent model (maintenance of agent information):

```
request(role(client_service_emp))
request(role(client_advisor))
```

These requests result in the following knowledge:

```
role_of(PA1, client_service_emp)
role_of(PA2, client_service_emp)
role_of(PA2, client_advisor)
```

I.e., the employee represented by Personal Assistant PA1 has the role `client_service_emp` and the employee represented by Personal Assistant PA2 has both roles. It is obvious that the Call Centre Agent has information about the client, thus no role information about the client needs to be retrieved.

All of this information is transferred to the component of cooperation management, and then to its internal component generate and modify project recipe. There, the selected activities are instantiated and commitment requests are generated. These requests are communicated to the Personal Assistants, together with the entire procedure: if the Personal Assistants know the entire plan and can schedule more intelligently. Instantiations of activities are denoted by an activity name followed by a number; e.g., advise_client-1. The Work Manager communicates a commitment request to PA1 and PA2:

```
commitment_request(PA1, activity(advise_client-1), duration(30), role(client_service_emp))
commitment_request(PA1, activity(administrative_closure-1), duration(30), role(client_service_emp))
commitment_request(PA2, activity(advise_client-1), duration(30), role(client_service_emp))
commitment_request(PA2, activity(close_deal-1), duration(30), role(client_advisor))
commitment_request(CC, activity(close_deal-1), duration(30), role(client))
commitment_request(PA2, activity(administrative_closure-1), duration(30), role(client_service_emp))
```

These requests are communicated to both Personal Assistants and the Call Centre Agent, together with the entire instantiated procedure. All agents are then completely informed. Each of the agents transfers this knowledge to its own cooperation management component. The agents look in their agendas and reply (again to all parties involved):

```
commitment_reply(PA1, activity(advise_client-1), earliest_start(300), earliest_finish(400))
commitment_reply(PA1, activity(administrative_closure-1), earliest_start(430), earliest_finish(460))
commitment_reject(PA2, activity(advise_client-1))
commitment_reply(PA2, activity(close_deal-1), earliest_start(100), earliest_finish(130))
commitment_reply(CC, activity(close_deal-1), earliest_start(200), earliest_finish(230))
commitment_reply(PA2, activity(administrative_closure-1), earliest_start(130), earliest_finish(160))
```

The Personal Assistants and the Call Centre Agent obviously answered without knowing what the others would do. No valid schedule is yet possible. The first step, however, is clear: advise_client-1 will need to be done by PA1. The Work Manager confirms this by communicating:

```
commitment_accept(PA1, activity(advise_client-1), earliest_start(300), earliest_finish(400))
```

With this new information, the agents provide the following replies:

```
commitment_reply(PA1, activity(administrative_closure-1), earliest_start(430), earliest_finish(460))
commitment_reject(PA2, activity(advise_client-1))
commitment_reply(PA2, activity(close_deal-1), earliest_start(400), earliest_finish(430))
commitment_reply(CC, activity(close_deal-1), earliest_start(400), earliest_finish(430))
commitment_reply(PA2, activity(administrative_closure-1), earliest_start(430), earliest_finish(460))
```

The Work Manager accepts the commitments of PA2 and the Call Centre Agent by communication of:

```
commitment_accept(PA2, activity(close_deal-1), earliest_start(400), earliest_finish(430))
commitment_accept(CC, activity(close_deal-1), earliest_start(400), earliest_finish(430))
commitment_accept(PA2, activity(administrative_closure-1), earliest_start(430), earliest_finish(460))
```

After this the schedule is fixed. Note that PA2 again rejected `advise_client-1`. This is not necessary. It is even possible that PA2 changes its mind if it sees that the procedure cannot be scheduled unless he takes on this activity. These changes of mind are allowed and cause backtracking to occur.

7.2 Monitoring and Control

What happens when an employee cannot meet his or her deadline, depends on monitoring and control. The employee knows the schedule for the entire procedure, thus he or she can see the effects on other activities within the procedure. For example, if the employee represented by PA1 is delayed with the activity `advise_client-1`, the activity `close_deal-1` will need to be delayed as well, and this impacts both the client and the client advisor (represented by the Call Centre Agent and PA2). Suppose, for example, that at time 340 (an hour before the appointment), the employee represented by PA1 becomes ill and goes home. At this point in time, PA1 notifies the Work Manager and PA2 (note that the client represented by the Call Centre Agent is not on line and would need to be called if the appointment (`close_deal-1`) needs to be delayed).

```
commitment_fail(PA1, activity(advise_client-1))
```

Basically, scheduling starts from scratch; all parties involved try to maintain their original commitments as much as possible and especially if appointments are involved. The Work Manager repeats its request to PA2:

```
commitment_request(PA2, activity(advise_client-1), duration(30), role(client_service_emp))
```

PA2 replied in the first scheduling round that it did not want to do `advise_client-1`. Now, PA2 again must decide whether or not to agree to taking on this activity, assuming there is time for it. PA2 accepts, not wanting to move the existing appointment with the client.

```
commitment_reply(PA2, activity(advise_client-1), earliest_start(370), earliest_finish(400))
```

The Work Manager accepts this commitment:

```
commitment_accept(PA2, activity(advise_client-1), earliest_start(370), earliest_finish(400))
```

8 Discussion

More and more organisations offer a 24 hour a day telephone service using a call centre to coordinate the service provided. Without support to really support clients, by, for example, being able to schedule appointments with a client, such a service is of limited value: only simple questions can be answered. This paper has presented a multi-agent system, introduced to

increase the value of 24 hour a day service by supporting call centres in making appointments and scheduling activities of employees in preparation of such appointments. This multi-agent system architecture has been applied to the banking domain, in co-operation with (and partially funded by) the Rabobank, one of the largest banks in the Netherlands. In this system scheduling is a distributed effort: each Employee is represented by its own Personal Assistant agent (that also maintains the Employee's agenda), and a Work Manager agent co-ordinates the schedules, and the client's requirements (through the Call Centre Agent).

The multi-agent system architecture described in this paper is based on a principled design, using the compositional development method for multi-agent systems DESIRE. Jennings' co-operation model (Jennings, 1995) for projects based on joint intentions, formalised in (Brazier, Jonker and Treur, 1997), and applied to project co-ordination in (Brazier, Jonker and Treur, 1996), provided the generic model upon which system design is based. DESIRE models specify processes and knowledge at different levels of abstraction. Information exchange between processes and process sequencing are explicitly defined at each of the levels distinguished. Different levels of abstraction within the knowledge composition structure information types and knowledge bases. The more specific model developed in this project refined the generic co-operation model both by extending generic knowledge, and by adding domain specific knowledge for the application at hand.

COOL (Barbuceanu & Fox, 1995) is an agent co-ordination language that focuses on the specification of co-ordination between agents (such as the co-ordination required in a supply chain, (Barbuceanu & Fox, 1996)). It uses finite state machines to describe the flow of communication between agents based on a fixed number of speech acts, such as propose, accept, and reject. The main difference between DESIRE and COOL is that COOL focuses on a specific way of modelling co-ordination between agents and not on the architecture of an agent itself, while DESIRE provides generic models to specify agent architectures, without prescribing specific protocols for specific functionality, such as, for example interaction between agents. The method used by COOL to describe the co-ordination between agents could be used to model the co-ordination in the Cooperation Management component of the DESIRE agent. The designer is however free to chose another coordination method.

The ADEPT model (Advanced Decision Environment for Process Tasks; see (Jennings, Faratin, Norman, O'Brien, Wiegand, Voudouris, Alty, Miah, and Mamdani, 1996)) is, in some ways, comparable to the refined model presented in this paper. The ADEPT model models business processes by a hierarchy of cooperative agents. The hierarchy ensures that communication overhead between agents and the autonomy of the agents are balanced. Within this model, agents have the following modules:

- a communication module
- an interaction management module (IMM)
- a situation assessment module (SAM)
- a service execution module (SEM)
- a self model (SM)
- acquaintance models (AM)

These modules correspond to the components within the generic DESIRE agent model: the module IMM may be viewed as the component cooperation management, the SAM may be viewed as part of the component own process control, the SEM is clearly related to the Agent Specific Task. The SM is also part of the component own process control and the module AM can be viewed as the component maintenance of agent information.

Also the ZEUS architecture of a generic agent (Nwana, Ndumu and Lee, 1998) is, to a certain degree, comparable to the generic DESIRE agent model. The ZEUS model distinguishes:

- Mailbox
- Message Handler
- Co-ordination Engine
- Execution Monitor
- Acquaintance Model
- Planner and Scheduler
- Task/Plan Database
- Resource Database

The Mailbox and the Message Handler together correspond to the component agent interaction management within the generic DESIRE agent model. The Co-ordination Engine is modelled by the component cooperation management. The Execution Monitor with the Planner and Scheduler, and the Task/Plan Database together provide the functionality provided by the component own process control. The Acquaintance Model is comparable to the component maintenance of agent information. Although interaction with the External World is not explicitly modelled within a ZEUS agent, the Resource Database may include some of this information. The ZEUS agent does not include models for specific types of tasks, but focuses on reusable components for interaction based on standard interface protocols.

The current DESIRE environment does not explicitly support standard communication languages, such as KQML (Finin, Labrou, and Mayfield, 1997) and the FIPA communication

language (FIPA, 1998), but allows the system designer the freedom to specify the language in which communication and co-ordination are expressed, as desired. The specification of standard components to support standard communication languages is possible within the environment.

Acknowledgements

The research reported here has been funded by Rabobank, Telemachos Planning and the Ministry of Economic Affairs. Close collaboration with domain experts within the Rabobank, and with Pieter van Langen (Rabofacet), Patrick Frants (MSc student) and Guy Kornblum (Telemachos Planning) in particular, has been of utmost importance to this research. Their contribution is greatly appreciated.

References

Barbuceanu, M. and Fox, M.S. (1995). COOL: A Language for Describing Coordination in Multi Agent Systems. In: V. Lesser (ed.), Proc. of the First International Conference on Multi-Agent Systems, ICMAS'95, MIT Press, Cambridge, MA, pp. 17-24.

Barbuceanu, M. and Fox, M.S. (1996). Coordinating Multiple Agents in the Supply Chain. In: Proceedings of the Fifth Workshops on Enabling Technology for Collaborative Enterprises, WET ICE'96, IEEE Computer Society Press, pp. 134-141.

Bradshaw, J. (1997) (e.d). Software Agents. AAAI Pres, 1997.

Brazier, F.M.T., Dunin-Keplicz, B.M., Jennings, N.R. and Treur, J. (1995). Formal specification of Multi-Agent Systems: a real-world case. In: V. Lesser (ed.), Proc. of the First International Conference on Multi-Agent Systems, ICMAS'95, MIT Press, Cambridge, MA, pp. 25-32. Extended version in: International Journal of Cooperative Information Systems, M. Huhns, M. Singh, (eds.), special issue on Formal Methods in Cooperative Information Systems: Multi-Agent Systems, vol. 6, 1997, pp. 67-94.

Brazier, F.M.T., C.M. Jonker, J. Treur (1996). Modelling Project Coordination in a Multi-Agent Framework. In: Proceedings of the Fifth Workshops on Enabling Technology for Collaborative Enterprises, WET ICE'96, IEEE Computer Society Press, 1996, pp. 148-155

Brazier, F.M.T., Jonker, C.M., Treur, J. (1997). Formalisation of a cooperation model based on joint intentions. In: (Müller, Wooldridge and Jennings, 1997), pp. 141-155.

Finin, T., Labrou, Y., and Mayfield, J. (1997), KQML as an agent communication language. In: (Bradshaw, 1997), pp. 291-316.

FIPA (1998). <http://www.cselst.stet.it/fipa>.

Genesereth, M.R., Fikes, R.E. (1992). Knowledge Interchange Format - version 3 - reference manual. Technical Report Logic Group, Logic-92-1, Stanford University, Stanford, CA.

Jennings, N.R. (1995). Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems using Joint Intentions, *Artificial Intelligence Journal* 74 (2), pp 195-240.

Jennings, N.R., Faratin, P., Norman, T.J. O'Brien, P. Wiegand, M. E. Voudouris, C., Alty, J. L., Miah, T. and Mamdani, E. H. (1996). ADEPT: Managing Business Processes using Intelligent Agents. In: Proc. BCS Expert Systems 96, Conference (ISIP Track), Cambridge, UK 5-23.

Müller, J.P., Wooldridge, M.J., and Jennings, N.R. (eds.) (1997). 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

Norman, T.J., Jennings, N.R., Faratin, P. and Mamdani, E.H. (1997). Designing and implementing a multi-agent architecture for business process management. In: (Müller, Wooldridge and Jennings, 1997), pp. 261-275.

Nwana, H.S., Ndumu, D.T. and Lee, L.C. (1998). ZEUS: An Advanced Tool-Kit for Engineering Distributed Multi-Agent Systems. In: Proceedings of the Third International Conference on the Application of Intelligent Agents and Multi-Agent Technology (eds. Nwana, H.S. and Ndumu, D.T.), The Practical Application Company, Blackpool, 377-391; also in this issue.

Wooldridge, M. and Jennings, N.R. (1995). Agent theories, architectures, and languages: a survey. In: (Wooldridge and Jennings, 1995), pp. 1-39

Wooldridge, M.J., and Jennings, N.R. (eds.) (1995). Intelligent Agents (Proc. of the First International Workshop on Agent Theories, Architectures and Languages, ATAL'94), Lecture Notes in Artificial Intelligence, Vol. 890, Springer Verlag, Berlin.

