# Bringing Users and Planning Technology Together. Experiences in SIADEX[*]

**Juan Fdez-Olivares, Luis Castillo , Óscar García-Pérez and Francisco Palao**
Dept. of Computer Science and Artificial Intelligence
ETSI Informática, University of Granada
18071, Granada, SPAIN
siadexwww@decsai.ugr.es

**Abstract**

This work describes the solutions adopted to the problems tackled during the development of SIADEX, an intelligent planning and scheduling application where users play a central role. The system is being developed under a research contract with the Andalusian Regional Ministry of Environment (Spain) and integrates planning and scheduling techniques in a service oriented architecture devoted to support decisions on crisis intervention planning.

## Introduction

A largely pursued objective in Intelligent Planning and Scheduling is to make it widely used, bringing its complex techniques closer to end-users, by helping them to solve common, everyday problems. Very recent successful AI Planning systems have been developed (Bresina *et al.* 2005; Ruml, Do, & Fromherz 2005; Nau *et al.* 2005) in domains where highly skilled staff (in both domain knowledge and computer knowledge) use the applications to solve problems on highly technological domains. As opposite to these successful systems, this work describes the solutions adopted to the problems tackled during the development of SIADEX, an intelligent planning and scheduling application where users (which are non AI experts) play a central role, and that has been integrated into their familiar working environment in order to solve their everyday problems about decision making. The system is being developed under a research contract with the Andalusian Regional Ministry of Environment and integrates planning and scheduling techniques in a service oriented architecture devoted to support decisions on crisis intervention planning.

The current domain application of SIADEX is forest fire fighting, where experts need to obtain valuable tentative fire attack plans, composed of a sequence of timed fighting operations, before making the most suitable decisions when defining the strategy to fight a forest fire. Therefore, in order to cover this need, the system obtains timed fire attack plans, providing support to monitor their execution, and allowing experts to revise their decisions, by re-defining the strategy, in a continuous interaction with the user in every stage of the crisis episode.

In order to understand the problems to be faced while developping such a system, we will start by introducing the decision making process that currently follows the technical staff.

## Decision making in crisis intervention planning

The management of crisis episodes such as forest fire fighting usually follows a sequence of stages like that shown in Figure 1, in which several stages of situation assessment, intervention planning, plan dispatching, plan execution and monitoring are followed up to completely overcome the crisis. Due to the dynamics and the many uncertainties of real world, a revision process of the intervention plans might be carried out, triggering again a new cycle.

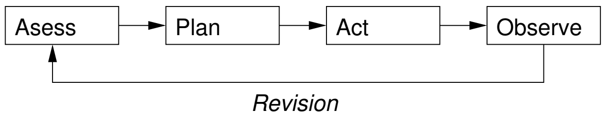


Figure 1: Life cycle of crisis management

This cycle usually relies in a chain of decision making stages from high skilled staff (in charge of assessing the situation, defining the strategic goals of the intervention, and the procedures to achieve these goals) to ground operators (in charge of carrying out operational activities that materialize procedures).

Technical staff related to SIADEX implement this lifecycle in the following steps: first, once a fire arises and its geographical initial point is identified, all the information about the fire local environment is gathered (geographical information such as orientation of the terrain, slope and access routes for fight means, forest fuel, and weather forecasts for the next hours such as temperature, humidity and wind speed and direction). This information, together with the output of a simulation of the fire spread, is used to roughly

[*]This work is being partially funded by the Andalusian Regional Ministry of the Environment, under research contract NET033957 and Spanish MCyT under project TIC2002-04146-C05-02

assess the best strategy to fight the fire in a given time horizon (usually no more than 12 hours).

The strategy consists in defining a fire scenario including both the definition of a group of operational geographical areas (called sectors) where the attack will be focused, and the statement of high-level fighting procedures to be performed at every sector. Experts also assign fighting means (such as terrestrial, aerial and human means) to these procedures, with no more information than that provided by their own personal expertise. At present, the only existing guide is provided by ”standard operating procedures” (called by technical staff *fighting protocols*), a specification at different levels of detail about which decisions have to be made and which tasks have to be carried out under these decisions. Protocols are then used to make subsequent decisions about what operational fighting activities (such as mobilization or deployment) should be carried out by assigned fighting means.

Thus, apart from the environmental information, and in order to make the best decisions on the strategy, experts must consider further crucial information (most of which is not yet available) about fighting resources such as current geographical positions, availability, fighting capacity (in order to determine the most appropriate ones regarding distance to the fire, experience, etc.), expected times of arrival to the fire, work shifts of the fighting brigades, periods of availability of aerial means, scheduled aerial water discharges, reservoir points to recharge both terrestrial and aerial means, etc. Considering the large amount of fighting means[1] and the short response time required, decision making becomes much harder and, in order to make an anticipated analysis of the consequences of their decisions, experts need to have a completely detailed *attack plan* composed of a sequence of timed fighting activities of the assigned means in the strategy.

At present, technical staff only have available some tools to record and manage the overwhelming amount of information gathered from the environment[2], but no tool is used to allow experts to evaluate their decisions about the best strategy to be defined by offering valuable tentative attack plans. Furthermore, due to the dynamics and uncertainty of the environment where the operations are executed, strategies and subsequent attack plans are revised every certain interval of time according to the evolution of the fire. Therefore, it becomes also necessary to allow experts to revise their decisions once a given plan is accepted, according to the life-cycle of Figure 1.

SIADEX is being developed in order to cover these needs and, therefore, it has been designed as a planning and scheduling system that obtains attack plans, providing support to monitor their execution, and allowing to revise experts' decisions by re-defining the strategy (if needed) by a continuous interaction with the user in every stage of the crisis episode. The development of such a system introduces some challenges to current state-of-art planning techniques that will be summarized next.

[1]At present, means devoted in Andalusia to forest fire fighting amount about 300 squads (grouping about 3000 people), 200 terrestrial vehicles (devoted either to transport or fire suppression) and 30 aicrafts

[2]Most of them based on ArcView (ESRI 2005), a GIS application intensively used by technical staff

## Problems to be faced

With respect to Intelligent Planning and Scheduling techniques, and regarding the above explained decision making life-cycle, temporal reasoning becomes one of the key aspects, since the system has to obtain temporal plans as a sequence of timed fighting activities, taking into account that top-level goals are constrained by deadlines (recall that experts define high-level procedures within a given time window). Additionally, in order to be suitable for validation and execution by technical staff, plans must follow standard fighting protocols, that is, the planning process should obtain plans by making decisions as if they were made by human experts. This has led us to develop a domain independent HTN planning system able to accept a hierarchical domain description based on the fighting protocols and perform a reasoning process that decomposes top-level tasks into subtasks, following the guidelines encoded in the domain. The novelty of our HTN planner, with respect to Intelligent Planning and Scheduling techniques, is two-fold: on the one hand, it accepts HTN domains based on a hierarchical extension of PDDL 2.2 level 3 (Edelkamp & Hoffmann 2004) and, on the other hand, it integrates temporal reasoning on tasks at every level of detail in the task hierarchy, using Temporal Constraint Networks as the underlying temporal representation formalism.

Furthermore, scalability and time processing requirements have also to be considered: fire attack plans range from one hundred fighting operations performed by about ten fighting means (in small extinction episodes), to several hundred actions and tens (or few hundred) of means involved. As the system has to provide several tentative plans, the planning process should allow to obtain several plans in a short time in order to be evaluated by experts, that will be able to define the most suitable strategy to be followed. As it will be shown, maintaining the causal structure in the plan allows to obtain an efficient HTN temporal planner that outperforms SHOP2 (Nau *et al.* 2003), an HTN planner widely used in several practical applications (Nau *et al.* 2005).

Considering the dynamics of the domain as well as the revision requirements, it is necessary to adopt a solution to the management of uncertainty in the execution of the plan. In order to provide support to execution failures we have designed a plan execution and monitoring process, integrated with the output of the deterministic planner.

With respect to the integration of AI Planning technology in a real application, the planning life-cycle of SIADEX must be seamlessly integrated and coupled with the current workflow and decision making life-cycle of technical staff, a strong practical requirement that decisively affects the success of the application. This requirement has led us to design an architecture that integrates the planning process within the familiar working environment of technical staff, thus enhancing the role of end-users. Furthermore, although users are not expected to have any background knowledge on AI, it is required that they could modify the knowledge of the planner and, therefore, the architecture must offer solutions to this issue as well as to the following ones: *flexible knowledge management an integration*, since the system must represent and provide access to the large amount of data coming from heterogeneous sources of information needed to make decisions, *support for ubiquitous access of end-users*, since the system is operated in a hostile and distributed environment and most of the inputs come from (and most of the outputs are directed to) end-users located at different places, and *integration with legacy software*, since the planning process income as well as the outcome must be redirected to legacy software so that end users may painlessly understand, process and deliver activity plans.

In the following we will describe how the aforementioned problems have been solved: first, the next section describes the architecture of the system, then we explain the different stages as well as the technical details of the planning process life-cycle.

## The architecture

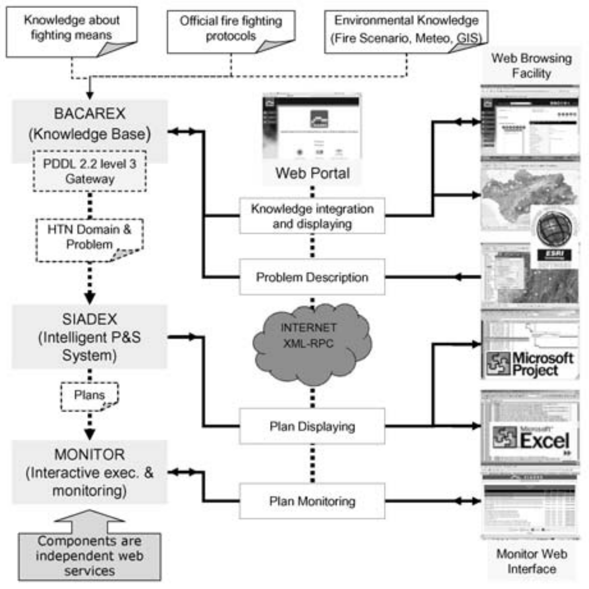

Knowledge about fighting means
Official fire fighting protocols
Environmental Knowledge (Fire Scenario, Meteo, GIS)
Web Browsing Facility
BACAREX (Knowledge Base)
PDDL 2.2 level 3 Gateway
HTN Domain & Problem
Web Portal
Knowledge integration and displaying
Problem Description
INTERNET XML-RPC
SIADEX (Intelligent P&S System)
Plans
Plan Displaying
MONITOR (Interactive exec. & monitoring)
Plan Monitoring
Microsoft Project
Excel
Components are independent web services
Monitor Web Interface

Figure 2: The architecture of SIADEX

SIADEX has been developed as a general, service-oriented architecture (Figure 2) to support decisions on crisis management episodes such as forest fire fighting. Its main components, implemented as web services, are the knowledge base (named BACAREX), that stores in Protégé (Protégé 2006) all the knowledge that would be useful for the planning engine, and the planning module (named SIADEX), the core of the architecture in charge of building fire fighting plans.

The planning algorithm and its knowledge representation are built as two independent servers (the planning server and the ontology server respectively) that provide services under a TCP/IP connection (OS independent) and can be accessed from any device with internet connectivity (a desktop computer, a laptop or a PDA). The architecture also includes a plan execution and monitoring web service in charge of tracking all the changes produced by the execution of the actions in a fire fighting plan. All the communication between the modules of the architecture is performed using the XML-RPC Protocol[3].

The system interface is composed of several front-ends, plugged into the legacy software that conforms the familiar working environment of technical staff, devoted to support interaction in every stage of the decision making process. Considering the staff's decision making life-cycle described in the first sections, the process followed by experts when making decisions (supported by SIADEX) includes the following steps:

**Problem description** In this step the fire fighting scenario is introduced by the technical staff through a user-friendly interface implemented as a software plug-in of the ArcView GIS (ESRI 2005).

**Knowledge integration** The scenario introduced is stored in BACAREX and integrated with additional knowledge (also recorded in BACAREX) needed by the planning process, such as standard protocols, fighting resources and other environmental information. Therefore, knowledge about resources and fire scenario share the same representation, and all this information is visible to other users by means of a web browsing facility.

**Requesting a plan** Once the scenario is integrated into the knowledge base, and by user request, the planning engine is invoked and a plan is obtained. The planning engine is not able to read the domain and the problem stored in Protégé, therefore a PDDL Gateway has been implemented that translates problems and domain into an HTN extension of PDDL 2.2 level 3.

**Displaying the plan** The obtained plan is delivered in XML format and it may be displayed in a number of "user-friendly" alternatives like Microsoft Excel, in the form of a chronogram, or Microsoft Project in the form of a Gantt chart.

**Plan execution and monitoring** The plan may be launched for execution, distributed amongst all the technical staff with some responsibility in the fire fighting episode, and concurrently monitored.

Each one of these steps will be detailed in the following sections.

## Problem description

Taking into account the description of the decision making cycle performed by fighting experts, this step starts after the environmental information has been gathered and the fire simulation is performed. This information is currently

[3]XML-RPC: XML Remote Procedure Call http://www.xmlrpc.com

recorded and managed using ArcView, a GIS application intensively used by technical staff. Therefore, we have developed a user-friendly interface implemented as a plug-in of Arcview, devoted to visually introduce the information about the fire scenario, thus maintaining the integration with legacy software. The fire director, after the simulation, defines through this interface in an effortless way (using basic drawing tools over a geographical map) the operational units of the forest fire: the sectors. These are relevant areas of the environment where the attack will be focused (see Figure 3). Every fire scenario may have several sectors and every sector may be composed into operational targets like fire lines, control lines and spraying areas.

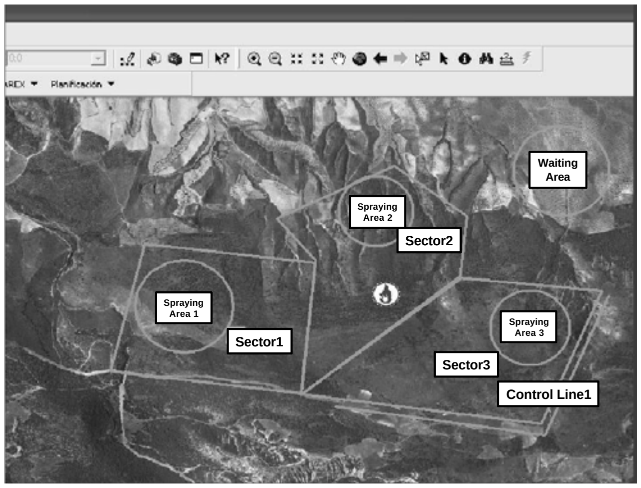


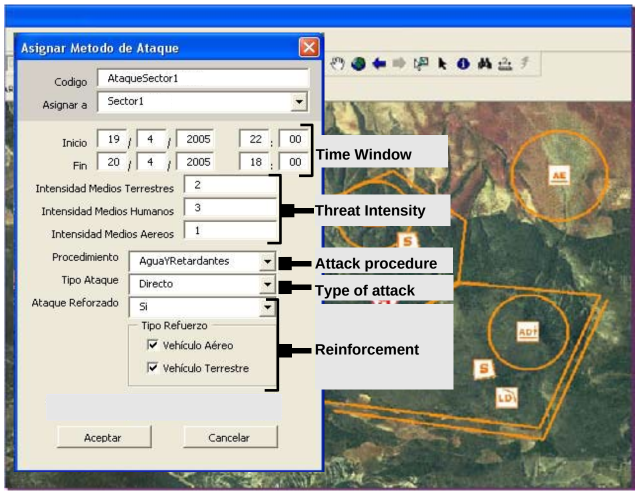
Figure 3: A simulated fire scenario with three sectors. Every sector contains a spraying area, Sector3 contains a control line. A waiting area, close to the fire, is also shown. Sectors, spraying areas and the control line are considered operational targets, the waiting area is devoted to logistics operations.

Once the sectorization has been defined, the fire director establishes the strategy to be carried out in the scenario. The strategy is defined as a set of *top-level tasks* to be accomplished at every operational target, a *time window* of activity associated to every task, and the *amount of resources* to be assigned to such tasks (see Figure 4). At present, and following the guidelines of technical staff, the amount of resources to be assigned is defined as three integer values: the number of human resources (squads), the number of aerial vehicles and the number of ground vehicles. These values are interpreted as a measure of the fire threat intensity and the planner is in charge of determining the instances of resources to be concretely assigned to every task.

The information introduced at this step is sent to the ontology server to integrate the fire scenario with the current contents in the knowledge base.

## Integrating knowledge

The ontology server performs all the processes related to knowledge integration. It contains an ontology that has been


Asignar Metodo de Ataque

| | |
|---|---|
| Codigo | AtaqueSector1 |
| Asignar a | Sector1 |
| Inicio | 19 / 4 / 2005 22 : 00 |
| Fin | 20 / 4 / 2005 18 : 00 |
| Intensidad Medios Terrestres | 2 |
| Intensidad Medios Humanos | 3 |
| Intensidad Medios Aereos | 1 |
| Procedimiento | AguaYRetardantes |
| Tipo Ataque | Directo |
| Ataque Reforzado | Si |

Tipo Refuerzo
☑ Vehículo Aéreo
☑ Vehículo Terrestre

Figure 4: A description of the strategy to be carried out in Sector1 described as a top-level task including the following information: **perform a direct attack**, on the **operational target** Sector1, using a combination of procedures with water and fire retardants, and reinforced by aerial and terrestrial vehicles; in order to mitigate the **fire with an intensity** of 2 terrestrial vehicles, 3 squads and 1 aircraft; within an **activity window** of 20 hours (start date: 19/04/2005 22:00, end date: 20/04/05 18:00).

designed with Protégé 2000 [4] that supports the integration and management of several categories of knowledge, including fighting resources, geographical information, fighting protocols and legal information. It also provides several knowledge interfaces depending on requester's needs. Details about the knowledge stored and the interfaces provided are explained in the following sections.

## Categories of knowledge

The most important part of the ontology is devoted to fire fighting means which are modeled either as material resources or human resources (Figure 5). Material resources may be facilities like operation bases, airports, etc. and they represent static objects since most of their attributes will remain unchanged although they are very important for the planning process like, for example, the geographic position of an airport, the availability of refueling and water recharging facilities, etc. The remaining resources, either vehicles or human workers, are very dynamic since many of their attributes (also called *operational slots*) change during the execution of plans, for example their geographic position, their state of availability, the work they are carrying out, etc.

Temporal knowledge is also represented in the ontology. The main source of such category of knowledge comes, on the one hand, from legal issues (such as the maximum duration of squads' shifts, *Legal Shifts* in Figure 5), or periods of availability for contracted aerial resources (*Contract Availability* in Figure 5). On the other hand, as previously

[4]At present it includes more than 130 classes and 2000 instances only for planning objects and excluding the representation of activities

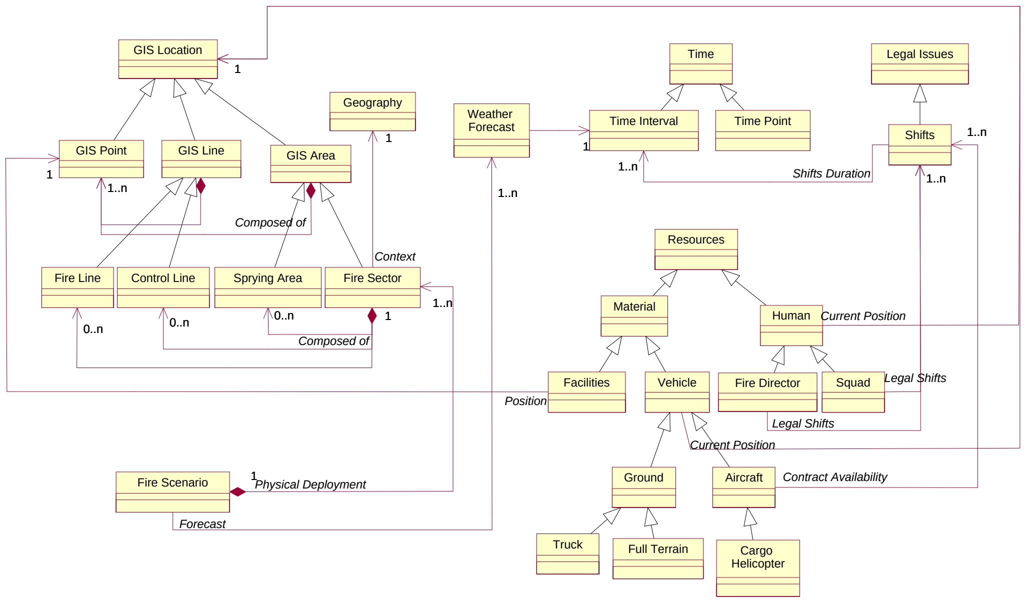


Figure 5: A UML representation of a simplified part of the ontology of SIADEX

seen, the strategy defined in the fire scenario contains top-level tasks to be accomplished in a given operational area, between a *time window* described as start and end dates that constraint all the low level fire-fighting operations to be carried out as part of the strategy. Therefore, this temporal information is also represented.

Geographical and environmental information, such as the sectorization of a fire scenario or weather forecasts, is also stored since decisions made by experts following the fighting protocols depend on this crucial information.

Finally, another part of the knowledge base is devoted to represent knowledge about operational tasks and fighting protocols. These standard operating procedures are directly represented as a hierarchy of tasks following an HTN domain description, and details an examples about this representation are shown in a later section.

## Knowledge interfaces

The ontology server also provides different services depending on requester's needs:

- From the user point of view it provides a common place to easily query, browse and update the information, as it integrates in a single model heterogeneous knowledge coming from different sources such as GIS or external resource management data bases (extracted by plug-ins and mappers). It also provides both offline and on-line access facilities. Offline access is performed by the standard Protégé framework, so that the development team may access the knowledge and carry out maintenance and validity checking operations with full operability. On the other hand, on-line access is done through the Protégé API on which an XML-based web service has been built. This on-line access is devoted to staff users that do not have skills on knowledge representation for planning but may easily access the knowledge in a web browsing fashion by means of a hierarchy of objects and activities close to their understanding of the problem.
- From the point of view of the plan execution and monitoring service, the ontology is used to maintain up-to-date the situation of means and resources as fighting operations are executed during the plan progress.
- From the planning server point of view the ontology is used to obtain the problem and the domain. As seen above the domain is directly represented in an HTN formalism. With respect to the problem, it is composed of an initial state and a top-level goal. The former is represented in the ontology as the values of the operational slots of every fighting resource instance, as well as the geographical deployment of the fire. The latter is obtained by querying the information about the top-level tasks introduced in the fighting strategy.

This last knowledge interface is indeed a process that carries out the translation of the problem defined by the user from the ontology representation to the representation used by the planner. This process is triggered when the user requests the planning service, the next step after storing the information of the scenario.

## Requesting planning services

When requested by the user, the planning server gets the domain and problem representation sent by the ontology server and triggers the planning process on these inputs. The planning algorithm implemented in this server is based on an HTN algorithm that integrates temporal reasoning using Simple Temporal Networks as the underlying temporal representation mechanism. This section describes the most relevant details of the planning process and the domain and problems representation used in SIADEX, explaining the reasons that led us to adopt this solution. Finally, the translation process that extracts the problem from the ontology to the planner is outlined.

### HTN and temporal reasoning

As previously noted, in order to be validated by experts, plans must follow standard operating procedures and, therefore, fighting protocols should be encoded in the domain description. Standard PDDL actions as well as PDDL 2.2 level 3 durative actions (Edelkamp & Hoffmann 2004) provide enough expressivity to describe operational, lowest-level fighting activities (mainly due to the use of fluents to support resource reasoning, and the use of temporal constraints such as action duration as shown in Figure 6), but durative-actions are not expressive enough to represent high-level processes such as those specified in fighting protocols. Protocols are a process specification at different levels of detail that also includes constraints on the order of tasks to be carried out. This kind of specification is best suited by the HTN formalism (Nau *et al.* 2003; Erol, Hendler, & Nau 1994) , where planning domains are designed in terms of a hierarchy of tasks representing compound and atomic activities (called *compound tasks* and *primitive tasks or operators* respectively). In such hierarchical domains, it is possible to describe how every compound task may be decomposed into different subtasks and the order that subtasks must follow, by using different *methods*.

```
(:durative-action move
  :Parameters (?g - Group ?v - Vehicle ?p1 ?p2 - GIS_Location)
  :duration (= ?duration (/ (distance ?p1 ?p2) (cruise ?v)))
  :condition (and (current_position ?v ?p1)
                  (current_position ?g ?p1))
  :Effect (and (not (current_position ?v ?p1))
               (not (current_position ?g ?p1)
               (current_position ?v ?p2)
               (current_position ?g ?p2)
               (decrease (current_autonomy ?v) ?duration))
```

Figure 6: A PDDL durative action representing the movement of a human group transported by a vehicle. Temporal (duration of the transport) and resource constraints (the autonomy of the vehicle decreases) have proved to be expressive enough in this practical application.

Motivated by the lack of expressivity found in PDDL durative-actions, we have developed an HTN extension of the PDDL standard in such a way that hierarchical domains in SIADEX encode primitive operators as PDDL 2.2 level 3 durative actions (Edelkamp & Hoffmann 2004) . In addition, basic issues of compound tasks included in our representation are inspired by the domain description language of SHOP2 (Nau *et al.* 2003), where the methods used to decompose tasks into subtasks include a precondition that must be satisfied by the world state in order for the method to be applicable by the planner. Therefore, the basic algorithm of our planner follows the HTN paradigm of SHOP: it is a state-based forward HTN planning algorithm that uses an ordered task decomposition search-control strategy. Thus tasks are decomposed according to the order in which they will be executed and methods are selected considering the current state.

But this is only the basics of SIADEX, and further extensions has been introduced to the basic representation for tasks, as well as in the planning algorithm, in order to overcome some recognized weaknesses of state-based forward planners (either HTN-based, like SHOP, or not) regarding practical applications. Usually this category of planners return plans as a totally ordered sequence of actions but, in many real world applications such as forest fire fighting, activities may be executed concurrently, and a total order of activities is not very appropriate for practical reasons. The domain description language of SIADEX and the planning algorithm support to explicitly represent and manage time and concurrency in both compound and primitive tasks (increasing the expressiveness of the planner), thanks to the handling of metric time over a Simple Temporal Network (See (Castillo *et al.* 2006) for more details).

This is a novel representation that provides a great expressivity power in order to support two fundamental requirements of experts: on the one hand, the representation of deadlines over top-level goals, as required in the top-level tasks formulation shown in the problem description in Figure 4, and on the other hand, it must also be possible to define complex control structures required by the representation of protocols such as sequencing, splitting and synchronization of high-level processes.

### Deadline goals

As shown in the section devoted to present the problem description process, the earliest stage of decision making requires to post one or more metric temporal constraints over both the start and the end of a top-level task. Furthermore, the underlying aim of a user is that its component subtasks also inherit them. SIADEX allows to post deadline constraints, and to inherit them in the case of subtasks of a given task, on either the start or the end (or both) time-points of any task either primitive or not. Any task has two special variables associated to it: `?start` and `?end` that represent its start and end time-points, and some constraints (basically `<=, =, >=`) may be posted to them. In order to do that, when describing a decomposition method, any subtask may be preceded by a logical expression that defines the desired deadline (Figure 7).

Deadline goals are easily encoded in the STN of a temporal plan as absolute constraints with respect to the absolute start point of the STN. They may also appear in the top-level goal and they are very useful for defining time windows of activity like the one shown in the example of Figure 4.

```
(define (problem FireOne)
(:domain encoded-protocols)
(= :time-start "19/4/2005 17:0:0")
(= :time-unit :minutes)

(:objects
  ...
  (is-part-of Fire Sector1)
  (is-part-of Fire Sector2)
  ... )

(:tasks-goal
:tasks ((and (>= ?start "19/4/2005 22:0:0")
             (= ?end "20/4/2005 13:0:0"))
         (attack Fire Sector1))
))
```

Figure 7: An example of a problem described following the HTN extension of PDDL where some information about the start time of the fire is included, as well as the top-level goal, that is formulated in terms of an instance of a top-level task `(attack ?f - GIS_Fire ?s - GIS_Sector)` with a deadline. According to the example of Figure 4 the attack to Sector1 starts 5 hours after the beginning of the fire and must end 20 hours later.

```
(:TASK attack
 :PARAMETERS (?f - GIS_fire ?s - GIS_Sector)
 (:METHOD not-reinforced
  :PRECONDITION (not (reinforced-attack ?s))
  :TASKS ((Select_Means ?f ?s)
          (Mobilize_Deploy_All_Groups ?f ?s)))
 (:METHOD reinforced
  :PRECONDITION (reinforced-attack ?s)
  :TASKS ((Select_Means ?f ?s)
          [ (Mobilize_Deploy_All_Groups ?f ?s)
            (Mobilize_Deploy_All_Aerial ?f ?s)
            (Mobilize_Deploy_All_Terrestrial ?f ?s)])))
```

Figure 8: An HTN task implementing a standard operating procedure to attack a sector ?s in a forest fire ?f. Doctrine establishes that, in the case of not reinforced attack, first select appropriate human means, then mobilize and deploy them. In case of reinforced attack, first select means, then deploy every human, aerial and terrestrial mean selected.

## Ordering constraints and synchronization between tasks

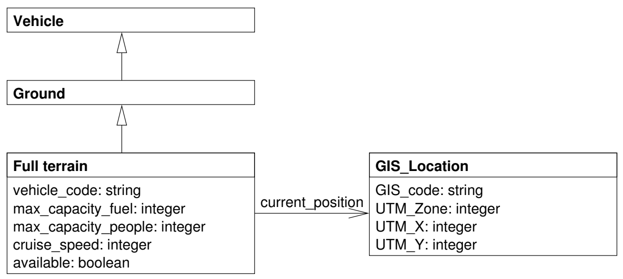
A natural description need for standard operating procedures such as fighting protocols used by technical staff is given by the use of different control structures for sequencing, splitting and synchronizing processes, included in almost all languages devoted to process specification such as workflow languages (OWL-S 2003). As shown in Figure 8, domain descriptions in SIADEX also allow to specify such control structures between the subtasks of a method. Tasks might be either sequenced, and then their signature appears between parentheses `(T1,T2)` , or splitted, appearing between braces `[T1,T2]`. For example, the second method of Figure 8 specifies that human, aerial and terrestrial means should be concurrently (if possible) mobilized and deployed after being selected and assigned to a sector. However, these order structures represent qualitative order constrains that are necessary but not sufficient in order to allow practical execution of concurrent tasks since, in practical applications, synchronization between tasks is needed. This can be done in SIADEX due to the quantitative, temporal constraints included in the STN of a temporal plan that are not only extracted from the order relations between tasks, but also from the causal structure allowing to manage timed causal links (See for more details (Castillo *et al.* 2006)). For example, as subtasks of the top-level task of Figure 8 inherit the temporal constraints posted in the problem (see in Figure 7 ), the method *reinforced* shown in Figure 8 represents a split-join structure as all the splitted subtasks must end at the same time that the top-level task.

Once the basic issues about the planner and the domain representation has been explained, next we will the describe how the problem, initially stored in the ontology server, is translated into a problem representation understandable by the planner.

## Getting the problem

The initial state of the problems managed by SIADEX maintains basically the same representation that in the PDDL standard, and it is expressed as a conjunction of literals that represent the set of facts that are known to be true. It may be obtained by querying the ontology about the main slots of the instances of classes like facilities, human resources, vehicles, water points, etc. This process is crucial when trying to maintain the planning knowledge as transparent as possible to end-users. In order to do that, a PDDL gateway has been designed so that, at the beginning of a planning episode and when requested by the user, this gateway iterates over the whole ontology and translates the content of the main instances into PDDL in a process whose main features are the following ones:

- The classes hierarchy of the ontology is translated as a hierarchy of types in the PDDL domain description. As illustrated above, the parameters of compound and primitive tasks are typed regarding this translated hierarchy of types.
- Predicates are easily translated since every operational slot of the ontology (that is, whose content might be relevant for the planning process) has a special template on how to translate the content of the slot into a PDDL literal. Binary slots of the form `Instance.slot=value` are easily translated into PDDL literals of the form `(slot instance value)`. Non binary predicates or slots that represent references to other instances have a similar translation process, and even some of the slots may produce multiple literals (see Figure 9). .
- Slots that contain numerical values that may change during a planning episode, mainly numerical resources like the current autonomy of a vehicle, are declared like fluents in the PDDL domain, so that the use of arithmetic operations and functions are permitted over them and they allow the planner to reason about the use of resources.
- Slots representing temporal knowledge (like shifts of workers) or environmental information (like weather forecasts of day/night events) are translated into PDDL 2.2 timed initial literals.



```
(current_position vehicle_code GIS_code) and
(coordinates GIS_code UTM_Zone UTM_X UTM_Y)
```

Figure 9: Translation of some references in the ontology

The top-level goal is also extracted from the ontology using the knowledge stored in the initial step of problem description. As shown in a previous example of this section, and considering the goal formulation of Figure 7, the top-level goal is composed of the top-level tasks defined by the user, and the time window defined by the user is translated as temporal constraints on this task. In addition, considering that the geographical deployment of the fire episode is stored as instances into the ontology, and that the slots of the instances are also considered as operational, the information about the sectorization is also translated as part of the initial state.

In addition, as also shown in Figure 7, every top-level task defined as part of the strategy contains information about the numeric evaluation of the power of resources that should be assigned to it and that is fixed by hand by the fire fighting director. From these values, the planner may pre-select several combinations of resources, that are submitted to the fight director who finally selects one of them. Later, the planner, and following standard protocols, will select a specific set of instances of fighting resources that fits within the selected combination.

Finally, thanks to the expressive power of using Temporal Constraint Networks as the underlying formalism to represent temporal knowledge, the planner is able to obtain temporal attack plans, that is plans whose actions are executed in a time line that is flexibly constrained to the time windows established in the definition of the goal. These plans are also suitable for being evaluated by users, as they are generated by following standard procedures encoded in the domain description, that has been proved to be expressive enough to describe complex ordering and synchronization mechanisms between tasks. The plans so obtained are served by the planning server in order to be displayed, analyzed and monitored, as discussed in the following section.

## Plan diplaying and monitoring

Plans obtained may be displayed in a number of "user-friendly" alternatives like Microsoft Excel, in the form of a chronogram, or Microsoft Project in the form of a Gantt chart (see Figure 10). These interfaces are used by experts in order to evaluate the suitability of the decisions made when defining the strategy in the problem description stage. If a plan is evaluated and accepted by the fire director, it is dispatched for execution, and distributed amongst all the technical staff with some responsibility in the fire fighting episode, and concurrently monitored.

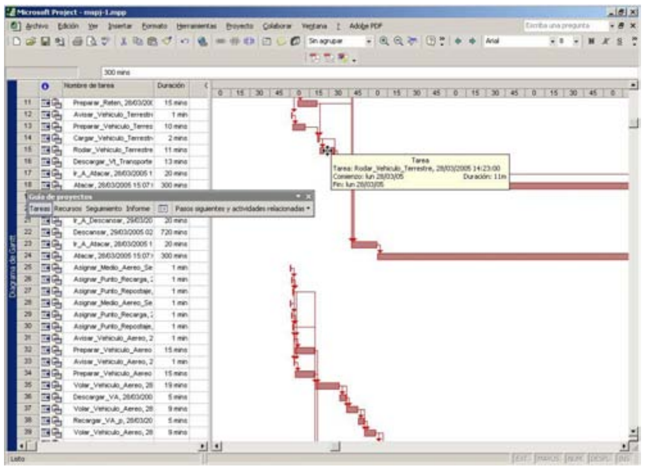


Figure 10: Gantt chart output

Plan execution and monitoring are also implemented as a web service and include a web interface. Through this interface experts might supervise in real time the execution of the plan, and track all the changes produced by the execution of the actions in the plan, the time they take to execute and it updates the ontology accordingly so that these changes are publicly available for any query at any time.

The monitoring algorithm (de la Asunción *et al.* 2004) is a real time algorithm that follows the execution of the temporal plan at the highest level of detail since temporal plans, as they are represented on top of a STN, still represent flexibly (by means of an interval of time) the time at which every effect of every action is achieved. As explained above it is possible to check that everything executes as predicted, otherwise, it is possible to detect and, in some cases repair, some of the problems without the need of starting a new planning episode[5]. The type of problems and their possible solutions are the following ones:*local delays* (they only affect to an isolated branch of the plan, and thus, a new local reschedule may be found only for that branch), *global delays* (they might affect all the remaining actions of the plan, and a whole new reschedule might be needed), and *infeasible delays* (no reschedule is possible, and a new planning episode considered). Finally, experts may also detect some perturbation in the execution of a plan and, in this case, a new cycle of decision making supported by SIADEX might be started.

## A short note on efficiency

Just to give an idea of the performance of SIADEX, we include here a short comparison in the ZENO domain of the International Planning Competition 2002 (Long & Fox 2003). Figure 11 shows the CPU times of SIADEX

[5]Clearly, the user may also interrupt the execution of the plan at any moment.

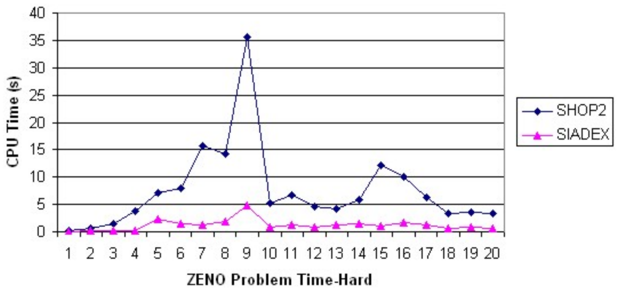


Figure 11: Performance of SHOP2 and SIADEX in zenotravel time problems. Y-Axis represents computation time in seconds and X-Axis represents the set of test problems.

and SHOP2 (Nau *et al.* 2003) running on the same machine and operating system, solving all the hard instances (time+numeric) of this domain, and it may be seen that SIADEX outperforms SHOP2 quite clearly.

Additionally, two other experiments are shown. Figures 12.a) and b) show the performance of SIADEX solving two real problems, named INFOCA-1 and INFOCA-2. The X-Axis shows the number of actions in the resulting temporal plan and the Y-Axis shows CPU time in seconds. In order to give a measure about the kind of real problems we are talking about it must be said that the size of the initial state is about 14000 literals, translated from about more than 130 classes and more than 2000 instances. At present the domain contains about 107 tasks, 121 methods, and 55 durative-actions [6].

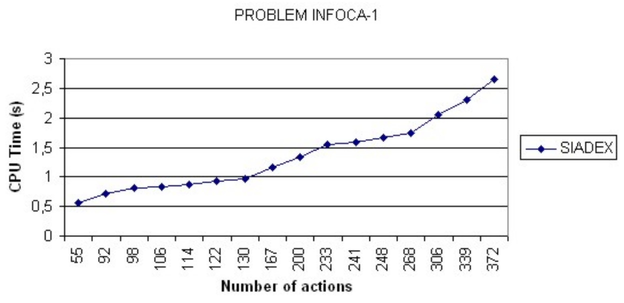
## Related work

Regarding general crisis situations, SIPE (Bienkowski 1995) and the I-X architecture (Siebra 2005) have been used, respectively, for obtaining plans of attack for oil spill threats in the sea and disaster relief domains. Although these architectures are very similar to SIADEX, their main drawback is that users are required to have a deep knowledge of planning techniques either to interact or to provide knowledge for the system.

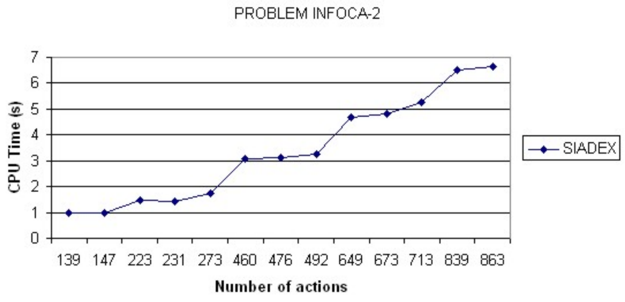
Another system used in civil crises was that described in (Biundo & Schattenberg 2001) where hybrid hierarchical techniques were developed to obtain plans of response for floods in Germany, but as far as authors know, it does not have monitoring or temporal reasoning integrated with HTN, what impedes severely its use in real environments.

Specifically in the field of forest fire fighting there are several approaches in the literature: PHOENIX (Cohen *et al.* 1989) (1989-1993) , CHARADE (Avesani, Perini, & Ricci 2000) (1992-1995), CARICA (Avesani, Perinni, & Ricci 1997) (1995-1997). However, they have failed in their application as assistants to real fire fighting scenarios. The reasons for these unsuccessful approaches are: (1) They have neglected the development of appropriated knowledge integration techniques, considering user interaction at edition level, not allowing users to affect the knowledge of the planner; (2)The lack of deliberative planning techniques able to flexibly generate new plans for a situation without the requirement of having a predefined skeleton or general plan previously stored; (3) None of these systems integrates HTN reasoning with temporal reasoning, what provides a great expressivity in order to encode fighting protocols; (4) Absence of plan execution and monitoring techniques to allow flexibility and responsiveness in the execution of the plan.

[6]Please consult http://siadex.ugr.es for more details on this real application



(a)



(b)

Figure 12: Performance of SIADEX in real problems extracted from past crisis episodes.

## Conclusions and Lessons Learned

In this work we have described the solutions adopted to the problems tackled during the development of SIADEX, an intelligent planning and scheduling application where users play a central role. SIADEX obtains temporal fire attack plans in a very time-efficient planning process, providing support to monitor their execution, and allowing to revise experts decisions by re-defining the strategy (if needed) by a continuous interaction with the user in every stage of the crisis episode.

The main lesson learned from the development of SIADEX is that, in order to be widely used, planning technology has to be developed assigning to users a central role, trying to bring the technology near to the user. This means that in order to provide solutions to the real, common problems of non AI expert users planning and scheduling processes must be part of a wider, overall architecture integrated with legacy software, such as the one presented here, allowing an easy to learn interaction process in every stage of the planning life-cycle. As users in forest fire fighting usually work in different kinds of (even hostile) environments, during the development of the architecture of SIADEX we have had to solve two additional problems: how to integrate a large amount of heterogeneous knowledge coming from different sources, and how to provide ubiquitous and concurrent access to users.

But the central role of the users not only affects to outside issues of planning and scheduling. This work also shows that, as plans must be finally evaluated by experts, they must follow standard fighting protocols and, thus, the planning process should obtain plans by making decisions as if they were made by human experts. This has led us to develop new techniques,extending current state-of-art planning techniques. The novelty of the planner we have developed, with respect to Intelligent Planning and Scheduling techniques, is two-fold: on the one hand, it accepts HTN domains based on a hierarchical extension of the PDDL standard and, on the other hand it integrates temporal reasoning on tasks at every level of detail in the task hierarchy, using Temporal Constraint Networks as the underlying temporal representation formalism. This provides support for two fundamental requirements of experts: on the one hand, the representation of deadline top-level goals, as required in the top-level tasks formulation when defining the fire scenario; on the other hand, it must also be possible to define complex control structures required by the representation of protocols such as sequencing, splitting and synchronization of high-level processes.

Furthermore, due to the response times demanded by users in their decision making cycle, time processing requirements have been also considered, and we have shown that our planner is extremely efficient regarding the type of planning and scheduling real problems to be solved (hundred of actions managing tens, even hundred of fighting resources).

This project is presently at a 75% of development and there are still pending issues, the most important is, the definition of a plan repairing and replanning process based on mixed initiative techniques. At present we are centering our efforts in this ongoing work.

## References

Avesani, P.; Perini, A.; and Ricci, F. 2000. Interactive case-based planning for forest fire management. *Applied Intelligence* 13(1):41–57.

Avesani, P.; Perinni, A.; and Ricci, F. 1997. CBET: a case-based exploration tool. In *Fifth Congress of the Italian Association for A.I.*

Bienkowski, M. 1995. Demonstrating the Operational Feasibility of New Technologies: The ARPI IFDs. *IEEE Expert* 10(1):27–33.

Biundo, S., and Schattenberg, B. 2001. From abstract crisis to concrete relief - a preliminary report on combining state abstraction and htn planning. In *6th European Conference on Planning (ECP-01).*

Bresina, J. L.; Jonsson, A. K.; Morris, P.; and Rajan, K. 2005. Activity planning for the mars exploration rovers. In *Proceedings of the ICAPS05*, 40–49.

Castillo, L.; Fernández-Olivares, J.; García-Pérez, O.; and Palao, F. 2006. Efficiently handling temporal knowledge in an HTN planner. In *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS06).*

Cohen, P.; Greenberg, M.; Hart, D.; and Howe, A. 1989. Trial by fire: understanding the design requirements for agents in complex environments. *AI Magazine* 10(3):32–48.

de la Asunción, M.; Castillo, L.; Fdez-Olivares, J.; García-Pérez, O.; González, A.; and Palao, F. 2004. Handling fuzzy temporal constraints in a planning framework. To appear in the special issue of Annals of Operations Research on Personnel Scheduling and Planning.

Edelkamp, S., and Hoffmann, J. 2004. The language for the 2004 international planning competition. http://ls5-www.cs.uni-dortmund.de/ edelkamp/ipc-4/pddl.html.

Erol, K.; Hendler, J.; and Nau, D. 1994. UMCP: A sound and complete procedure for hierarchical task-network planning. In *AIPS-94*.

ESRI. 2005. Arc view gis software http://www.esri.com.

Long, D., and Fox, M. 2003. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research* 20:1–59.

Nau, D.; Au, T.; Ilghami, O.; Kuter, U.; Murdock, J. W.; Wu, D.; and Yaman, F. 2003. SHOP2: An HTN Planning System. *Journal of Artificial Intelligence Research* 20:379–404.

Nau, D.; AU, T.; Ilghami, O.; Kuter, U.; Wu, D.; and Yaman, F. 2005. Applications of shop and shop2. *IEEE Inteligent Systems* 34–41.

OWL-S, T. 2003. Owl-s:semantic markup for web services. technical white paper (owl-s version 1.0). Technical report, http://www.daml.org/services/owl-s/1.0/owl-s.html.

Protégé. 2006. Stanford medical informatics. http://protege.stanford.edu/.

Ruml, W.; Do, M. B.; and Fromherz, M. 2005. On-line planning and scheduling for high-speed manufacturing. In *Proceedings of the ICAPS05*, 2–11.

Siebra, C. 2005. Planning requirements for hierarchical coalitions in disaster relief domains. *Expert Update* 8(1):20–24.