

UN PROTOTIPO DI SISTEMA DI SUPPORTO ALLE DECISIONI PER LA PIANIFICAZIONE AUTOMATICA NELLA GESTIONE DELLE EMERGENZE INDUSTRIALI.

G. Di Costanzo, C. Balducelli, A.M. Gadomski
ENEA Casaccia
Via Anguillarese 301 00060 Roma

Sommario

Il prototipo proposto nella memoria è un prodotto della linea di ricerca perseguita in ENEA da diversi anni nell'indagine e nella realizzazione di strumenti di tipo *attivo* per rispondere efficacemente alle situazioni di emergenza di tipo industriale.

Tramite lo strumento sviluppato, possono essere configurati diversi scenari incidentali e chiedere un supporto nell'ottimizzazione della pianificazione, in termini di occupazione delle risorse e di miglioramento nell'efficacia delle azioni da effettuare. Quindi l'obiettivo dello strumento è di collaborare con i gestori della emergenza nella identificazione e validazione dei piani di intervento disponibili per l'area in esame, piani che spesso debbono essere applicati in modo flessibile e dinamico durante l'emergenza reale.

Il caso di studio presentato si riferisce al dominio di un Porto Petroli, di cui è stato riutilizzato il modello strutturale sviluppato in un precedente progetto di ricerca europeo; lo strumento realizzato è comunque facilmente riconfigurabile ad altri settori e riadattabile a differenti utenti di diverso ruolo manageriale.

Nel progetto, denominato IDA (Intelligent Decision Assistant), sono state impiegate le tecnologie degli *agenti intelligenti*, il *Reinforcement Learning (RL)*, il *Case Base Reasoning (CBR)*, il modello di pianificazione alla STRIPS, l'*object-oriented design*. Queste tecnologie sono state integrate per offrire al potenziale utilizzatore un'interfaccia *user-friendly* di tipo misto grafico e testuale.

L'approccio di tipo *goal-oriented* è volto a fornire all'end-user la generazione automatica dei goals in funzione degli eventi generati sul dominio, intesi come situazioni finali da raggiungere. I goals sono quindi passati all'*agente pianificatore*, che realizza ricerche attraverso lo spazio degli stati-azioni, al fine di ottenere il miglior piano applicabile con le risorse a disposizione. Il lavoro descrive le funzionalità principali del prototipo da un punto di vista dell'utente, mostrando come le differenti finestre dell'interfaccia possano condurre l'utente dalla messa a punto d'una configurazione di incidente fino alla scelta dell'azione migliore disponibile, in quanto adeguata alla situazione corrente dell'emergenza.

1. Introduzione

A causa della complessità crescente degli impianti industriali e delle infrastrutture, il compito di una corretta gestione delle emergenze industriali su grande scala diventa sempre più difficile. Particolarmente per quelle emergenze che richiedono la partecipazione ed il contributo di numerose ed eterogenee organizzazioni.

Di conseguenza è sempre più sentita la necessità di disporre di *strumenti di supporto* che possano effettuare grandi quantità di elaborazioni e facilitino la valutazione della situazione durante l'emergenza.

D'altra parte a questi strumenti, che fanno uso delle tecnologie più avanzate nel campo dell'informatica e delle telecomunicazioni, è richiesta un'alta affidabilità e facilità d'uso. Tali strumenti devono essere usati inizialmente in fase di addestramento e soltanto in seguito, quando l'utente ne ha preso familiarità e confidenza, anche durante le emergenze reali.

Nel campo della gestione di emergenza diventano sempre più importanti i sistemi IDSS (Intelligent Decision Support Systems), per aiutare i decisori a trovare la scelta disponibile migliore in un contesto difficile e con stringenti vincoli temporali. Il responsabile dell'emergenza deve affrontare spesso situazioni in cui, per i vincoli temporali rigorosi e per la complessità dei fenomeni in gioco, non riesce a rispondere efficacemente alla situazione. Il lavoro presentato integra il documento progettuale[1], nel quale sono state principalmente illustrate metodologie e concetti, mentre ora vengono più approfonditamente discusse le funzioni ed analizzati i risultati di progetto.

2. Stato dell'arte.

Il progetto IDA si inserisce nell'ambito della linea di ricerca perseguita da diversi anni in ENEA. In questo ambito sono stati sviluppati una serie di sistemi di supporto alle decisioni per la gestione delle emergenze su vasta scala territoriale, tali da richiedere l'intervento di autorità preposte alla protezione civile della popolazione

Il sistema IDA, pur seguendo questa linea di ricerca, rappresenta una novità sia per l'applicazione di metodologie innovative nel campo della *pianificazione automatica*, sia per la definizione di specifiche funzionali e la formalizzazione della conoscenza con utilizzo di tools di supporto quali UML (Unified Modeling Language). Esso rappresenta un IDSS (Intelligent DSS), che vuole offrire all'utente un supporto più attivo e più vicino al ruolo dei decisori, che lo aiuta a implementare il proprio modello di ragionamento e a ottimizzare l'impiego delle risorse in campo

L'attività del progetto IDA è iniziata nel 1998, ed ha portato, a conclusione di una prima fase, la definizione delle specifiche tecniche e dei requisiti utente[1]. E' stata a tal fine riutilizzata parte delle conoscenze e delle esperienze maturate nell'ambito del progetto europeo *ENVIRONMENT MUSTER* [2].

Come prima applicazione dello strumento di supporto all'emergenza si è quindi utilizzato il dominio del porto petroli di Genova, e come scenari incidentali i possibili incendi ed esplosioni su serbatoi e/o petroliere. Il porto è stato modellato in termini di layout grafico, oggetti a rischio, risorse, ruoli dei coordinatori in campo, piani di intervento etc.

Il sistema sarà di ausilio a gestori delle emergenze di diverso ruolo. Esso consente di configurare e simulare scenari incidentali, al fine di dare la possibilità ai gestori delle emergenze di validare le loro scelte decisionali. Il sistema, pur facendo riferimento a uno specifico dominio applicativo, può essere facilmente riconfigurato per domini applicativi diversi. Ciò comporta però la ridefinizione delle classi utilizzate nel prototipo. Per poter rendere il sistema di tipo del tutto indipendente dal dominio, occorrerà un maggiore sforzo per la definizione di classi di livello più generale.

La tecnologia degli Agenti Intelligenti è stata utilizzata per la definizione dell'architettura complessiva del sistema. Ciò ha consentito la separazione e il raggruppamento di funzionalità comuni in moduli indipendenti ed autonomi comunicanti tra di loro. Come estensione futura si potrebbe pensare di distribuire ciascun agente in nodi differenti, rendendo il sistema distribuito sulla rete Internet. Rendere il sistema di tipo distribuito può simulare meglio la gestione di emergenza come attività di collaborazione tra diversi utenti con diversi ruoli, diversi domini di attività e diverse dislocazioni geografiche.

L'attività del progetto IDA è proseguita in una seconda fase, con la progettazione e lo sviluppo di due prototipi basati su tecnologie diverse. Il primo è stato sviluppato su piattaforma G2 ed ha avuto il ruolo principale di testare i modelli di generazione dei goals e delle azioni sulla base degli eventi incidentali generati. Il secondo, sviluppato in C++ e Microsoft Access ha avuto il compito principale di testare la architettura multi agente, le connessioni alle basi di dati ed i protocolli di comunicazione.

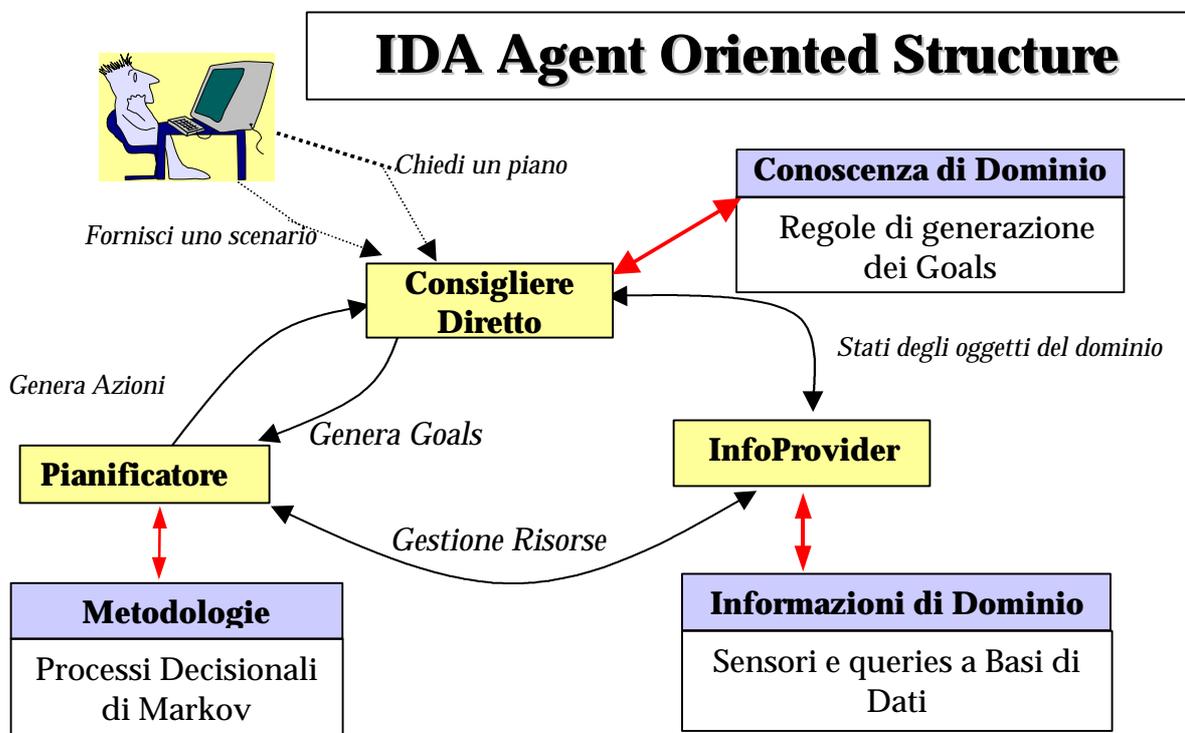


Fig 1 - Architettura ad agenti del sistema IDA

3. Componenti del sistema IDA

Il sistema IDA si compone principalmente di 3 agenti (fig.1), ciascuno dei quali svolge funzioni differenti e ognuno è in grado di scambiare informazioni con gli altri.

I tre agenti sono il Consigliere Diretto (CD), il Pianificatore Automatico(PA) e l'InfoProvider (IP). Ogni agente è preposto a particolari funzioni. CD si preoccupa dell'interazione con l'utente, del quale riceve le richieste, e fornisce i dati; inoltre gestisce direttamente l'I/O con i simulatori.

IP accede direttamente al data base e alla base di conoscenza, e fornisce tali dati agli altri agenti. PA è responsabile dei suggerimenti riguardo alle azioni più opportune da svolgere.

Le funzionalità principali del sistema sono le seguenti. All'inizio l'utente seleziona il database da cui caricare i dati dello scenario iniziale e chiede all'IP di caricare tali dati. L'utente poi *configura* un'emergenza, scegliendola da quelle previste per ogni categoria di oggetto a rischio.

Con la creazione dell'emergenza verrà modificato lo stato dell'oggetto direttamente coinvolto dall'emergenza: se ad esempio si genera un'emergenza di tipo *incendio per un serbatoio*, il suo stato cambierà da *normale* a *incendiato*. Il cambiamento di stato agirà come trigger per il relativo simulatore dell'irraggiamento. I risultati forniti dal simulatore dell'irraggiamento, verranno utilizzati da IP per modificare gli stati degli oggetti coinvolti nell'emergenza, in quanto posizionati nelle immediate vicinanze del serbatoio incendiato.

IP genera quindi i goals che l'utente vuole raggiungere nella situazione di emergenza in esame. Essi, assieme agli stati, vengono analizzati da PA, il cui compito è quello di fornire l'azione o le azioni più efficaci, scegliendole tra quelle attuabili in base alle risorse disponibili. Può inoltre fornire una sequenza di azioni (piano), o prevedere la situazione futura più probabile.

IP offre agli altri due agenti i servizi informativi ed attuativi di lettura e scrittura sul Data Base.

3.1 L'agente InfoProvider.

L'agente *Infopvider* ha come funzione quella di fornire agli altri agenti e all'operatore umano la possibilità di accedere in modo controllato alle informazioni relative al dominio di riferimento. A tale scopo, esso usa, come *repository* un database relazionale contenente tutte le informazioni statiche del dominio. Le classi di interfaccia del package sono state suddivise in 3 parti: *Physical, Abstract e State Domain*. Nel *Physical domain* sono comprese tutte le classi che si riferiscono a rappresentazioni fisiche (petroliere, bacini, serbatoi, fasci tubieri, risorse fisse, risorse mobili); nell'*abstract domain* sono comprese tutte le classi di tipo astratto come azioni, fatti, goals; mentre nello *state domain* viene definita la struttura a stati degli oggetti. In sintesi quindi tale agente fornisce tutti i metodi necessari a creare, leggere, modificare, scrivere o cancellare sia gli oggetti fisici che astratti del dominio. Inoltre provvede, nelle chiamate provenienti dal CD e smistate al PA a realizzare le opportune conversioni per rendere i dati compatibili al formato richiesto dal Pianificatore Automatico stesso

3.2 L'agente Pianificatore.

L'agente *Pianificatore* rappresenta la parte "intelligente" dell'intero sistema in quanto suggerisce le azioni da eseguire in funzione della situazione di emergenza. Esso fa uso di un *framework* basato sui processi decisionali di Markov, che considerano la realtà come composta da oggetti che possono assumere solo stati discreti, e da azioni che fanno modificare gli stati degli oggetti con una certa probabilità, al variare del tempo che assume anch'esso valori discreti o step. Inoltre ogni stato non dipende dalla storia precedente ma solo dalla situazione al momento. A questo schema sono stati applicati i concetti del Reinforcement Learning (RL) [6], che tentano di trovare l'azione ottimale tra quelle che consentono di raggiungere lo stato goal, attraverso la valutazione del massimo di una funzione, la *Q-learning* function. Tale funzione rappresenta la sintesi della bontà di un'azione nel processo di raggiungimento del goal, ed è ottimizzata in funzione di *rewards* attesi, valutati in funzione della distanza dello stato attuale dallo stato goal, e in funzione dei *costi* considerati come *reward* negativi.

Gli operatori STRIPS [7] sono utilizzati per trovare la sequenza di transizioni di stato necessarie per passare dalla situazione iniziale a quella finale: ogni azione è suddivisa in una o più transizioni necessarie per passare dallo stato iniziale a quello finale. Infine è stata prevista, ma non utilizzata nel prototipo, l'applicazione di metodologie Case Base Reasoning (CBR), come alternativa al metodo del RL, per il reperimento dell'azione ottima, scegliendola da una base di casi storici, cioè in situazioni in cui l'esperienza passata ha dato prova di esito positivo nell'applicazione dell'azione considerata.

3.3 L'agente Consigliere Diretto.

L'agente Consigliere Diretto (CD), ha il compito di raccogliere le richieste dell'utente, di smistarle chiamando le opportune funzioni, e di presentare le informazioni ottenute; funge quindi da interfaccia tra l'utente e il resto del sistema. Fa quindi uso di una serie di finestre, anche di tipo grafico, suddivise in aree di visualizzazione e in aree di invio comandi.

Il sistema inizialmente presenterà una finestra iniziale, con un menu e i comandi principali; da questa finestra principale saranno richiamate altre finestre per gestire in maniera più dettagliata le funzionalità specifiche. Le varie finestre o viste sono state generate usando il tool IlogViews, che genera il codice in linguaggio C++ per ogni vista. Per ogni comando associato ad un pulsante sullo schermo, sarà generata una *callback*, cioè una funzione vuota per eseguire il comando, che andrà riempita con il codice della funzione che si vuole richiamare per eseguire il comando.

3.4 L'architettura "agent oriented".

Come modello del dominio di riferimento è stato preso l'area del porto petroli di Genova, in quanto tale dominio è stato già utilizzato nel precedente progetto MUSTER [2], è quindi non è stato necessario reperire la conoscenza dagli esperti del dominio, fase che in genere è molto laboriosa e difficile. Inoltre è stato possibile riutilizzare sia parte del database che i file grafici del layout della zona considerata.

Dal diagramma generale dello schema logico delle classi di alto livello (fig.2) si può vedere che il I livello è costituito dalla classe FIPA. FIPA, che sta per Foundation for Intelligent Physical Agents, è un'associazione che ha l'obiettivo di promuovere lo sviluppo di sistemi basati su agenti, attraverso l'emissione di specifiche sui diversi aspetti della tecnologia [9]. In Ida la classe FIPAgents è stata creata come superclasse, che modella il concetto di agente FIPA in quanto contiene i metodi richiesti dalla specifica per poter costituire una Agent Platform, che è l'architettura che contiene gli agenti. Tali metodi però non sono stati sviluppati nell'implementazione, per cui gli agenti IDA o IDAgents, pur essendo in linea con la specifica FIPA, in realtà

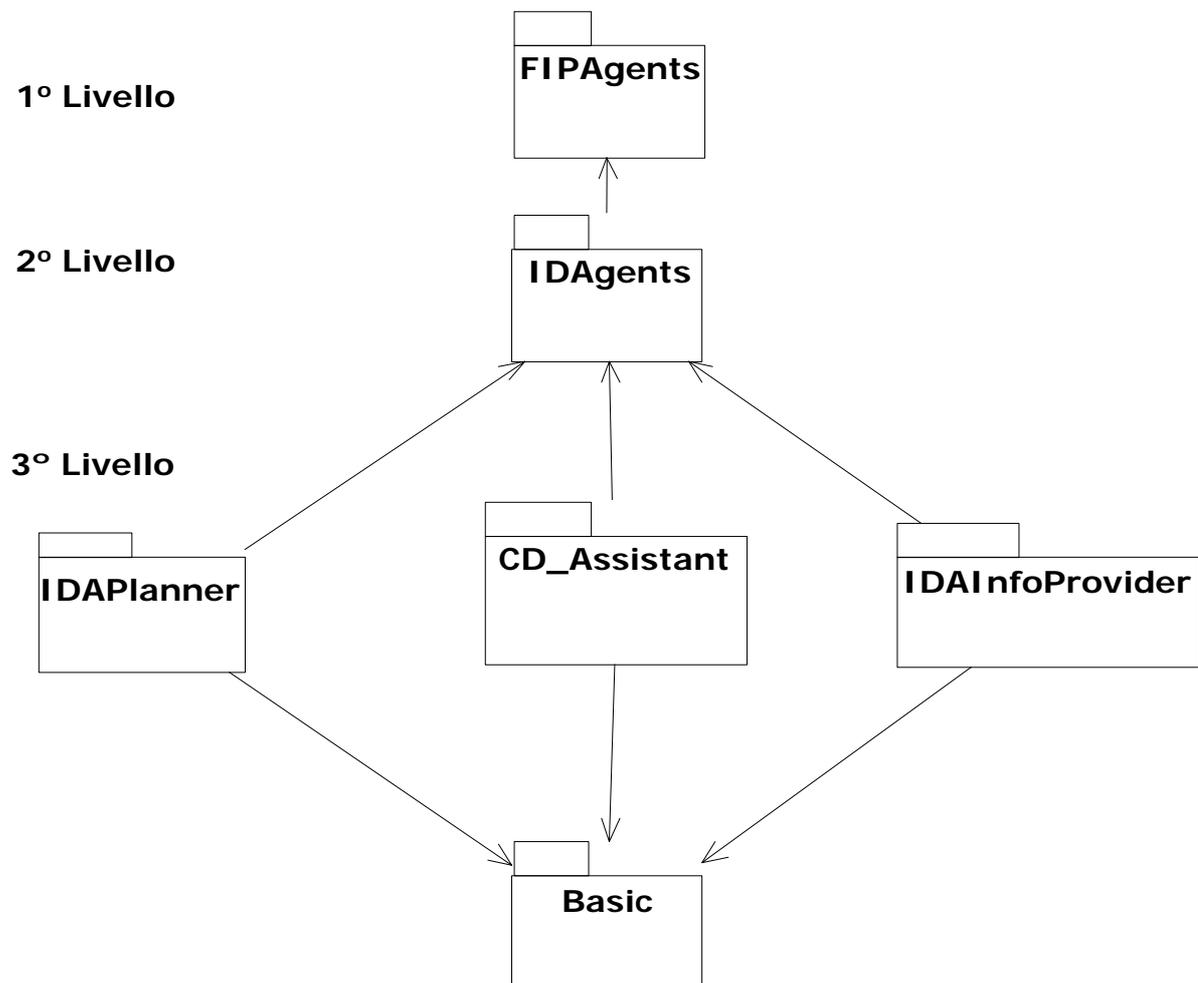


Figura 2 - Vista generale delle classi

non implementano i metodi di tale specifica. In futuro potranno essere sviluppate le piattaforme relative a tali funzionalità FIPA, come pure si potranno introdurre le piattaforme relative ad altri attori di dominio, che potranno interagire tra di loro anche in maniera distribuita, cioè allocati in differenti nodi.

I 3 agenti IDA CDAssistant, IDAInfoProvider e IDAPlanner, modellizzano le competenze dei tre agenti specifici relativi a ciascun gruppo in cui sono state accorpate le relative funzionalità. Lo scambio di informazioni avviene attraverso i metodi di interfaccia messi a disposizione da ogni agente. Ciascun agente è stato definito in package logici secondo la notazione UML, a cui è stato aggiunto il package Basic, contenente le classi di dominio comuni; tali classi modellizzano oggetti utilizzati da tutti e tre gli agenti di dominio.

Il database utilizzato dal sistema IDA contiene i dati relativi allo scenario preso come riferimento per il prototipo. E' stato utilizzato il database ACCESS 97 della Microsoft. L'accesso al Database avviene attraverso i drivers ODBC, cioè utilizzando esclusivamente le chiamate dello standard Open Data Base Connectivity. Tale scelta rende possibile l'utilizzo di altri database commerciali che seguono lo stesso standard al posto di ACCESS.

3.5 I simulatori

Al fine di valutare gli effetti nell'area interessata dagli incendi (*incendio su serbatoio, bacino o petroliera*), sono stati utilizzati codici di previsione degli irraggiamenti termici. Questi codici calcolano l'intensità della radiazione termica prodotta dagli incendi sui vari oggetti coinvolti. I codici assumono che la sezione della fiamma coincida con la sezione degli oggetti (serbatoi, bacini etc.) su cui hanno luogo. L'altezza della fiamma viene stimata mediante una correlazione proposta in letteratura da Thomas; per calcolare l'inclinazione della fiamma dovuta alla velocità del vento viene utilizzata un'equazione proposta dall'American Gas Association (A.G.A.), mentre per il calcolo del diametro apparente di fiamma (diametro della fiamma soggetta a vento), si ricorre ad un'equazione di Moorhouse. L'intensità della radiazione termica viene predetta mediante il modello del corpo solido emittente (C.S.E.).

Il codice di simulazione utilizza due file di input, il primo contiene il nome della sostanza e le caratteristiche geometriche dell'oggetto incendiato; il secondo contiene i dati relativi alle condizioni meteorologiche: temperatura, velocità del vento, umidità relativa, categoria di stabilità.

Come dati di output il codice fornisce l'andamento dell'intensità della radiazione termica in funzione della distanza del ricettore dall'oggetto incendiato a varie altezze rispetto al suolo.

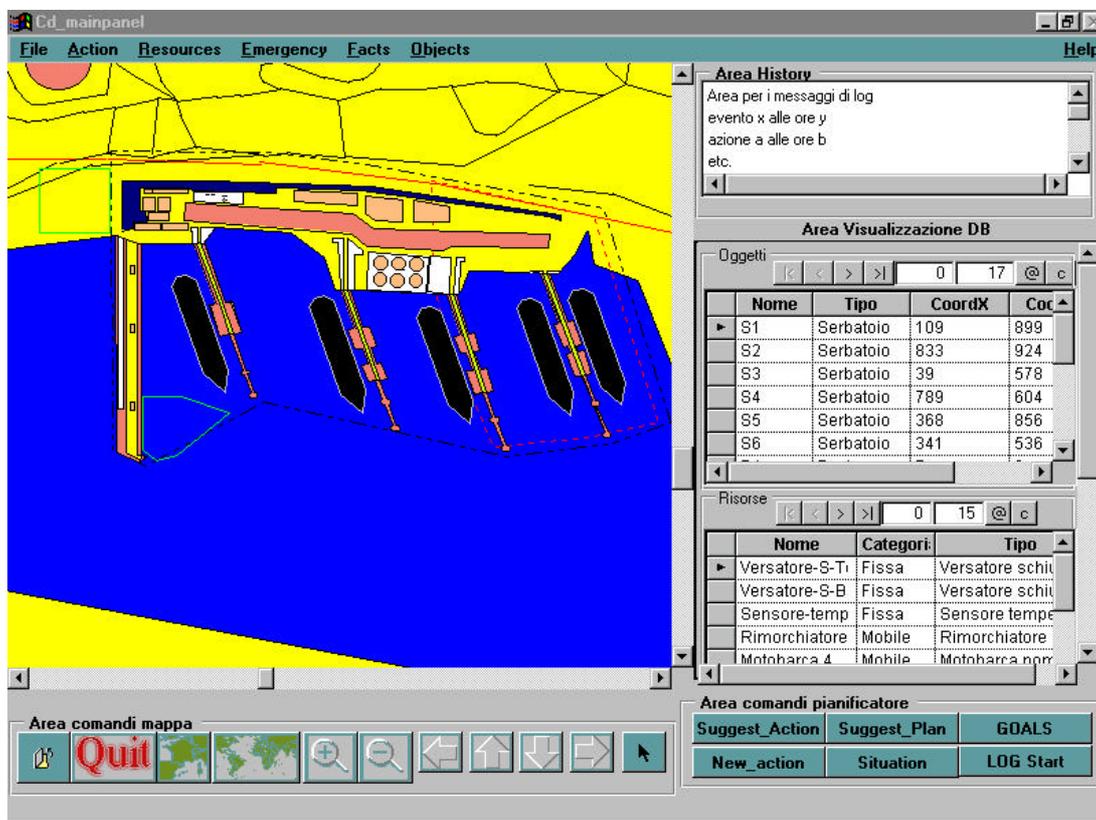


Fig. 3 - Finestra del Menu iniziale

4. L'interfaccia utente.

Una sequenza di viste permette all'utente di inserire facilmente i dati di input richiesti per la situazione di emergenza da analizzare. La gestione della interfaccia segue un approccio *goal-driven* rispetto al più tradizionale paradigma *menu-driven*.

Il menu principale è suddiviso in quattro aree (fig.3). La prima riguarda l'area menu, che comprendono anche sottomenu a tendina. Nell'area in alto a sinistra è visualizzata l'area geografica in esame (2^a area), ed è la parte maggiore in termini di occupazione dello schermo. I comandi in basso permettono di modificare l'immagine grafica visualizzata. In alto a destra è la 3^a area, di cui l'*Area History* mostra i messaggi relativi a tutte le operazioni avvenute nella sessione (non implementato nel prototipo), mentre l'*Area Visualizzazione DB* è dedicata alla visualizzazione di dati provenienti direttamente dal database. Nella 4^a area, in basso a destra, sono riportati, in forma di bottoni rettangolari, i comandi per richiamare le funzioni del pianificatore, come *situation*, *goals*, *suggest_actions*, *suggest_plan*, *new_action*, *log_start*.

4.1 Situation View

La vista *Situation* consente di valutare la situazione degli oggetti a rischio del dominio, visualizzando per ognuno di essi le variabili di stato, che possono variare o per effetto dell'emergenza generata nella vista relativa, o per effetto del calcolo di un simulatore, come oggetto coinvolto nelle fasce di irraggiamento. L'utente inoltre, può variare direttamente il valore di una variabile di stato, se lo ritiene opportuno, per creare una nuova situazione di emergenza da analizzare, al fine di valutare la risposta del sistema sia sotto forma di goals generati che di azioni suggerite dal pianificatore. Una volta modificato tale valore, con il comando *apply*, tale variazione sarà resa operativa.

La vista *Situation* consente di visualizzare la situazione degli oggetti del dominio del porto petroli. Viene presentata una tabella con i valori degli attributi, diversi a seconda del tipo di oggetto.

4.2 Emergency View

La vista *Emergency* consente di creare un'emergenza scegliendola tra quelle previste e associandola a un oggetto a rischio, scegliendolo tra quelli soggetti a quel tipo di emergenza. Ad esempio per eventi incidentali "incendio della corona di serbatoio", sono proposti come oggetti da associare all'evento, solo quelli del tipo serbatoio, e così per gli altri eventi incidentali previsti. Scelto l'evento e il nome dell'oggetto su cui far avvenire l'evento, il sistema creerà l'emergenza modificando la o le variabili di stato (relative all'evento) dell'oggetto colpito da evento incidentale.

Le variazioni delle variabili di stato dovute alle diverse emergenze considerate avvengono secondo la seguente tabella.

Nome evento	Tipo	Coinvolge	Attributo	Val.inizi	Val.final
Incendio della corona di serbatoio	Incendio	Serbatoio	fireTop	0	1
Incendio nel bacino di serbatoio	Incendio	Serbatoio	irradied	0-4	5
Incendio del tetto di serbatoio	Incendio	Serbatoio	irradied	0-4	5
Avaria impianti fissi schiumatura	Avaria	Serbatoio	fireRisk	0-1	1
Sversamento su bacino	Sversamento	Bacino	spilled	0	1
Incendio su bacino	Incendio	Bacino	irradied	0-4	5
Incendio su petroliera	Incendio	Petroliera	irradied	0-4	5
Incendio su bacino a mare	Incendio	Bacino	irradied	0-4	5
Avaria impianti fissi schiumatura	Avaria	Bacino	fireRisk	0-1	1
Avaria impianti fissi raffreddamento	Avaria	Serbatoio	fireRisk	0-1	1
Avaria impianti fissi raffreddamento	Avaria	Racks	fireRisk	0-1	1
Avaria impianti fissi raffreddamento	Avaria	Petroliera	fireRisk	0-1	1

Tab. 1 - Range dei valori delle variabili di stato

Il simulatore inserito nel prototipo è un simulatore di incendio, che calcola il raggio delle fasce di irraggiamento circolari relative ai valori di irraggiamento(fig 4):

- Fascia 1 rossa : livello di irraggiamento superiore a 12 Kw/m²
- Fascia 2 rosa : livello di irraggiamento compreso tra 7 e 12 Kw/m²
- Fascia 3 gialla : livello di irraggiamento compreso tra 5 e 7 Kw/m²
- Fascia 4 azzurra : livello di irraggiamento compreso tra 3 e 5 Kw/m²

Il simulatore si attiva automaticamente all'atto dell'inserimento di una nuova emergenza tramite la vista *Emergency*. I dati relativi all'oggetto incendiato, che vengono considerati dal simulatore sono il nome, la posizione, il livello di greggio contenuto, il diametro, l'altezza. I dati meteorologici sono la pressione atmosferica, la temperatura, la velocità del vento, l'umidità relativa e categoria di stabilità. dal. Il valore delle

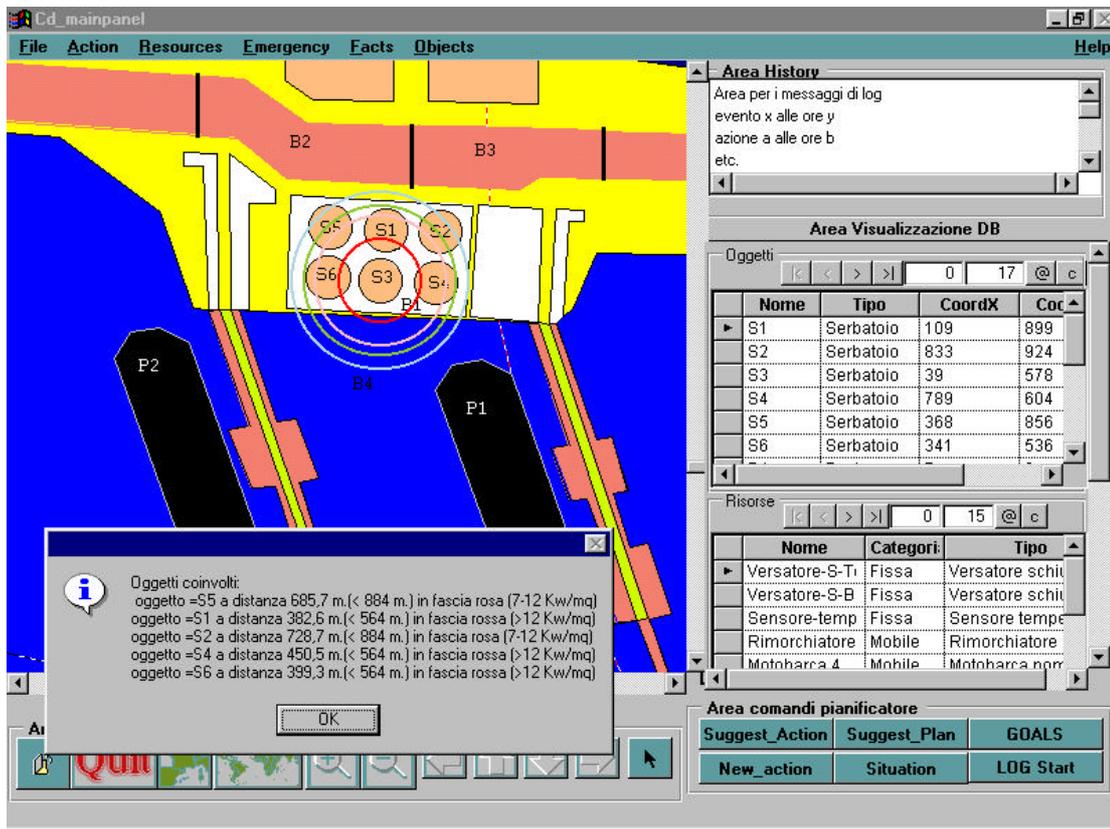


Fig. 4 - Visualizzazione fasce di irraggiamento

variabili di stato degli oggetti coinvolti sarà settata in accordo all'appartenenza dell'oggetto nelle varie fasce di irraggiamento. I codici di simulazione utilizzati è stato sviluppato al di fuori del progetto IDA. E' altrettanto semplice introdurre in maniera modulare nuovi simulatori eventualmente disponibili, qualora fosse necessario per l'analisi dei fenomeni in gioco durante l'emergenza.

4.3 Goals View

La vista *Goals* permette di generare i goals prima che questi siano passati al pianificatore, insieme con lo stato corrente del dominio, al fine di ottenere le azioni da fare.

I goal possono essere *elementari*, cioè costituiti da un unico elemento nella tabella dei goal, oppure possono essere *goal composti* da un insieme di goal elementari: in questo caso i parametri *idgoal* e *nomegoal* saranno uguali per tutti i goal elementari appartenenti alla stesso goal composto. Questo perché per il pianificatore in generale il goal è composto da una lista di stati da raggiungere.

I goals possono essere ottenuti in 3 modalità differenti. Si è lasciata la massima libertà perché la tecnica di scelta dei goals è da investigare ulteriormente anche al fine di arrivare ad una scelta del tutto automatica che l'agente CD dovrà effettuare in ogni diversa situazione di emergenza.

- 1^a modalità : **genera da default**. In questo caso i goals sono generati a partire dai dati di default, cioè dai dati dello scenario iniziale; quindi per ogni stato di qualunque oggetto diverso da quello iniziale sarà generato un goal.

- 2^a modalità: **genera da regole**. In questo caso i goals saranno generati non a partire dai valori iniziali delle variabili di stato, ma da regole introdotte sulla base dell'esperienza di un esperto del dominio.
- 3^a modalità : **direttamente dall'utente**. L'utente può sia modificare o cancellare i goals generati nelle modalità precedenti, sia inserirne di nuovi a seconda delle sue personali valutazioni della situazione di emergenza in atto.

Una volta generati, i goals possono essere memorizzati con il comando *apply*, per poter essere poi trasmessi al pianificatore. Il comando *carica*, permette di caricare in tabella i dati relativi ai goals presenti in memoria; mentre il comando *ordina*, consente di ordinare i goals in tabella in base alla strength, cioè alla diversa priorità.

4.4 Action View

Il pannello di visualizzazione delle azioni (fig. 5) informa l'utente sulla azione o sulla sequenza ottimale di azioni suggerita dal Pianificatore nello scenario incidentale corrente. In un primo tempo vengono visualizzati i goals generati sul Goal View panel. Per ognuno di essi l'utente può spedire al Pianificatore il comando *suggest_action*, che rende disponibile la migliore azione suggerita.

Allo stesso tempo, vengono fornite su una tabella azioni la descrizione delle azioni suggerite, i tipi di oggetto sui quali le azioni debbono essere effettuate, i tipi di risorse da impegnare, la durata in minuti delle azioni stesse, ed i costi necessari per ognuna di esse.

Allo stesso modo può essere inviato il comando *seq_of_actions* a cui il sistema risponde fornendo la sequenza suggerita di azioni.



Fig. 5 Actions View

Ogni volta che una azione viene eseguita dall'operatore (comando *dispatch_action*), il sistema produce una sequenza di *Action Transitions*. Una stessa azione può produrre una o più transizioni. Le transizioni operano modificazioni nello spazio degli stati degli oggetti, cambiando le relative variabili di stato: esse sono definite in termini di Operatori Probabilistici dello spazio degli stati e sono una estensione dei classici operatori STRIPS. Sono generalmente formate dalla tripla: *Prelist*, *Dellist*, *Addlist*. *Prelist* rappresenta la condizione iniziale degli stati per i quali la transizione può avere luogo, *Dellist* sono le condizioni degli stati che verranno eliminate dalla transizione stessa mentre *Addlist* sono le condizioni degli stati che debbono essere create dalla transizione. Si può inoltre definire una *probabilità* che la transizione avvenga come conseguenza della esecuzione di una azione; all'inizio questa probabilità viene fissata uguale per tutte le transizioni, mentre in seguito può essere modificata sulla base della valutazione del comportamento e dell'efficacia delle risposte ottenute dal sistema.

5. Risultati del Progetto e Sviluppi Futuri

Il progetto IDA, pur non essendo terminato definitivamente (manca da completare la parte di test su casi specifici) ha rappresentato una fase di ricerca innovativa per diversi aspetti.

L'utilizzo del tool Rational Rose 98, che si sta affermando sempre più come uno standard per l'analisi e sviluppo di sistemi complessi, riducendo fortemente i costi e facilitando sia il riuso, che le modifiche in corso d'opera. Il tool ha reso possibile la collaborazione di 3 diversi team di analisti e/o sviluppatori, che hanno potuto prima analizzare insieme il sistema da realizzare (fase che ha richiesto notevole tempo, come d'altra parte avviene normalmente); una volta realizzata una prima architettura, che definisce la decomposizione delle funzionalità, e identificati i requisiti, successive modifiche o aggiunte sono state possibili facilmente in quanto inserite in un quadro architettonico chiaro e predefinito.

Anche l'adesione allo standard FIPA, seppure non dettagliato sufficientemente, ha consentito la disponibilità di un framework di riferimento per il progetto foriero di sviluppi interessanti ed innovativi per quanto riguarda la possibilità di estensioni sia a piattaforme multi-agents, che a sistemi distribuiti in senso generale.

Entrando più in dettaglio nel sistema, il risultato maggiore è la possibilità di avere in output dalla vista *Actions*, sia la singola azione che una loro sequenza da effettuare nelle particolari fasi dell'emergenza. Tale risultato è stato ottenuto grazie all'impiego sia di tecnologie in parte ancora in fase di studio, come il *Reinforcement Learning*, sia di metodi più affermati come il pianificatore STRIPS o il CBR. Un maggiore sforzo di ricerca è necessario per la valutazione del metodo di integrazione proposto.

Dal punto di vista dell'utente, è necessario semplificare sia l'introduzione del set di azioni possibili (attualmente inserite tramite file), che la loro modifica durante l'emergenza. Inoltre l'utente dovrebbe poter seguire in maniera più trasparente il metodo di ragionamento seguito dal pianificatore, attraverso una spiegazione del link tra goal e sequenza di stati da percorrere con le azioni suggerite. Anche l'impiego e ottimizzazione delle risorse può essere migliorato attraverso una migliore interfaccia e attraverso un legame più diretto con le azioni disponibili. Ad esempio, quando una risorsa non è più disponibile, l'azione collegata dovrebbe automaticamente disabilitarsi.

Inoltre il sistema potrebbe essere reso più indipendente dal dominio di riferimento generalizzando la definizione delle classi. Ciò naturalmente renderebbe necessario riprendere la fase di analisi e riconsiderare le emergenze di tipo non solo portuale ma di vari tipi. Questo porterebbe all'introduzione di un ulteriore livello nella gerarchia delle classi, di tipo più generale, da cui deriverebbero le classi più specializzate per i singoli domini. La possibilità di individuare per queste classi metodi più astratti, renderebbe poi più facile l'adattamento al particolare dominio di emergenza considerato.

L'impiego di tecnologie object-oriented insieme con i tools ad alto livello della ILOG per lo sviluppo dell'interfaccia utente, ha certamente semplificato tale fase, in quanto ha reso il processo di inserimento di nuove viste o loro modifiche, piuttosto semplice. Inoltre nel prototipo non è stata utilizzata tutta la potenzialità del prodotto Ilog Views che consente l'inserimento di file grafici di diverso tipo, e l'animazione di oggetti grafici su di essi. Queste animazioni possono essere molto utili al gestore delle emergenze per poter seguire sullo schermo, con buona approssimazione realistica, lo svolgersi degli eventi e le conseguenze delle azioni.

6. Referenze

- [1] **Requisiti Funzionali e Specifiche di un Sistema Intelligente di Supporto alle Decisioni nel campo della Gestione delle Emergenze IDA (Intelligent Decision Advisor)** C.Balducelli,G. Di Costanzo, A.M. Gadomski, R. Iannucci .*Rapporto Interno ENEA EIIIE98024*.
- [2] **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*
- [3] **A Prototype of an Active Decision Support System for Automatic Planning Support in Emergency Management.** C. Balducelli, G. Di Costanzo, A. M. Gadomski. *Proceedings of TIEMS2000*
- [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*.
www-anw.cs.umass.edu/~rich/book/3/node1.html
- [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] **Visual Modeling with Rational Rose and UML.** Terry Quatrani. *Addison Wesley 1998*.
- [10] **Rational Rose 98.** User's Manuals. Rational Software Corporation. 1998
- [11] **FIPA Web site :** <http://www.fipa.org/>
- [12] **An Approach to the Intelligent Decision Advisor (IDA) for Emergency Managers** Adam M.Gadomski, S. Bologna, G. Di Costanzo, A. Perini, M. Schaerf. *Proceedings of TIEMS99. The Sixth Annual Conference of The International Emergency Management Society ,Delft, Netherlands, June 8-11, 1999*