A Generic Multi-Agent Architecture for Interactive Diagnostic Tasks with Clarification

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

Abstract. A reusable multi-agent architecture is presented for interactive diagnostic tasks. The multi-agent system interacts with the user at two levels: at the level of the task itself and at the level of clarification of the process. The three agents distinguished are: a user, a diagnostic support agent and a clarification support agent. The user and the diagnostic support agent co-operate on the basis of a shared diagnostic task model; the user and the clarification support agent co-operate on the basis of a shared clarification task model.

1. Introduction

The design of systems to support experts in complex decision tasks requires analysis of the task as a whole, including the co-operation between the user and the decision support system. During analysis, the acquisition of a shared task model of a task can play an important role, a model shared by the user (who also is an expert) and the designer. This shared task model is used to design the architecture of a support system, and the interaction between this system and the user. A shared task model may not necessarily be 'the' expert's mental model of the task, but it is a model on the basis of which knowledge acquisition can be performed during the development of the system (cf. [4]), and on which the co-operation between the user and the support system can be based when the system is used (cf. [5]).

The agent paradigm provides a useful perspective to model co-operation between a user and a system on a complex task, such as diagnosis; e.g., [3]. [11][14][16] When human agents work together with automated agents in a co-operative task, clarification is often needed. For example, clarification may be needed about the meaning of terms used in the task, but also on the process; e.g., on the overall problem solving method that is followed in the task, or on a specific sub-task that is performed. Clarification can be considered as a complex task, which is at a meta-level with respect to the (object) task on which clarification is generated. Usually, human agents can clarify many aspects by themselves, but sometimes they may need support from another agent which co-operates with the human agent to perform the clarification task.

This perspective applied to a diagnostic task results in a multi-agent architecture consisting of three agents: the *user*, the *diagnostic support agent*, and the *clarification support agent*. Each pair of these agents has interaction: the user and the diagnostic support agent co-operate in the object task, the user and the clarification support agent co-operate in the clarification task, and the clarification support agent monitors and inspects the diagnostic support agent, the user, and their interaction. In the approach introduced here, the user and the diagnostic support agent have a shared model of the diagnostic task, whereas

the user and the clarification support agent have a shared model of the clarification task. The architecture introduced in this paper has been applied to chemical process diagnosis, in co-operation with Dutch chemical industry.

In Section 2 the overall multi-agent architecture is introduced. In Section 3 the (shared) task models for both the diagnostic object task and the clarification task are briefly introduced and related to the three agents. In Section 4 the agent models of the diagnostic support agent and the clarification support agent are discussed in some more detail. In Section 5 a brief overview of the knowledge structures used is given. In Section 6 the interaction pattern between the agents is discussed. In Section 7 the different types of questions for clarification are discussed, and the types of knowledge required. Section 8 addresses example clarification processes and shows the knowledge used.

2. The Multi-Agent Architecture

In this section a multi-agent architecture is proposed in which three agents are distinguished: the user, the diagnostic support agent and the clarification support agent. As both the user and the diagnostic support agent may need information from the external world (for example on results of inquiries or tests performed to acquire additional information), interaction with the external world needs to be explicitly modelled. In Figure 1 the top level composition (i.e., abstracting from the internal structure of the agents) is depicted.

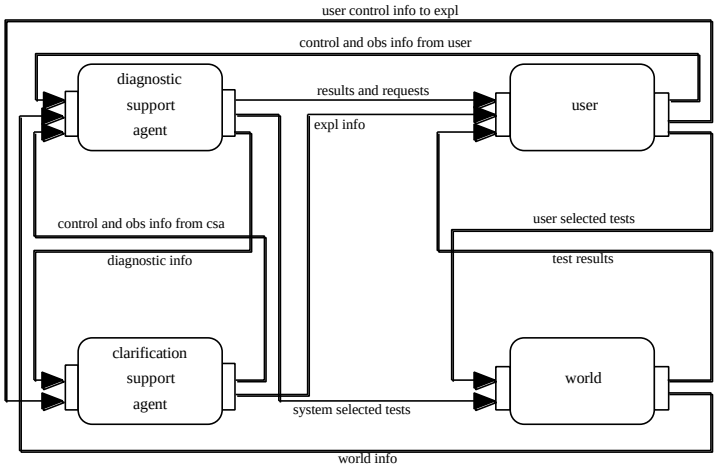

Figure 1: Top level composition of the multi-agent system

By considering the user and the diagnostic support system both as agents, a number of notions from the area of multi-agent systems can be exploited, such as autonomy, pro-activeness and reactiveness of both agents, and communication and co-operation between the agents [6], [11], [14], [16]. The user, the diagnostic support agent and the clarification support agent are autonomous, and in principle run in parallel. They all reason and react on the basis of their own knowledge, but also on the basis of information provided by communication with other agents and observation in the external world. The clarification support agent almost continually interacts with the diagnostic support agent: the diagnostic support agent keeps the clarification support agent informed of the status of the diagnostic process. The clarification support agent can explicitly request additional information from the diagnostic support agent, for example, to analyse a specific line of reasoning. The diagnostic support agent and the clarification support agent communicate information to the user, who in turn may communicate additional information on the basis of which a

diagnostic and/or clarification process may continue. The *weak agent notion* (cf. [16]) provides a useful concept to model the interactive concurrent processes involved. On the basis of this agent notion the required knowledge and behaviour is specified at a conceptual level. The weak agency characteristics autonomy, reactiveness, pro-activeness and social abilities all apply in the context of application.

3. The Shared Tasks Models for Diagnostic Task and Clarification Task

The shared task models for diagnosis and clarification will be discussed in relation with the agents in which they occur: the diagnostic support agent in Section 3.1, the clarification support agent in Section 3.2, and the user in Section 3.3.

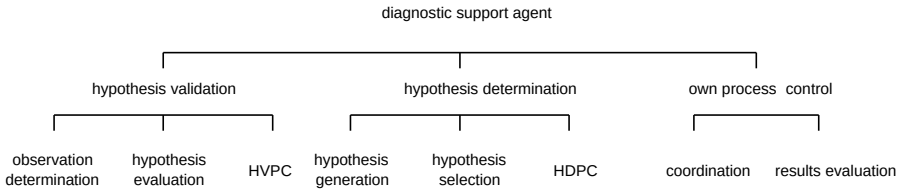

Figure 2: Task composition of the diagnostic support agent

3.1 The Tasks of the Diagnostic Support Agent

The diagnostic support agent has diagnosis as its main task. Its task composition is depicted in *Figure* 2. The diagnostic task has three main tasks: (1) to control the problem solving process (2) to determine hypotheses to be considered and (2) to validate these hypotheses. For comparable models in the knowledge-based systems literature, see [7], [8], [9]. To determine which hypotheses are to be considered, a number of hypotheses are generated and one or more are selected. To validate hypotheses, a number of observations are determined and performed, and the hypotheses evaluated on the basis of the observation results.

3.2 The Tasks of the Clarification Support Agent

The following global categories or levels of clarification can be distinguished: *terminological* clarification, clarification of *syntax* (of expected input), clarification of the diagnostic *process.* Each of the different levels of clarification needed is modelled as a separate task both within the user and within the clarification support agent.

The actual clarification process is performed by the two agents in co-operation. In addition the clarification support agent has an own process control task. In Figure 3 the clarification support agent's task composition is depicted. Note that the task model for process clarification has an object-meta relation with the task model of diagnosis. For each of the subtasks of diagnosis a meta-task is modelled that monitors and clarifies this specific subtask. This task composition is useful to structure the process of process clarification.

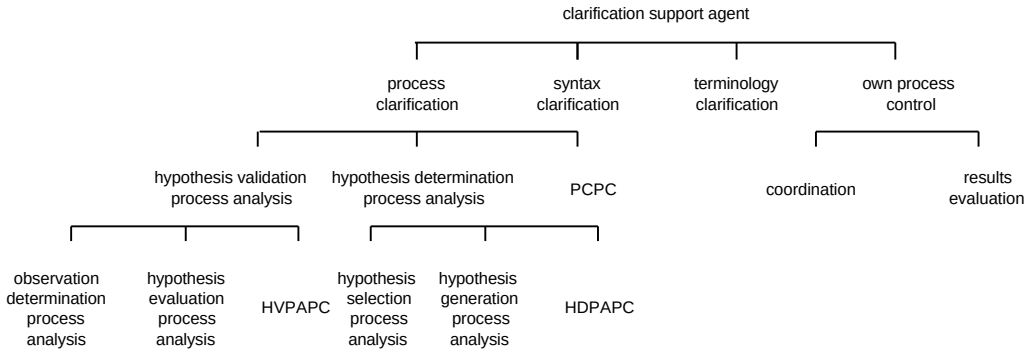

Figure 3**:** Task composition of the clarification support agent

3.3 The Tasks of the User

The tasks of the user are twofold: diagnosis and clarification. The former is performed by the user in co-operation with the diagnostic support agent, the latter in co-operation with the clarification support agent. To obtain insight in the agents, their tasks, and their interactions, also the assumed tasks of the user can be modelled. In Figure 4 a tree representation of the assumed user task composition of the diagnostic support agent (it is the intersection of Figures 2 and 4). The task model for clarification within the user is shared with the clarification support agent (it is the intersection of Figures 3 and 4). Which of the tasks in a shared task model actually is performed by the one agent and which by the other (and which by both), depends on the task division. In Section 5 an example task division and interaction pattern are discussed.

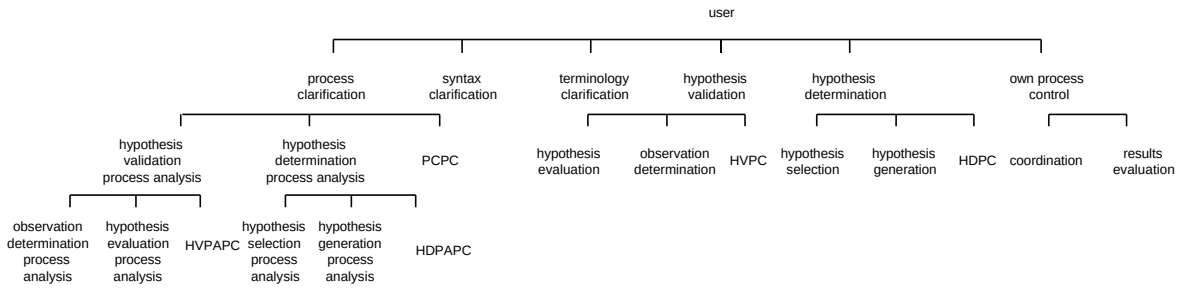

Figure 4: Assumed task composition of the user

4. The Diagnostic Support Agent and the Clarification Support Agent

In this section the agent architectures for the diagnostic support agent and for the clarification support agent are discussed.

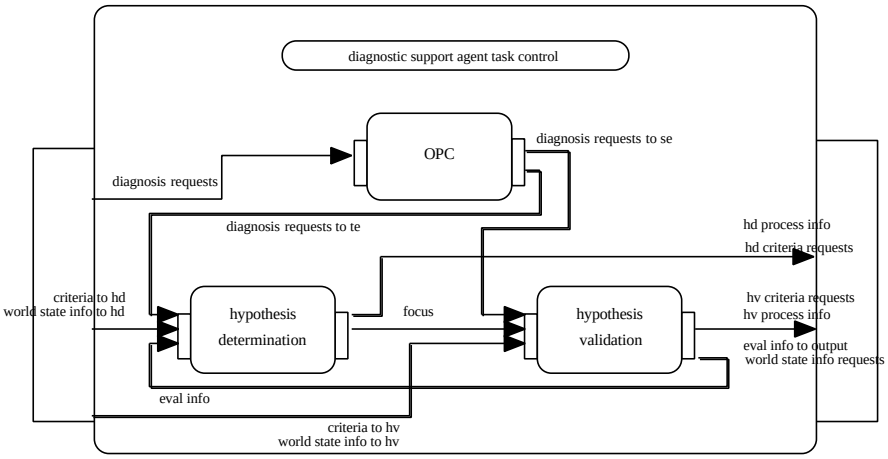

Figure 5: Top level composition of the Diagnostic Support Agent

4.1 The Diagnostic Support Agent

In contrast to most of the literature on diagnosis, the model of diagnosis used in this paper is in the first place a *process* model. It models the process of focusing on appropriate hypotheses to be investigated and determines which observations are to be performed for the focus hypotheses. Thus the decision making in the course of the diagnostic process is modelled, whereas literature such as [15] addresses diagnosis from a static point of view, assuming that all relevant observation results are already given.

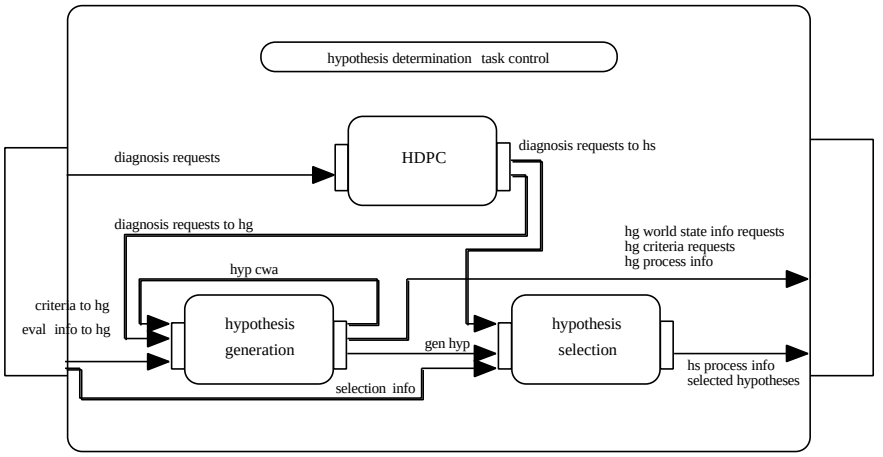

Figure 6**:** Composition of Hypothesis Determination

Following the task composition explained in Section 3.1, the diagnostic support agent is composed of three components (see Figure 5): own process control, hypothesis determination (which determines the focus hypotheses: those that should be investigated) and hypothesis validation (which takes care of investigation whether a focus hypothesis should be confirmed or rejected). The component own process control is composed of two sub-components: process coordination and results evaluation. The component hypothesis determination is composed of the three sub-components hypothesis determination process control, hypothesis generation (which generates the options for focus hypotheses) and hypothesis selection (which selects one of the options); see Figure 6.

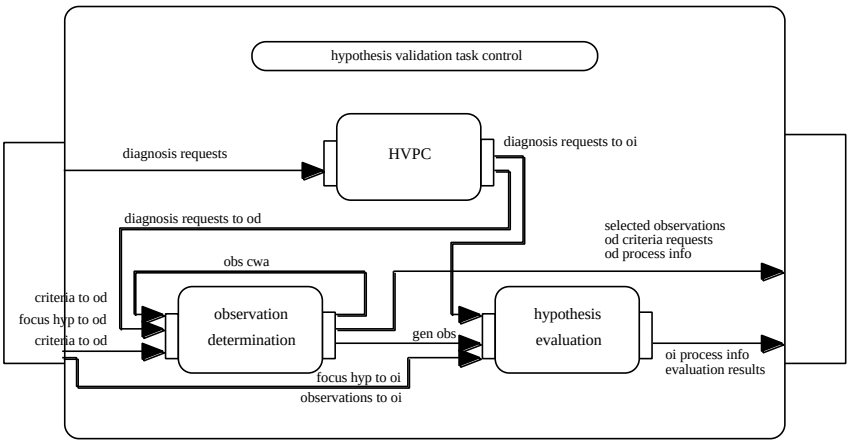

Figure 7: Composition of Hypothesis Validation

The component hypothesis validation is also composed of three sub-components (see Figure 7): hypothesis validation process control, observation determination (which determines the observations that are relevant for the focus hypothesis, and are not yet performed), and hypothesis evaluation (which evaluates the focus hypotheses in the light of the results of observations that have been performed). In the component hypothesis evaluation any of the static approaches to diagnosis available in the literature (e.g., [15]) can be incorporated.

4.2 The Clarification Support Agent

Composition at the top level of the clarification support agent is depicted in Figure 8. When required, the clarification support agent receives specific requests for clarification from the user. Furthermore, the clarification support agent receives information from the diagnostic support agent on the basis of which the diagnostic process can be fully (continually) monitored. The clarification support agent's own process control analyses an incoming request, and determines which type of clarification is required: terminology clarification, syntax clarification or process clarification. After this, the right component addresses the clarification request. Terminology clarification and syntax clarification are not discussed in further detail in this paper.

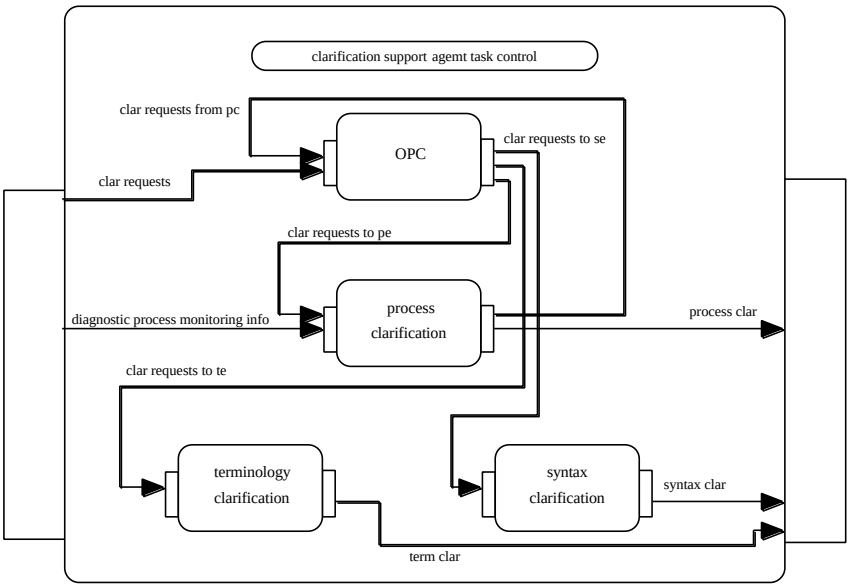

Figure 8: Composition of the top level of the Clarification Support Agent

Explanation on the process can be generated by the component process clarification at any point in the diagnostic process. Often clarification of the diagnostic process involves clarification of previous steps in the process in addition to the current status, and possibly includes the future process as well. A request often refers to the current step in the process, but may also refer (either explicitly, or implicitly) to previous (or subsequent) steps.

In some cases, clarification of the entire diagnostic process is requested. During process clarification a user may request further clarification on the syntax or on terms. The process clarification task co-ordinates the design of the contents of clarification in interaction with the user. The architecture of the process clarification component of the clarification support agent mirrors the model of the diagnostic process within the diagnostic support agent. It is composed of three components (see Figure 9): process clarification process control, hypothesis determination process analysis and hypothesis validation process analysis.

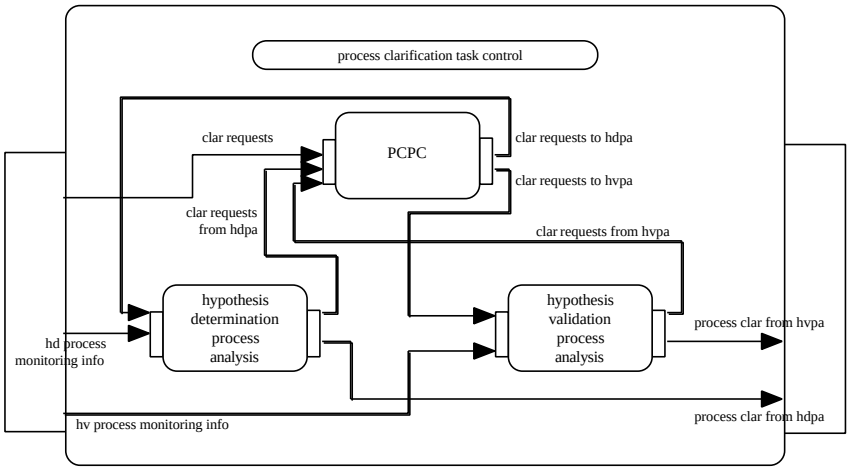

Figure 9: Composition of Process Clarification

The component hypothesis determination process analysis is composed of three sub-components: hypothesis determination process analysis process control, hypothesis generation process analysis and hypothesis selection process analysis (see Figure 10).

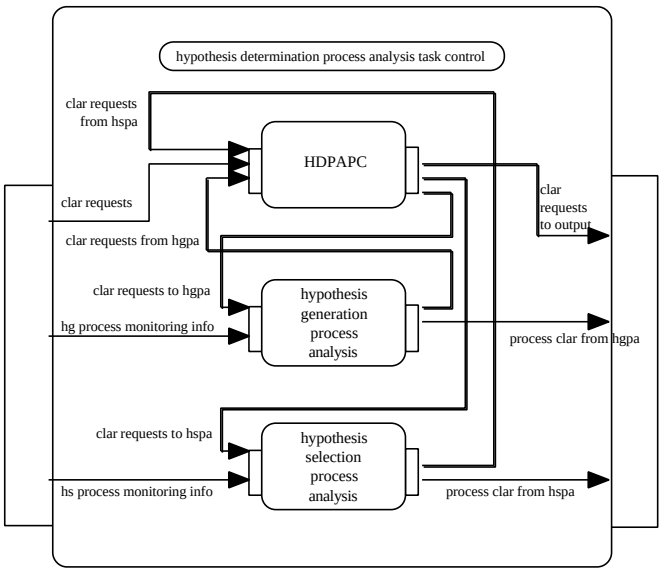

Figure 10: Composition of Hypothesis Determination Process Analysis

The component hypothesis validation process analysis is also composed of three sub-components: hypothesis validation process analysis process control, observation determination process analysis, and hypothesis evaluation process analysis (see Figure 11).

The component process clarification first classifies the clarification request according to whether it concerns clarification of the overall tasks or clarification of the process within a specific task. This classification is performed by the component process clarification proces control. If clarification of the overall task is requested the component process clarification proces control generates a global explanation. If clarification of the process within a task is required, this is delegated to the specific component within the clarification support agent related to that task. For example, the component hypothesis generation process analysis determines clarification of the process within the task hypothesis generation of the diagnostic support agent. The components hypothesis selection process analysis, observation determination process analysis, and hypothesis evaluation process analysis of the clarification support agent play a similar role with respect to the components hypothesis selection, observation determination, and hypothesis evaluation of the diagnostic support agent.

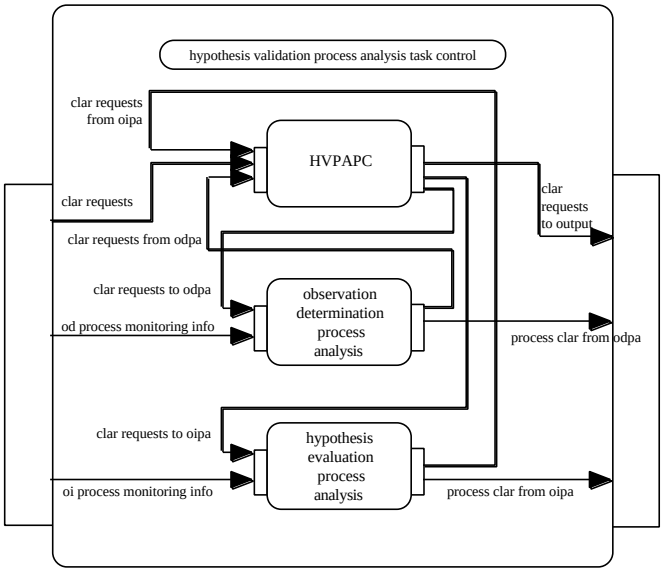

Figure 11: Composition of Hypothesis Validation Process Analysis

Process explanation has been designed to be concise. An explanation, however, can refer to concepts that have been used in a previous task. For example, if a user asks why a specific symptom has to be observed, the explanation refers to the selected hypothesis. The user then may want to know the origin of this selected hypothesis. In this case the user may ask for additional clarification. Additional clarification is provided by the clarification agent. In this approach the clarification process can be interactively and iteratively deepened, depending on the detail desired by the user.

5. Knowledge Structures for Clarifications

The model for diagnostic process clarification can be used to answer various process clarification requests. For each of these clarification requests knowledge structures are presented with which the clarification can be generated. The explanation instances (texts) generated are of sort Explanation_texts. Some of these texts are general; they are ‘precanned’ fragments that are dynamically combined in a flexible manner during an actual clarification

process; they are specified as objects of the sort Explanation_texts. These objects are the names of the texts. In the implementation the actual text to which the name refers is displayed. Most texts consist of a generic template text in which reference is made to a specific symptom or hypothesis. These texts are parameterised and are specified by functions with parameters as arguments. The ontology depicted below is used to define text names.

sorts
Explanation_texts
objects
general_task_model_text
functions

task_text : Tasks -> Explanation_texts

abstract_symptom_text : Abstract_symptoms * Hyps * Signs -> Explanation_texts
abstract_symptom_text : Abstract_symptoms -> Explanation_texts
hyp_generation_text : Hyps * Abstract_symptoms -> Explanation_texts

config_text : Config * Hyps -> Explanation_texts
displayed_hyp_text : Hyps -> Explanation_texts
focus_hyp_text : Hyps -> Explanation_texts
selectable_hyp_text : Hyps -> Explanation_texts
selectable_but_not_selected_hyp_text : Hyps -> Explanation_texts

sym_for_hyp_text : Symptoms * Hyps -> Explanation_texts

lit_for_lit_text : Lits * Lits -> Explanation_texts
confirmed_hyp_text : Hyps -> Explanation_texts
rejected_hyp_text : Hyps -> Explanation_texts
ever_selectable_hyp_text : Hyps -> Explanation_texts
neither_confirmed_nor_rejected_hyp_text : Hyps -> Explanation_texts
rejected_hyp_text : Hyps -> Explanation_texts
confirmed_hyp_text : Hyps -> Explanation_texts
from_the_obs_no decision_was possible_text : Hyps -> Explanation_texts
you_never_selected_this_selectable_hyp_text : Hyps -> Explanation_texts

General information:

sorts
Origins, Tasks
objects
abstract_symptoms, config_info, hyp_generation, hyp_selection, symptoms, hyp_assessments : Origins
hypothesis_generation, hypothesis_selection, observation_determination, observation_interpretation : Tasks

6. Task Division and Interaction between the Agents

In this section, the task division and interaction pattern between the agents are discussed by means of an example pattern.

6.1 An Example Task Division between User and Diagnostic Support Agent

For the application to chemical process diagnosis, on which the example is based, the following task division between user and diagnostic support agent was used:

- **hypothesis generation**
  user: provides initial symptoms

diagnostic support agent: generates hypotheses on the basis of initial symptoms

- **hypothesis selection**
  user: provides information on the chemical process configuration
  diagnostic support agent: determines a first selection (which is presented to the user)
  user: selects one of the hypotheses from the presented list
- **observation determination**
  diagnostic support agent: suggests an observation of a symptom to the user
- **hypothesis evaluation**
  user: provides the observation result
  diagnostic support agent: derives conclusions about the focus hypotheses
  user: result interpretation

6.2 An Example Interaction Pattern

The user may perform both clarification tasks and diagnostic tasks and switch between them at any moment. Both types of tasks are supported: the diagnostic tasks by the diagnostic support agent, and the clarification tasks by the clarification support agent. The following interaction pattern shows how these three agents and their contributions to the process are co-ordinated. In this example, the point in the process at which the user is expected to provide input on the hypothesis to be selected is discussed. The process proceeds as shown in Figure 12. Notice that this only shows one example pattern of interaction. The user is allowed to initiate an interaction with the clarification agent at any point in time and on any subject. The choices of these time points and subjects heaviliy afffect the further interaction pattern.

active components and interactions	*explanation*
dsa - hypothesis selection	The diagnostic support agent determines a list of hypotheses as as first selection
dsa -> user	The diagnostic support agent communicates this list of hypotheses to the user
user - process clarification	The user reads the list of hypotheses and tries to clarify the current state of the process, determines a lack of clarification, and generates a clarification request
user -> csa	The user communicates the clarification request to the clarification support agent
csa - proces clarification	The clarification support agent analyses the clarification request and generates an explanation
user <- csa	The clarification support agent communicates the explanation to the user
user - process clarification	The user reads the explanation and integrates it with his or her own clarifications
user - hypothesis selection	Clarification is satisfactory, the user selects one of the hypotheses
dsa <- user	The user communicates the selected hypothesis to the diagnostic support agent

Figure 12: An example interaction pattern between the three agents

7. Clarification Requests and Required Knowledge

In this section the types of clarification requests that can be handled by the clarification support agent are discussed, and the clarification process is illustrated by showing for some example clarification requests how they are handled.

7.1 Types of Clarification Requests that can be Handled

The model for diagnostic process clarification can be used to answer, among others, the following process clarification requests (in these example questions the pointers 'this' and 'these' refer to specific instances that can be indicated by the user):

- *Where are we in the process ?*
- *What are we doing ?*
- *Why do I need to give initial symptoms here ?*
- *Why do I need to give these particular initial symptoms ?*
- *Why is this hypothesis in the list of displayed hypotheses ?*
- *Why do I need to select a hypothesis ?*
- *Why was this hypothesis focussed upon ?*
- *Which hypotheses were selectable ?*
- *Which selectable hypotheses have not (yet) been selected ?*
- *Why is this hypothesis not in the list of displayed hypotheses ?*
- *Why do I need to perform a test here ?*
- *Why is this test suggested ?*
- *Why is this test not suggested ?*
- *How was this hypothesis found (why is this hypothesis correct) ?*
- *Why was this hypothesis rejected ?*
- *Which hypotheses were rejected ?*
- *Which hypotheses that were selected were neither rejected nor confirmed ?*
- *Why was this selectable hypothesis not confirmed ?*
- *Which hypotheses were not considered ?*
- *Why was this hypothesis not considered ?*
- *Why was this selected hypothesis not confirmed nor rejected ?*

For some of these clarification requests it is shown how the clarification can be generated and which knowledge structures are used. The generated explanation instances (texts, but in some cases also graphics) are of sort Explanation_texts. Some of the instances are generic; they are specified as objects of the sort Explanation_texts: the names of the texts. In the implementation the actual explanation instance to which the name refers is displayed. Most explanation instances consist of a generic template text in which reference is made to one or more specific symptoms or hypotheses. These texts are parameterised and are specified by functions with parameters as arguments.

7.2 Knowledge Used

To be able to provide clarification, the clarification support agent requires at least the following types of meta-information and meta-knowledge:

1. *Status information about the actual and planned diagnostic process*

Information about the diagnostic process refers to information on the diagnostic process as performed by the user and the diagnostic support agent:

- about the previous process until the time point at which the request for clarification is put forward
- about the planned diagnostic process.

This information is *dynamically* acquired by the clarification support agent during the process by monitoring the user and the diagnostic support agent during their diagnostic reasoning and interaction. For example, the generated list of hypotheses is recorded, together with information on which of them have been chosen previously, and which were confirmed or rejected.

2. *Knowledge about relations between the concepts in the diagnostic reasoning process*

To be able to generate explanation about the possible and actual reasoning steps in the diagnostic process the following types of meta-knowledge are required:

- relations between initial symptom information and generated hypotheses
- relations between hypotheses and the tests required to confirm or reject them

This *static* knowledge is part of the specification of the clarification support agent.

8. Example Clarification Processes

In this section an example clarification is shown to illustrate the clarification process and the knowledge used within the clarification support agent.

8.1 Clarification of the overall process

At any moment in the diagnostic process, it is assumed that the user can also perform his or her clarification task, and if the clarification generated by the user is not sufficient, for example one of the following two questions can be generated.

- *Where are we in the process ?*
- *What are we doing ?*

These questions are input for the clarification support agent in the form

```
required_process_clar(where_are_we)
required_process_clar(what_are_we_doing)
```

For the first question the clarification support agent's component process clarification process control uses the knowledge

```
if      required_process_clar(where_are_we)
  and   current_task(T:Tasks)
then    selected_process_explanation(task_text(T:Tasks), where_are_we)
```

It derives

```
selected_process_clar(task_text(t), where_are_we)
```

where t is the current task in the process. The text referred to by task_text(t) is given in the form of a graphical representation of the overall diagnostic task model, with colour or blinking to indicate which task is currently being performed, and a short description of the aim of the task (see below for instances of these task descriptions).

From the second question also the first question is derived within component process clarification process control, using the knowledge

if required_process_clar(what_are_we_doing)
then required_process_clar(where_are_we)

In addition, by the rule

if required_process_clar(what_are_we_doing)
then selected_process_explanation(general_task_model_text, what_are_we_doing)

it is derived

selected_process_clar(general_task_model_text, what_are_we_doing)

The explanation text referred to by general_task_model_text is given in the form of a graphical representation of the overall diagnostic task model, with colour or blinking to indicate which task is currently being performed, and, in addition, from each of the displayed components a hyperlink to a short text to explain the task.

8.2 Clarification of generated hypotheses

Another clarification request can arise during hypothesis selection, at the moment that the user receives the list of hypotheses selected by the diagnostic support agent:

- *Why is this hypothesis* H:Hyps *in the list of generated hypotheses ?*

This question is handled by the component hypothesis generation process analysis; it is represented as

required_process_clar(why_generated(H:Hyps))

First, by the component process clarification process control the question is classified as one to be treated by the component hypothesis generation process analysis: Using the knowledge

if required_process_clar(why_generated(H:Hyps))
then required_process_clar_type(hypothesis_generation)

the component process clarification process control generates

required_process_clar_type(hypothesis_generation)

Note that this is a case in which the question goes back to an earlier task in the process.

Next, hypothesis generation process analysis processes the question using the knowledge

if required_process_clar(why_generated(H:Hyps))
and is_positive_support_for(S:Symptoms, H:Hyps)
and has_been_observed(S:Symptoms)

then selected_process_explanation(hyp_generation_text(H:Hyps, S:Symptoms),
why_generated(H:Hyps))

For a number of symptoms, conclusions that are drawn contain the expression

hyp_generation_text(H:Hyps, S:Symptoms)

Based on these conclusions the following text is displayed:

> *Because of the initial symptoms* S:Symptoms *that were observed, and the fact that hypothesis* H:Hyps *can cause these symptoms, the hypothesis was one of the generated possible hypotheses*.

8.3 Clarification of the task hypothesis generation

In the part of the process in which the user performs both the task hypothesis generation and the task hypothesis generation process analysis, a lack of clarification can be found and one of (or both) the following questions can be generated:

- *Why do I need to give initial symptoms here ?*
- *Why do I need to give these particular initial symptoms here ?*

The clarification support agent receives them as instances of atoms of the form

required_process_clar(why_do_we_need_initial_symptoms)
required_process_clar(required_info(S:Initial_symptoms))

Using the knowledge

if required_process_clar(why_do_we_need_initial_symptoms)
then required_process_clar_type(hypothesis_generation)

if required_process_clar(required_info(S:Initial_symptoms))
then required_process_clar_type(hypothesis_generation)

the component process clarification process control generates

required_process_clar_type(hypothesis_generation)

as its conclusion. By the task control knowledge, hypothesis generation process analysis is provided with the input on required clarification and activated. In this component, for the first question, based on the knowledge

if required_process_clar(why_do_we_need_initial_symptoms)
then selected_process_explanation(task_text(hypothesis_generation),
why_do_we_need_initial_symptoms)

the conclusion

selected_process_explanation(task_text(hypothesis_generation), why_do_we_need_initial_symptoms)

is generated. Here task_text(hypothesis_generation) refers to the following text to be displayed:

> *Initially the complaints are needed to be able to focus on a set of possible hypotheses about the cause of the problems. For example, if the complaint is .. then the hypothesis ... will not be taken into account .*

For the second question, and the specified initial symptom, the component hypothesis generation process analysis uses the knowledge

if required_process_clar(required_info(S:Initial_symptoms))
and is_positive_support_for(S:Initial_symptoms, H:Hyps)
then selected_process_explanation(initial_symptom_text(S:Initial_symptoms, H:Hyps, pos),
required_info(S:Initial_symptoms));
if required_process_clar(required_info(S:Initial_symptoms))
and is_negative_support_for(S:Initial_symptoms, H:Hyps)
then selected_process_explanation(initial_symptom_text(S:Initial_symptoms, H:Hyps, neg),
required_info(S:Initial_symptoms))

to derive for a number of hypotheses H:Hyps conclusions of the form:

selected_process_explanation(initial_symptom_text(S:Initial_symptoms, H:Hyps, S:Signs)),
required_info(S:Initial_symptoms))

Here initial_symptom_text(S:Initial_symptoms, H:Hyps, S:Signs) refers to a text explaining

(1) if S: Signs is pos: *The initial symptom* S:Initial_symptoms *positively supports the hypothesis* H:Hyps;

(2) if S:Signs is neg: *The initial symptom* S:Initial_symptoms *negatively supports the hypothesis* H:Hyps.

8.4 Clarification related to hypothesis selection

Within the user task hypothesis selection the following clarification request can arise.

- *Why do I need to select a hypothesis ?*

By the component hypothesis selection process analysis the conclusion

selected_process_explanation(task_text(hypothesis_selection),why_do_we_need_to_select_a_hypothesis)

is drawn: the text to which task_text(hypothesis_selection) refers (see above) is displayed.

* *Why can I select just one hypothesis ?*

In a similar manner this question is processed: by the component hypothesis selection process analysis the same conclusion

selected_process_explanation(task_text(hypothesis_selection), why_can_I_select_just_one_hypothesis)

is drawn and the same text is displayed.

8.5 Clarification within hypothesis validation

Within the user task hypothesis validation the following question can arise

- *Why was this hypothesis focussed upon ?*

This question is handled by the component hypothesis selection process analysis, using the knowledge

if required_process_clar(hyp_selection, focus_hyp(H:Hyps))
then selected_process_explanation(focus_hyp_text(H:Hyps), focus_hyp(H:Hyps))

The conclusion contains focus_hyp_text(H:Hyps), which refers to the following text to be displayed:

The hypothesis H:Hyps *was on the list of generated hypotheses and you selected it .*

Subsequently, the following follow-up question can arise:

- *Which hypotheses were selectable ?*

This question is treated by the component hypothesis selection process analysis, using the knowledge

if required_process_clar(hyp_selection, which_hyps_were_ever_selectable)
and ever_selectable_hyp(H:Hyps)
then selected_process_explanation(selectable_hyp_text(H:Hyps), which_hyps_were_ever_selectable)

>> *display list of the hypotheses that were considered*

Next, the following question could be put forward:

- *Which selectable hypotheses have not (yet) been selected ?*

This question is treated by the component hypothesis selection process analysis, using the knowledge

if required_process_clar(hyp_selection, which_selectable_hyps_were_never_selected)
and ever_selectable_hyp(H:Hyps)
and not selected(H:Hyps)
then selected_process_explanation(selectable_but_not_selected_hyp_text(H:Hyps),
which_selectable_hyps_were_never_selected)

>> *display list of the considered hypotheses that have not yet been selected*

A question that was left out of consideration is:

- *Why is this hypothesis not in the list of displayed hypotheses ?*

This question could be answered similarly.

8.6 Clarification of observation determination

Within the user task observation determination the following question can arise:

- *Why do I need to perform a test here ?*

This question is answered by the component observation determination process analysis, using the knowledge

if required_process_clar(symptoms, why_do_you_need_symptoms)
then selected_process_explanation(task_text(observation_determination), why_do_you_need_symptoms)

Displayed text task_text(observation_determination):

> *The diagnostic process has been focussed on a hypothesis that is expected to be the cause. By performing relevant tests the hypothesis can be confirmed or rejected.*

A folow-up question can be:

- *Why is this test suggested ?*

Also this question is answered by the component observation determination process analysis, this time using the knowledge

if required_process_clar(symptoms, required_info(S:Symptoms))
and current_hyp(H:Hyps)
then selected_process_explanation(sym_for_hyp_text(S:Symptoms, H:Hyps), required_info(S:Symptoms))

Displayed text sym_for_hyp_text(S:Symptoms, H:Hyps):

> *If the focus hypothesis* H:Hyps *is the fault, then this can have effect on the values measured by this test for* S:Symptoms.
> *If the test results are available, then this forms the basis for confirming or rejecting the focus hypothesis.*

After this the possibility can be offered to make a new selection:

> *Do you want to revise your hypothesis selection ?*

The question

- *Why is this test not suggested ?*

has not been considered, but could be answered in a similar way.

8.7 Clarification of hypothesis evaluation

Within the user task result interpretation the following questions can be put forward:

- *How was this hypothesis found (why is this hypothesis correct) ?*
- *Why was this hypothesis rejected ?*

These questions are answered by the component observation interpretation process analysis, using the knowledge

instances of:
rule_for(R:Rules, L:Lits)
in_body_of(L:Lits, R:Rules)

if rule_for(R:Rules, L:Lits)
and In_body_of(L1:Lits, R:Rules)
then relevant_for(L1:Lits, L:Lits)

if required_process_clar(hyp_assessments, confirmed(L:Lits))
and relevant_for(L1:Lits, L:Lits)
and symptom(L1:Lits)
then selected_process_explanation(sym_for_hyp_text(L1:Lits, L:Lits), confirmed(L:Lits))

if required_process_clar(hyp_assessments, rejected(L:Lits))
and relevant_for(L1:Lits, not(L:Lits))
and symptom(L1:Lits)
then selected_process_explanation(sym_for_hyp_text(L1:Lits, not(L:Lits)), rejected(L:Lits))

if required_process_clar(hyp_assessments, rejected(L:Lits))
and relevant_for(L1:Lits, L:Lits)
and **not** symptom(L1:Lits)
then selected_process_explanation(lit_for_lit_text(L1:Lits, not(L:Lits)), rejected(L:Lits))

if required_process_clar(hyp_assessments, confirmed(L:Lits))
and relevant_for(L1:Lits, L:Lits)
and **not** symptom(L1:Lits)
then selected_process_explanation(lit_for_lit_text(L1:Lits, L:Lits), confirmed(L:Lits))

Based on the conclusions, text of the following type is displayed:

> *The test information .... implies ..... (...) .... implies(by domain knowledge) that this hypothesis is/is not the fault.*

An opposite question is:

- *Which hypotheses were rejected ?*

This question is treated by the component observation interpretation process analysis, using the knowledge

if required_process_clar(hyp_assessments, which_hypotheses_were_rejected)
and ever_rejected(H:Hyps)
then selected_process_explanation(rejected_hyp_text(H:Hyps), which_hypotheses_were_rejected)

Based on the conclusion the following is displayed:
>> display list of the hypotheses that were rejected
Another question is:

- *Which hypotheses that were selected were neither rejected nor confirmed ?*

This question is treated by the component observation interpretation process analysis, using the knowledge

if required_process_clar(hyp_assessments, which_hypotheses_were_neither_confirmed_nor_rejected)
and **not** ever_confirmed(H:Hyps)
and **not** ever_rejected(H:Hyps)
then selected_process_explanation(neither_confirmed_nor_rejected_hyp_text(H:Hyps),
which_hypotheses_were_neither_confirmed_nor_rejected)

Based on the conclusion the following is displayed:

>> *display list of the selected hypotheses that were neither confirmed nor rejected*

A further question:

- *Why was this selectable hypothesis not confirmed ?*

This question is treated by the component observation interpretation process analysis, using the knowledge

if required_process_clar(hyp_assessments,
why_was_this_selectable_hypothesis_not_confirmed(H:Hyps))
and ever_selected(H:Hyps)
and ever_rejected(H:Hyps)
then selected_process_explanation(rejected_hyp_text(H:Hyps),
why_was_this_selectable_hypothesis_not_confirmed(H:Hyps))

if required_process_clar(hyp_assessments,
why_was_this_selectable_hypothesis_not_confirmed(H:Hyps))
and ever_selected(H:Hyps)
and **not** ever_rejected(H:Hyps)
and **not** ever_confirmed(H:Hyps)
then selected_process_explanation(from_the_obs_no decision_was possible_text(H:Hyps),
why_was_this_selectable_hypothesis_not_confirmed(H:Hyps))

if required_process_clar(hyp_assessments,
why_was_this_selectable_hypothesis_not_confirmed(H:Hyps))
and ever_confirmed(H:Hyps)
then selected_process_explanation(confirmed_hyp_text(H:Hyps),
why_was_this_selectable_hypothesis_not_confirmed(H:Hyps))

if required_process_clar(hyp_assessments,
why_was_this_selectable_hypothesis_not_confirmed(H:Hyps))
and **not** ever_selected(H:Hyps)
then selected_process_explanation(you_never_selected_this_selectable_hyp_text(H:Hyps),
why_was_this_selectable_hypothesis_not_confirmed(H:Hyps))

depending on the conclusions, display text:
- *the hypothesis was selected and rejected*
- *this hypothesis was selected but neither rejected nor confirmed*
- *actually this hypothesis was confirmed*
- *the hypothesis was not selected yet*

In addition, the user can be offered the possibility to select a conclusion

Do you want to select it now ?

The following questions have been left out of consideration, but could be answered as well.

- *Which hypotheses were not considered ?*

- *Why was this hypothesis not considered ?*
- *Why was this selected hypothesis not confirmed nor rejected?*

9. Conclusions

The multi-agent architecture presented in this paper has been developed to support a user both at the level of the diagnostic task he or she is performing and at the level of clarification. In co-operation with the Dutch chemical industry DSM the architecture has been applied to diagnosis of chemical processes. The objective is to make it easier for a user to work with a diagnostic support system by adding the clarification support agent, and thus to contribute to more successful application of diagnostic support systems in chemical industry. The genericity of the architecture makes it reusable for various tasks and domains, for diagnostis and other tasks, within chemical industry and beyond.

To design multi-agent systems, the component-based design method DESIRE [2] has turned out quite useful. DESIRE also supports task modelling abstracting from (multi-) agent structures. The architecture presented here could also have been modelled as a component-based knowledge-based system with components user, diagnosis support task and clarification support task. Then some form of control over the three task shouls be specified, for example some loop through the three. In our approach the multi-agent perspective was chosen to make explicit the different interactions of the parties involved, and to support dynamic, event-driven behaviour. The co-operating agents can behave in a reactive manner, but can also be pro-active by taking the initiative. Thus the architecture as presented provides more flexibility. In the perspective put forward in [10], among others, it was argued that in interactive processes involving knowledge-based support systems, explanation is crucial and that agent-based modelling could turn out useful for that. In [12], interface agents are described that support search for information. A difference with our approach is that no explicit task model is involved and that no explanation is given on the process on the basis of such a task model. The same type of difference can be found with the approach described in [1]. Another difference with the references as mentioned is that in our approach the architecture is based on an explicit component-based architecture, which supports maintainability and reuse.

Acknowledgements

DSM provided funding for parts of the reported research. The authors have had stimulating discussions with Frances Brazier (Vrije Universiteit Amsterdam), Rob Faessen (DSM Research), Bart Gras (DSM Research/Bolesian) and Gert Poppe (DSM Research).

References

[1] Barrett, R., Maglio, P.P., Kellem, D.C., "How to personalize the Web", in Pemberton, S., ed., Human Factors in Computing Systems., Proceedings of the CHI'97. ACM. Addison Wesley, 1997, pp. 75-82.

[2] Brazier, F.M.T., Dunin-Keplicz, B., Jennings, N.R., Treur, J., "Formal specification of Multi-Agent Systems: a real-world case", in V. Lesser, ed., Proceedings of the First International Conference on Multi-Agent Systems, ICMAS-95, MIT Press, Cambridge, MA, 1995, pp. 25-32. Extended version in: International Journal of Co-operative Information Systems, M. Huhns, M. Singh, eds., special issue on Formal Methods in Co-operative Information Systems: Multi-Agent Systems, vol. 6, 1997, pp. 67-94.

[3] Brazier, F.M.T., Treur, J., "User-centered knowledge-based system design: a formal modelling approach", in L. Steels, G. Schreiber, W. Van de Velde, eds., A future for knowledge acquisition, Proceedings of the 8th European Knowledge Acquisition Workshop, EKAW'94, Lecture Notes in AI, vol. 867, Springer Verlag, 1994, pp. 283-302.

[4] Brazier, F.M.T., Treur, J., Wijngaards, N.J.E., "The Acquisition of a Shared Task Model", in N. Shadbolt, K. O'Hara, G. Schreiber, eds., Advances in Knowledge Acquisition, Proceedings of the 9th European Knowledge Acquisition Workshop, EKAW'96, Lecture Notes in AI, vol. 1076, Springer Verlag, 1996, pp. 278-289.
[5] Brazier, F.M.T., Treur, J., Wijngaards, N.J.E., "Modelling Interaction with Experts: the Role of a Shared Task Model", in W. Wahlster, ed., Proceedings of the European Conference on AI, ECAI'96, John Wiley and Sons, 1996, pp. 241-245
[6] Castelfranchi, C., "Commitments: From individual intentions to groups and organizations", in V. Lesser, ed., Proceedings of the First International Conference on Multi-Agent Systems, ICMAS-95, MIT Press, Cambridge, MA, 1995, pp. 41-48.
[7] Chandrasekaran, B. "Generic tasks in knowledge-based reasoning: high-level building blocks for expert system design", in *IEEE Expert*, Vol. 1, 1986, pp. 23–30.
[8] Clancey, W.J. "The epistemology of a rule-based expert system", in *Artificial Intelligence*, vol. 7, 1983.
[9] Clancey, W.J., Bock, C., "Representing control knowledge as abstract tasks and metarules", in Bolc, Coombs, eds., *Expert System Applications*, 1988.
[10] Dieng, R., Giboin, A., Tourtier, P.A., Corby, O., "Knowledge Acquisition for Explainable, Multi-Expert, Knowledge-Based Design Systems", in Th. Wetter, K.D. Althoff, J. Boose, B.R. Gaines, M. Linster, F. Schmalhofer, eds., Current Developments in Knowledge Acquisition, Proceedings of the 6th European Knowledge Acquisition Workshop, EKAW'92. Lecture Notes in AI, vol. 599, 1992, pp. 298-317.
[11] Jennings, N.R., "Controlling Co-operative Problem Solving in Industrial Multi-Agent Systems using Joint Intentions", in Artificial Intelligence Journal 74 (2), 1995.
[12] Lieberman, H., "Autonomous Interface Agents", in Pemberton, S., ed., Human Factors in Computing Systems. Proc. CHI'97. ACM. Addison Wesley, 1997, pp. 67-74.
[13] Pemberton, S., ed., Human Factors in Computing Systems. Proc. CHI'97. ACM. Addison Wesley, 1997.
[14] Reiter, R., "A Theory of Diagnosis from First Principles", in Artificial Intelligence, vol. 32, 1987, pp. 57-95.
[15] Rao, A.S., Georgeff, M.P., "Modeling rational agents within a BDI architecture", in R. Fikes, E. Sandewall, eds., Proceedings of the Second Conference on Knowledge Representation and Reasoning, Morgan Kaufman, 1991, pp. 473-484.
[16] Wooldridge, M., Jennings, N.R., "Agent theories, architectures, and languages: a survey", in Wooldridge, M., Jennings, N.R., eds., Intelligent Agents, Proceedings of the First International Workshop on Agent Theories, Architectures and Languages, ATAL'94, Lecture Notes in AI, vol. 890, Springer Verlag, 1995, pp. 1-39.
[17] Wooldridge, M., Jennings, N.R., eds., Intelligent Agents, Proceedings of the First International Workshop on Agent Theories, Architectures and Languages, ATAL'94, Lecture Notes in AI, vol. 890, Springer Verlag, 1995.