An Approach to the Intelligent Decision Advisor (IDA) for Emergency Managers

Adam M.Gadomski*, Sandro Bologna*, Giovanni.DiCostanzo*, Anna.Perini **, Marco Schaerf***

* ENEA, CR Casaccia, 00060 Rome, Italy, gadomski_a@casaccia.enea.it
** IRST, Loc. Pante' di Povo, 38050 Trento, Italy, perini@irst.itc.it
*** Universita` di Roma "La Sapienza",via Salaria 113, 00198 Roma, Italy, schaerf@dis.uniroma1.it.

Kaywords: Intelligent Decision Support System, Emergency, Management, Intelligent Agent,Plan.

Abstract

The paper presents a next ENEA's step towards development Intelligent Decision Support Systems (IDSS). The prototype IDA (Intelligent Decision Advisor) for emergency management in an oil port is analyzed as a test-case. The work was performed in frame of the national R&D MICA project under the umbrella of the ENEA’s long term strategic MINDES Program synchronized with indications of the worldwide GEMINI (Global Emergency Management Information Network Initiative) of the G7 Committee. IDA is an approach to the design of intelligent-agent based kernel of IDSS. In frame of the generic TOGA(Top-down Object-based Goal-oriented Approach) model of abstract intelligent agent, IPK (Information Preferences Knowledge) architecture is employed. The IDA objectives were to develop and verify the properties of information managed agent and knowledge managed agent, where the last should suggest an action or plan after every new significant event in the emergency domain. The IDA functional kernel is composed with three simple agents: -*DirectAdvisor,* it interacts with human user and emergency domain, -*InfoProvider,* it manages of information and intervention goals, and - *IDAPlanner,* it plans adequate interventions. For the design, UML (Unified Modeling Language) has been employed. MDP (Markov Decision Process) and CBR (Case Base Reasoning) are used for plans construction. Owing to a generic agent model and object-based conceptualization, the IDA system should be adaptable to the different roles of emergency managers. The IDA conceptual solutions can be also seen as a successive step towards high-intelligent DSSs.

" New thinks get started by evolution of chance, not design"
(-)Allan Newell

1. Introduction

The first prototype of an Intelligent Decision Advisor **(**IDA**)** for Emergency Management has been the objective of the MICA R&D project 2.8.3.1.F focused on the development of an active computer support for large scale industrial and territorial emergencies.
The project has been realized under the umbrella of the ENEA’s long term strategic MINDES Program synchronized with the worldwide GEMINI (Global Emergency Management Information Network Initiative) of the G7 Committee [18], [21].
The main objective of MINDES is to develop an intelligent decision-support kernel for the computerized nodes of an Emergency Management Network. Local decision support systems

should be connected by the Internet with another similar emergency management centers in frame of an international Global Emergency Management Information Network. The accepted software solutions are based on the intelligent agent theories and technology [Internet]. The MINDES Program profits the past experience of ENEA in the field of emergency management and plant-operator support systems. Especially, the program scope is to reduce the probability of human managerial errors during decision-making.

The following strategic objectives of MINDES were defined:

◊ To provide real-time data and a global situation awareness necessary for the emergency managers decision-making during emergency situations.
◊ To use data from the available information systems present in- and out- of the test site. The end-user test site could be, for instance, a chemical plant system (oil refinery) or a regional emergency management organization.
◊ To have a ***user-friendly interface*** to assist a typical staff of high level industrial or administration managers. Easy to use communication interface will be supported by multimedia techniques, a GIS system, and a voice commands option. Therefore, ***the use of the system should not require any support of the computer specialists*** **.**

It should be a tool for periodical managers' training sessions of *emergency games*.

In the above context, the IDA contribution to the MINDES-GEMINI program has been defined as a one year R&D work performed with the contribution of IRST - Trento and the Rome University "La Sapienza". The project was focused on the modeling of the domain of emergency and a verification and validation of some intelligent agent technologies in a selected class of concrete emergency management cases.

The paper deals with three subjects related to the ENEA's experiences and on-going results:

◊ Motivations for the development of an intelligent multipurpose DSS and IDA objectives.
◊ Theoretical and methodological frameworks
◊ IDA; a prototype of IDSS.

2. Theoretical Background

2.1. IDA contexts

One of the key concepts of IDA is "agent". It is considered as a functional software unit with the capability to execute a pre-defined class of tasks autonomously, i.e. without help of its user. Its different types, in the subject matter literature [Internet], depend on implementation software environments, different definitions of task, autonomy, and on the selected domains of expected intervention.

Intelligent agent is here understand as an agent with high autonomy which enables self-modification of intervention goals, learning, and tasks/actions planning. In general, the concept "intelligence" is not well defined in the AI literature.

The problems of the construction of IDSS (Intelligent Decision Support System) for different emergency domains and operator/manager role were discussed and illustrated in the previous ENEA's papers since 1993, see for example [13], [15], [16], [20]. The main idea is based on the TOGA (Top-down Object-based Goal-oriented Approach) conceptual framework, proposed and theoretically developed by Gadomski since 1989, where an abstract intelligent agent with the hierarchical multilevel IPK (Information Preferences Knowledge) architecture, called personoid, is employed in the reasoning kernel of IDSS. The TOGA hypothesis has required an experimental verification of the IPK in frame of a IDSS structure. The project has been a first approach to the application of some aspects of the personoid concept, defined as an abstract intelligent agent with a *structural intelligence* [13],[14], and also seen, as a hypothesis of a *reusable, incremental, repetitive*, *recursive* and *iterative* architecture of an abstract *intelligence.*

The concept *intelligence*, here assumed, is discussed in the paper [13], [14] in a wide sense, ***intelligence is a capability of a system to use possessed knowledge and preferences in order to achieve new objectives.***
The above definition is roughly congruent with major part of psychological IQ tests [Internet].
According to TOGA, others, more complex properties of intelligent agents, should be possible to infer from this definition if a proper definitions of information, knowledge and preferences are chosen.
Concluding, the verification of the hypothesis above formulated, requires a theoretical specification of an *abstract intelligent agent* , i.e. abstracted from realization and application domains, and its experimental implementation in the kernel of a decision -support system. In practice, different specific active DSS prototypes (not very "intelligent") were just designed and implemented, as subsequent conceptual iterations towards an multipurpose, domain independent IDSS architecture, see, for example CIPRODOS [19], GEO [20].

2.2 IDA objectives

In order to present the IDA project objectives we need to recall the generic functional personoid architecture composed with the triangle IPK modules [14], as presented on the Fig. 1.

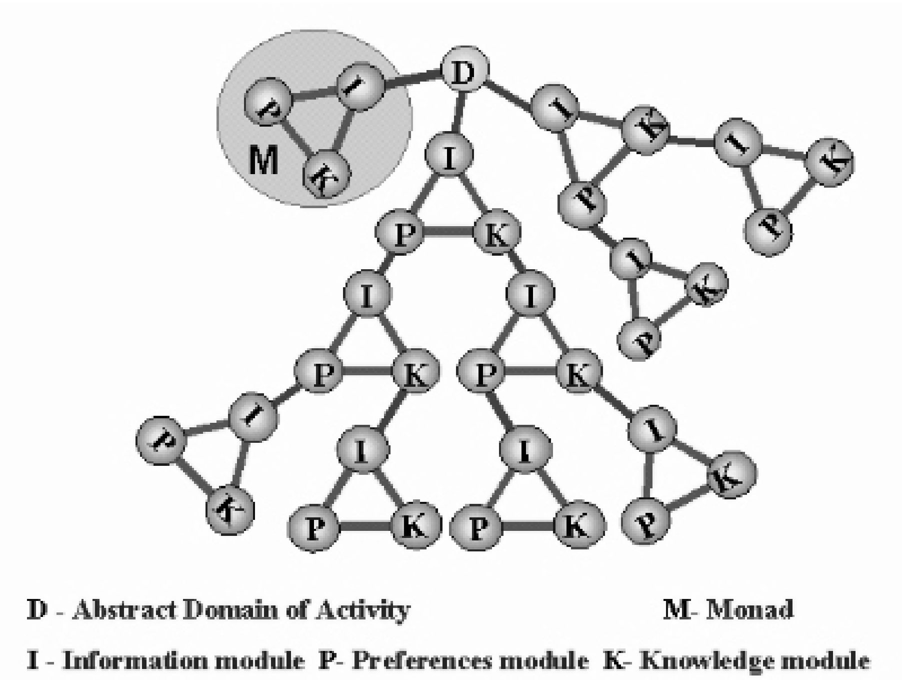

Fig. 1. An example of abstract intelligent agent structure: the IPK architecture of personoid.

Here, *information* concept represents every state of pre-selected intervention domain, *knowledge* is every mental/abstract entity/system which is able to transform information in another information, and *preferences* are relative rules which order domain states according to a subjective importance scale accepted by a human domain-expert.
A simplified basic mechanism of personoid is the following:
New Data from the Domain modifies Information.- Information activates Preferences.-Preferences produce Goal. - Goal activates Knowledge. - Knowledge modifies Information.
If Preferences or Knowledge are not able to execute such task then it produces Data for the higher level triangle and itself becomes the new Domain, for instance, for a planning function.

In this way, personoid has many possible abstract domains of activity on different abstraction/meta levels of reasoning. They are: an image of real emergency domain, agent preferences bases and agent knowledge bases. The concepts: intervention domain, abstract intelligent agent, information, preferences and knowledge consist a basic ontological platform of the TOGA theory.

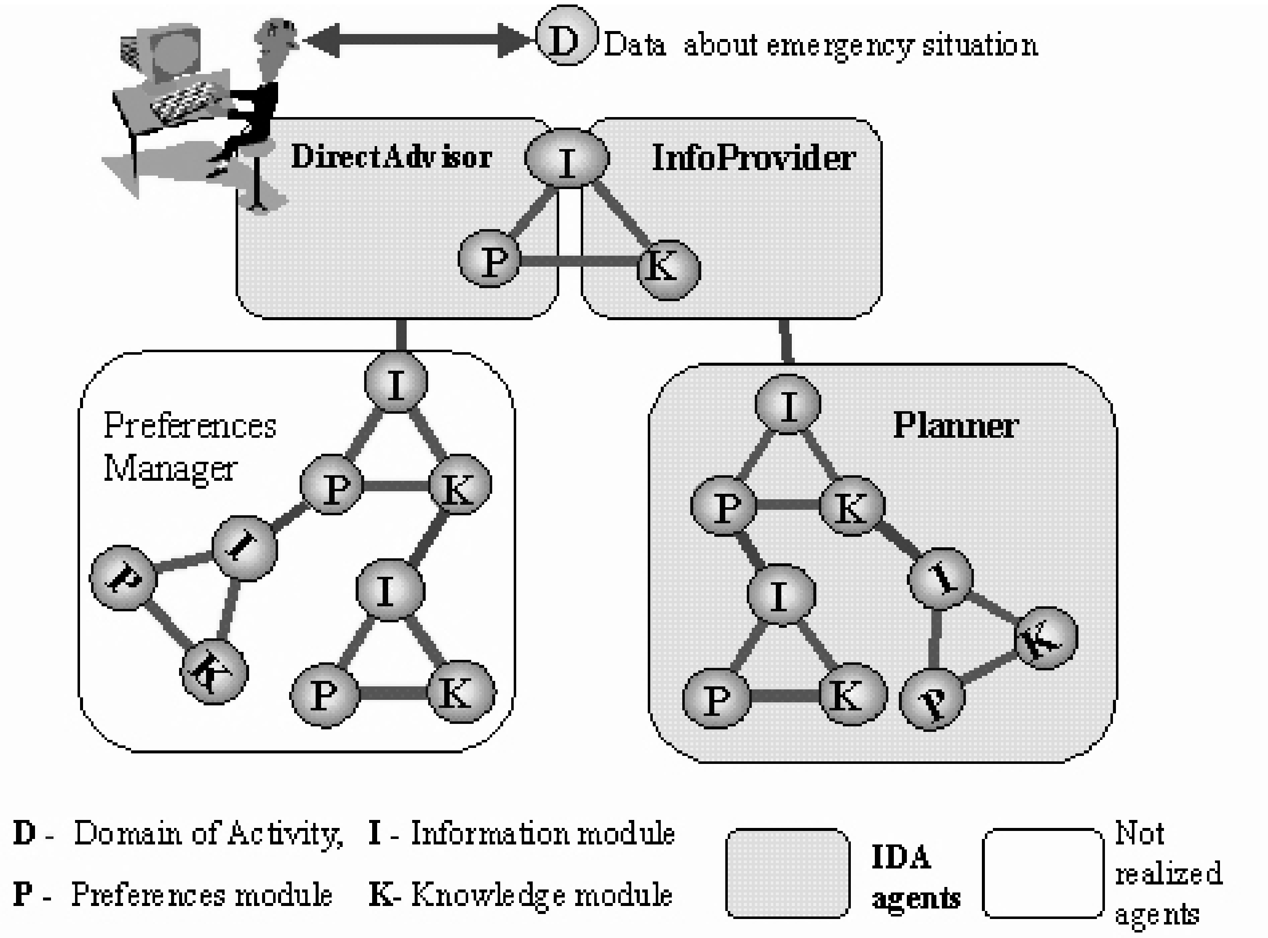

Fig.2. Substitution of the personoid structure by the IDA tree agents for the verification of information management and planning functions.

In such context, the main IDA objectives [1] were to develop and verify information management function by InfoProvider agent, and a knowledge management by Planner agent, where the last should suggest an action or plan (a sequence of actions) after every new significant event in the emergency domain. For the reason of numerous technical problems which ought to be solved, the PreferenceManager agent has been substituted by the fixed set of the possible goals of intervention. Its modeling and realization requires yet a theoretical analysis. It should be added to the system in the IDA-2 version. From the technical point of view, the work has been concentrated on: - Modeling and formalization of emergency domains, in particular, on their representation by a generic world of objects with events, resources and facts. The dynamics of the domain has to be introduces by qualitative relations between components of the domain. - Choice, modeling and verification of planning methods and their implementation.
The fig. 2. illustrates the functinal decomposition of peronoid on the prototipal IDA agents. As a design conceptual platform the UML language has been adopted [Internet].

3. Information, Knowledge and Goals

3.1 Information and the Test-Case

The test-case describes the emergency domain from the EU MUSTER project [16], it is an emergency in an oil port. Every global or local state of the domain communicated to the manager is represented as an information and is available explicitly. The map with the initial situation of the emergency game is illustrated on the Fig. 3.

The emergency-management top-goal is to stop losses generation process and to minimize total losses, i.e. the innitial emergency state has to be changed to the state accepted by the port manager and losses and risk generation process caused by the fire of one of the six tanks must be stoped. The local emergency manager must use in the best way his/her own resources, he has for disposition a set of operative units which everyone can execute a predefined set of tasks.

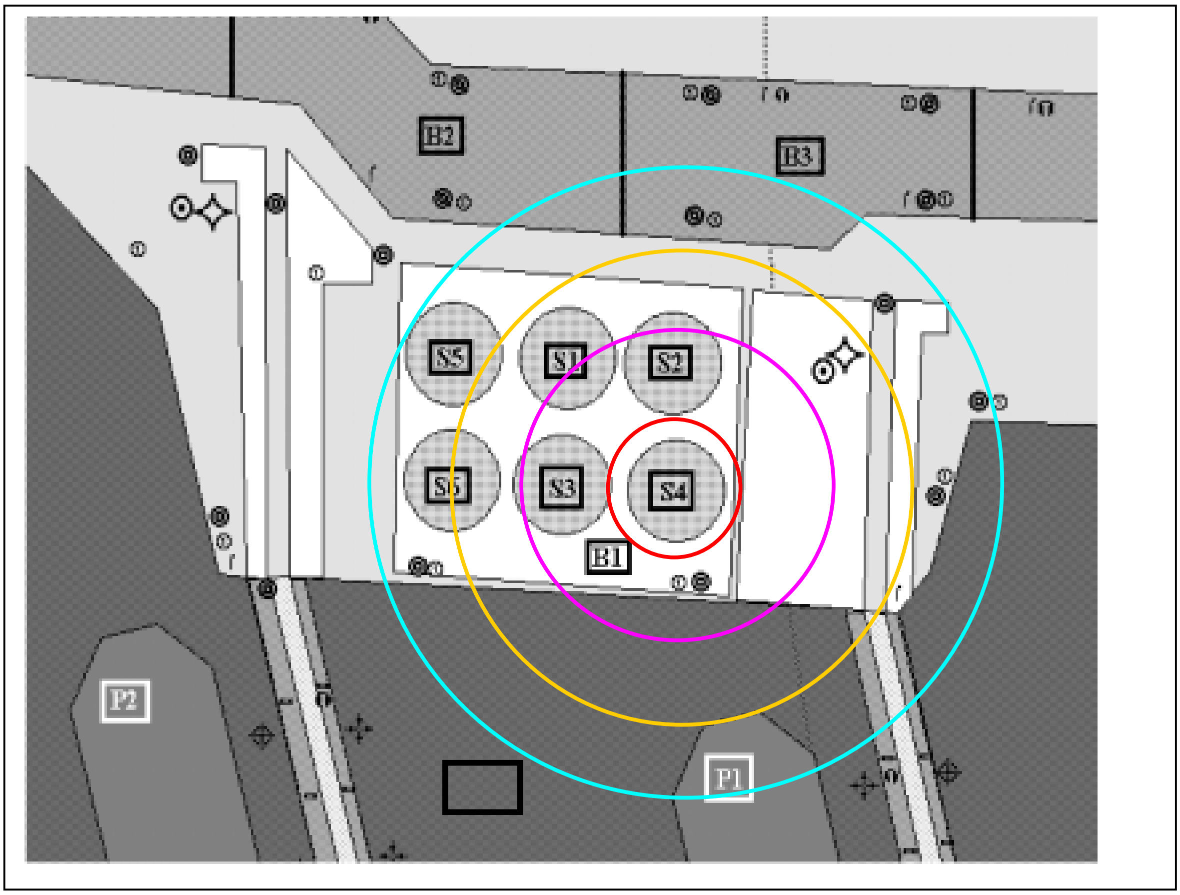

Fig. 3 Test case: A tank fire in an oil port, where: S1-S6 are tanks, B2, B3 are docks, P1, P2 are oil tankers.

3.2. Descriptive and Operational Knowledge

<u>Descriptive knowledge: domain model</u>

In this section we discuss how domain dependent information is transformed. In particular we will describe the state and the action representation used for the system planing function. A generic emergency domain model referred to some class of the emergency domains is considered as a descriptive domain-knowledge, formally it works as follows:

$$\mathbf{KD}_j : \; I_i \rightarrow I_{i+1}, \; i=1,2,\ldots$$

where I_i, I_{i+1} are two information which are components of a specific description of emergency domain state, $\mathbf{KD}_j$ is a domain descriptive knowledge which is a part of the relations among classes of abstract objects present in the domain model.

Every concrete element of the domain is represented as an instance domain objects and resources. For instance, oil port domain objects classes are tanks, tankers, dock and racks.
The system states are described by a finite set of state variables that takes on discrete values, they model features of the object domains. For instance the temperature of a tank is modeled by a boolean variables that states if the object temperature is above or below a critical value or the level of coverage by foam of an object is an integer ranging from 0 (no foam) to 3 (fully covered). The table 1 illustrates the classes of state variables (attributes) currently used for the oil port domain.

Object / Type	dock	tank	tanker	rack
4	irradied	irradied	irradied	irradied
B	spilled	spilled	spilled	spilled
B	temperature	temperature	temperature	temperature
4	foamCover	foamCover	foamCover	foamCover
B	fireRisk	fireRisk	fireRisk	fireRisk
10	eventSize	eventSize	eventSize	eventSize
B	-	fireTop	-	-
4	-	productLeve	productLevel	-
B	-	-	tankerAtWharf	-
B	-	-	inDock	-
B	defiled	-	-	-
B	wharfAcces			

Table 1: IDA descriptive domain knowledge in the form of table: State variables are defined for the domain objects. The first column (type) indicates if the variable is boolean (B) or the number of discrete values that it can assume.

So for given set of variables (=object attributes) is used for describing the state of the environment, for instance {Tank1.irradied, Tank1.temperature, Tank1.foamCover} and a specific state (information) can be s = { Tank1.irradied= 3, Tank1.temperature=0, Tank1.foamCover=0}.
In this way the descriptive knowledge can be enriched simply adding new state variables to the domain mode**l.** This operation can be performed by the system user.

Operational Knowledge: Actions
Every action possible to the execution for the specific emergency manager (E-M'er) is considered his/her operational knowledge:
$\mathbf{KO}_j$: $I_i \rightarrow I_k$**,** where the subsequent informations describing the domain before and after the action execution.
$\mathbf{KO}_j$ is an action j which is included in the specification of E-M'er role.

In the IDA system, actions representation is similar to the *probabilistic state space operators* (PSOs) [7, 9] an extension of the classical STRIPS operators [12]. A PSO α is a triple (φ,ρ,ω) where φ e ω are conjunctions of atomic formula $(x=v)$ where $x \in X$ is a variable state and v is one of the possible variable values. φ represents action preconditions that must be satisfied in order to be able to apply the operator α, resulting in environment transition to the state described by ω with probability ρ. ω is also called postconditions. In practice, α is a set of STRIPS operators [12], enriched by a probability value associated to each transition.
For instance the complex action of spreading foam on a tank is described as follows:

Action: *Foam the top ring of a tank when irradied or burning or spilled*
1. Preconditions: {obj(T,t) $\wedge$ spilled(T,1) $\wedge$ fireRisk(T,1) $\wedge$ foamCover(T,0)}

Postconditions: *Delete List:* fireRisk(T,1) $\wedge$ foamCover(T,0)
Add List: fireRisk(T,0) $\wedge$ foamCover(T,3)

2. Preconditions: {obj(T,t) $\wedge$ irradied(T,3) $\wedge$ foamCover(T,0)}
Postconditions: *Delete List:* irradied(T,3) $\wedge$ foamCover(T,0)
Add List: irradied(T,2) $\wedge$ foamCover(T,3)
3. Preconditions: {obj(T,t) $\wedge$ irradied(T,4) $\wedge$ foamCover(T,0)}
Postconditions: *Delete List:* irradied(T,4) $\wedge$ foamCover(T,0)
Add List: irradied(T,2) $\wedge$ foamCover(T,3)
4. Preconditions: {obj(T,t) $\wedge$ irradied(T,5) $\wedge$ foamCover(T,0)}
Postconditions: *Delete List:* irradied(T,5) $\wedge$ foamCover(T,0)
Add List: irradied(T,2) $\wedge$ foamCover(T,3)

In the IDA system actions describe techniques of emergency management that rest on the use of specific means (for instance means for spreading foam or water) and squads. Each technique requires information like: - an estimate of the minimum duration time, - the domain objects on which it can be applied (for instance Tank) and the relative state variables (attributes) that will be affected, - the class of means that can be employed, - a qualitative estimate of the action cost (for example 0=null, 1=low, 2=medium, 3=high).

4. General IDA Architecture

The IDA prototype is an intelligent agent with capability to the execution of a certain class of tasks autonomously. It is composed with the following functional agents:
- DirectAdvisor, - InfoProvider and - Planner.
All together have the capability to:

- the representation of the current emergency domain state and updating it while new events are notified by the InfoProvider agent;
- the representation of user goals notified by the InfoProvider agent;
- the suggestion to the DirectAdvisor, an action or a plan;
- the forecasting to the DirectAdvisor the environment state on the request and upon execution of a given action.

4.1 InfoProvider

The Infoprovider agent is a common interface for other agents and for the IDA user. The main tasks performed by the InfoProvider can be divided into two main groups, since we pointed out two main functionalities in the IDA system. First of all, the InfoProvider provides current information to the two other agents on the actual state of the emergency domain and resources. These data are stored in a relational database (using DBMS Ms Access 97). It manages all domain dependent information and provides them in a requested form to the other agents, it acts as a mediator for Planner and for Direct Advisor. All information related to emergency domain map represent an abstract domain of activity for the IDA. The second group of functions is related to the updating and management of the DataBase on the request. The InfoProvider has a capability to reasons on these data in order to provide information at a higher level of abstraction.
The DataBase contains information, goals and the user operational knowledge in the following relational tables:
Objects* -** domain objects list divided on fixed classes; ***ObjectState* -** attributes of resources in any given moment; ***Resources* -** resource objects lists ; ***Resource State* –** attributes of resources in any given moment; ***List of Goals - plausible goals of local interventions. ***Set of Actions* -** it contains the actions for the selected roles of emergency manager**;**

New components of these tables can be added at the run-time of the system.
DataBase can also be seen as three interacting functional components:

1. Interface to other modules;
2. Abstract components: facts, actions and goals. The abstract components contain the knowledge on the usage of the resources and the evolution of the emergency (historical data);
3. Actual components: objects and resources. They represent current state of the domain.

The analytical results obtained here suggest that the temporal intervals, non-monotonic and default reasoning [22], [23], [24] could also effectively support a generic inference tool employed for information, knowledge and preference management in frame of the IPK architecture.

4.2 Planner agent

Emergency management or crisis mitigation planning problems requires to be modeled as an *interactive decisional process* where the human and the machine reasoning activities interleave. For instance, if given an emergency situation such as a fire on a tanker at an oil port dock then the following interactions between the human and the decision advisor system can be described:
the human decision maker can pose specific goals devoted to maintain under control/survelliance a critical zone of the port, such as neighboring tanks, in order to avoid the evolution of the emergency into a catastrophe. So, for instance he can express goals such as maintaining the temperature of neighboring tanks under critical values or reducing the amount of inflammable liquids near the crisis epicenter. As the IDA response, the system suggests to the human decision-maker appropriate actions/tasks to perform in order to manage the emergency situation taking into account the specific intervention goals and, in parallel, evaluating actions costs versus their benefits (in the IPK architecture these criteria are isolated in the meta-preference rule-base). So, for instance, the system can suggest to take the tanker away from the dock and let it burn instead of trying to extinguish the fire at the dock maintaining a high risk of fire propagation to the other port resources; the IDA can be asked to forecast the state of the oil port after the execution of a given action; the decision maker decides which action to perform. This kind of planning problem is also known as *Mixed-initiative planning* [3].
Moreover the Planner agent was built according to basic requirements of system extensibility and of applicability to different application domains. In particular:

- the requirement of realizing the agent competencies,(the planner meta-knowledge and meta – preferences) using different approaches ranging from deterministic to non-deterministic AI planning approaches.
- the requirement of building a planner whose domain model could be improved or extended or substituted with a different emergency domain.

The first requirement was met by defining the class hierarchy (a C++ class) depicted in figure 3. The *IDAPlanner* class is the planner module interface. Its functions implement the Planner agent competencies described above.
The classes that specializes the IDAPlanner class represent different ways of their possible realizations. The MDP Planner class implements these competencies following a decision-theoretic scheme based on the Markov Decision Processes (MDP) techniques [11], but other specializations can be built. The MDP Planner will be described in more details in the following.
The second requirement has been taken into account by isolating domain dependent information from an operational knowledge (actions) into the action library, in this way, a module that is domain and situation dependent can be extended or replaced by another one.

The MDP Planner.
Planning for an emergency in an oil port requires to deal with a complex dynamic environment whose behavior is determined by phenomena dependent on many parameters, often not directly observable. The environment dynamics is hard to represent by the deterministic domain models. So

the effects of a specific action in a given state are not known a priori. When the reasoning on specific *goal states* to be reached becomes not meaningful, the problem solving to be deal with calls for, so called, process-oriented [8,9,10] approach. In other words the goals in the EM planning are also localized on higher abstractions level. For instance, one of such goals (top, maximal preferences) can require to maintain always the whole plant under control.

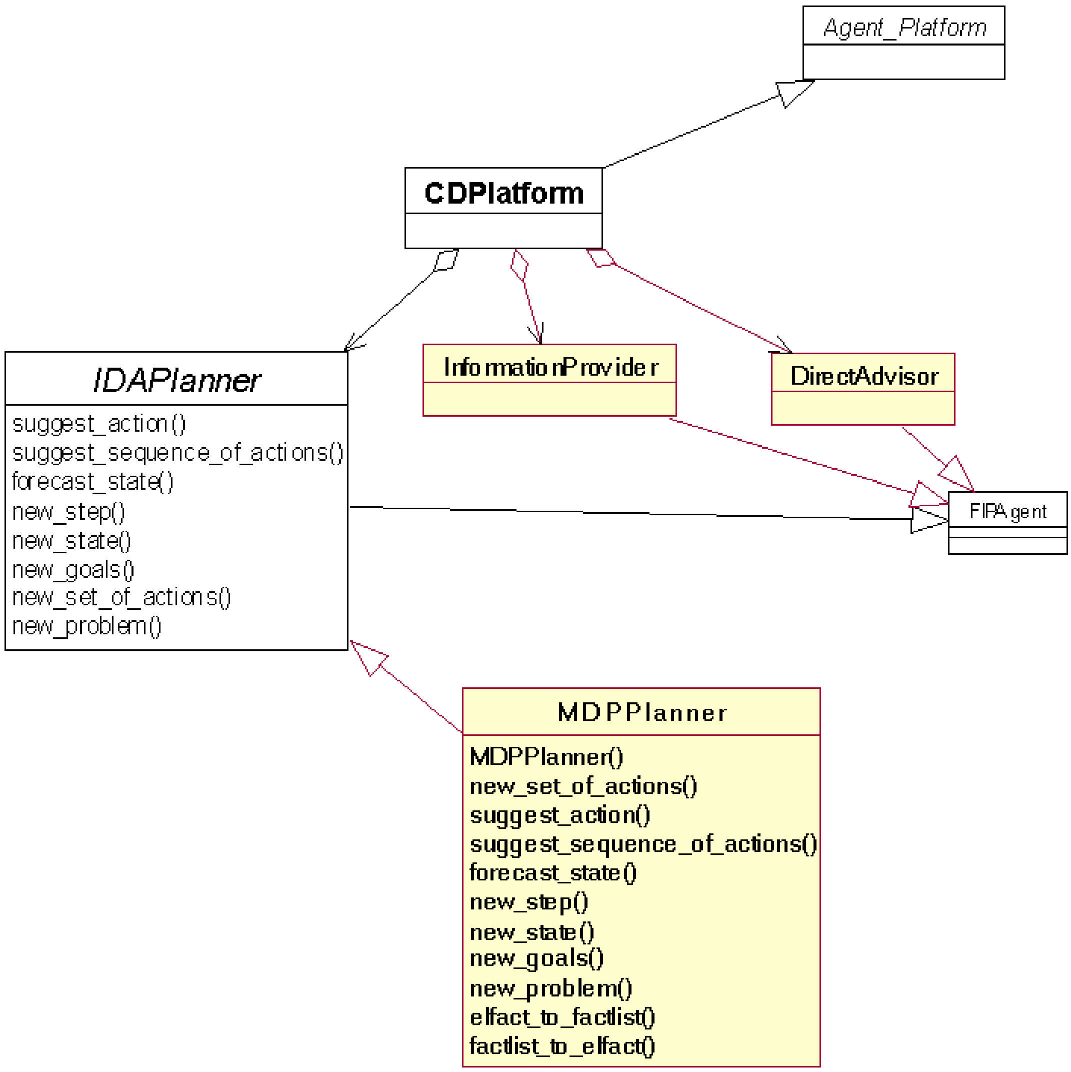

Fig. 4. The MDPPlanner class Diagram, UML notation [Internet].

Another aspect of the emergency management planning that we took into account is the role of past experience in decision making. The practical knowledge about utility of the past plans and their adopting to similar emergency situations has motivated our choice of Case-Base reasoning (CBR) techniques[12]. In particular, in the MDP Planner an architecture similar to that of Dyna-Q, described in [9] is implemented. The Planner provides a policy computed upon the optimization of a value function. If a memory of value function estimates, defined respect to a given set of goals, is available. The CBR techniques is used to start the planner with a better value function estimate, possibly resulting in a faster optimization process. The architecture of the MDP Planner is depicted in the Fig. 5, its basic components are:

- the controller that manages the learning process of a new experience both from the real environment and from the simulated environment;
- the simulator that executes a sequence of simulation steps following a given strategy;
- the simulated environment that exploits a model of the environment based on transition probabilities between two environment states, upon execution of a given action, the reward (R) and the state-action value function (the Q function);
- the CBR component that manages a case base of actions and a case base of Q functions.

More detailed information related to the architecture of the IDA Planner are included in the report [25]. Modeling an intervention planning respect to the MDP framework posed also some interesting problems such as how to model complex actions with various duration time and actions that can be executed in parallel, see for example [22]. Concluding, from the personoid perspective, the planning process can use different criteria for internal choices and can be realizable by different planning methods. These criteria were identified during the project and could be inserted as an independent meta-preferences base in the next IDA version.

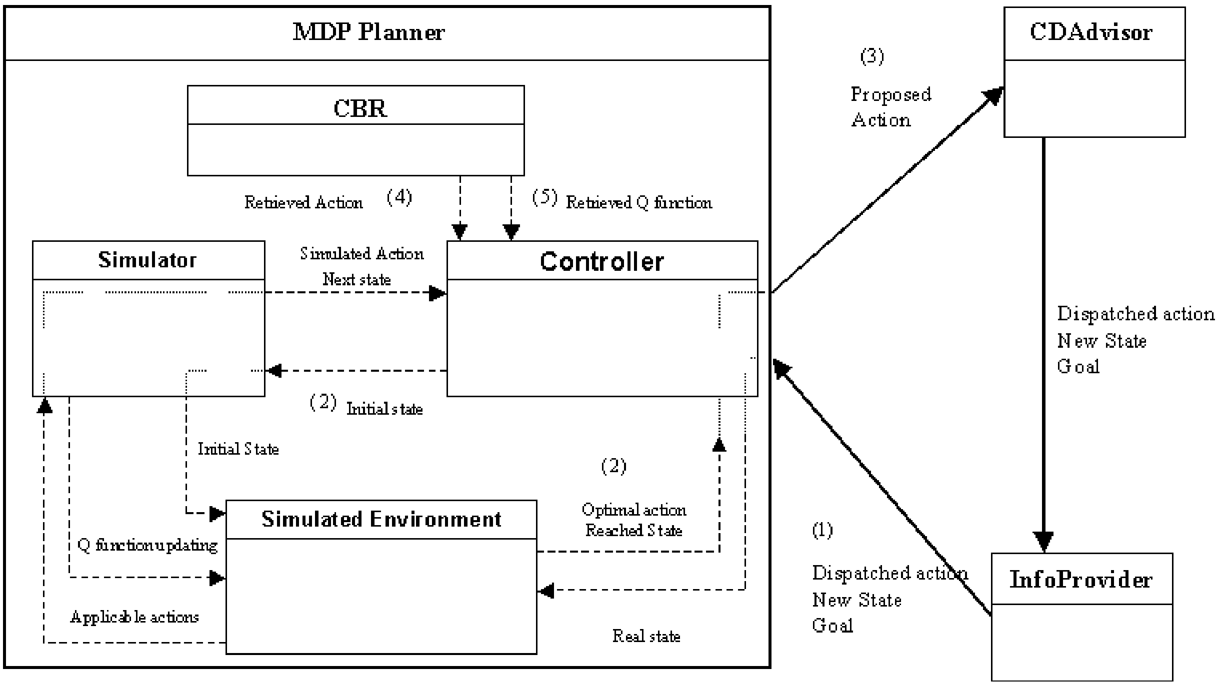

Fig. 5. The IDA structure with the Planner agent architecture (CDAdvisor = DirectAdvisor Agent)

4.3. DirectAdvisor Agent

Every managerial intervention in the Emergency Domain is executed by subordinated human agents, as firemen, policemen, captain of ship, plant operator. These activity requires messages exchange.The messages can includes: *Tasks in the Domain, Tasks for Experts* and *Tasks for Executors*, for instance: - ask an expert about Z, - send an information X , - request (command) to perform an action A. The objective of the DirectAdvisor is to be an interface among information source, i.e. InfoProvider and intervention Planner, and human manager.
The users has three types of interactions with IDA:

1. Map set-up session, when various domain maps are inserted
2. Emergency Set-up session when the type of emergency, initial emergency situation , possible intervention goals and the managerial role is edited
3. IDA Demo Emergency-Management Interactive session**.**

Demo illustrates a scenario of a simple industrial emergency, some kind of an “emergency game” . From the perspective of the game theory it can be calIed a *game with nature*.

The domain and rules of this game can be modifiable/updated by the player, using interface module of the DirectAdvisor. . The *player* is an emergency manager with a predefined role.
Here, we illustrate how, by an interactive generic conversation scenario between IDA and its user, IDA may help to make decisions.
Let us assume now a play convention

- He/she plays with a simulated *emergency*.
- He/she tends to achieve *emergency management goals*.

<u>Simple scenario of a generic emergency-game</u>

In parallel to the system interface specification, the definitions of the used concepts have to be done. We have introduced:

1. map - i.e. a map which represent an emergency territory with infrastructure.
2. current state (state of the map) - a template for the map state.
3. possible operations (i.e. manipulations=interventions=actions) on the map**.**

The player activities are focused on:

- current emergency state identification, - cause searching (backward- propagation)
- consequences searching (forward -propagation), - objectives searching (a max. preferred state) in preferences base, - elaboration of a suggested intervention (an action), as a plan, instruction, task.

In general, we have player decisions: ***- Without support***; where the player have to choice data sources, data/information, action or to elaborate and to execute actions/intervention himself. ***- With support***; where the player needs to choice only between actions suggested by the system. In the IDA current version the map, user role and emergency initial situation are pre-loaded.
The generic IDA scenario is as follows:

1. E-M'er (emergency manager) inserts new information (facts) about a change in the.
2. IDA provides data about current state of emergency domain (menu-driven).
3. E-M'er choices intervention-goal from the possible goals list.
4. IDA presents intervention-plan as a sequence of actions.

E-M'er inserts new facts; for instance, when the objective of the suggested action was not achieved the user should decide: goto 3 or go to 4.

<u>IDA User Interface</u>

The IDA User Interface has a form of a set of goal-oriented hierarchical windows it is composed of a main window, acting as a starting panel and of a series of panels each one designed for input/output of a specific kind of data (information, goals and knowledge).
The main window has been divided in four different area (Fig.6), reserving each area to a particular type of operations or commands. The left high area is for displaying continuously updated map; this is the major area in term of surface occupancy on the screen. At the lower level of this area are placed a series of map commands, as for zoom in, zoom out, movements up, down, left, right, changing of visualization at regional or local level; faster movements are also obtained using vertical and horizontal scroll. A button for loading the graphic file with an emergency map, to be displayed after introducing his name, is also present. In the right high area are displayed data table coming directly from databases like risk objects, resources, actions. In the right low area a series of button commands designed specifically for the planner are grouped. These commands perform a series of functions aimed at obtaining information about the most important kind of data (situation of state objects, goals, suggested actions or plans). On requiring each of these commands, the system answers providing a new panel, that groups most detailed information according to the kind of command. For example if “GOALS” button is pressed, a new panel will display a list of goals already present for a particular situation of emergency. Also buttons for generating new local intervention goals, their deleting and reordering are provided. Others more specific commands are included in each detailed panel.

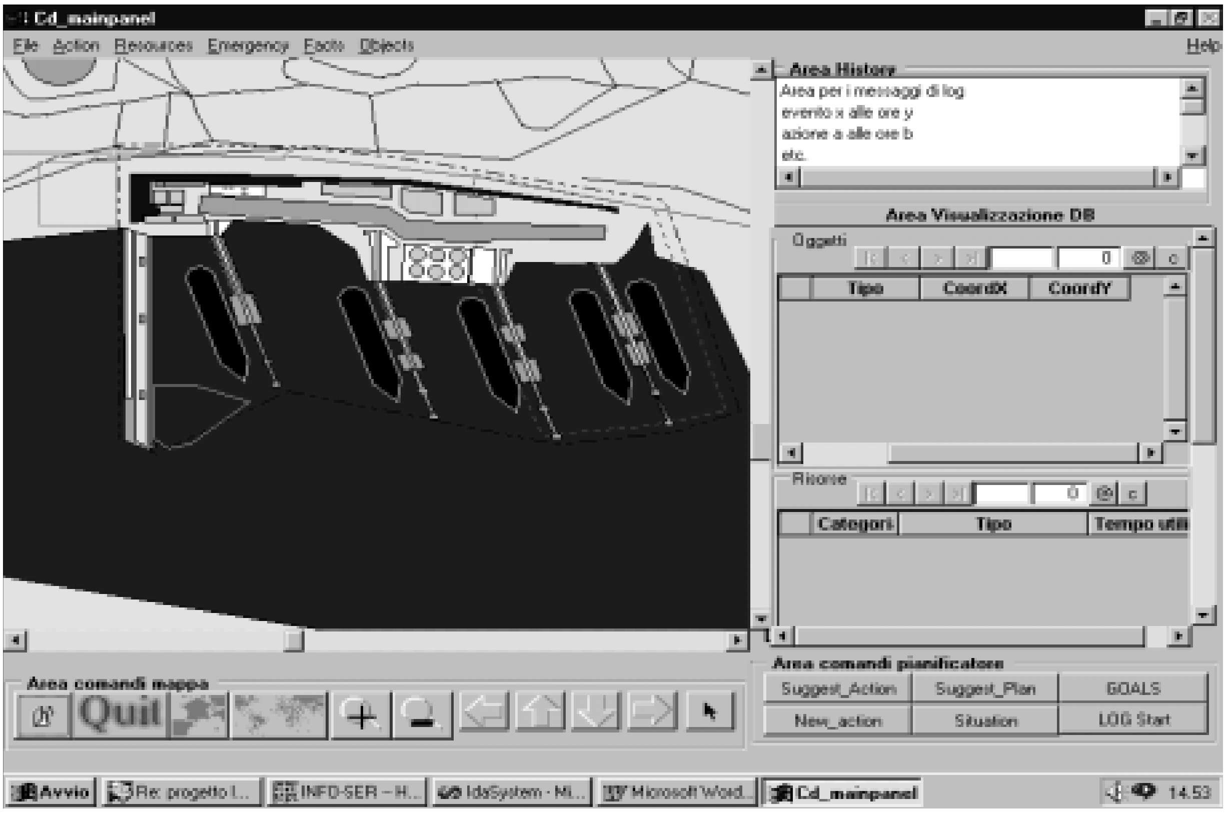

Fig. 6. An example of the IDA interface windows.

5. Conclusions

Summarizing, IDA is a prototypal, intermediate, demo-system focused on the validation of pre-selected properties of a multipurpose IDSS. It has been constructed taking under consideration general framework and paradigms of the IPK architecture. and personoid framework, as a specific abstract intelligent agent. Main attention in this first version of IDA, has been concentrated on the representation of the emergency-domain descriptive knowledge conceptualized as an objects world ontology[1], and on the platform of the Object-Oriented languages. The domain operational knowledge has been conceptualized as the STRIP' like operators. Chosen planning method, considered as meta-knowledge, with implicitly included knowledge meta-preferences, has been realized by Markow Decision Process method. The obtained results should enable to separate structurally preferences and knowledge on the different IDA reasoning levels. This planning knowledge has been supported by CBR method.

In the next version of IDA, we expect to implement a Preference Management Agent with a goal choice, and to separate domain dependent information, knowledge and preferences on the system structural level . However, the realized prototype satisfied our expectations related to the current research phase of the ENEA's MINDES Program.

The adopted ontology models and software solutions should be reused in other more applicative projects focused on active/Intelligent Decision Support Systems in various particular emergency domains.

[1] ontology - a set of axiomatic assumptions and apriori accepted concepts, which are used for the representation and conceptualization of the domain of activity of an intelligent agent (human or artificial). In a cooperation task, ontology is shared between cooperated agents.

Acknowledgment

We need to mention the colleagues which also participated in the developing of the IDA system, they were: F.Ricci, C.Balducelli, L.Senter, R.Sensi and R.Iannucci. We are very grateful for their indispensable contributions.

References

[1] C.Balducelli, G.DiConstanzo, A.M.Gadomski, R.Iannucci, *Requisiti Funzionali per un Sistema Intelligente di Supporto alle Decisioni nel campo della Gestione delle Emergenze*, Enea Report,1997.
[2] F. Ricci, P. Avesani and A. Perini *Cases on Fire: Applying CBR to Emergency Management* Submitted to New Review of Applied Expert Systems 6-98
[3] DARPA projects. AI Magazine, Winter 1998.
[4] M.T.Gervasio, W.Iba & P.Langley, pub/aaai98.ps. Proceedings of the Fifteenth National Conference on Artificial Intelligence *(pp. 721-726), 1998.*
[5] M.des Jardins and M. Wolverton, http://www.erg.sri.com/people/marie/papers/dcp-symp.ps, AAAI Fall Symposium on Distributed, Continual Planning ,AAAI Technical Report FS-98), AAAI Press, 1998.
[6] M. Veloso, /afs/cs/user/mmv/www/papers/arpi-book.ps.gz. Advanced Planning Technology, A. Tate (ed.), AAAI Press, May 1996, pgs. 277-282
[7] ARPA/RL Planning Initiative [see Internet].
[8] C. Boutilier, T. Dean, and S. Hanks. Planning under uncertainty: structural assumptions and computational leverage. In Proceedings, 2nd European Planning Workshop, 1995.
[9] T. Dean, M.Wellman . Planning and Control, Morgan Kaufmann, San Mateo 1991.
[10] L. P. Kaebling. *Learning to achieve goals*. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Chambéry, France, pages 1094—1098, 1993.
[11] M. Puterman, *Markov Decision Processes*, John Wiley and Sons, 1994.
[12] A.Aamodt, E.Plaza, *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches,* AICom. - Artificial Intelligence Communications, IOS Press, Vol. 7: 1, pp. 39-59, 1994.
[13] A.M.Gadomski. TOGA: A Methodological and Conceptual Pattern for modeling of Abstract Intelligent Agent. Proceedings of the "First International Round-Table on Abstract Intelligent Agent". A.M. Gadomski (editor), 25-27 Jan.1993, Rome, Published by ENEA, Feb.1994. As well as: http://wwwerg.casaccia.enea.it/ing/tispi/gadomski/gad-agen.html
[14] A.M.Gadomski, IPK: Information, Preferences, Knowledge. 1997, http://wwwerg.casaccia.enea.it/ing/tispi/gadomski/gad-dict.htm
[15] A.M.Gadomski **,** S. Bologna, G. Di Costanzo. Intelligent Decision Support for Cooperating Emergency Managers: the TOGA based Conceptualization Framework. The Proceedings of "TIEMEC 1995: The International Emergency Management and Engineering Conference", J.D. Sullivan, J.L. Wybo, L. Buisson (Editors), Nice, May, 1995.
[16] C.Balducelli, S. Bologna, G. Di Costanzo, A. M. Gadomski, G. Vicoli. Computer Aided Training for Cooperating Emergency Managers: Some Results of the MUSTER Project. The Proceedings of The MemBrain Conf. TIEMC, J.L.Wybo, M.Terrien, F.Guarnieri (Eds.), Montreal, May, 1996.
[17] C.Balducelli, S. Bologna, G. Di Costanzo, A. M. Gadomski, G. Vicoli. Case Base Reasoning Approach in Industrial Assessment and Management. The Proceedings of International Emergency Management and Engineering Conference on International Aspects of Emergency Management and Environmental Technology, Oslo, 1995.

[18] S.Bologna, A.M. Gadomski. MINDES Program: Managerial Intelligent Node of Decision-support for Emergency Supervision. The Proceedings of the GEMINI (Global Emergency Management International Network Initiative) Meeting, Rome, Sept.,1996.
[19] A.M. Gadomski, G.Di Costanzo. Intelligent Decision Support System for Industrial Accident Management. Proceedings of 8th European Simulation Symposium,SCS, Genoa, Oct.,1996.
[20] A.M. Gadomski, C. Balducelli, S. Bologna, G. DiCostanzo. Integrated Parellel Bottom-up and Top-down Approach to the Development of Agent-based Intelligent DSSs for Emergency Management. Proceedings of the International Emergency Management Society Conf. TIEMS'98: Disaster and Emergency Management. Washington, May 1998.
[21] A.M. Gadomski. MINDES Umbrella Program Managerial Intelligent Node for Decisional Emergency Support, 1997, http://wwwerg.casaccia.enea.it/ing/tispi/gadomski/gad-prog.html
[22] J.F. Allen "Maintaining Knowledge about Temporal Intervals", Communications of the ACM, 26(11), pp. 832-843, November 1983.
[23] J. McCarthy "Circumscription - A form of Non-monotonic Reasoning", Artificial Intelligence, Vol. 13 pg. 27-39, 1980.
[24] R. Reiter "A Logic for Default Reasoning", Artificial Intelligence, Vol. 13 pg. 81-132, 1980.
[25] A. Perini, L..Senter, F.Ricci. L' Agente Pianificatore del sistema di supporto alla gestione di emergenze ambientali ad uso del Consigliere Diretto di un porto petroli. The IDA Project Report, 1998.
[Internet] You can easy find adequate information on the web.

Adam M.Gadomski works as a Senior Research Scientist and R&D Expert. Employed at ENEA. He is an author & co-author about 115 scientific papers. He serve as referee, program committee member and chair at Scientific Conferences (since 1986) related to Intelligent Decision Support Systems, cognitive intelligent agents, emergency and risk management. Before, Assistant Professor and Head of Identification and Diagnostic Lab. in the Inst. of Atomic Energy, Poland. His interest is focused on interdisciplinary R&D projects, meta-reasoning, knowledge-management, high-risk systems specification. A member of different professional associations and working groups.

Sandro Bologna is graduated in Physics at University of Rome. From 1972 up today he has been working at ENEA, where he has covered different positions as Researcher, Head of Research Units, Head of Research Projects at national and international levels. His main research activities deal with the achievement and assessment of software safety and reliability, software quality, operator decision support systems for emergency management, plant control room design and assessment. In this field he has co-authored several publications and books.

Giovanni Di Costanzo, graduated in Nuclear Engineering, is working at ENEA Casaccia as researcher. His job consist in design and development of new decision support systems. He has been author and of several papers for international conferences especially in the field of emergency management, decision support system, intelligent agents. He is also contract professor of Information Technology at the post-graduated school of Sanitary Physics at the University of Rome.

Anna Perini received her doctoral degree in physics from the University of Trento in 1981. She worked in statistical mechanics and computer simulation applied to solid state physics problems, with a fellowship from the Istituto per la Ricerca Scientifica e Tecnologica, Trento (itc-IRST), Italy. Since 1985 she has been a Research Scientist in the Arificial Intelligence Department at IRST, working at the application of AI techniques to environmental decision support systems devoted to agricultural and to forest fire management problems. Her research interests include constraint satisfaction problems and temporal reasoning, case-based reasoning and planning.

Marco Schaerf is Associate Professor of Computer Science at the Department of Computer and Systems Sciences of the Universita' di Roma ``La Sapienza''. His research interests include knowledge representation, belief revision, non-monotonic reasoning, geometric reasoning, database theory, computational complexity and applications of AI to computer graphics. He is a member of IEEE and ACM and founding member of the AI*IA (Italian Association for Artificial Intelligence).