Supporting Interaction in the ROBOCARE Intelligent Assistive Environment

Amedeo Cesta, Gabriella Cortellessa, Federico Pecora, and Riccardo Rasconi

ISTC-CNR, National Research Council of Italy
Institute for Cognitive Science and Technology
Via S.Martino della Battaglia 44, I-00185 Rome, Italy
{name.surname}@istc.cnr.it

Abstract

The ROBOCARE project is aimed at creating an instance of integrated environment with software and robotic agents to actively assist an elderly person at home. In particular the project has synthesized a proactive environment for continuous daily activity monitoring in which an autonomous robot acts as a main interactor with the person. This paper accounts for the combination of features that create interactive capabilities for personalized and mixed-initiative interaction with the assisted person.

A project with an intelligent assistant

The use of intelligent technology for supporting elderly people at home has been addressed in various research projects in the last years (Pollack *et al.* 2003; Pineau *et al.* 2003; Haigh *et al.* 2003; Pollack 2005).

Increasing attention is also given to the synthesis of Cognitive Systems to produce aids that enhance human cognition capabilities. As an example, the project CALO (Myers 2006; Myers & Yorke-Smith 2005) has as its primary goal the development of cognitive systems able to reason, learn from experience, be told what to do, explain what they are doing, reflect on their experience, and respond robustly to contingencies. Other projects somehow connected to different aspects of this research topic are CMRADAR (Modi *et al.* 2004), whose aim is to develop an intelligent scheduler assistant, and CAMEO (Rybski *et al.* 2004) whose main objective is to build a physical awareness system to be used by an agent-based electronic assistant.

All these projects have required the orchestration of different intelligent software technologies and have highlighted a number of important issues that need to be addressed: the problem of coordinating the distributed components as well as the problem of providing intelligent interaction with the user, are undoubtedly among the most critical.

The ROBOCARE project shares several of the challenges with the above mentioned projects. Indeed ROBOCARE's main motivations can be summarized as follows[1]:

> "The objective of this project is to build a distributed multi-agent system which provides assistance services for elderly users at home. The agents are a highly heterogeneous collection of fixed and mobile robotic, sensory and problem solving components. The project is centered on obtaining a virtual community of human and artificial agents who cooperate in the continuous management of an enclosed environment."

The project has involved research groups with different background with the goal of investigating how state of the art AI techniques could be combined to create new home-service integration for elderly people (Cesta *et al.* 2003; Cesta & Pecora 2006; Bahadori *et al.* 2004).

As a target domain we have chosen a prototypical home environment where the presence of an intelligent assistant would be of concrete help in the daily life of a cognitively impaired user through the integrated performance of advanced distributed components.

The most important capability of an intelligent assistant is the continuous maintenance of a high level of situation awareness. This objective is obtained through the interaction of a number of intelligent physical and/or software agents: among others, vision-based sensors, which ensure the acquisition of continuously updated data from the environment; a schedule management software agent, which analyzes the status of every activity being performed within the monitored space; a mobile robotic platform able to behave robustly and continuously in the environment.

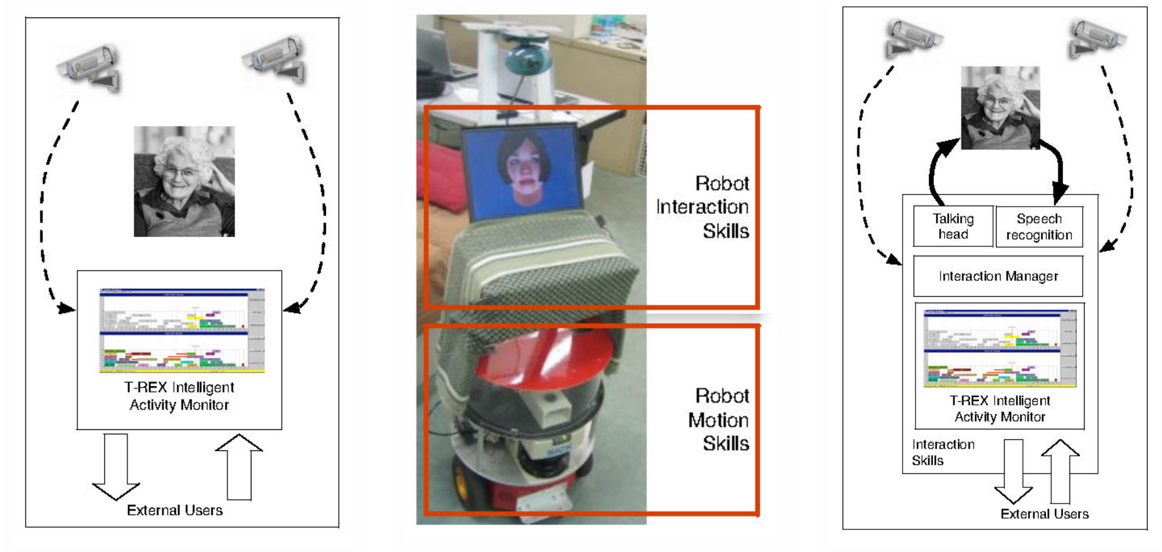
The ultimate goal of the overall system is to provide cognitive support both *on-demand*, by guaranteeing a real-time question&answer service situated to the contextual knowledge of an assisted person, and *proactively*, by providing an event-driven support again grounded on what is going on in a daily living environment.

Our efforts to integrate an autonomous robot in the intelligent environment have immediately made clear that the latter might naturally play the role of attention focalizer and main interactor with the user. Therefore we have concentrated most of the interaction capabilities in the robot, and, additionally, have chosen verbal communication as the main interaction modality.

This paper specifically concerns the aspects related to the system's context-awareness as well as interaction capabilities. In particular we describe how the constraint-based scheduling technology is used to maintain a knowledge repository aimed at supporting on-demand specific in-

Copyright © 2007, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

[1] Quote from the original project proposal.



(a) The "silent observer" (b) The current robotic assistant (c) The "proactive interactor"

Figure 1: Evolving ROBOCARE demonstrators.

teractions as well as enabling autonomous system initiative.

The chosen domain and the addressed problem

After having studied features of different physical environments for elderly people[2], the home environment has been selected to develop the fielded integrated demonstrator. This choice is supported not only by the aim of improving home technology personalization, but also by recent studies, e.g., (Giuliani, Scopelliti, & Fornara 2005), that underscore the relevance of the attachment of elderly people to their home and the beneficial effects of increasing their independence at home.

Figure 1(a) shows our first step toward the creation of an effective combined support service. We integrated sensors that monitor users' activities[3] with a schedule management environment called T-REX, (Tool for schedule Representation and EXecution (Pecora *et al.* 2006)), through which it is possible to represent of a set of activities and their quantitative temporal connections (i.e., a schedule of activities that the user is expected to carry out). The broad idea is to allow the specification and then the execution monitoring of a set of activities that the person usually performs or needs to perform due to prescription (on suggestion from his personal doctor for example).

T-REX integrates a constraint-based scheduler (Cesta *et al.* 2001) with additional features for knowledge engineering, in particular for problem modeling. Particular attention has been given to the definition of "user-oriented terminologies" in order to easily synthesize both the basic elements of a target domain as well as different problem instances in the particular domain (i.e., in the form of activities and constraints among activities). For example, in the ROBOCARE context, T-REX allows the definition of "home domain activities" like *breakfast*, *lunch*, *go-for-walk*, and also temporal/causal links among activities like *meal-bound-medication* to express rules like "aspirin cannot be taken before eating". Through this high level language an external user (a doctor, a relative of the assisted person, etc.) may define a network of activities, a *schedule*, that the observed person is supposed to carry out in the home environment during the day. Such a schedule is dispatched for execution and monitored using the underlying schedule execution technology. Information coming from the sensors is used for maintaining an updated representation of what is really happening in the environment. Even if human activity recognition[4] is outside the scope of the project, it is worth highlighting how the sequence of observations from the artificial vision sensors allows to follow the evolution of the activities of the observed person (e.g., if and when she took a pill, when she had lunch, etc.). Based on the synthesis of these observations, the system is able to generate a report for the external users that underscores when the person's activities have been performed within "reasonable" temporal boundaries or when important anomalies or even violations on their execution have been detected (Cesta *et al.* 2005; Bahadori *et al.* 2004). In this light the demonstrator in Figure 1(a), is a basic example of home *Activity Monitor*[5] grounded on scheduling technology. Notice that, on its own, the domestic activity monitor acts as a "silent observer" and does not take initiative with respect to the elder person in any way. In this paper, we show how its indications can be

[2]See (Cesta & Pecora 2006) for a broad presentation of the diversified project initiatives.

[3]In particular we used stereo cameras, endowed with specific software for people locatization and tracking (Bahadori *et al.* 2005).

[4]There is plenty of recent research on activity recognition with sensors, e.g., (Pentney *et al.* 2006), that could potentially impact this class of applications.

[5]Further details on the Activity Monitor will be presented later in this paper.

employed to trigger system initiatives.

In parallel with research on the home environment that "observes a person" the project has worked on setting up an autonomous robotic platform able to robustly behave in a home environment. Our colleagues from the University of Rome "La Sapienza" have set up a Pioneer 2 platform with a Sick laser scanner for localization. Additional work has been requested to both integrate advanced SLAM algorithms (Grisetti, Stachniss, & Burgard 2005) and obtain robust navigation abilities by integrating a topological path planning and a reactive obstacle avoidance module. We summarize the situation of the robotic platform with what is referred to as "robot motion skills" in the lower part of Figure 1(b).

The software/robotic assistive environment

While pursuing the project's goal of actively integrating the robotic platform in the home environment, we had to decide a role for the robotic agent in accordance with the choices already made for the initial demonstrator built in the first step.

The choice has been to move from a "silent observer" to an "active interactor" that could spontaneously decide to intervene in the scene and interact with users. The robot's ability to independently move within the environment inspired our next steps in the development of the intelligent assistant. The robot was the natural candidate to be in charge of managing interaction with the assisted person. Therefore we have enhanced the robot with an additional level of competence, referred to as "interactive skills" in Figure 1(b), which group and provide access to the capabilities of the overall intelligent system.

The sketchy view in Figure 1(c) shows how the interaction skills integrate the capability of the T-REX Activity Monitor, with a simplified Interaction Manager and a front-end for the interaction module consisting in a Talking Head and a Speech Recognition subsystem. The"talking head", called Lucia[6] is also endowed with speech synthesis functionalities based on the elaboration of a specific "content files" in text format.

The specific speech recognition system we are using is Sonic, a tool developed at the University of Colorado[7]. The Talking Head and Sonic endow the robotic assistant with the basic capability of understanding and producing speech in both Italian and English.

Before dedicating the rest of the paper to how different forms of interaction have been generated, the next section describes how we combined together the specialized modules previously introduced. Indeed, we have always pursued the idea of having a multi-agent architecture for supporting the integration of heterogeneous intelligent components.

The coordination subproblem

ROBOCARE requires the combination of various intelligent tools to ensure a comprehensive behavior of the enhanced physical environment. Our goal is to achieve an environment which acts as a *proactive assistant for daily activity monitoring*. This section explains how heterogeneity is kept under control by synthesizing a coordinated behavior.

Coordination of multiple services is achieved by solving a Multi-Agent Coordination problem. This problem is cast as a Distributed Constraint Optimization Problem (DCOP), and solved by ADOPT-N (Pecora, Modi, & Scerri 2006), an extension of the ADOPT (Asynchronous Distributed Optimization) algorithm (Modi *et al.* 2005) for dealing with n-ary constraints. Figure 2 gives an intuition of the chosen approach[8]. We call $Application_i$ the generic intelligent subsystem that is to be integrated in the overall multi-agent system, and Var_j one out of a set $\mathcal{V}$ of variables in terms of which the coordination problem is defined. Each variable has an associated domain of Values D_j. Variables are bound by constraints like in regular Constraint Satisfaction Problems (CSP). Conversely, while constraints in CSP evaluate to *satisfied* or *unsatisfied*, in the optimization case constraints evaluate to costs, and can thus express what are often refered to as "soft constraints". Such constraints are useful for modeling prefrences, and in general requirements which have a "degree of satisfiability". Constraints may involve an arbitrary subset of the variables (n-ary constraints): a constraint among the set $C \subset \mathcal{V}$ of k variables is expressed as a function in the form $f_C : D_1 \times \ldots \times D_k \rightarrow \mathbb{N}$. For instance, a constraint involving the three variables $\{Var_1, Var_3, Var_7\}$ may prescribe that the cost of a particular assignment of values to these variables amounts to c, e.g., $f_{Var_1,Var_3,Var_7}(0, 3, 1) = c$. The objective of a constraint optimization algorithm is to calculate an assignment $\mathcal{A}$ of values to variables while minimizing the cost of the assignment $\sum_{C \in \mathcal{C}} f_C(\mathcal{A})$, where each f_C is of arity $|C|$.

In ROBOCARE, the valued constraints are decided in order to orient the solution of the DCOP toward preferred world situations (broadly speaking those situations in which the person is maximally helped by the intelligent system). The system is composed of a number of heterogeneous applications: (a) T-REX activity monitor, (b) the dialogue manager plus the speech I/O modules, (c) the robot mobile platform, (d) one application for each of the cameras, each of them with the appropriate software for people localization and tracking, (e) a PDA with specialized software that acts as a specialized I/O device for the assisted person.

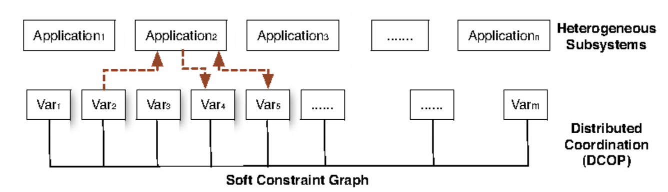


Figure 2: DCOP to maintain distributed coherence.

Each application manages one or more of the variables which are part of the DCOP. A variable may represent (a part of) the input to an application, its output, or both (see the

[6]The head called Lucia is due to our colleague Piero Cosi, ISTC-CNR, Padova, http://www.pd.istc.cnr.it/LUCIA/

[7]See `http://cslr.colorado.edu/beginweb/speech_recognition/sonic.html` for details.

[8]Further details are given in (Cesta *et al.* 2006).

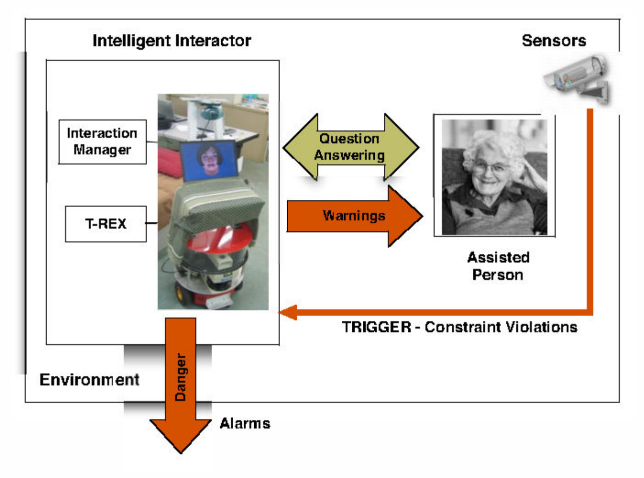


Figure 3: Interaction summary.

dashed lines in Figure 2 as an example). When the DCOP resolution algorithm (ADOPT-N) is called into play, the values of the application-output variables are taken into account in the distributed resolution. When resolution reaches a fixed-point, the input variables will reflect an updated input for the applications. The correspondence between the applications' output at the i-th iteration and their input at iteration $i+1$ is a result of the propagation rules specified in the DCOP. Overall the decisions of the applications constitute the input for the subsequent iteration of the cycle $\langle$*DCOP-resolution; read-variable; application-decision; write-variable*$\rangle$.

It is worth underscoring that the multi-agent solution based on DCOP guarantees continuous control over the whole environment. Additionally, the value functions f_C allow to bias the produced solution towards aggregate behavior which is helpful for the assisted person.

Managing assistant/assisted interaction

As already mentioned, *interaction* within ROBOCARE is a multifaceted problem that presents numerous interesting challenges. All the agents operating within the environment exchange information amongst each other, overall contributing to the creation of the intelligent assistant. Nonetheless, given that we choose to rely on an embodied robotic assistant as the main interactor, we focus here on the "communication channel" between the robot and the assisted person. While presenting the interaction features of the system we will make a distinction according to *who takes the initiative* to start a dialogue. In this light we distinguish between:

On-Demand interaction in which the user takes the initiative first. The assisted person can start interaction through two different modalities. She can use a PDA endowed with a very simple interface developed for the specific domain, to convey commands to send the robot to a specific location, relay streaming video from the robot to the PDA and/or stop the robot. Alternatively, she can use verbal communication to interact with the assistant. In particular the user can ask questions and obtain answers back (e.g., "at what time should I take my pill?").

Proactive interaction in which the intelligent environment commences interaction guided by its internal reasoning. Indeed, an important feature for an intelligent assistant is to understand when to interact in order to effectively and properly support the user. Our work explicitly focuses on the development of active and, at the same time, unobtrusive services to integrate within the artificial assistant. Within ROBOCARE, constraint violations have been considered as a *trigger* for the system to take the initiative and perform some actions (e.g., approach the assisted person, issue an alarm) or verbalizations (e.g., issue warnings and suggestions).

The on-demand interaction PDA $\leftrightarrow$ Robot is managed through the DCOP and will not be further detailed here. Conversely, the rest of the interaction services rely on the Interaction Manager. This module essentially consists in a rule-based system that fires *situation-action* rules. In other words, it continuously assesses the situation and activates a particular submodule as an action.

Situation assessment is carried out with a number of variables under the manager's responsibility. A subset of variables participate in the DCOP coordination, namely those receiving input from sensors and from the T-REX Activity Monitor. Other variables are internal to the agent itself and are associated to the manager and I/O module for speech. The main "interaction occasions" managed in the current version of the intelligent assistant are shown in Figure 3: *Question/Answer*, belonging to the on-demand interaction category, *Danger* and *Warning* as instances of proactive interaction type.

Question/Answer. This is an activity triggered by a speech input from the assisted person. The generation of the answer is managed mostly internally to the manager that has information on the activities' history or on the current state of the environment, to answer questions like "Have I taken the pill?" or "What time is it?" etc. This is clearly the point of the system in which the current solution is quite *ad-hoc* and can be in the future substituted with a state-of-the-art dialogue system.

Danger Scenario. Undoubtedly, one of the important task for assistance is to recognize emergencies for the monitored person. The emergency trigger is fired by particular combinations of the input provided by the sensors that monitor the environment and the assisted person. As an example we can discriminate as a dangerous situation the case in which a person is "lying down in the kitchen floor" or "lying down in bed half and hour after usual wake up", as opposed to "lying down in bed within a given period" which is recognized as a regular situation. The danger trigger is dealt with by a specific behavior of the multi-agent system that interrupts the usual flow of activities and undertakes an action: the robot is sent to the assisted person, a specific dialogue is attempted, and if no answer from the assisted person is obtained, an *Alarm* is immediately fired to the external world (call to a relative, to an emergency help desk, etc.).

Warning Dialogue. This is the case of constraint violations detected by the T-REX activity monitor. Broadly speaking the activity monitor decides the values for the variables that are used by the interaction manager to trigger a proactive dialogue with the assisted person. The content of the dialog is synthesized on the basis of the monitor's internal knowledge, through a mechanism that is described in the rest of the paper.

Overall the Interaction Manager is a quite simple planner that supervises the initiative of the "interactor" towards the assisted person. It is also interesting to underscore how the combination of this manager and the activity monitor endows the whole assistive environment with capabilities of proactive participation in mixed-initiative interaction.

Generating assistive interaction

The ability to detect and follow the actions of the assisted person is one of the central features of the ROBOCARE assistant. The main goal is in fact to guarantee cognitive and physical support while continuously maximizing the user's independence, well-being and safety.

In the ROBOCARE context we employed our scheduling technology as a specific knowledge component. In fact, the T-REX Activity Monitor plays the role of activity *supervisor* and continuously tries to maintain situation awareness by updating the representation of the set of activities that the assisted person performs in the environment.

As mentioned above, the desired behavior the assisted person should adhere to is initially decided by a caregiver (a doctor, or a family member) in terms of a set of activities to be monitored, i.e., the *schedule*. Activities in the schedule are bound to one another through potentially complex temporal relationships.

An important role of the intelligent assistant in this context is played by the management of all the temporal constraints present in the schedule. As the environment sensing cycle commences, the system periodically checks the state of the monitored area, trying to detect and recognize the execution state of all the activities. Regardless of the prescribed behavior described in the baseline schedule, the assisted person is obviously free to act as she likes: this basically means that at each detection cycle, the system is called to precisely assess the possible differences between the actual and desired state. Assessing such differences does not necessarily entail the need for a system reaction. Conversely, when a true constraint violation occurs, reaction is triggered in order to issue suggestions and warnings.

In the ROBOCARE context, we obviously have no control whatsoever in the actions the assisted person is going to perform, despite the caregiver's prescriptions. While monitoring the person's behavior, the system takes note of the evolution of the environment, continuously keeping an updated internal representation of the latter, and possibly reacting to some significant events, if deemed necessary. The monitoring efforts will therefore focus upon: (1) keeping the internal representation of the real world consistent with the behavioral decisions of the assisted person at all times, and (2) performing the necessary rescheduling actions so as to satisfy a maximum number of temporal constraints originally imposed by the caregiver.

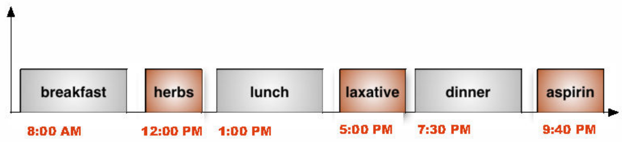


Figure 4: Example of desired behavior specified by the care giver for the assisted person in form of a *schedule*.

An example. To be more concrete, let us consider the monitoring of a behavioral pattern described by a schedule composed of activities $A = \{a_1, a_2, \ldots, a_n\}$, and let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of temporal constraints existing among the activities.

In particular suppose we have 6 different activities corresponding to *breakfast*, *lunch*, *dinner*, as well as taking one of three different medicines, the execution of which should be monitored; due to medical requirements, let us also suppose that these activities must satisfy the following temporal requirements:

1. *breakfast*: should not begin before 8:00 AM; its nominal duration should be 30 minutes;
2. *lunch*: should not begin before 1:00 PM; should not begin before at least 4 hours after the end of *breakfast*; should not begin later than 6 hours after the end of *breakfast*; its nominal duration should be 1 hour;
3. *dinner*: should not begin before 7:30 PM; should not begin before at least 5 hours after the end of *lunch*; should not begin later than 6 hours after the end of *lunch*; its nominal duration should be 2 hours;
4. *taking herbs*: should not begin before 12:00 AM; should not begin after 8:00 PM; its nominal duration should be 10 minutes;
5. *taking laxative*: should not begin before 5:00 PM; should not begin after 9:00 PM; its nominal duration should be 5 minutes;
6. *taking aspirin*: should not begin before the end of *dinner*; should not begin later than 20 minutes after the end of *dinner*; its nominal duration should be 5 minutes.

Figure 4 shows one possible solution to the problem defined above. The schedule represents a solution because, as can be easily confirmed by inspection, all the temporal constraints in the original problem specification are respected. The above schedule represents the behavior the patient should adhere to, and contains all the information regarding the temporal relationships among the activities, the consistency of which should be constantly checked during execution. The aim of the Activity Monitor is to supervise the execution of these activities and check its adherence to the nominal schedule, possibly reacting with warnings and alarms in case of danger.

The execution monitoring algorithm. Once the monitoring starts, the sensors are periodically queried and the nominal schedule is adjusted in accordance with the patient's detected behavior. At each detection cycle, the execution status of each activity is checked: among the possible cases, some activities may be reported as under execution before their nominal start time, the execution of other activities may be reported as delayed, the duration of some activities may exceed the nominal value, and so on; each deviation from the nominal schedule may entail a conflict which has to be reacted upon.

As an example (see figure 4), let us suppose that the patient, after having dinner, sits on the sofa and starts watching TV: at each new sensor detection cycle, the system assesses the situation and delays the *aspirin-taking* activity. But since, according to the medical requirements, the aspirin should be taken no later than twenty minutes after dinner, delaying the aspirin activity beyond the prescribed time will eventually determine a time constraint insertion conflict in the data base of constraints; upon the occurrence of this conflict, the intelligent assistant will possibly respond by issuing a warning to the patient as a reminder for the forgotten action.

Algorithm 1 shows the execution monitoring algorithm employed in the ROBOCARE context. As shown, an "environment sensing" action is periodically performed (line 2). This occurs by accessing the symbolic representation of the current situation (S_t), that is obtained by reading the DCOP variable described previously. As a result, the set $Events_t$ of the occurred events is periodically acquired. By *event* we mean any mismatch between the expected situation, according to the caregiver's prescriptions, and the actual situation (i.e., a planned action which fails to be executed, is considered as an event).

Algorithm 1 The Execution Monitoring Algorithm.

1. **while** true **do**
2. $Events_t \leftarrow S_t$
3.   **if** $Events_t \neq \emptyset$ **then**
4. $C_{r,t} \leftarrow$ removeConstraints()
5.   insertContingencies($Events_t$)
6. $K_t \leftarrow \emptyset$
7.   **while** $C_{r,t} \neq \emptyset$ **do**
8. $c_j \leftarrow$ chooseConstraint($C_{r,t}$)
9.   **if** $\neg$ re-insertConstraint(c_j) **then**
10. $K_t \leftarrow K_t \cup c_j$
11.   **end if**
12.   **end while**
13.   **end if**
14. **end while**

If events are detected, the first action is to remove all the active constraints present in the schedule (line 4). By *active* constraints, we mean those which do not completely belong to the past, with respect to the actual time of execution t_E[9]. All constraints that are not idle are active. Obviously, idle constraints do not take part in the analysis because they will not play any role in the evolution of the future states of the world.

[9]More formally, given an execution instant t_E and a constraint c_k binding two time points t_a and t_b, c_k is considered *idle* if and only if $(t_a < t_E) \wedge (t_b < t_E)$.

In the next step (line 5) all the detected contingencies, properly modeled as further constraints, are inserted in the plan. This is the step where the system updates the internal representation of the schedule in order to preserve consistency with the world's true state.

Lines 7–12 implement the constraint re-insertion cycle, where the algorithm tries to restore as many caregiver requirements as possible given the current situation. Notice in fact that it is probable that not all the original constraints will be accepted at this point: the occurrence of the contingencies might in fact have changed the temporal network constrainedness, so as to make impossible the complete re-insertion of the constraints removed at the previous step. During the cycle, all the constraints which are rejected are stored in the set K_t. Constraints insertion (and rejection) is an extremely delicate issue, for many reasons:

– System reaction may consist in verbal suggestions or warning. The information conveyed by these messages strongly depends on the contents of the set K_t. As we will show, the analysis of all the rejected constraints quantitatively and qualitatively determines the system's response. Given a temporal network TN underlying the current schedule, the set $K_t = \{k_{t,1}, k_{t,2}, ..., k_{t,r}\}$ must be such that: (1) the insertion of each $k_{t,j}$ in TN causes a propagation failure; (2) the cardinality of K_t is maximum. Condition (1) ensures that every constraint in K_t plays a role in determining system's reaction, ruling out false-positive situations; condition (2) ensures that no contingency escapes system's attention.

– The acceptance of each constraint c_j (and complementarily, the contents of K_t), is generally dependent on the particular order chosen for re-insertion. In general, a number of different choice heuristics (chooseConstraint() method) can be envisaged, leading to different approaches for contingency management. To clarify this issue, let us consider a temporal network TN and two constraints c_1 and c_2 such that the attempt of posting both of them in TN would determine an inconsistency: in this case, if the insertion order is $\{c_1, c_2\}$, then c_2 is going to be rejected; if the opposite order is used, c_1 is rejected. Since in the ROBOCARE context it is essential that the reaction be related to the closest contingency with respect to execution time t_E, the particular heuristic employed for re-insertion is backward-chronological. The result of this choice is that the rejected constraints will be the ones which are temporally closer to the actual instant of execution, therefore meeting the condition of reaction urgency. In other terms, the ROBOCARE monitoring system is oriented towards synthesizing a suggestion regarding the primary cause of a violation, rather than forming one based on a distant effect of the assisted person's behavior. The constraints are chronologically ordered taking into account the values of the time point pairs they are connected to. More formally, given a set of constraints $\{c_1(t_{1,s}, t_{1,e}), c_2(t_{2,s}, t_{2,e}), \ldots, c_n(t_{n,s}, t_{n,e})\}$, where each $c_i(t_{i,s}, t_{i,e})$ connects the time points $t_{i,s}$ and $t_{i,e}$, the constraint $c_i(t_{i,s}, t_{i,e})$ chronologically precedes the constraint $c_j(t_{j,s}, t_{j,e})$, if $min(t_{i,s}, t_{i,e}) < min(t_{j,s}, t_{j,e})$.

From the scheduler knowledge to the interaction

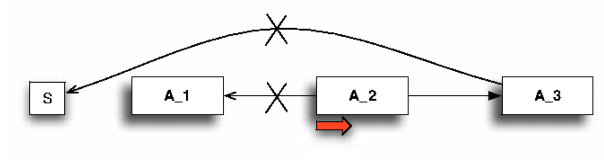
The main issue here is how to translate constraint violation information into semantically meaningful verbalizations that the user may immediately understand. We will initially present the building blocks of this semantic analysis. At this level, all semantic inference will have to be derived exclusively from the analysis of information of temporal nature; later on, we will show how temporal data can be integrated with different types of environmental information.

Each element in the violated constraints set K_t is either a *minimum* or a *maximum* constraint. *Duration* constraints are defined through both a minimum and a maximum constraint insisting between the start and end time points of an activity and representing, respectively, the minimum and the maximum duration allowed. At a basic level, the violation of each constraint is immediately given a semantic interpretation: (a) violation of the minimum constraint c_{min}^{ij} between activities A_i and A_j (where A_i is the *SOURCE* activity), directly involves the following interpretation: "*A_j is taking place too soon.*"; similarly, violation of the maximum constraint c_{max}^{ji} between activities A_j and A_i (where A_i is the *SOURCE* activity), enables the possible semantics interpretation: "*A_j is being delayed too much.*". Duration constraints undergo a slightly different analysis: in fact, a violation of a duration constraint on activity A_i might either entail the violation of the minimum or of the maximum constraints involved. In the first case, we imply the semantics: "*A_i was too brief.*"; in the second case, "*A_i is lasting too long.*".

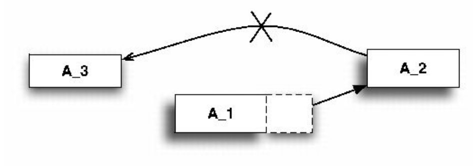
As anticipated, these represent the building blocks for higher level interpretations of the events related to constraint violations. Let us see how the integration of other kind of information might be exploited in order to improve semantic precision. Again, the meaning of the violation of the constraint c_{min}^{ij} might take a different interpretation depending on the execution status of activity A_i. In fact, in case A_i has not yet been executed, it is easy to see that the violation of c_{min}^{ij} directly implies that the activities A_i and A_j have been temporally swapped, against the caregiver's prescriptions. Therefore, a general verbalization consistent with the situation is: " *Shouldn't A_i be performed first?*". Obviously, in case A_i has been executed, a more realistic verbalization is: "*Shouldn't you wait a little longer before performing A_j?*". Regarding the maximum constraints c_{max}^{ji}, a different interpretation might be given depending if activity A_i is the *SOURCE* or not. When A_i = *SOURCE*, we are describing an *absolute* time limit which involves the start time of A_i; in the opposite case, the time constraint is *relative* to the end time of A_j. This difference might influence the verbalization related to c_{max}^{ji} violation: in the first case, "*Expedite the execution of A_i.*" might suffice; in the second case, "*Too much time is passing between A_i and A_j.*" is more appropriate.

Another source of information that is used to enhance verbalization meaningfulness is related to the causal domain theory. In other words, information about casual links between direct neighbors of the activities involved in a constraint violations can be exploited to deliver explanations.

An example is given in figure 5 (a): in this case, the delay on A_2 involves also a delay on A_3, as both activities are causally linked; as shown, two maximum constraints might be simultaneously violated, and causal link analysis can interpret the situation according to the following semantics: "*Commence A_2, as it cannot be executed too late with respect to A_1, and A_3 cannot begin later than a certain hour.*". Figure 5 (b) shows another example: A_1 is currently under execution and the causal link between A_1 and A_2 eventually gets the maximum constraint between A_2 and A_3 violated. Again, the deduced verbalization would be: "*Stop A_1, as A_2 must be immediately executed because it cannot be performed too late with respect to A_3.*".



(a)



(b)

Figure 5: Exploiting causal information for meaningful semantics deduction.

Discussion and conclusions

The ROBOCARE project has addressed one of the open challenges in AI, namely that of integrating diversified intelligent capabilities to create a proactive assistant for everyday life in a domestic environment. Our work has shown how a "silent observer" system capable of passively monitoring the execution of activities can be turned into a "proactive interactor" able to perform consistent advice-giving dialogue.

In order to contextualize interaction we mainly rely on the T-REX activity monitor that is the principal source of updated knowledge in the ROBOCARE environment. It is worth highlighting the particular use we have made of the internal knowledge of a constraint-based scheduler as well as its capability of managing changes in the environment. Specifically, we have shown how constraint violations determine *when* the system has to interact (i.e., violation that correspond to dangerous situation for the assisted person entails the need for synthesizing suggestions or alarms). The analysis and interpretation of the violation contribute to determine *how* to interact with the user (i.e., the content of the verbalization).

At present, a basic interpretation has been implemented for synthesizing the verbal interaction with the user that is based on a semantic elaboration of the information contained in the scheduler. The interpretation of constraint violations is also in line with current research in explanation that fosters the development of "user-oriented" explanations on the basis of constraint-based internal representations (Smith *et al.* 2005; Bresina & Morris 2006).

Acknowledgements. This research has been partially supported by MIUR (Italian Ministry of Education, University and Research) under project ROBOCARE: "A MultiAgent System with Intelligent Fixed and Mobile Robotic Component", L. 449/97 (http://robocare.istc.cnr.it).

References

Bahadori, S.; Cesta, A.; Iocchi, L.; Leone, G.; Nardi, D.; Pecora, F.; Rasconi, R.; and Scozzafava, L. 2004. Towards Ambient Intelligence for the Domestic Care of the Elderly. In Remagnino, P.; Foresti, G.; and Ellis, T., eds., *Ambient Intelligence: A Novel Paradigm*. Springer.

Bahadori, S.; Iocchi, L.; Leone, G. R.; Nardi, D.; and Scozzafava, L. 2005. Real-Time People Localization and Tracking Through Fixed Stereo Vision. *Lecture Notes on Artificial Intelligence* LNAI 3533:44–54.

Bresina, J., and Morris, P. 2006. Explanations and Recommendations for Temporal Inconsistencies. In *Proceedings of the 5^{th} International Workshop on Planning and Scheduling for Space, IWPSS'06*.

Cesta, A., and Pecora, F. 2006. Integrating Intelligent Systems for Elder Care in RoboCare. In Mann, W., and Helal, A., eds., *Promoting Independence for Older Persons with Disabilities*. IOS Press. 65–73.

Cesta, A.; Cortellessa, G.; Oddi, A.; Policella, N.; and Susi, A. 2001. A Constraint-Based Architecture for Flexible Support to Activity Scheduling. *Lecture Notes on Artificial Intelligence* LNAI 2175:369–381.

Cesta, A.; Bahadori, S.; Cortellessa, G.; Grisetti, G.; Giuliani, M.; Iocchi, L.; Leone, R.; Nardi, D.; Oddi, A.; Pecora, F.; Rasconi, R.; Saggese, A.; and Scopelliti, M. 2003. The robocare project: Cognitive systems for the care of the elderly. In *Proceedings of International Conference on Aging, Disability and Independence (ICADI), Washington D.C., USA*.

Cesta, A.; Cortellessa, G.; Pecora, F.; and Rasconi, R. 2005. Monitoring Domestic Activities with Scheduling Techniques. In *Proceedings of the 2^{nd}* ROBOCARE *Workshop, Rome, Italy*.

Cesta, A.; Iocchi, L.; Leone, G.; Nardi, D.; Pecora, F.; and Rasconi, R. 2006. Robotic, Sensory and Problem Solving Ingredients for the Future Home. Technical report, ROBOCARE Project RC-TR-0806-5.

Giuliani, M.; Scopelliti, M.; and Fornara, F. 2005. Elderly people at home: technological help in everyday activities. In *IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2005)*, 365–370.

Grisetti, G.; Stachniss, C.; and Burgard, W. 2005. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Proceedings of ICRA-05*, 2443–2448.

Haigh, H. Z.; Kiff, L. M.; Myers, J.; Guralnik, V.; Kirschbaum, K.; Phelps, J.; Plocher, T.; ; and Toms, D. 2003. The independent lifestyle assistant (ILSA): Lessons learned. Technical report, Technical Report ACSPO3023, Honeywell Laboratories, Minneapolis, MN, USA.

Modi, P. J.; Veloso, M. M.; Smith, S. F.; and Oh, J. 2004. CMRADAR: A Personal Assistant Agent for Calendar Management. *Lecture Notes in Computer Science* LNCS 3508:169–181.

Modi, P.; Shen, W.; Tambe, M.; and Yokoo, M. 2005. Adopt: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence* 161:149–180.

Myers, K., and Yorke-Smith, N. 2005. A Cognitive Framework for Delegation to an Assistive User Agent. In *Proceedings of AAAI 2005 Fall Symposium on Mixed-Initiative Problem Solving Assistants*.

Myers, K. 2006. CALO: Building an intelligent personal assistant. Invited Talk. *The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference (AAAI-06)*.

Pecora, F.; Rasconi, R.; Cortellessa, G.; and Cesta, A. 2006. User-Oriented Problem Abstractions in Scheduling, Customization and Reuse in Scheduling Software Architectures. *Innovations in Systems and Software Engineering* 2(1):1–16.

Pecora, F.; Modi, P.; and Scerri, P. 2006. Reasoning About and Dynamically Posting n-ary Constraints in ADOPT. In *Proceedings of 7th Int. Workshop on Distributed Constraint Reasoning (DCR-06), at AAMAS'06*.

Pentney, W.; Popescu, A.-M.; Wang, S.; Kautz, H. A.; and Philipose, M. 2006. Sensor-based understanding of daily life via large-scale use of common sense. In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference (AAAI-06)*.

Pineau, J.; Montemerlo, M.; Pollack, M.; Roy, N.; and Thrun, S. 2003. Towards Robotic Assistants in Nursing Homes: Challenges and Results. *Robotics and Autonomous Systems* 42(3–4):271–281.

Pollack, M. E.; Brown, L.; Colbry, D.; McCarthy, C. E.; Orosz, C.; Peintner, B.; Ramakrishnan, S.; and Tsamardinos, I. 2003. Autominder: an Intelligent Cognitive Orthotic System for People with Memory Impairment. *Robotics and Autonomous Systems* 44(3–4):273–282.

Pollack, M. 2005. Intelligent Technology for an Aging Population: The Use of AI to Assist Elders with Cognitive Impairment. *AI Magazine* 26(2):9–24.

Rybski, P. E.; De la Torre, F.; Patil, R.; Vallespi, C.; Veloso, M.; and Browning, B. 2004. CAMEO: Camera Assisted Meeting Event Observer. In *Proceedings of ICRA-2004*, 1634–1639.

Smith, S.; Cortellessa, G.; Hildum, D.; and Ohler, C. 2005. Using a scheduling domain ontology to compute user-oriented explanations. In Castillo, L.; Borrajo, D.; Salido, M.; and Oddi, A., eds., *Planning, Scheduling, and Constraint Satisfaction: From Theory to Practice*. IOS Press.