# A Prototype of an Active Decision Support System for Automatic Planning Support in Emergency Management

Claudio Balducelli, Giovanni Di Costanzo, Adam M. Gadomski
ENEA, CR Casaccia, 00060 Rome, Italy
balducelli_c@casaccia.enea.it, giovanni.dicostanzo@casaccia.enea.it, gadomski_a@casaccia.enea.it

**Keywords:** Decision Support System, Emergency Management, Intelligent Agent, Reinforcement Learning, Automatic Planning.

**Abstract**

The proposed prototype is a product of our group's lengthy research investigating more active and helpful tools to improve emergency response from decision-makers during crisis situations. The developed tool allows the user to configure different accidental scenarios, then seek support to efficiently plan the deployment of resources and to improve the efficacy of scheduled emergency actions. The tool's objective is to collaborate with emergency managers to identify and validate the best intervention plans, and to respond to the dynamics and pressures of a real emergency. Although the adopted case-study refers to an oil port domain, a model that has been imported from a previous EU project, the tool is easily reconfigurable for different domains and other emergency managerial roles. Several new technologies as *intelligent agents, reinforcement learning, case base reasoning, STRIPS operator, object-oriented design*, have been employed and integrated in order to establish an active, user-friendly, mixed text-graphical interface. In addition, goal-oriented systems provide the user with an automatic manual that reaches toward the *"desired situation."* Then these goals are passed to the *"planner agent,"* who performs a search through the *state-action* space, to determine the best available plan.

The paper also presents the main prototype functions from a user's perspective. We show how the different panel windows can lead the user from the set-up of an accident configuration to the choice of the best available response for the current emergency.

## 1. Introduction

Due to the growing complexity of industrial plants and infrastructures, the proper management of large-scale industrial emergencies becomes increasingly difficult, especially for those emergencies that involve several organizations. The need for support tools aimed at performing significant elaboration and situation assessments in emergency situations are also increased, and such tools are expected to simplify the decision-making's task during real emergencies. On the other hand, these tools are required to be reasonably reliable, using the most advanced technologies in the field of informatics and telecommunications.

The demand is growing for more general tools for emergency management, and the availability of user-oriented technologies is strongly required in such fields. Such instruments are to be used in actual emergencies by competent operators. As Emergency Management becomes increasingly important, likewise developing IDSS (Intelligent Decision Support System) is critical to aid Decision-Makers in finding the best available choice in a pressing context.

This conference paper integrates last year's presentation *[ref.1]* that was limited to merely illustrating methodological and conceptual issues. However, this current presentation advances last year's project by analyzing the methodology's results.

## 2. State of the Art

The IDA project is part of the research pursued by our group for several years. Inside this line a set of decision support systems had been developed for emergency management of a large-scale territories; such scale requires the involvement of authorities responsible for civil protection of the population. This project, although following previous research lines, represents a novel approach in the field of automatic planning and represents an example of IDSS (Intelligent DSS). It offers the customer a more active support, closer to the role of Decision-Maker, allowing the implementation of the reasoning model for optimum deployment.

The IDA Project started in 1998 and was the first step that detailed the technical specifications and user's requirements for a decision support system directed to an emergency manager *[ref.2]*. It redevelops the knowledge and experience gained from our group in the European Union *ENVIRONMENT* project MUSTER *[ref.3]*. The first application of the decision support tool used was the Genoa Oil Port. This project was already modeled (in the MUSTER project) in terms of *graphic layout, allowed actions, risk objects, resources, roles of the on field/on site emergency coordinators*, etc.

Great attention has been paid to the modeling of the emergency domain. UML (Unified Modeling Language) has been used and particularly the Rational Rose *[ref.11]* tool, which allows automated code generation from system specifications, thus simplifying and supporting the software development phases.

The realized system assists emergency managers of different roles. It allows the configuration and simulation of several accidental scenarios, and provides managers the possibility of validating their decisions.

The user, after the selection of a reference scenario, manages the emergency with the aid of the realized tool. Therefore the objective of IDA system is to collaborate with the emergency manager to identify and validate the best available emergency plans. Although the realized prototype refers to a specific application domain, it can easily be reconfigured for different domains. That involves however, the redefinition of the object classes used in the prototype. A major effort will be to define the higher-level classes, in order to make the system domain-independent.

Intelligent Agents technology has been used to define the global architecture of the system. It allows separation and grouping of common functionalities inside independent modules, communicating with each other. A future extension could be the distribution of every agent in different network nodes that makes possible the communication through Internet. *[ref.4]*.

The usefulness of such scenarios will be improved by the capacity of simulating in detail the behavior of the domain objects. For such purpose various simulators will be used, numerical or qualitative, in order to define the behavior of the critical objects for the emergency (for example a ship or a tank that burns, an industry that emits toxic substances in the atmosphere, etc.).

**3. IDA Components Architecture**

IDA system is mainly composed of three software agents, each of them performing different functions and each one exchanging information. As shown

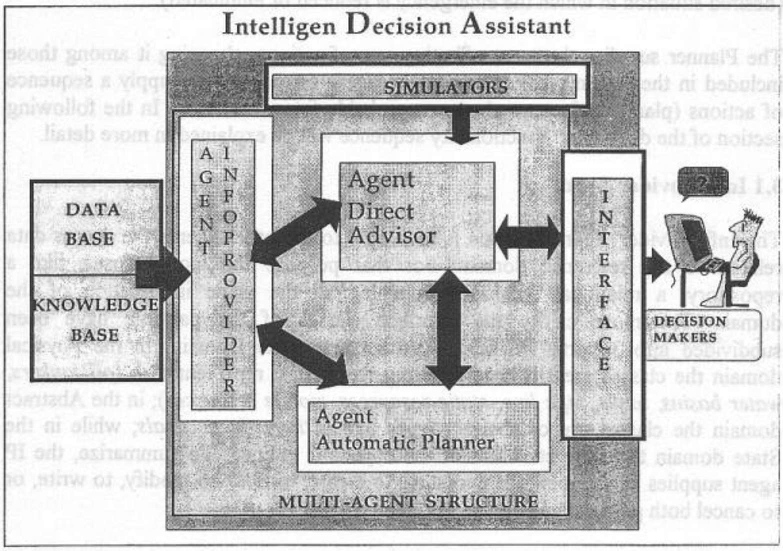


Fig 1 IDA System Architecture

in fig. 1, the three agents are the Direct Advisor agent (DA), the Automatic Planner (AP) agent, and the InfoProvider (IP) agent.

Every agent is in charge of a set of particular functions. The DA is dedicated to the interactions with the user, from which it receives requests, and supplies data and results; moreover it manages directly the I/O with the simulators. The IP directly accesses Data Base and Knowledge Base, and provides the retrieved data to the other agents. The Planner is responsible for the suggestions regarding the more suitable action to be performed.

The system functionalities could be decomposed as follows: At the beginning the user chooses from the database the initial scenario data sets, and asks the IP to load them. At this point the user creates an emergency instance, choosing it from possible ones for every category of risk objects. With the creation of the emergency, the directly involved object-state will be modified. For example, if we create an emergency type of *fire* for a tank, its state will be changed from "idle" to "set fire."

Whenever an object state change is generated (eventually), a certain domain simulator must be running. In this case, the fire simulator is activated and, as a result, the values of the states of the objects involved in the emergency will be modified, since they were placed near the set fire tank. The next step will create the set of "goals" in a form of *desirable states* in the current emergency situation. Such results are passed to the Planner, including the states that represent the starting situation (of emergency), and the goals defined as the state to be reached (desired situation in which the emergency is reduced or eliminated).

The Planner supplies the most effective sets of actions, choosing it among those included in the action table of the database. He can moreover supply a sequence of actions (plan), or forecast the most probable future situation. In the following section of the document, functionality sequence will be explained in more detail.

### 3.1 InfoProvider Agent

The Infoprovider Agent function is to supply to the other agents the access data related to the reference domain. For that purpose the module uses, like a repository, a relational database containing all the static information of the domain. For more clarity the interface classes of the package have been subdivided into 3 parts: Physical, Abstract and State Domain. In the Physical domain the classes are all those referring to physical representation (*oil tankers, water basins, tanks, pipe line, static resources, mobile resources*); in the Abstract domain the classes are of abstract types like *actions, facts, goals*; while in the State domain the state structure of the objects is defined. To summarize, the IP agent supplies all the methods necessary to create, to read, to modify, to write, or to cancel both physical and abstract domain objects.

Moreover it realizes, in the calls coming from the DA, and transmitted to the Planner, the appropriate conversions in order to make the data compatible to the format requested by the Planner.

### 3.2 Planner Agent

The Planner Agent represents the "intelligent" part of the entire system as it provides the actions to be executed in a particular emergency situation. It uses a framework based on the decisional processes of Markov *[ref.5]*, in which the reality is modeled as objects that can only assume discrete states, and by actions that modify the object states with a certain probability, assuming discrete values or steps. Moreover, every state does not depend on the previous history but only on the present situation. Reinforcement Learning is added to this outline (RL) *[ref.6]*, which allows optimal action to reach the most desirable goal, through the evaluation of the Q-learning function. Such function represents the synthesis of the prudent goal-oriented action, determined by *expected rewards* that are estimated as a function of the *distance* of the actual state from the goal, and the *action costs* considered as negative rewards.

The STRIPS operators *[ref.7]* are used in order to find the sequence of state transitions necessary for passing from the initial to the final situation. The operator subdivides every action in one or more elementary transitions. Finally, the application of Case Base Reasoning (CBR) *[ref.8]* is an alternative to the RL method in determining an optimal action. The method chooses from a base of historical cases that have produced positive outcomes.

### 3.3 Direct Advisor Agent

The Direct Advisor Agent (DA) task is to collect the user requests, sort them, choosing the appropriate functions, and to present the information. Therefore, he acts as an interface between the user and the rest of the system. It is composed by a series of user graphic windows subdivided in visualization panels, with dispatching commands panels. At the starting time, the system opens an initial window, with menus and main commands. From this main window, other windows will be recalled to manage in more detail the specific functionalities. The several windows or panels have been generated using IlogViews [*ref.9*] tool, which generates code in C++ language for every window. For every command associated to a push-button on the screen, an empty function *callback* will be generated; that function will be filled up with the necessary code of the function to be recalled in order to execute the command.

## 4. User Interface

A sequence of view panels allows the user to easily input the required data about the emergency simulation, shifting the focus from a *menu-driven* paradigm to a *goal-driven* approach.

#### 4.1 Situation View

The Situation View panel estimates the risk objects in the oil port domain. The panel illustrates every object: the relative state variables resulting from the generated emergency scenario, or as a result of the simulator evaluation. Moreover, the customer can directly change the value of a state variable, in order to create new emergency situations, with the aim to estimate the goals set by DA, or the actions set suggested by the Planner.

#### 4.2 Emergency View

The Emergency View panel supports the user to create an emergency, selecting it among possibilities and associating it to a risk object that can be chosen among the types of emergency. The variable values change in relation to different emergencies, as shown in the following table:

| Event name | Type | Involves | Attribute | Initial.value | Final.value |
|---|---|---|---|---|---|
| Fire in tank crown | Fire | Tank | fireTop | 0 | 1 |
| Fire in tank dock | Fire | Tank | irradied | 0-4 | 5 |
| Fire in tank top | Fire | Tank | irradied | 0-4 | 5 |
| Failure in foaming fixed plants | Breakdown | Tank | fireRisk | 0-1 | 1 |
| Spilling in dock | Spilling | Dock | spilled | 0 | 1 |
| Fire in dock | Fire | Dock | irradied | 0-4 | 5 |
| Fire in tanker | Fire | Tanker | irradied | 0-4 | 5 |
| Fire in sea dock | Fire | Dock | irradied | 0-4 | 5 |
| Failure in foaming fixed plants | Breakdown | Dock | fireRisk | 0-1 | 1 |
| Failure in fixed cooling plants | Breakdown | Tank | fireRisk | 0-1 | 1 |
| Failure in fixed cooling plants | Breakdown | Racks | fireRisk | 0-1 | 1 |
| Failure in fixed cooling plants | Breakdown | Tanker | fireRisk | 0-1 | 1 |

The principal simulator integrated in the IDA prototype is a fire simulator, which evaluates the range of the radiation levels related to the different fire thermal energy values. The simulator-input data is classified according to an index such as: *name, coordinates, amount of crude oil, diameter, height and inclination*. Further, the information is relative to the combustion of products such as: *crude oil type, gasoline type etc.* Lastly meteorological data is categorized such as: *atmospheric pressure, temperature, speed of the wind, relative humidity, and category of stability*.

#### 4.3 Goal View

The Goal View panel displays the goals generated by DA before they are passed to the Planner, together with the current domain state values, with the aim to identify the best actions to be performed.

As the algorithms used for choosing goals need further investigations in the prototype, it has been left maximum freedom in the goal-generation process. The goal can be elementary, if it is relative to a single element of the previous table, or it can be composed by a list of more elementary goals. For the Planner, in any case, the goal is a list of desired states to be reached (fig.2).

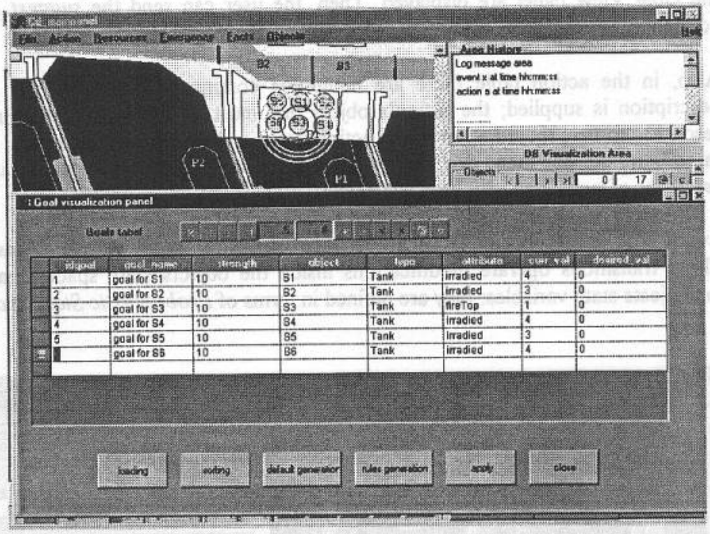


Fig. 2 - Goal view

There are three different ways for the goal-generation process to occur:

- <u>First method</u> goals are generated considering the *desired state* value of the variables, which are defined from the data of the initial non-emergency situation. Therefore, for every state value of any object that is different from its initial value a goal will be generated.
- <u>Second method</u> generates goals by rules. In this case the goals will be generated not from the initial values, but by rules introduced on the basis of the knowledge of a domain expert, in order to generate *protection goals* of the objects, in case they are in a risk situation.
- <u>Third method</u> goals come directly from the user. The user can both modify (or cancel) the goals generated in the previous modalities, and insert new goals according to his personal assessment of the current emergency situation.

After generating the goals they can be memorized, then transmitted to the Planner. They can also be sorted on the basis of specific strength or priority.

**4.4 Action View**

The Action View panel visualizes the sequences of optimal actions produced by the Planner in the current emergency situation. At first the goals generated from the Goal View panel are displayed. Then, the user can send the *suggest_action* command, which makes the action to be carried out available for every goal.

Also, in the action table there are numerous commands displayed: the action description is supplied; the action's object that must be performed; the kinds of resources to use; the duration of the action itself, and the *cost* of the action. In the same way, the command *seq_of_actions* can be performed, and in such case the *sequence* of suggested actions is visualized.

Every IDA action can produce a sequence of one or more *Action Transitions* These transitions operate modifications inside the objects state space, changing the objects state variables: they are defined in terms of *Probabilistic State Space*

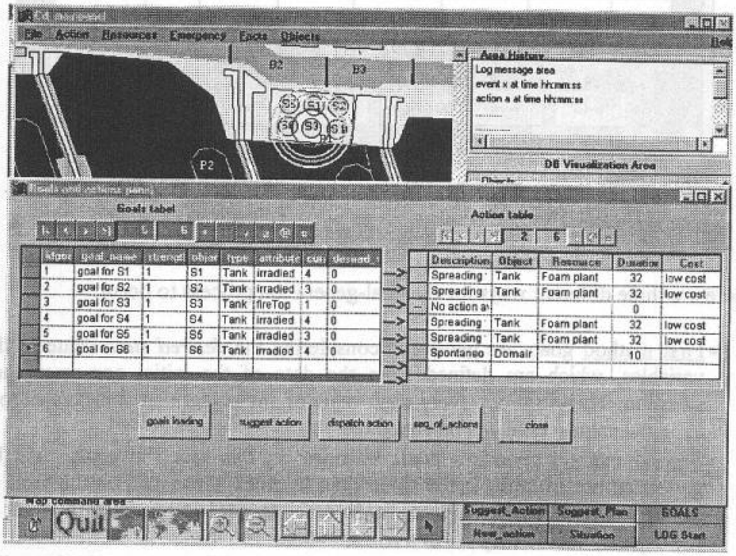


Fig. 3 Actions View

*Operators* (PSOs) are an extension of the classics STRIPS operators, and are formed of the following: *Prelist*, *Dellist*, *Addlist*.

*Prelist* represents the initial states for which the transition can take place. The *Dellist* are the states that will be eliminated due to the transition. And the *Addlist* are the states created after the transition. Moreover, a *probability* is the transition that occurs as a consequence of the action. In the beginning this probability has been fixed for all the transitions equal to 1. After the evaluation of the prototype response by the expert, the probability values could be changed.

**5. Project Result and Future Development**

The IDA project represents a phase of innovation. The development has been carried out using UML notations, with the support of Rational Rose 98, which significantly reduced strongly the development costs and facilitated reusing and modifications of the working phase.

The UML tool facilitated the collaboration of 3 different teams of analysts and developers. After the realisation of a first architecture, with a defined decomposition of functionalities and user requirements, the following implementations were easily integrated inside the pre-defined schema. For the future, a test and validation phase is necessary, with well-defined end-users, by which the system will gain further focalizations.

Entering more detail into the system's function shows the Action View panel, with suggestions of a single action or a sequence of actions to be executed during an emergency phase. Such results have been obtained by employing technologies that need further study, such as the Reinforcement Learning methods, and other well-known methods like STRIPS planner or CBR.

From the user's point of view, it is necessary to simplify both the input phase of the set of possible actions (currently inserted by external files), and their modification during the emergency drill. Moreover, the customer ought to clearly follow the reasoning pursued by the planner, and the link between goals and sequence of states to follow with the suggested actions. Also, the use and optimization of resources could be improved through a better interface and a direct link with the available actions. Furthermore, the system should be more independent from the specific reference domain by generalising the definition of the classes. That, of course, necessitates reviewing the analysis phase and several types of emergencies; this would introduce further hierarchical levels, of the more general type, from which the specialized classes of the single domain originate.

The use of object-oriented technologies together with high-level tools such as ILOG, which develops the user interface, has surely simplified the insertion or modifications phase. Lastly, the potential offered in animation tools remains unrealized by the prototype. These tools could improve the manager's response, providing a realistic approximation virtualized by graphic depictions showing emergencies and the consequences of the user's actions.

## 6. References

[1] An Approach to the Intelligent Decision Advisor (IDA) for Emergency Managers Adam M.Gadomski, Sandro Bologna, Giovanni Di Costanzo, Anna Perini, Marco Schaerf. *Proceedings of TIEMS99. The Sixth Annual Conference of The International Emergency Management Society, Delft, Netherlands, June 8-11, 1999.*

[2] Functional Requirements of an Intelligent Support System for Emergency Management. C.Balducelli, G.DiConstanzo, A.M.Gadomski, R.Iannucci. *Enea Report, 1997.*

[3] A Computerized Support System to Cooperative Training in Emergency scenarios Management and its application to an oil Port Domain. C. Balducelli, S. Bologna, M. Boero, G. Di Costanzo, G. Vicoli- *Proceedings of TIEMEC95.*

[4] Genetic Agent in an EDSS System to optimize resources management and risk objects evacuation. C. Balducelli, C. D'Esposito. *Proceedings of TIEMEC99.*

[5] Markov Decision Process. Martin Puterman. *John Wiley and Sons, 1994.*

[6] Reinforcement Learning Problem. Richard Sutton and Andrew G.Barto. *The MIT Press 1998.*

[7] STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. R.E. Fikes, N.Nilsson. *Artificial Intelligence Journal,2(3-4):189-208,1971.*

[8] Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches. A.Aamodt, E.Plaza. AICom – *Artificial Intelligence Communications, Vol. 7 : 1, pp. 39-59, 1994.*

[9] Ilog Views 2.4 e Ilog Inform 1.1 User's Manuals. *Ilog Corporation.*

[10] FIPA Web site : e

[11] Visual Modeling with Rational Rose and UML. Terry Quatrani. *Addison Wesley 1998.*