Applied Intelligence 8, 235–245 (1998)

An Intelligent Man-Machine Dialogue System Based on AI Planning

BART KUIJPERS*
Department of Mathematics and Computer Science, University of Antwerp (UIA), Universiteitsplein 1, B-2610 Antwerpen
kuijpers@uia.ua.ac.be

KRIS DOCKX
Expert Systems Applications Development Group, Department of Chemical Engineering, Katholieke Universiteit Leuven, de Croylaan 46, B-3001 Leuven
Kris.Dockx@cit.kuleuven.ac.be

Abstract. We describe the modular architecture of a generic dialogue system that assists a user/operator in performing a task with a tool. This coaching system is named CALLIOPE after the Greek goddess of eloquence. It aims at being an active partner in an intelligent man-machine dialogue. The intelligent dimension of the coaching system is reflected by its ability to adapt to the user and the situation at hand. The CALLIOPE system contains an explicit user model and world model to situate its dialogue actions. A plan library allows it to follow loosely predetermined dialogue scenarios.

The heart of the coaching system is an AI planning module, which plans a series of dialogue actions. We present a coherent set of three dialogue or speech actions that will make up the physical form of the man-machine communication.The use of the AI planning paradigm as a basis for man-machine interaction is motivated by research in various disciplines, as e.g., AI, Cognitive Science and Social Sciences. Starting from the man-man communication metaphor, we can view the "thinking before speaking" of a human communication partner as constructing an underlying plan which is responsible for the purposiveness, the organisation and the relevance of the communication.

CALLIOPE has been fully implemented and tested on theoretical examples. At present, also three tailored versions of CALLIOPE are in operational use in different industrial application domains: operator support for remedying tasks in chemical process industry, operator support for a combined task of planning, plan execution and process control in the area of chemical process development, and thirdly decision support in production scheduling.

Keywords: man-machine interaction, dialogue planning, AI planning, scenarios

1. Why Planning as a Basis for Intelligent Dialogue?

We will describe the design of a generic, i.e., application independent, planning-based, user adaptive dialogue system that supports or assists a user (or operator) who has to perform a (number of) task(s) with a tool (or on a machine). In the following we will refer to this coaching system as CALLIOPE. Its main goal is to facilitate a flexible dialogue and to smoothen the communication between man and machine.

Because man-machine communication is such a fundamental issue, it is the subject of research in many (overlapping) disciplines as e.g., Artificial Intelligence, Cognitive Science, Computer Science, Social

*At the time the research presented in this paper was performed this author was with the Expert Systems Applications Development Group, Department of Chemical Engineering, Katholieke Universiteit Leuven, de Croylaan 46, B-3001 Leuven.

Sciences, Philosophy, Pedagogy, and Ergonomics. Hence, there exists also a lot of controversy about the subject.

When we think about communication, we also have to consider the actions that constitute the communication, especially the coherence and understandability of these actions. In man-man communication, the communication partners think before they speak. This *thinking before acting/speaking* can be considered as underlying planning that orients the dialogue or the interaction towards a goal, gives the interaction a certain structure and adapts the dialogue to the context and to the communication-partner. So, in an intuitive way, we can see a plan as a "basis" for communication.

We now look at different motivations from different disciplines to use planning as a basis for man-machine communication.

In the past decades, "planning" and "problem solving" have received a great deal of research interest in the AI community. This research focuses on domain independent planning, without addressing directly problems of communication.[1]

Research on planning in AI has reached a certain maturity, the results are logically well founded, and a number of reliable paradigms for planning are available (see [25] for a survey of the field). Since planning is generally considered as a basis for intelligent behaviour, many systems in the recent AI history have been enhanced with planning capabilities to give them more intelligence and also more autonomy. So, AI planning gives us formally well founded paradigms to attack also problems concerning communication [5].

From Cognitive Science (see e.g., [14, 21]), a new approach to the problem of Man Machine Systems (MMS) has given new insights. In the old view on MMS, humans are forced to interact with the machine on the machine's terms. This is not merely inconvenient. More importantly, because it is an unnatural mode of interaction, it is the primary cause of human errors. Now, Cognitive Science looks at MMS as systems which produce intelligent actions, in the sense that they are goal oriented, adaptive, and that they use knowledge about themselves and their environment to replan their actions. An important idea is that by carrying a model of its environment and its user, the system is able to try out various alternatives, to conclude which is the best, to react to future situations before they arise, to utilise the knowledge of past events in dealing with the present and the future, and in every way to react in a much fuller, safer, and more competent manner to emergencies. This point of view is very similar to the AI point of view on planning. Furthermore, the concept of "user model" is introduced as an incorporated part of the MMS program mode.

In [24], Suchman, Suchman gives a correction to the above views on interaction, which are all heavily based on an underlying planning paradigm. Instead of viewing plans as the basis for the interaction she looks at plans merely as (weak) resources for interaction. She views goal-oriented actions always as being "situated" (see the introductory Turkese fisherman's story in [24]). As projective and retrospective accounts of action, plans are themselves located in the larger context of some ongoing practical activity. Since the context is subject to a continuous change and since the context of an action can never be anticipated completely, actions have to be considered in their very concrete context. A plan can therefore, at best, be seen as a (weak) basis for the concrete "ad hoc" activity.

For the reasons mentioned above, it is best to start from a vaguely formulated plan that is only completed during execution. In the AI community, this approach to planning is referred to as "planning and execution". Replanning after unforeseen events and reactivity of planning systems are also to be located in this area of research. Human errors are an example of unforeseen events. They become an increasingly important issue in contemporary systems. Rasmussen argues that human errors should not be avoided, but systems should be designed in such a way that human errors are absorbed [21]. The above presented views are incorporated in CALLIOPE and are summarised in Fig. 1.

Our research can be situated in the larger body of work on *communication planning* and also deals with issues like plan recognition, intentions in communication, and theories and pragmatics of speech acts. A standard reference is [5]. We also refer to the work of Grosz and Moore and references therein [12, 17].

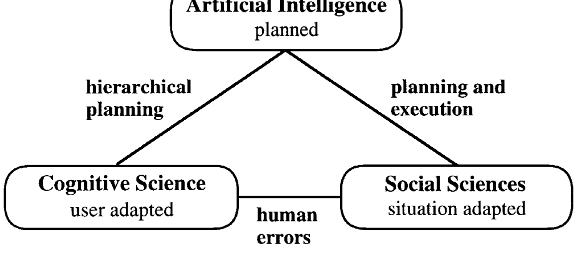

Figure 1. Three approaches to intelligent dialogue and their links.

Since our objectives are narrowed by our focus on *task-oriented dialogue*, the main issues addressed in the above mentioned fields are, however, not our primary subject but rather our inspiration. The focus on task-oriented dialogue is motivated by the applications we envisage for CALLIOPE. These applications are mainly situated in the area of chemical process control and production planning and scheduling where operators have to perform a number of tasks to obtain a desired state of the plant or to achieve certain production goals. In this sense, our work is closer related to the development of intelligent help systems and intelligent tutoring systems (see e.g., [13, 19, 20, 27]). From a planning point of view our main interest is in applications which require a tight integration between plan generation and plan execution. Our work is also situated in between open-loop AI planning and reactive planning. In this respect we refer to the work of Kambhampati and Musliner [15, 18].

The remainder of this paper is organised as follows. In Section 2, we describe the modular architecture of our planning-based, user and situation adapted dialogue system CALLIOPE. In Section 3, we look in more detail at the three speech or dialogue actions that constitute the (physical) realisation of the dialogue system: the so-called I.I.I.-system. In the following section, we look at an example of CALLIOPE in a real world application, taken from a remedying task in chemical process industry, and we discuss some further features of the system. In Section 5, we discuss the implementation and use of CALLIOPE in different applications. The final section summarises the key ideas incorporated in CALLIOPE.

2. Modular Architecture of the Intelligent Dialogue System

In this section, we describe the architecture of CALLIOPE. The main functionality of the system is to assist or coach a user/operator in performing a task with some physical tool. This is accomplished by conducting an intelligent dialogue with the operator. The communication between CALLIOPE and the user consists of two phases.

In a *first phase*, the task or tasks are determined through a question and answer (Q&A) session. For our generic implementation of CALLIOPE the Q&A phase is completed by the user/operator by giving the system a complete description of the physical world that he wants to achieve. This is done by supplying the system in a declarative way with a number of *facts* that completely describe the desired world state. Because computer systems are limited in their sensory perception of the physical world the user is also asked to supply the system with additional data about the state of the current world. According to Suchman this inability of systems to monitor their physical environment and their "situation" constitutes one of the main differences between normal human communication and man-machine communication [24].

For the application-tailored versions of the system we have made this Q&A more application-dependent, more smooth and more user-friendly. For one particular remedying application in a chemical production plant, for instance, where malfunctioning systems have to be brought back to their normal functioning state or where emergencies like fires have to be dealt with, this Q&A can be considered as a diagnosis phase [6].

We have implemented this diagnosis task in such a way that the user/operator is also used as a sensor to the physical world. He is asked to answer a number of multiple-choice questions concerning visually observable facts (smoke, no smoke, fire, etc.) and to provide the system with a number of parameters he can read from control panels (pressures on pipes, temperatures of vessels, etc.). For this particular application, these inputs are given to an expert system which returns a number of possible and probable causes and a series of tasks to remedy the emergency at hand. The successful completion of these tasks are the objectives for the second phase.

For a production planning and scheduling application, we have implemented the Q&A as a module where the user can input new production orders. These consist of product identifiers, custumor identifiers, amounts, due dates and priority rankings. The added production orders determine the optimal schedule that the system has to help to achieve in the second phase of the dialogue. Also here the user has the opportunity to give the system additional information about the current state of the physical environment (e.g., unforeseen maintenance, altered personnel availability).

At the end of this Q&A phase the goals of the *second phase* of the interaction, where the user becomes an operator who wants to achieve the goals specified in the first phase, are determined and fixed. In the second phase, CALLIOPE behaves more or less like a human *coach*: it gives instructions, explanations (it helps), and asks the user/operator to provide it with information.

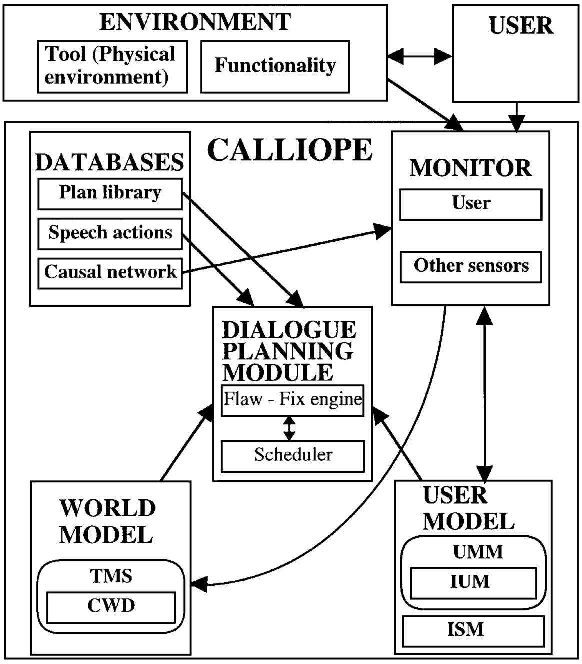

Figure 2. The modular architecture of CALLIOPE in its physical environment.

We remark that, as a consequence of the particular AI planning paradigm that we have adopted in CALLIOPE, the above two phases do not need to be strictly separated. Interleaving the goal determination phase with the goal satisfaction phase is perfectly possible in our approach. The latter is in particular supported by using the user as a physical sensor of CALLIOPE (see modular description of the CALLIOPE architecture). In the following description, however, we assume that the first phase has already been completed and that the goals of the coaching session are known to the system.

As a guide for the description of CALLIOPE we will use Fig. 2 which depicts the overall structure of the system and its physical environment. The design of the system is highly modular, mainly because CALLIOPE aims to be generic.

- *Tools* are part of the physical environment of the system. A tool may, in this context, be any physical device with which an operator can perform a task. Examples of tools are control systems, plumber tools, intelligent or conventional computers and their interfaces. We will refer to the combination of a tool and one of its possible functionalities as the *application*. CALLIOPE, as a generic coach, has a wide range of possible functionalities on a wide range of tools.
- The *user/operator* is the other component of interaction that resides outside CALLIOPE. Examples of operators in the applications we envisage for CALLIOPE are chemical plant operators, plant engineers, production managers, and sales managers.

We now describe the different modules of CALLIOPE.

- The central module of the CALLIOPE system is the *dialogue planning module*. This module contains the engine of the intelligent dialogue system. To go into more detail we need to explain how this engine works and how it determines the way in which time proceeds for the entire system. The dialogue planning module is essentially a domain independent, hierarchical, nonlinear (AI) planning system, which gives the possibility to interleave planning, execution and monitoring. In this module we avoid the frame problem through the STRIPS assumption [9]. Other assumptions correspond to the assumptions made in TWEAK [4]. In CALLIOPE, the planning module consists of a simplified version of the planning shell IPEM [2]. IPEM is an acronym for Integrated Planning, Execution and Monitoring. IPEM traverses a search space of partial plans (plans with flaws) by sequentially improving (i.e., by fixing the flaws of) an initial plan.
  A *scheduler* is at the heart of the system. This scheduler is the internal clock of CALLIOPE and is depicted in Fig. 3. It determines the problem solving strategy by deciding which flaw has to be put highest on its agenda, i.e., which flaw has to be fixed next. An example of a flaw in a partial plan of IPEM is "unexecuted action *A*". This flaw can be fixed by executing action *A*.

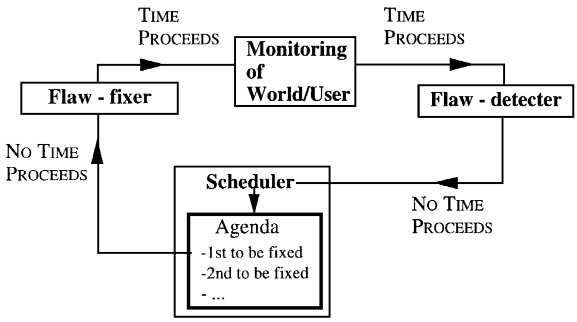

Figure 3. The dialogue planning module of CALLIOPE.

The scheduler receives input from the *flaw-detection component*, which scans the partial plan at hand for flaws. The scheduler updates its *agenda* with the newly received input, and sends the top flaw to be fixed to the flaw-fixing part of the engine. The *flaw-fixing component* is responsible for the execution of the fixes, and, therefore, also for the execution of actions. An important example of a flaw that may have to be fixed is an unexecuted dialogue action. This can be fixed by the physical presentation of dialogue or speech actions to the user. During this three-step process of the planning component CALLIOPE has no contact with the outside world, which by assumption remains unchanged for the system. Therefore, we mark this physical period of time during which nothing changes for CALLIOPE as "no time proceeds" in Fig. 3.

- The period in which time does proceed is characterised by *monitoring* the physical environment of the system. CALLIOPE has at its disposal a number of monitors on the physical world, one of which is its human user/operator. Other sources of information are physical sensoring devices. Especially in remedying applications, the operator is used as a sensor to the physical world, where he has to provide observable facts to CALLIOPE. In these applications, other sensors are, for instance, the process computer of a chemical plant which can, at fixed intervals, send updates of certain physical parameters, like temperatures and pressures, to the system. The realisation of the user as a sensor on the world is explained in more detail in Section 3, where we focus on the concretisation of the dialogue, and the way in which the dialogue system communicates with the human user.
- These monitored observations are transferred to the *World Model* where they are recorded. It is this model that notifies the planning component of possible changes in its view on the world. The system's model of its environment is updated by newly observed facts that are passed to the World Model. The sensored data are transferred by the physical sensors or by the user to the *Current World Description* (CWD) which is a database of facts about the world that are stored in a declarative, Prolog-like, way. At any time the CWD consists of a list of facts. A causal network enhances the pool of monitored facts by deriving unmeasurable consequences from the sensored data. Typically, a causal network is a set of application-dependent rules. A Truth Maintenance System (TMS) makes sure that the CWD database remains consistent. The CWD is only updated with monitored data and can therefore be used to verify if instructions that were given to the operator have been successfully executed by checking if the expected results of the instructions have been observed.
- For the part of the *User Model* an analogous process takes place. For our example applications there are typically several users involved. In chemical process industry, these users are the operators that fill the different work-shifts during a day. The skills and the experience of these users varies, as some operators are novices while others are very experienced. Therefore, the behaviour of CALLIOPE is tailored to the user/operator at hand. For each user known to the system, an *Individual User Model* (IUM) is maintained. This IUM contains data about the user's capability to perform specific tasks, his experience in performing certain operations, about his skills and his knowledge of certain physical facts. His experience in performing certain tasks is, for instance, determined by his success-rate in performing equal or similar tasks in the past. When, for instance, an operator has informed the system of a physical fact *A* (how a user can do this is explained in the following sections) the IUM's belief that this particular user knows *A* is very high. The IUM's are also updated by monitored data in a similar fashion as the CWD is updated. Here again, the operators themselves are used as sensors (e.g., if they state that they do know some fact) and also the user's capability to execute given instructions is monitored. This is achieved by monitoring if the state of the physical environment that should have resulted from a given instruction is monitored within a reasonable time. If this happens, the User Model's belief that the user is capable of performing this instruction increases. Concretely, the IUM contains a list of numerical ratings which express CALLIOPE's belief in statements about the user's knowledge and skills. The IUM is updated during an interaction with monitored data and with additional information coming from other information sources (e.g., rules and stereotypes), that are at the disposal of the user modelling component. The user modelling component consists of a *User Model Manager* (UMM) and an *Information Sources Manager* (ISM), and it sends news to the planning module in the same way as the world description module does.

- A final component of CALLIOPE is a set of three databases. The first is a library of *skeletal plans*. Skeletal plans are stored in a *plan library* and are used as scenarios, as will be explained further on in this section. Another database contains the three classes of *speech actions*. These will be discussed in more detail in Section 3. A third library contains a *causal network*. This library is largely application-dependent.

We now return to the functioning of CALLIOPE. At the start of the second phase of the dialogue, when a task is determined for the user/operator to perform, this task is mapped on a *skeletal plan* taken from the plan library. Skeletal plans may be the result of logging previous sessions with CALLIOPE on the same application, or they may be predefined by the system's designer. Scenarios are typically application-dependent. The mapping of a goal on a skeletal plan is always possible since the plan library contains a very simple partial plan consisting of two actions. In the terminology of IPEM, this simple plan consists of the BEGIN action with the initial CWD as its effects and the END action with the goal state of the world, i.e., the tasks, as its preconditions. In some other cases, the skeletal plan might already contain some steps in the interaction.

After the first skeletal plan is loaded from the system's libraries, the planning module of CALLIOPE refines this partial plan taking the user/operator (e.g., his level of expertise, his skills, and his dialogue preferences) and the situation of the current world (e.g., the situational complexity, crisis or no crisis, and operator's errors) into account. Not all actions that the planning module plans result in a communication with the user/operator. In order to fix some partial plan flaws, however, the planning module plans *dialogue actions*.

For our implementation of the system we have chosen a set of three dialogue actions: *INFORM*, *INSTRUCT*, *INSTANTIATE*. In the next section, we will take a closer look at these dialogue actions which we refer to as the I.I.I.-system. The execution of speech actions triggers a procedure in the physical interface of CALLIOPE. Thus far, we have neglected the interface part of the system.

Of course, the CALLIOPE approach forces strong implications on the physical interface it needs. For this reason a *generic graphical interface* (GGI) has been designed [16]. Firstly, the interleaved planning and execution approach enhances the communication between the system environment and the user, and in particular it makes the GGI necessarily user-driven. For instance, user actions should be allowed at any time, including in the course of the planning/execution. Secondly, the user model permits CALLIOPE to adjust the communication style and form of assistance during the session. Thus, the initial GGI might need to evolve significantly by changing both its appearance and its functionality. Dynamic configuration facilities and an interpreter are embedded into the GGI in order to secure its run-time adaptability to changing user models. We are not going into depth here, because it is of limited relevance to the problems discussed in this paper. For the sake of argument, we can simply assume that the execution of speech actions is a message appearing on a screen.

3. I.I.I.-Dialogue or Speech Actions

In this section, we present a system of three dialogue or speech actions which correspond to the normal modes of speech: the declarative, the imperative and the interrogative mode. They are:

- **INFORM**,
- **INSTRUCT**, and
- **INSTANTIATE**.

CALLIOPE *informs* the user of facts that the user needs to know to be able to fulfill some task correctly or to achieve some subgoal. The system *instructs* the operator to perform certain physical actions or operations (this terminology motivates the name operator for a user). CALLIOPE uses the operator as a sensor to the world by asking him to *instantiate* the values of some parameters that are unknown to the system and that the system may need, e.g., to fix further flaws.

As argued in the Introduction, we need only a fragment of what is usually considered appropriate for a man-machine dialogue since we restrict ourselves to *task-oriented dialogue*. Therefore, these three speech actions suffice from the system's point of view. From the user's point of view, CALLIOPE can also handle user requests by simply adding them as goals to the goal state description.

We take a closer look at the anatomy of the three speech actions by using the graphical representation system introduced by Steel [23]. In the design and implementation of CALLIOPE, we use a representation based on predicates to describe facts and propositions in a declarative way concerning the world and the user.

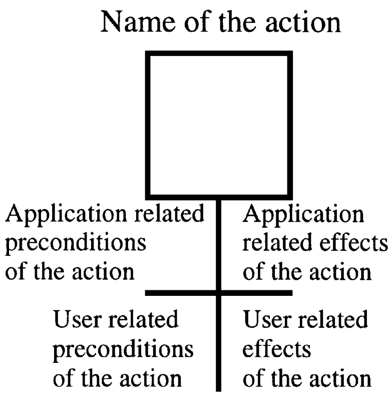

Figure 4. Most general graphical representation of an action.

Figure 4 specifies, in a graphical formalism, the partitioning of preconditions and effects of an action in application related propositions and user related propositions. Further in this section, we will focus on these preconditions and effects for the three dialogue actions. This graphical notation will also be used in the discussion of an example in the next section.

Figure 5 gives INSTRUCT, INFORM and INSTANTIATE in the notation of Fig. 4.

For the INSTRUCT action, a physical operation *Op* is instructed. The application related preconditions of such an action consist of the physical conditions to perform *Op*. If, for example, the instruction concerns the closing of a valve, an example of such a precondition may be $pressure(valve) < 75$. A user related precondition is a condition on the user/operator's knowledge and skills. An example is the fact that the user is supposed to know that the valve is a right-turn and not a left-turn valve. Application related effects of such an action include changes in the values of physical parameters. These changes are monitored by the user or by the system's sensors and are passed to the CWD. User related effects are, for instance, the fact that the user's capability to close valves increases.

The system informs the user by means of the INFORM action of some application related fact a. Typical preconditions in this case are the fact that CALLIOPE, through its user model, believes that the user is ignorant of a but is capable of understanding it when a is presented to him. Another precondition is that the system believes that a is true, which means that a is in the CWD of the system. The information is presented to the user in a "professional language", i.e., a language which any user/operator that is active in the particular application domain is assumed to understand. Effects of an INFORM action only concern the user: the user model's belief that the user knows a will increase after the user has acknowledged the information contained in the INFORM action. Heuristics are used to adjust parameters in the IUM of the present user.

Before turning to the INSTANTIATE action we have to introduce two special constants (see [2]). $x?$ denotes a constant value which is yet unknown to the dialogue planning system but which the dialogue planning system wants to retrieve by using the user as a sensor on the world and by letting him instantiate the exact value. $x!$ is the variable x instantiated with a constant that is unknown until the execution of the plan action in which $x?$ appears. The INSTANTIATE dialogue action is performed by the system when it wants to be informed by the user of some fact concerning its physical environment or concerning its user. Preconditions of this dialogue action concern the user's capability to know $a(x!)$. After the dialogue action is successfully completed, i.e., when the system has received the answer of the user, the system knows $a(x!)$ and this is added to the CWD of the system.

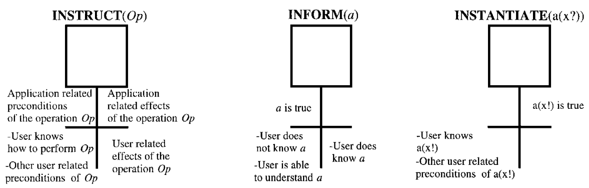

Figure 5. Graphical representation of the I.I.I.-actions.

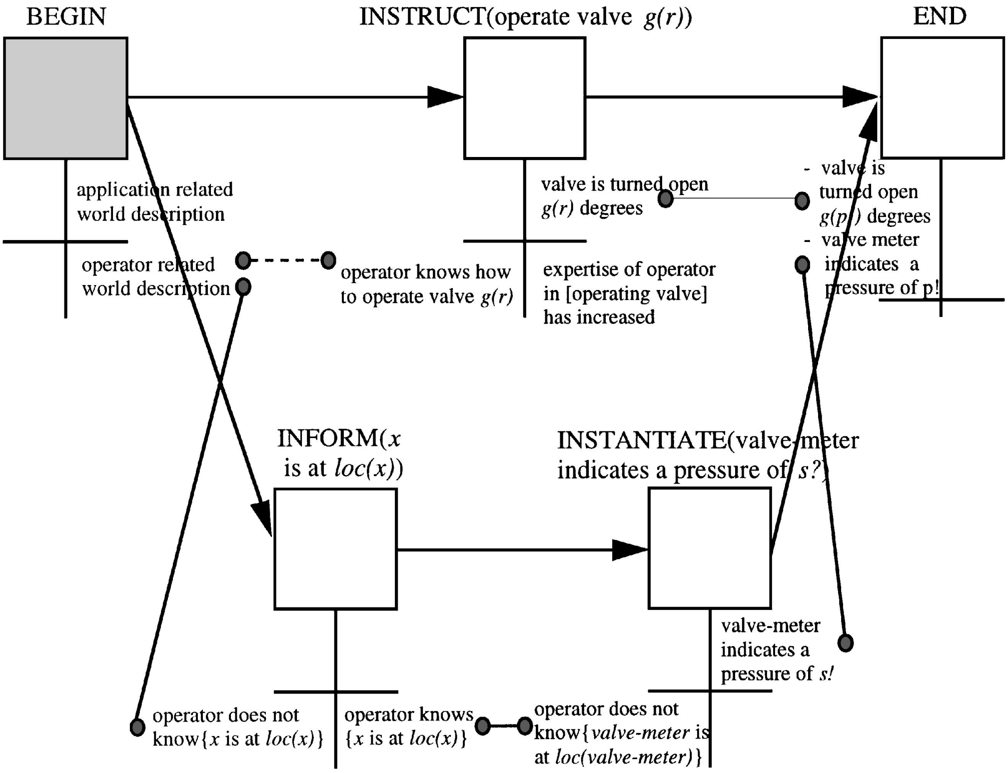

Figure 6. The example: a partial plan.

4. An Example Dialogue: Coaching an Operator in a Chemical Plant

In this section, we illustrate the I.I.I.-formalism by means of a real-life example. The application is taken from a remedying task in a chemical process plant and the dialogue fragment we take as an example concerns the closing of a right-turn valve over $g(p_{\text{valve}})$ degrees, where p_{valve} is the pressure of that valve. We first describe what happened before the situation in Fig. 6 was reached. CALLIOPE started to plan from an empty initial partial plan, i.e., an initial plan consisting of two actions only—BEGIN with the CWD as application related effects and the IUM as user/operator related effects, and END with the tasks to be completed as preconditions. The initial planning actions of CALLIOPE were guided by the fact that the actual pressure on the valve was (yet) unknown to the system. Therefore, an INSTANTIATE(valve-meter indicates a pressure of s?) was inserted in the partial plan. If the preconditions of this INSTANTIATE were supported by the CWD and IUM, planning the INSTANTIATE would have effectuated the support of one of the two goals of the coaching session. As a second step INFORM (x is at loc(x))—loc being a predicate that gives information about the spatial location—was inserted, since the single precondition of this INSTANTIATE action was not supported by the IUM in the effects of BEGIN. For similar reasons, the INSTRUCT(operate valve $g(r)$) has been inserted. After these planning steps, we arrive at the situation of Fig. 6: also the second goal is supported. In Fig. 6, all the executed actions have been marked grey. Obviously, at this stage only BEGIN is executed, since its effects coincide with the CWD at the beginning of the coaching session. In the situation depicted in Fig. 6, there are two possible continuations. If "operator knows how to operate valve $g(r)$" is in the current IUM (i.e., if this precondition is supported by the BEGIN action), the scheduler can decide to put the execution of INFORM(x is at loc(x)) at the top of its agenda (i.e., as the first flaw to be fixed). On the other hand, if "operator knows how to operate valve $g(r)$" is not in the IUM, a hierarchical expansion of INSTRUCT(operate valve $g(r)$) is desired. The planner will replace the original INSTRUCT action by a sequence of three

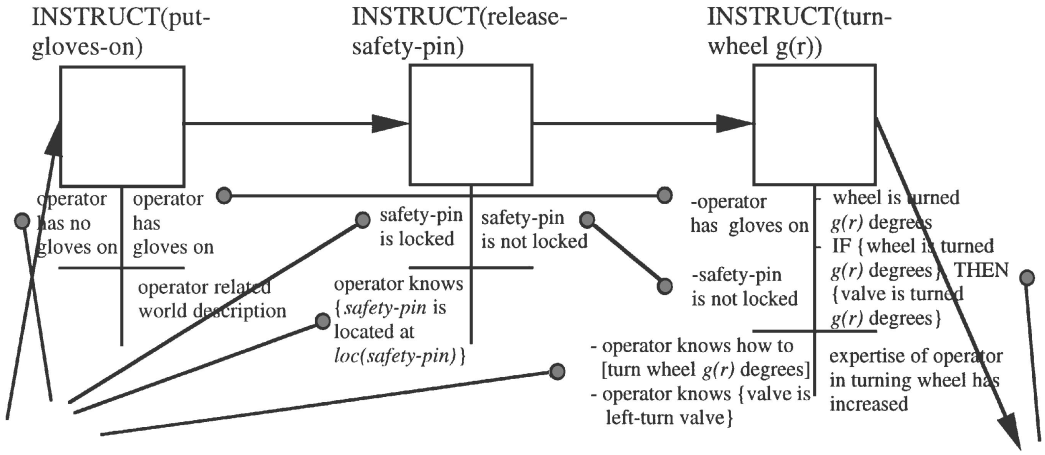

Figure 7. An hierarchical zoom on the INSTRUCT of Fig. 6.

INSTRUCTs. This hierarchical zoom-in is illustrated in Fig. 7 where the three INSTRUCT actions are shown that replace the INSTRUCT(operate valve $g(r)$) of Fig. 6. Unsupported, user related preconditions of the type "operator knows how to perform *Op*" typically trigger hierarchical zooms. For more details of this mechanism of hierarchical expansions we refer to [2].

When the INSTRUCT(put-gloves-on) action has been executed successfully, the operator is presented with the INSTRUCT(release-safety-pin) dialogue action. If at this point the operator fails to unlock the safety-pin, CALLIOPE will detect this human error, e.g., because "safety-pin is locked" will not be negated by any monitored data. This error will lower the numerical ratings expressing the belief of CALLIOPE in the user's ability to unlock safety-pins. This could have several consequences. Either CALLIOPE's confidence in the operator remains sufficiently high and the INSTRUCT(release-safety-pin) is simply repeated or an even deeper hierarchical expansion is warranted. For example, an INFORM(safety-pin is located at loc(safety-pin)) may be planned to increase the user's knowledge and skills.

5. Implementation Issues

CALLIOPE has been implemented in C.

To tailor CALLIOPE to an application, the empty databases (see Fig. 2) have to be filled with the appropriate knowledge. Because of its generality the CALLIOPE system can be used for different communication purposes: as a coach; a "on the job" training system in a real or simulated environment; an intelligent tutoring system; a consulting system; an intelligent interface; and as a help system.

We have tested CALLIOPE's performance and compared it to, e.g., the performance of IPEM in a block world environment. In these tests, CALLIOPE's libraries contained the block world environment and speech actions appropriate to guide a user to perform robot-like tasks in the block world. Compared to IPEM, the Sussman anomaly is solved instantaneously. The gain in speed in comparison to the shell IPEM is mainly due to employing a more or less fixed search strategy allowing only backtracking on the order of subgoals to be satisfied. Also, domain-dependent heuristics for determining primary effects of operators are used (see e.g., [10, 11]).

At present, three-tailored versions of CALLIOPE are in operational use in different application domains: operator support for remedying tasks in chemical process industry, operator support for a combined task of planning, plan execution and process control in the area of chemical process development, and thirdly decision support in production scheduling.

In the first application, CALLIOPE coaches operators of different expertise who have to perform manual remedying tasks in a chemical plant. This implementation is described in [6]. The example in Section 4 has been extracted from this application.

In the second application, the dialogue planning module of CALLIOPE has been tightly coupled with a tool that supports an operator in performing a combined task of planning, plan execution and process control. The planning module in this application has been augmented with the conjunctive use of constraint satisfaction techniques together with stochastic techniques. Essentially, the system distinguishes between two levels of user expertise. There are novice users and expert users. The latter are given more extensive control of the quality of response and thus become a more (inter)active partner in the overall problem solving process.

In a third application, the decision support tool is strongly integrated with the CALLIOPE planning module. On the basis of demands and resources production schedules are generated cooperatively by the scheduler and the user even to such a large extend that the user can be considered as being an integral part of the system definition. The central concept of the interactivity of the tool is the concept of scenario. A scenario is nothing more than a parametrised schematic skeletal plan. It contains user intuitions about plant priorities in a given situation. The user himself is responsible for selecting the scenario that fits best the situation at hand.

A user model models a hierarchy of authority and expertise among the users and enforces this hierarchy on the scenario-base.

The scheduling tool itself is based on the integration of classical mathematical optimization, stochastic optimization, and constraint satisfaction [7, 8].

6. Conclusion

We have successfully used the domain independent AI planning paradigm as a basis for a generic man-machine dialogue. However, we have taken the justified comments from the Cognitive and Social Sciences into account: taking a plan as a sole basis for communication is inadequate. Therefore, we consider the underlying plans as vague scenarios which get their completion and adjustments during their execution. Thus, we are achieving real interleaving of planning execution and monitoring. To enhance the limited access that machines have to their environment we make use of a causal network and we employ the user as one of the sensors. Furthermore, we make use of a user model which, in combination with an interleaved planning, execution, and monitoring approach, is able to cope adequately with the so-called human errors. In combination with the hierarchical planning employed in CALLIOPE, the user model is also responsible for communicating with the user at his level of expertise. In this manner, we achieve a dialogue that is adapted to the moment and to the user.

Acknowledgments

The above text presents research results from the project IT/IF/7 under the supervision of Prof. M. Rijckaert and executed in the framework of the Belgian Incentive Program "Information Technology"—Computer Science of the future, initiated by the Belgian State—Prime Minister's Service, Science Policy Office. The scientific responsibility is assumed by its authors. The authors are indebted to the referees for numerous valuable pointers to the literature and for suggesting clarifications to earlier versions of the text.

Note

1. There are however exceptions: (1) NOAH [22], which deals with the problem of interactive hierarchical instruction, and (2) work in the field of natural language understanding [1, 3], where the planning paradigm is adopted as a basis for communication in the sense that language is a form of action, which is analysed with respect to an underlying plan.

References

1. J.F. Allen and C.R. Perrault, "Analysing intentions in utterances," *Artificial Intelligence*, vol. 15, no. 3, pp. 143–178, 1980.
2. J.A. Ambros-Ingerson and S. Steel, "Integrated planning, execution, and monitoring," in *Proceedings of AAAI 1988*, 1988, pp. 65–69.
3. D.E. Appelt, "Planning English referring expressions," *Artificial Intelligence*, vol. 26, pp. 1–33, 1985.
4. D. Chapman, "Planning for conjunctive goals," *Artificial Intelligence*, vol. 32, pp. 333–377, 1987.
5. P.R. Cohen and C.R. Perrault, "Elements of a plan-based theory of speech acts," *Cognitive Science*, vol. 3, pp. 177–212, 1979.
6. K. Dockx and R. Timmermans, "Adding a temporal dimension to expert systems working in a real-time environment," *Communication and Cognition—Artificial Intelligence*, CCAI, vol. 9, nos. 2–3, pp. 253–257, 1992.
7. K. Dockx and K. Meert, "Production scheduling in a chemical process environment based on a simulated annealing algorithm," *Journal A*, vol. 34, pp. 41–46, 1993.
8. K. Dockx, K. Meert, and B. Kuijpers, "Production scheduling in a chemical environment based on simulated annealing," in *Proceedings of AAAI SIGMAN Workshop*, New Orleans, 1994.
9. R.E. Fikes and N. Nilsson, "STRIPS: A new approach to the application of theorem proving to problem solving," *Artificial Intelligence*, vol. 5, no. 2, pp. 189–208, 1971.
10. E. Fink and Q. Yang, "Automatically abstracting effects of operators," in *Proceedings of the First International Conference on AI Planning Systems*, 1992, pp. 243–251.
11. E. Fink and Q. Yang, "Characterizing and automatically finding primary effects in planning," *IJCAI-93*, pp. 1374–1379, 1993.
12. B.J. Grosz and C.L. Sidner, "Plans for discourse," in *Intentions in Communication*, edited by P.R. Cohen, J. Morgan, and M.E. Pollack, MIT Press: Cambridge, MA, pp. 417–444, 1990.

13. R.W. Hill, Jr. and W.L. Johnson, "Situated plan attribution for intelligent tutoring," in *Proceedings of the Twelfth National Conference on Artificial Intelligence*, *AAAI-94*, 1994, pp. 499–505.
14. E. Hollnagel and D.D. Woods, "Cognitive systems engineering: New wine in new bottles," *International Journal of Man-Machine Studies*, vol. 18, pp. 583–600, 1983.
15. S. Kambhampati, "On the utility of systematicity: Understanding tradeoffs between redundancy and commitment in partial-order planning," *IJCAI-93*, pp. 1380–1385, 1993.
16. K. Kanev and K. Dockx, "A framework for graphically-oriented human computer interactions in intelligent operator support systems," *Comput. and Graphics*, vol. 18, no. 4, pp. 563–570, 1994.
17. J.D. Moore, *Participating in Explanatory Dialogues. Interpreting and Responding to Questions in Context*, MIT Press: Cambridge, 1995.
18. D.J. Musliner, "Using abstraction and nondeterminism to plan reaction loops," in *Proceedings of the Twelfth National Conference on Artificial Intelligence*, *AAAI-94*, 1994, pp. 1036–1041.
19. S.H. Nwana, "Intelligent tutoring systems: An overview," *Artificial Intelligence Review*, vol. 4, pp. 251–277, 1990.
20. L. Quinn and D.M. Russel, "Intelligent interfaces: User models and planners," in *Proceedings CHI '86*, April 1986, pp. 314–320.
21. J. Rasmussen, "Cognitive systems analysis for risk management," in *Proceedings of SITEF 91—International Symposium on Cognitive Interactions*, 1991, pp. 24–25.
22. E. Sacerdoti, "Planning in a hierarchy of abstraction spaces," *Artificial Intelligence*, vol. 5, no. 2, pp. 115–135, 1975.
23. S. Steel, "The bread and butter of planning," *Artificial Intelligence Review*, vol. 1, pp. 159–181, 1987.
24. L.A. Suchman, *Plans and Situated Actions. The Problem of Human Machine Communication*, Cambridge University Press, 1987.
25. A. Tate, J. Hendler, and M. Drummond, "A review of AI planning techniques," in *Readings in Planning*, edited by J. Allen, J. Hendler, and A.Tate, Morgan Kaufmann Publishers: California, pp. 26–49, 1990.
26. P. Van Bael, "Scheduling chemical batch processes with transfer times and sequence dependent set-up times in ZW or NIS policy," *The Second International Conference on the Practical Applications of Constraint Technology*, London, UK, 1996, pp. 453–464.
27. R. Wilensky, D.N. Chin, M. Luria, J. Martin, J. Mayfield, and D. Wu, "The Berkeley UNIX consultant project," *Computational Linguistics*, vol. 14, no. 4, pp. 35–84, 1988.