JAISCR, 2012, Vol. 2, No. 2, pp. 125 – 132

A LAYERED MULTIAGENT DECISION SUPPORT SYSTEM FOR CRISIS MANAGEMENT

Fahem Kebair
Optimization Strategies and Intelligent Computer Science Laboratory,
Higher Institute of Management,
41, Rue de la Liberté, Cité Bouchoucha 2000, Le Bardo, Tunis, Tunisia

Abstract

Decision Support Systems are powerful tools to help support making decisions. However, they are known to be customized for a specific purpose and can rarely be reused. Moreover, they do not support complex situations sufficiently. Our work addresses this challenge and consists in building a DSS that aims to help emergency managers to manage cases of crisis. The DSS is designed to be flexible and adaptive, so that it may be applied on different subjects of studies and whose behaviour may change with the change of its environment. We endowed it therefore with a multiagent layered core whose role is to represent dynamically and in real time the current situation, to characterize it and to compare it with past known scenarios. The final result of the DSS will help decision-makers to analyze the current crisis and its possible evolution. The RoboCupRescue simulation system is chosen as a test bed to illustrate and to test this approach.

1 Introduction

Risk and crisis management are one of the most complex problems raised by the scientific community currently. The efforts devoted to this research area consists of changing the classical disaster management methods by using new means. This is already realized and accepted as a high priority task by many organizations, governments and companies in Europe and all over the world [1].

We are interested in our works in the risk detection and management in emergency situations. Decision Support Systems (DSSs) are an appropriate solution for this kind of problem, since they are able to complete the knowledge of the decision makers and to support them to deal with particular problems. However, DSSs are well known to be customized for a specific purpose and can rarely be reused. Moreover, they only support circumstances which lie in the known and knowable spaces and do not support complex situations sufficiently [4]. Thereby, our main goal is to develop a system that must be sufficiently independent of the treated problem in order to be adjusted easily to different cases of studies. Moreover, we propose an original approach based on a mechanism of perception, representation, characterization and assessment that enables the system to operate autonomously and to adapt its behaviour according to the change of its environment. We use the multiagent systems (MAS) technology to achieve this objective. In fact, intelligent agents [16] are able to self-perform actions and to interact with other agents and their environment in order to carry out some objectives and to react to changes they perceive by adapting their behaviours.

The proposed system is made up of several agent organizations whose core is operating on three levels. A first level, in which a factual agent organization has as role to represent dynamically and in real time the evolution of the current situation. This step is fundamental in the final assessment of the situation. Indeed, the system creates its own representation of the environment state in order to extract the significant facts that may reveal the existence of risks. It compares therefore the current situation with previous known ones stored as scenarios. That way, the system may have a generic and adaptive mechanism and may learn during its functioning.

In our approach, it is necessary to test the MAS on several case studies to illustrate it and to validate it. The work presented here is addressed to the RoboCupRescue Simulation System (RCRSS) [8][10]. We provide here a brief description of this application and we present and discuss the related experimentations.

2 DSS Role and Design

2.1 DSS Definition and Role

DSSs are interactive, computer-based systems that aid users in judgment and choice activities. They provide data storage and retrieval but enhance the traditional information access and retrieval functions with support for model building and model-based reasoning. They support framing, modelling, and problem solving [2]. More precisely, the purposes of a DSS are the following [6]:

- Supplementing the decision maker.
- Allowing better intelligence, design, or choice.
- Facilitating problem solving.
- Providing aid for non structured decisions.
- Managing knowledge.

In our context, the DSS could be used either to prevent a crisis or to deal with it. In both cases, the main internal aim of the system is to detect a crucial event. From the system point of view, detecting a crisis implies representing it, characterizing it and comparing it with other crises, permanently stored in scenarios. The result of this comparison is provided to the user as the answer of the global system. The system chooses to highlight parts of scenarios similar to the current situation. The information thus obtained will help decision-makers to analyze the current crisis and its possible evolutions.

The DSS has to evaluate a dynamic situation. Monitoring the situation generates dynamic parameters which vary all the time. The system must be dynamic in order to be able to take into account the changes in the description of the evolving situation. This requires a system able to be reconfigured when necessary, thus benefiting from a sufficiently flexible and adaptive architecture. Complexity and dynamics of the situation to be treated, lead us to choose MAS paradigm for its modelling.

The observed situation generally contains a great number of dynamic parameters, that is to say parameters whose value change over time. Systems allowing the management of such situations must be dynamic in order to be able to handle these evolutions. As a consequence, to design these systems, a flexible and adaptive architecture is needed. Such a system must not only represent the observed situation, but must also make it possible to evaluate it. Evaluating the situation can be performed by anticipating its possible consequences. This can be carried out using previous situations whose consequences are used relying on the following hypothesis: if situation A looks like situation B, the consequences of situation A ought to be similar to those of situation B. This mechanism is similar to a Case-Based Reasoning (CBR) [9] which is a methodology based on the re-using of past experiments for solving new problems.

2.2 Role of the DSS in Crisis Management

A crisis may be defined as a major unplanned occurrence [7] with a potential negative outcome [12]. We are interested, in our research, in natural and technological crisis. Crisis management (also known as emergency management or emergency response) is a dynamic process that begins well before the occur of the critical event and continues over its conclusion. The process involves a proactive, responsive and reflective component. Each stage of a crisis poses challenges for managers and decision makers and requires a different approach depending on the phase in question. This process is complex and exceeds widely the human abilities. Thus, DSSs may help to manage this process. Indeed, the DSS we present here must insure the following functionalities:

– Evaluation of the current situation: the system must detect/recognize an abnormal event.

– Evaluation/Prediction of the consequences: the system must assess the event by identifying the possible consequences.

– Planning interventions: the system must help the emergency responders in planning their interventions thanks to an actions plan (or procedures) that must be the most appropriate to the situation.

2.3 DSS Architecture and Design

Fig. 1 shows the global architecture of this DSS. The user interfaces allow the dialogue between all users authorized to access the DSS and its core. This interface also displays the final results provided by the core. The latter also needs to access outside distributed information systems (DIS). The specific information on the domain stores persistent data such as the ontology and scenarios. Proximity measure is also an aspect of this specific information.

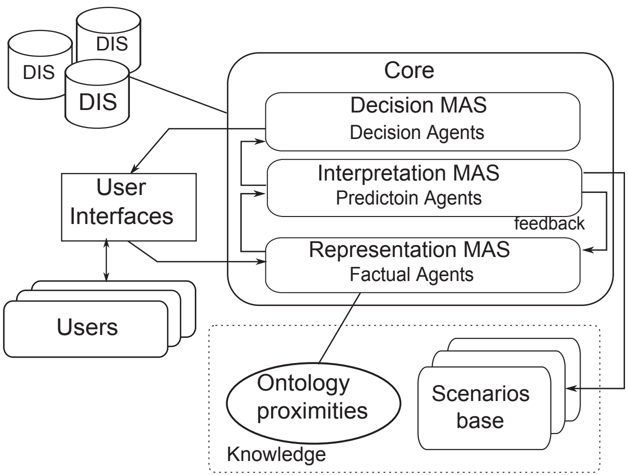

Figure 1. Whole DSS architecture

2.4 Dss Application: RoboCupRescue Case Study

We chose the RCRSS in order to apply the proposed approach. The RCRSS is an agent-based simulator which intends to reenact the rescue mission problem in real world. It reproduces an earthquake scenario which includes various kinds of incidents as the traffic after earthquake, buried civilians, road blockage, fire accidents, etc. A set of heterogeneous agents (RCR agents) coexist in the disaster space: rescue agents that are fire brigades, ambulance teams and police forces, and civilian agents. A model of the RCR disaster space and the properties of its components, and the RCR agents are detailed in [11]. We use this model in order to extract knowledge and to formalize information.

As in real case, RCR agents play the actors role here, they send their perceived information to the DSS in order to get a sequence of actions to perform. The DSS builds, based on these information, an overall knowledge which allows the evaluation of the whole situation. We defined two kinds of factual agents for this case study:

– Factual agents describing phenomena (fires, injuries, building collapses,...).

– Factual agents describing the states and the events related to the RCR agents. More precisely, these agents manage the evolution of the states and the actions of the RCR agents.

We focus, in this paper, on the fires incidents and their related facts. The work concerns therefore the perception and the representation of both the fires propagation and the behaviours of the fire brigades. Hence, we present here the implementation of the fire factual agents and the fire brigade factual agents.

3 Perception and Representation MAS

3.1 Factual Semantic Features

The system receives and analyses permanently elementary information coming from the environment. These information are presented in the shape of a Factual Semantic Feature (FSF). The noun given to this message content provides an explication to our approach: we stress observed and punctual elements that are the facts. A fact is a knowledge or information based on real occurrences; it may be an event, an action, a phenomenon, etc.

Each FSF describes a fact and consequently a state change of an observed object issued from the environment. This may be modelled as a state-transition diagram. A transition represents an instantaneous transit from a state to another. It is triggered by an event (message), followed by the performance of one or several actions in the new state. The observation of this change is sent to the system in the shape of an FSF. An FSF has a generic structure which is composed of $< key, (qualifier, value)^{+} >$. The key is a unique identifier related to the observed object to which are associated some characteristics described by qualifiers and their related values. We associated also time and spatial values to an FSF to describe the temporal and the spatial aspects of the observation. An example of an FSF is the following: <fire♯1, intensity, strong, localisation, building♯12, time, 10:00pm>. This fact describes a strong fire, located in building♯12 and which is observed at 10:00pm.

Dedicated to the observed environment, the ontology serves as a mean for establishing a conceptually concise basis for communicating knowledge. The vector of this communication is the FSF, with the taxonomy structuring and defining the meaning of the observed facts. The measure functions use the ontology to compare FSFs. This comparison is coupled with temporal and spatial data carried by the FSF to obtain a normalised proximity measure in [-1, 1]. A value of -1 means a complete opposition between the two compared FSFs. A value of 0 means neutral or not comparable. A value of 1 means identity between the two FSFs and any other value in this interval means a semantic connexion in the range from opposite to identical.

3.2 Factual Agents

The system is permanently fed by information describing the state of the environment. These information is handled thereafter by agents. The system needs knowledge about the environment such as the ontologies of the domain and proximity measures. The representation layer is made of factual agents whose main aim is to represent the current situation dynamically. A Factual Agent (FA) is a reactive and a proactive agent according to Wooldridge in [15]. Each FA reflects a partial part of the observed situation. Fig. 2 shows the internal structure of an FA. Each agent has:

- An FSF that represents its knowledge.
- An Augmented Transition Newtork (ATN) of four states that describe its behaviour. ATN transitions carries conditions and actions, and are specific to the FA type.
- Specific indicators that reflect its dynamics.
- An Acquaintances Network (AN).

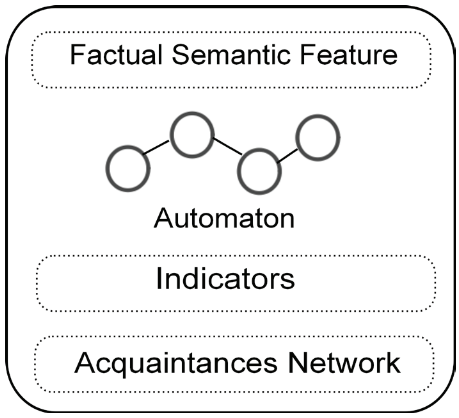

Figure 2. Internal structure of a factual agent

When a given FSF reaches the system, either an FA–whose FSF has the same key–exists and this FA will replace its internal FSF with the new one, or a new FA is created. Thus, at any time, the whole population of FAs with their embedded FSFs reflects the current view of the situation.

FAs represent the dynamic evolution of the situation thanks to their internal indicators. These indicators must reflect as much as possible the reality, their definition depend therefore on the treated application. We defined two indicators for the RCR case study:

- *Action Indicator (AI):* represents the potential and the efficiency to solve problems for FAs related to RCR agents, and the damage and the hazard degree for FAs related to phenomena.
- *Plausibility Indicator (PI):* means the ability to discover new problems in the disaster space for factual agents related to RCR agents, and the solving probability and the worsening impediment for FAs related to phenomena. For phenomena factual agents, PI means the solving probability and the worsening impediment of a phenomenon.

The acquaintances network of every FA is generic and dynamically constructed. It contains all the other FAs which are semantically connected (proximity not equal to 0). The FA of a given acquaintances network is close when its proximity is positive and is opposite when its proximity is negative. The evolution of the FA is managed by strengthening and weakening mechanisms. Internal indicators reflect this evolution.

3.3 Application on a Fire Scenario

We discuss here two different fire scenarios in order to illustrate the functioning and the adapting of the representation MAS. In the first one, we have inactivated the police forces who has as goal to clear the blocked roads. This made therefore more difficult the moving of the fire brigades to achieve the fires. In the second scenario, we let police forces do their work, thus the roads are less blocked and the accessibility to the fires is much easier.

In Fig. 3, the green chart illustrates the activities number of the representation MAS during the whole scenario. The activities include the state changes, the indicator values variations and the messages sent by the FAs. The red area represents the fire spreading, expressed by the number of the perceived fires over time. In this scenario the fire brigades could not reach the fires, they still moving to reach the fires without success. This is expressed by several oscillations of the FAs activities.

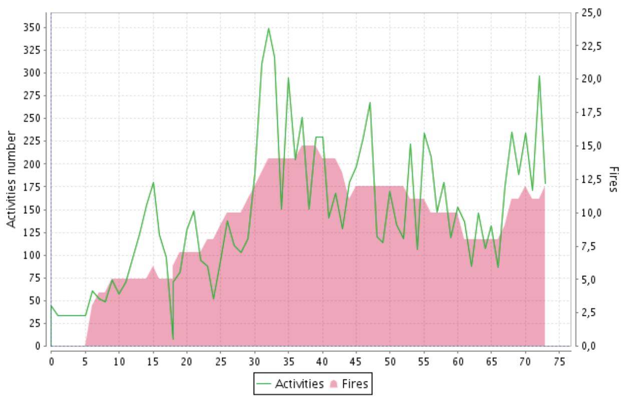

Figure 3. Fire spreading and FA activities without police forces aid

Fig. 4 shows the FA activities and the fire spreading in the second scenario. We note here a smaller area of fires compared to the previous scenario. Moreover, we have less oscillations and a higher number of the FA activities. This reflects the system stability on the one hand and the efficiency of the fire brigades on the other hand. Indeed, the system reacts in a moderate way at the beginning of the scenario, in which the fires are isolated. By dint of receiving more and more information, describing the fires propagation and the mobilization of the fire brigades, the factual agents react by intensifying their activities. The values and the oscillations of the activities number depend strongly on the behaviours of the fire brigade agents. Indeed, the activities number grows when the fire brigades are fighting fires. Inversely, it drops when the fire brigades are potentially far from fires or are searching new ones. This explains the perfect synchronization between the activities evolution and the fire spreading. To summarize, we can say that there is an activities peak when there is a high level of risk and emergency, due to the rapid spreading of fires and the struggle of the fire brigades that try to restore the situation.

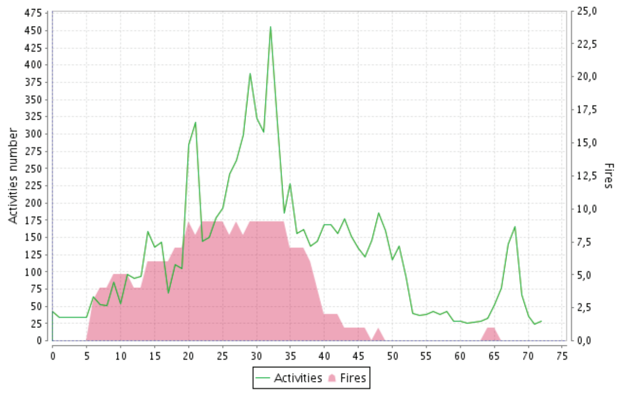

Figure 4. Fire spreading and FA activities with police forces aid

At the end of the scenario. the system knows an evident bending result of the fires extinction. The factual agents become less meaningful since there are not important facts related to fires that come stimulating them. However, the system still in a warning state in order to alert every notable change in the environment. We may notice this at the 63rd second of the simulation, when a fire reappears suddenly. The system reacts immediately to this fact and resumes its activities, then it becomes again stable after the fire were put out.

4 Characterization and Interpretation MAS

We used FAs in the representation situation level to reflect the dynamic change of the situation and to let emerge, from this view, agent subsets. The analysis of these subsets is based on geometric criteria, insuring thus the independence of the treatment from the subject of study. Each FA exposes behavioural activities that are characterized thanks to its internal indicators. The latter form a behavioural vector that draws, by its variations, the dynamics of the agent during its live. This gives a meaning to the state of the agent inside its organization and consequently to the prominence of the semantic character that it carries.

4.1 Clustering Algorithms for the Characterization of the Situation

The goal of our approach is to characterize the factual agents organization by forming dynamically agents clusters and comparing them with stored scenarios. The clustering algorithms are an appropriate approach to this objective, since they are able to create objects groups in an unsupervised way. The basic element of these algorithms is the measure of (dis)similarity between the compared objects, that corresponds often to a geometric distance. In our case, the FAs evolve in an n-dimensional space, corresponding to the n internal indicators of the FAs. We have therefore a two dimensional space in the current case study. AI and PI are quantitative, hence it is possible to establish distances between agents. We used Cosine Similarity (CS) as (dis)similarity measure. Indeed, this measure is the most suited to our problem since we try to compare vectors that express the changes activities of the FAs. The value provided by CS is included in a range of [0, 1]. A value of 1 means the perfect equality between the two vectors (consequently the similarity of the related FAs), whereas 0 means their total divergence (dissimilarity of the FAs).

$$CS(V_1, V_2) = \frac{AI_1 AI_2 + PI_1 PI_2}{\sqrt{AI_1^2 + PI_1^2}\sqrt{AI_2^2 + PI_2^2}} \quad (1)$$

With V_1 and V_2 two vectors.

We experimented two of the most used clustering algorithms: DBScan algorithm [3] and Kmeans algorithm [5]. However, Kmeans presents an important inconvenient, since it requires to determine beforehand the number of the created clusters, that we do not know in advance. We kept finally DBScan for its concordance with our approach, but also for its efficiency.

4.2 DBScan Implementation

Fig. 5 shows an example of a DBScan experimentation. A number of parameters should be specified, such as the radios of the neighbourhood and the threshold density. These parameters are determined based on pattern scenarios that we define for each case study and on which relies the DSS scenarios base. The two axis of the chart represent AI values and PI values. We have two formed clusters in this example which are colored by red and blue, and a third set of unclassified agents, called ”noise”. At this stage, these agents do not have sufficiently evolved to integrate other clusters, consequently they do not have significant semantic characters.

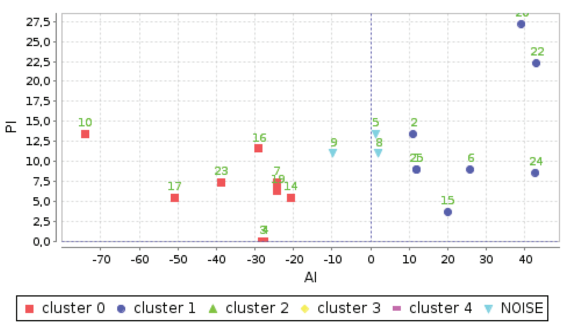

Figure 5. Display 2D of DBScan algorithm

Clusters are dynamic and change according to information coming from the environment. Indeed, each entering FSF may change the form of the clusters by altering the internal state of one or several agents.

4.3 Interpretation MAS

According to the type of the treated application, each interpretation agent is associated to one or several scenarios stored in the SB. The aim of these agents is to identify, among the emergent FA clusters, those which are enough close to the stored scenarios. They have as role also to store the new scenarios and to deal with them. The final goal of the interpretation agents is to compare the current situation with past situations and to generate thereafter response elements, that help users to evaluate the situation and to make decisions.

Scenarios are past experimented and memorised situations. They are used by the system to evaluate the treated situation and the potential risks that may produce. We defined scenarios based-on the characterization provided by Rolland and al. in [11] and which includes:

- *The content:* a scenario represents partial knowledge describing a particular situation or a particular state of the environment.
- *The purpose:* a scenario allows the system to recognize a current situation, evaluate its consequences and determine the appropriate decisions to deal with.
- *The form:* scenarios are stored in a scenarios (or cases) base in the form of cases which are made of couples of problem, solution.
- *The life-cycle:* scenarios are managed by a set of interpretation agents. Their mechanism is similar to a CBR.

Each scenario, stored in the base, represents a problem resolution episode. The problem contains the description of the scenario. It is a memorized image of an agent cluster that has been experimented. It is constituted by a set of elements describing the characteristics of the FAs at a given moment of their evolution. This elements are the FSFs and the indicators values of their associated FAs. The solution is composed of two parts:

- the first part covers the possible evolution of the situation. It may describe therefore a risk appearing or evolution,
- the second part provides the target objective which consists in managing the described risk. It is often formulated by a combination of actions and the persons who perform these actions, and used means (for example: extinguishing a building, rescuing a person, pushing a button,...).

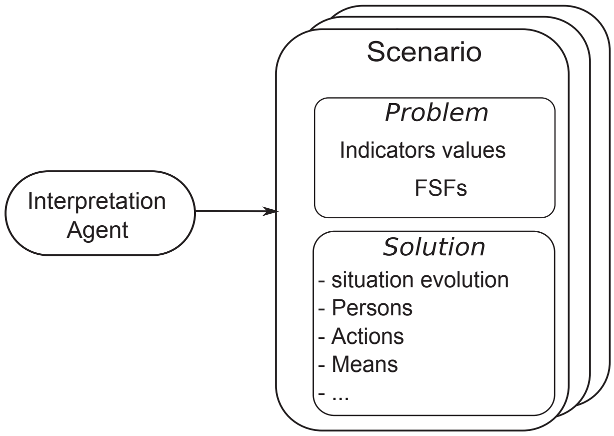

Figure 6. Scenario structure

5 Decision MAS

The decision MAS is composed of decision agents. The role of these agents is to automate the decision transmission process between the system and its environment, which must be appropriate to the treated application. Decision agents get the responses sent by the interpretation agents and adapt them to produce final decisions and send them finally to the users. In the case of the RoboCupRescue, each agent manages a type of a problem. We have four agents:

- an agent dealing with decisions related to the fire incidents,
- an agent dealing with decisions related the rescue problem of the injured civilians,

- an agent dealing with decisions related to the blocked roads problem,
- an agent dealing with default decisions. These decisions concern default actions, for example RCR agents may receive instructions to ride through the city in order to search new incidents.

The decision agents access to a common knowledge base in order to accomplish their tasks. This knowledge base includes the world model of RoboCupRescue and the current state of the disaster space, which is updated continuously by all the RCR agents. In addition, the decision agents have an intelligent module, which allows them to make computing operations and to combine certain actions to make them more suited to the current situation.

6 Conclusion

This paper aimed at addressing the problem of the decision support in crisis situations. A layered multiagent DSS has been presented here, whose goal is to help decision-makers to evaluate and to manage crisis situations. The core of the DSS represents dynamically and in real time the current situation using factual agents, then characterizes it and compares it with past known scenarios to provide finally results to decision-makers.

The factual agents played a fundamental role in all the process, since it allow the emerge of the noteworthy occurred facts of the environment. Moreover, they insure the system adaptivity thanks to their flexible internal structure. The DBscan algorithm is used to characterise the situation and to extract FAs subsets. Indeed this method is powerful to form dynamic clusters based-on geometric criteria. This insure the independence of the system from the treated application. The parameters are defined based-on typical scenarios, but we contemplate using probabilistic methods aiming to change these parameters during the process.

The next step in this work, is to introduce the other types of the factual agents related to the RoboCupRescue and to enrich consequently the scenario base to deal with all the captured events. It is also necessary to carry out the approach to other applications in order to test and to validate the multiagent core.

References

[1] S.L. Cutter, D.B. Richardson and T.J. Wilbanks, The Geographical Dimensions of terrorism, Taylor and Francis, New York, 2003.

[2] M.J. Druzdzel and R. Flynn, Decision Support Systems, Encyclopedia of Library and Information Science, 67, 2000, 120-133.

[3] M. Ester, H.P. Kriegel, J. Sander and X. Xu. A, Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, In: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996, 226-231.

[4] S. French and C. Niculae, Believe in the Model: Mishandle the Emergency, Journal of Homeland Security and Emergency Management, 2, 2005, 1-18.

[5] J.A. Hartigan and M.A. Wong, A k-means clustering algorithm, 28, 1979, 100-108.

[6] C.W. Holsapple and A.B. Whinston, Decision support systems, West publishing company, New York, 1996.

[7] G. King, Crisis management & team effectiveness: A closer examination, Journal of Business Ethics, 42, 2002, 235-249.

[8] H. Kitano, RoboCup Rescue: A Grand Challenge for Multi-Agent Systems, In: ICMAS, Washington, DC, USA, 2000, 5-12.

[9] J. Kolodner, Case-based reasoning, Morgan Kaufmann, Boston, 1993.

[10] RoboCupRescue, http://www.robocuprescue.org/cue , RoboCupRescue official web site.

[11] C. Rolland and al., A proposal for a scenario classification framework, 3, Secaucus, NJ, USA, 1998, 23-47.

[12] P. Schrivastava, Crisis Theory/Practice: Towards a Sustainable Future, Industrial and Environmental Crisis Quarterly, 7, 1, 1993, 23-42.

[13] T. Takahashi, RoboCupRescue Simulation Manual, http://sakura.meijou.ac.jp/ttakaHP/kiyosu/robocup/Rescue/manual-English-v0r4/index.html.

[14] W.A. Woods, Transition network grammars for natural language analysis, In: Commun. ACM, 13, 10, New York, NY, USA, 1970, 591-606.

[15] M. Wooldridge and N.R. Jennings, Intelligent Agents: Theory and Practice, Knowledge Engineering Review, 1994.

[16] M. Wooldridge, An Introduction to MultiAgent Systems, John Wiley & Sons, 2002.