Intelligent system for disasters management using Boolean Delay Equations models

Theodora Dumitrescu, Razvan Popescu, Daniel Merezeanu, Radu Dobrescu

Abstract— The paper proposes an architecture for an Intelligent System for Disaster Management. It is envisioned as a Multi-Agent System. By including a Model Integration component to form a hybrid system, it aims to offer support for an as wide as possible range of decisions. A final section addresses the dynamical modeling of disasters (defined as extreme events), focusing on Boolean Delay Equations (BDEs) models and its application in a case study on seismic phenomena.

Keywords— Decision Support System, Disasters Management, Intelligent System, Multi Agent System, Boolean delay equations

I. Introduction

DISASTERS management is a combined term encompassing all aspects of preparation for and responding to disasters, including prevention, mitigation, preparedness, response, and recovery [1]. The coordination of these emergency operations must be performed by a command center which needs to efficiently allocate the available resources, communicate information and take decisions regarding the planning and execution of the operations. There are different Decision Support Systems (DSS) developed for various categories of disasters and these systems are based on specific models. Due to different decision support needs that arise in disaster management area, one single model is not sufficient to cope with all of them. The first objective of this paper is to present a framework for a hybrid DSS model, which integrates different DSS models and propose the adaptation to a given scenario. Based on this model, intelligent techniques will be used to improve disaster management processes such as monitoring, controlling, and decision making. Therefore the second objective is to propose the architecture of an Intelligent System for Disasters Management (ISDM) able to offer support for various decisions.

This work was supported by the Sectorial Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labor, Family and Social Protection through the Financial Agreement POSDRU/159/1.5/S/132395.

Theodora Dumitrescu is PhD student at the Doctoral School of the Faculty of Control and Computers, Politehnica University of Bucharest, in the field of Systems Engineering (corresponding author: phone: +40214029105; e-mail: theodora.dumitrescu@gmail.com).

Razvan Popescu is PhD student at the Doctoral School of the Faculty of Control and Computers, Politehnica University of Bucharest, in the field of Systems Engineering (e-mail: codecrz@gmail.com).

Daniel Merezeanu, PhD, is lecturer in the Department of Automatic Control and Industrial Informatics of the Faculty of Control and Computers, Politehnica University of Bucharest, (e-mail: danmerezeanu@gmail.com).

Radu Dobrescu, PhD, is professor in the Department of Automatic Control and Industrial Informatics of the Faculty of Control and Computers, Politehnica University of Bucharest, (e-mail: rd_dobrescu@yahoo.com).

II. Model Integration

Model integration can be described as a way of developing decision models from existing models, by adapting a specific paradigm according to a given disaster situation. It produces a composite model, developed by merging or combining two or more models. To address this problem, we propose a dynamic integrated model which is based on a group of subroutines selected by an intelligent technique. Such a group, based on a specific disaster scenario, can be considered as a dynamic integrated model for disaster management decision support systems.

In order to provide effective decision making, modularity has been suggested as one of the possible solutions to the problems in developing decision support systems for disaster management [2]. The design of this system offers a variety of technical and theoretical aspects such as modularity and model reusability approaches to model decomposition. In order to improve this model, we suggest considering the modular routines utilized in the integrated DSS model as agents in a multi-agent system (MAS). In this new framework, the integrated model is realized in three steps:

1. Selection of the intelligent technique proper for events representation
2. Events correlation
3. Implementation of a knowledge base with dynamic relationships between the subroutines for a particular disaster scenario and the subsequent development of the domain base.

The integrated model will represent from here an Intelligent System for Disasters Management (ISDM).

We consider event correlation to be one of the key technologies in recognizing complex multi-source events. The task of event correlation can be defined as a conceptual interpretation procedure in the sense that a new meaning is assigned to a set of events that happen within a predefined time interval. The conceptual interpretation procedure could stretch from a trivial task of event filtering to perception of complex situational patterns. The act of recognition of a new situation by the correlation procedure could be formally handled as a synthetic event, and as such, it is a subject for further correlation. The process of building correlations from correlations allows the formation of complex inter-connected processes. In Fig. 1 are shown several basic connections between different correlation processes, proposed in [3], which can be mixed to create a flexible and scalable environment for complex situation modeling.

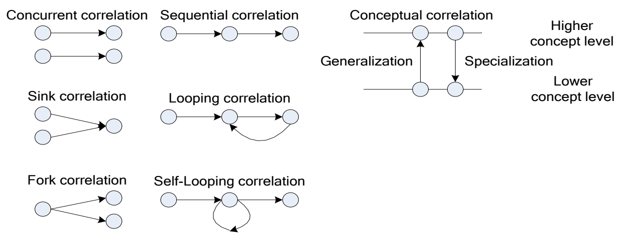

Fig. 1. Interconnections between correlation processes (after [3])

The conventional ISDM practice involves various coupled databases, event monitoring and public emergency announcement systems. During the last years significant progress was made in the development and deployment of integrated disaster information monitoring systems, and new emerging solutions of cognitive information processing, situation management, distributed computing and agent technologies has opened opportunities for new architectures for ISDMs.

III. Using a Multi-agent System for Disasters Management

A. *Basic Principles of the Approach*

The Multi-Agent System (MAS) has been widely recognized as an effective solution in modeling large number of dynamic interacting entities due to (a) the distributed organization of MAS, (b) the use of perceptual and reasoning models of mobile intelligent agents, and (c) the natural fit to model collaboration between the teams of agents. Such characteristics of MAS directly fit the requirements of ISDM. Several different architectures of MAS have been proposed, including the Belief-Desire-Intention (BDI) agent architecture [4]. Since its introduction, the BDI model has experienced several functional advancements; however especially for large-scale distributed dynamic systems this model presents some weakness, namely the lack of an adequate capability to cope with complex operational situations [5].

In the case of the proposed ISDM we focus on its cognitive aspects that require a cognitive-level MAS that is organized in a reactive situation-driven architecture, supports varying populations of agents, and scales too many interacting agent systems, where each system might have many agents. The difference between this new approach and the previous cited [3-5] is the use of a specific model of reasoning called case-oriented reasoning (COR), where each case is a template for a generic situation. The library of standard case templates that represent typical generic situations allows the construction of specific ISDM models by selecting the appropriate case templates and modifying (adapting) the selected cases with actual parameter values deduced from previous experience. In fig. 2 is represented the relation between two main processes involved in decision making, one for Situation Recognition (SR) enabled by Event Correlation (EC) which operate with the Correlation Memory and the other for Plan Reasoning (PR) driven by Case-Oriented Reasoning (COR) which operates with the Case Memory. Both processes work in a main loop, where the primary situations recognized by EC might be refined and combined by the COR and EC might get context-sensitive meta-situations in order to proceed with the event correlation process. In case of incomplete information, EC might pass queries to event collection procedures for additional information. A secondary loop appears in the PR process, where sections of a plan can trigger an iterative deliberation process.

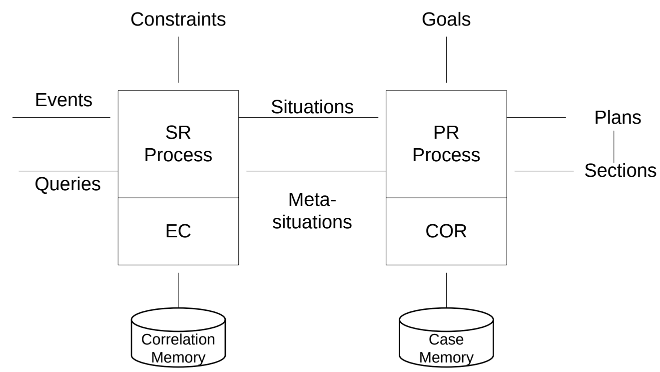

Fig. 2. Reactive relations in the decision making process

One can consider the structure in fig.2 as a situation-aware agent for Disaster Situation Management (DSM). One of the important aspects of using MAS for DSM is that the concept of an agent takes two embodiments: the physical embodiment of material resources and the virtual embodiment of software agents. Consequently, the DSM environment allows mapping the physical agents (vehicles, robots, human teams, etc.) into the abstract framework of MAS. This task involves several engineering considerations, including energy consumption, relative autonomy of physical agents, information sharing, security, etc. In particular, this architecture allows the application of the Event-Situation-Plan (ESP) paradigm, which drives invocation of a plan in a Belief-Desire-Intension (BDI) model [6].

B. *BDI agent architecture*

The Belief-Desire-Intension (BDI) model was conceived as a relatively simple rational model of human cognition. It operates with three main mental attitudes: beliefs, desires and intentions, assuming that human cognitive behavior is motivated by achieving desires (goals) via intentions providing the truthfulness of the beliefs [7].

Beliefs are the knowledge about the managed operational space (the World) that the agent possesses and believes to be true. Beliefs could be specifications of the World entities, their attributes, relations between entities, and states of the entities, relations. In many cases, the agent's beliefs include the knowledge about other agents as well as models of itself.

Desires are agent's motivations for actions. Two kinds of activities are associated with the desires: (a) to achieve a desire, or (b) prove a desire. In the first case, by applying a sequence of actions the agent wants to reach a state of the World, where the corresponding desire formula becomes true, while in the second case, the agent wants to prove that the World is or isn't in a particular state by proving that the corresponding belief formula is true or not. Often desires are called goals or tasks.

Plans are operational specifications for an agent to act. An agent's plan is invoked by a trigger event (acquisition of a new belief, removal of a belief, receipt of a message, and acquisition of a new goal). When invoking a plan, an agent tests whether the plan invocation preconditions are met, and tests run-time conditions during the plan execution. Actions could be external ones, essentially procedure calls or method invocations, or internal ones of adding and removing of beliefs. Abstract plans are stored in the agent's plan library.

Intentions are sequences of instantiated plans that an agent is committed to execute. Always while responding to a triggering external event, an agent is invoking a plan from the plan library, instantiating it and pushing into a newly created stack of intentions. Contrary to that, when an agent responds to an internal triggering event, i.e., an event created by an internal action of some previous plan instance, then the new plan instance is pushed onto the stack of the previous plan that caused the invocation of the new plan instance.

The architecture of a BDI agent is presented in Figure 3.

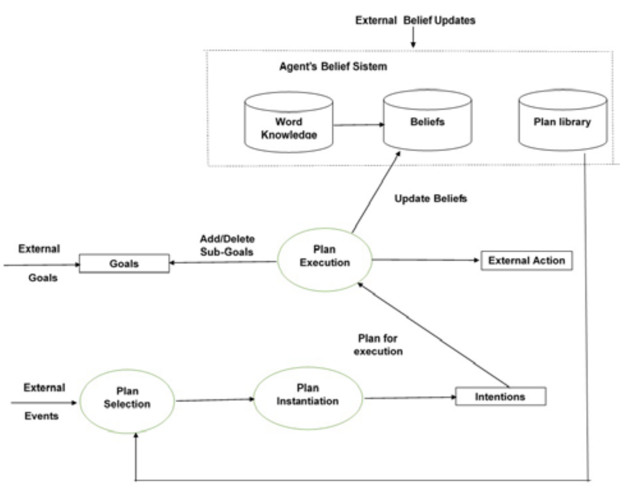

Fig. 3. Architecture of a Situation – Aware BDI Agent

IV. ISDM Concept and Design

A. Main Tasks of ISDM

The major activities with decision-making needs (shown inside brackets) in disaster management are as follows:

- Hazard assessment (vulnerability analysis, frequency of hazard occurrences)
- Risk management (analysis of disaster risks, evaluating risks and treating risks)
- Mitigation (developing mitigation plan, analysis of measures)
- Preparedness (planning and resource management)
- Response (emergency response plans, analysis and evaluation)
- Recovery (assessments, re-settlement issues)

In order to offer the possibility to combine and adapt different strategies, we decided to build our Intelligent System for Disaster Management (ISDM) using a Decision Support System (DSS) able to help decision-makers by cooperative work of several intelligent agents, included in a multi-agent structure. Multi-agent systems are ideally suited to representing problems that have multiple problem solving methods and multiple perspectives. Intelligent agents take initiative where appropriate, and socially interact, where appropriate, with other artificial agents and humans in order to complete their own problem solving and to help others with their activities.

The most important responsibilities of the agents involved in Decision Support for disaster management usually are:

- Monitoring: observe the environment and detect problematic behaviors;
- Alarm generation: raise alarms if there is a critical situation;
- Warning: warning respecting undesired consequences of "bad" actions and potentially suggesting better ones.

The specific tasks for the implementation of the above mentioned activities must be carefully designed. In methodologies that go back to the knowledge engineering field, a task is usually conceived as an abstract description of how the world (or an agent's "mental model" of it) needs to be transformed in order to achieve a desired behavior or functionality. To generate answers for the different classes of actions in our management framework, we have identified four essential tasks:

- Problem identification: From the analysis of the information received from a communication infrastructure or directly from the operator, the classifier chooses the state of the monitored system;
- Diagnosis: The presence of unacceptable events or situations requires an explanation in terms of causal features of the situation.
- Action planning: Once a problem has been identified, a possible sequence of actions applicable on the causes may be established.
- Prediction: The consequences of events and operator actions are simulated.

Let us consider a set of system components S, a set of external events E and a set of operator actions A.

By combining the above tasks in different manners, several questions that a decision maker typically faces can be answered. For instance:

"What is happening in S?" represents: *problem identification + diagnosis.*

A diagnosis D for some potential malfunction is produced:

"What to do on D in S?" represents: *action planning + prediction*

Decision options are shaped and their potential effects evaluated:

"What may happen if E in S?" represents: *prediction + problem identification + diagnosis*

Potential future problems in evolution of the system are identified:

"What to do if *E* in *S*?" represents: *prediction + problem identification + diagnosis + planning.*

A common way of dealing with these issues it to conceive the ISDM itself as a multi-agent system, where each distributed entity is controlled by an agent. Hence, any of the aforementioned tasks of problem identification, diagnosis, action planning and prediction can be performed locally by each agent within the multi-agent system. These local tasks will be of less complexity, but they are also interdependent. The co-ordination task that such a multi-agent system faces, refers to the management of these dependencies between local tasks.

B. Specific Methods for ISDM

Most knowledge-oriented methodologies make use of the concept of problem-solving methods in order to cope with tasks. In particular, such methods indicate how a task is achieved, by describing the different steps by which its inputs are transformed into its outputs. The problem-solving process associated to a task is structured as follows: each of its steps may set up several subtasks, which again are to be solved by simpler methods and so on, until some elementary tasks can be achieved directly.

Problem identification methods

A classification method with two options may be applied:

• Identification of a reference situation and classification of the differences between the reference and the current situation.

• Direct classification of the current situation based on a predefined taxonomy where problems of different types are described.

The first approach requires: (1) to infer from the current situation the evolution of parameters consistent with the functional and structural constraints which optimizes a collection of predefined criteria and (2) to classify the differences between the observed situation and the resulting class of situations according to a hierarchy similar to the one previously commented.

For the first subtask, the method is applied in two steps. The first step derives a possible new state from the current situation that may be supported by an ad hoc procedure, adapted to the characteristics of the domain model. For the second subtask a primary representation based on rules and/or frames may be applied in a hierarchical establish & refine model. The second approach is similar to the first one, but in this case a complete description of the situation is required, not only the differences with the reference situation.

Diagnosis methods

This task infers a collection of causes explaining the problems identified by the previous one. Several methods may be directly applied:

(1) The classification method, which extends problem type frames by additional cause attributes in such a way that once a problem pattern has been selected, the cause features assumed for this problem type are assumed.

(2) A version of the cover & differentiate method [8] where a hierarchical approach to an explanatory set of causes is generated through the following reasoning steps: (i) from the attributes of the type of problem detected a collection of possible causes may be inferred covering these values; (ii) since this first set of causes may be too large, a deeper analysis to differentiate subsets explanatory enough is necessary.

Action planning methods

After the problem identification and diagnosis tasks, some scenarios of causes of problems have been deduced together with its impacts. The action planning task must generate a consistent set of actions oriented toward the reduction or elimination of causes and/or toward the reduction of impact damages where no possible cause reduction may be produced. Specifying this task in a general way requires defining the elementary actions that will be the basis for definition of acceptable decision plans together with their models [9].

Behavior prediction methods

This task has as main goal to propose scenarios of short-term future behavior of the different components of the model. There may be specific simulation methods performing this type of task. A library could be considered to support a class of applications including a collection of typical physical components. The model of reasoning may take the current state from the information system and the assumptions about the external actions and match it with some node in the graph [10]. As a result, for every matched situation the predictable short-term changes are described by the downstream connected states.

Coordination methods

Coordination is best conceived of as the management of dependencies between activities. Methods that perform this type of management usually comprise three steps:

• Dependency detection: using domain knowledge about the different dependencies that may occur (producer-consumer relationships, resource limitations etc.) positive and negative relationships between the different local tasks of the agents are detected.

• Option generation: for every dependency, the set of possible management actions is generated.

• Management decision: finally, a decision must be taken respecting the dependency management action to be applied.

V. System Architecture

In a well-known publication [11], Meystel and Albus stipulate that any intelligent system consists of two parts:

1. Internal, or computational, which can be decomposed into four internal subsystems of intelligence as follows:

a) Sensor processing - inputs are provided to an intelligent system via sensors and are processed to create a consistent state of the world. Sensors are used to monitor the state of the external world and the intelligent system itself.

b) World modeling - is the estimation of the state of the world; it includes knowledge databases about the world and contains a simulation module that provides information about future states of the world.

c) Behavior generation – is the decision making module that selects goals and plans, and executes tasks.

d) Value judgment – it evaluates both the observed state and predicted state; it provides the basis for decision making.

2. External, or interfacing; input and output from the internal part of the intelligent systems are generalized via sensors and actuators that can be considered external parts.

We adapted the system architecture referenced in [11] which is based on the real control system techniques. Figure 4 shows the basic components of the ISDM.

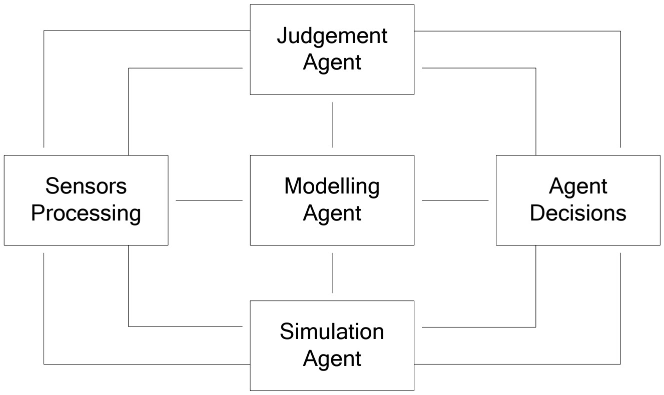

Figure 4. Block scheme of ISDM

The general diagram in Fig. 5 is constructed in concordance with the principles of a multi-agent system organization. Software agents are computational units that are repeated many times within an intelligent system at many different levels as the units of information in all of the subsystems are aggregated into entities, events, situations, and goals are decomposed into sub-goal tasks and generate actions or commands. Within each loop, data processing maintains a knowledge database with a characteristic range and resolution. At each level, plans are made and updated with different planning horizons. At each level, short term memory traces sensory data over different historical intervals. At each level, feedback control loops have a characteristic. This model of a multi-resolution hierarchy of computational loops ushers deep insights into the phenomena of behavior, perception, cognition, problem solving and learning.

The architecture of an intelligent system is a specific framework of agents and each agent has its own architecture. In the core of any intelligent system, there is also the concept of a generalized agent. Agents with similar functions can be gradually lumped in a group type agent, which basically is a generalized agent. The group agent gives a new world representation in terms of granularity or resolution. Furthermore, group agents can be aggregated into an even more generalized agent, in a hierarchical structure.

The proposed architecture includes elements of intelligence to create functional relationships and information flows between different subsystems. The elements of intelligence are based on components using one or more AI techniques: natural language processing, artificial neural networks, fuzzy logic, cellular automata (in particular Boolean Networks for solving Boolean Delay Equations - BDE). The following section of the paper is dedicated to the use of BDE models.

Figure 5 depicts the proposed architecture of the intelligent assistance system, based on the integration of traditional statistical methods and various AI techniques to support a general system that operates automatically, adaptively, and proactively.

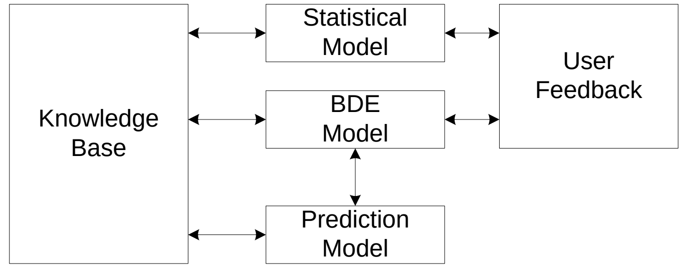

Figure 5. Intelligent model components

The system's architecture is based on a hybrid approach that yields both the robustness and depth of understanding of decision making using intelligent models. This system aims to improve monitoring and decision making processes with an effect size that is higher than a human expert. In addition, this system provides mechanisms to enhance the active construction of knowledge about threats, policies, procedures, and risks. The model is adaptive and supports processing and classification of events and data that leads to the prediction of anomalies or even extreme events. One major component in design is the development of an intelligent model for the analysis and correlation of events and data in real-time to increase the detection and prevention capabilities. The hybrid system is an integration of different models for the disaster and extreme events management to include AI techniques and other methods based on statistical and traditional procedural approach. The basic idea of the multiple models is to independently perform different functions with different measures, and to complement the weaknesses of one model with the strengths of another model.

Furthermore, the ISDM can interpret the data for human. Because the outputs of the models are uncertain and imprecise in some situation, and because human experts may have some intuition or additional knowledge on the characteristics of the presented information, ISDM could interpret the outcomes of other models in a form that humans relate better to. The system should include functions for the automation of tasks such as data collection, data reduction, filtering, and event correlation based on multi-agent technologies. The system may generate commands to end processes or move the processing to another device when signs of suspicious behavior or failures are detected.

In addition, our system is an intelligent assistant to provide feedback to the user such as help on making decisions and taking actions. The system includes a user interface based on multimedia for supporting network administrator's operations, and a knowledge base for maintaining trustworthiness as systems change and adapt. This knowledge base must be adaptive and shared via the network. The validation of the computer generated decisions can be performed by comparing with the decisions of experts. The user feedback module provides different feedback to a network administrator. The type of feedback available is important. Direct feedback entails specific information about the results and impact of each possible feedback. Indirect feedback is situated on a higher level, with no specific information about individual change or predictions but rather proposing new strategies and systemic changes.

An important issue is to maintain the functionality of the system in situation of risk and hazards, so to provide the management of uncertainties to ensure improved design, robust operation, accountable performance and responsive risk control. Therefore, ISDM can provide the ability to adaptively assign emergency-role and permissions to specific subjects and inform subjects without explicit access requests to handle emergency situations in a proactive manner. In this aim the concepts of emergency-group and emergency dependency [12] were introduced. Emergencies are processed in sequence within the group and in parallel among groups.

VI. Boolean Delay Equations for dynamic modeling

A. *Theoretical background*

Boolean delay equations (BDEs) are a modeling framework especially tailored for the mathematical formulation of conceptual models of systems that exhibit threshold behavior, multiple feedbacks and distinct time delays [13]. BDEs are intended as a heuristic first step on the way to understanding problems too complex to model using systems of partial differential equations at the present time. BDEs may be classified as *semi-discrete dynamical systems*, where the variables are discrete — typically Boolean, *i.e.* taking the values 0 (“off”) or 1 (“on”) only — while time is allowed to be continuous. Systems in which both variables and time are continuous are called *flows*. Vector fields, ordinary and partial differential equations (ODEs and PDEs), functional and delay-differential equations (FDEs and DDEs) and stochastic differential equations (SDEs) belong to this category. Systems with continuous variables and discrete time are known as *maps* and include diffeomorphisms, as well as ordinary and partial difference equations. In automata both the time and the variables are discrete; cellular automata (CAs) and all Turing machines (including real-world computers) are part of this group. These kind of BDE are used in our approach and are described with the following general form.

Given a system with n continuous real-valued state variables $\vec{v} = (v_1, v_2, ..., v_n) \in R^n$ for which natural thresholds $q_i \in R$ exist, one can associate with each variable $v_i \in R$ a Boolean variable, $x_i \in B$ = {0, 1}, *i.e.*, a variable that is either “on” or “off,” by letting

$$x_i = \begin{cases} 0, v_i \le q_i \\ 1, v_i > q_i \end{cases}, i = 1, ..., n \tag{3.1}$$

The equations that describe the evolution of the Boolean vector $\vec{x} = (x_1, x_2, ..., x_n) \in B^n$ due to the time-delayed interactions between the Boolean variables $x_i \in B$ are of the form:

$$x_i = f_i(x_1(t-\theta_{i,1}), x_2(t-\theta_{i,2}), ..., x_n(t-\theta_{i,n})$$

The functions $f_i : B^n \to B$, $1 \le i \le n$, are defined via Boolean equations that involve logical operators and delays. Each delay value $\theta_{i,j} \in R$, where $1 \le i, j \le n$, is the length of time it takes for a change in variable x_j to affect the variable x_i. Fig. 6 presents the time diagram of a system described by a two dimensional Boolean vector , $x_1, x_2 \in B^2; 0 < \theta \le 1$ with:

$$\begin{cases} x_1(t) = x_2(t-\theta), \theta = 1/2 \\ \quad x_2(t) = \bar{x}_1(t-1) \end{cases}$$

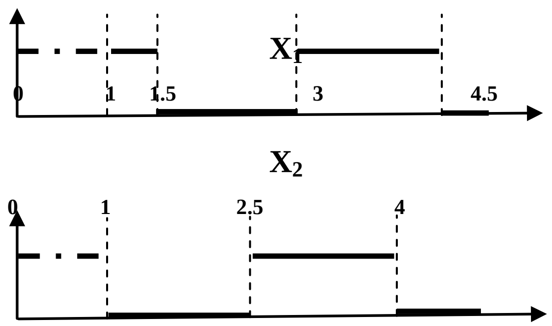

Fig.6. Time diagram of a bi-dimensional BDE system

The basic theoretical results from BDE theory appears in Ghil and Mullhaupt [14]. They classified BDE systems as follows: all systems with solutions that are immediately periodic for all rational delays are *conservative*, while systems that for some rational delays exhibit transient behavior before settling into eventual periodicity are *dissipative*. The differential dynamical systems analogs are conservative (*e.g.*, Hamiltonian) dynamical systems versus forced-dissipative systems. For example, a system described by the equation model $x(t) = \bar{x}(t-1)$ is conservative, while if the model is $x(t) = \bar{x}(t-1) \wedge x(t-\theta)$ is dissipative. Another characteristics of BDEs system is the asymptotic behavior. The following types of asymptotic behavior were observed in BDE systems: *(a) fixed point* — the solution reaches one of a finite number of possible states and remains there; *(b) limit cycle* — the solution becomes periodic after a finite time elapses; and *(c) growing complexity* — certain classes of BDEs with incommensurable delays were shown to have solutions with growing complexity, as measured by the number of jumps per unit time. This number grows like a positive, but fractional power of time t, with superimposed log-periodic oscillations.

B. *A BDE Model for Seismicity*

Lattice models of systems of interacting elements are widely applied for modeling seismicity, starting from the pioneering work of Allegre *et al*. [15]. The state of the art is well summarized in [16], which also refers to the colliding cascade models, able to reproduce a wide set of observed characteristics of earthquake: (i) the seismic cycle; (ii) intermittency in the seismic regime; (iii) the size distribution of earthquakes; (iv) long-range correlations in earthquake occurrence; (v) a variety of seismicity patterns premonitory to a strong earthquake. Colliding cascade models [17] synthesize three phenomena that play an important role in many complex systems: (i) the system has a hierarchical structure; (ii) the system is continuously loaded (or driven) by external sources; and (iii) the elements of the system fail (break down) under the load, causing redistribution of the load and strength throughout the system. Eventually the failed elements heal, thereby ensuring the continuous operation of the system. The load is applied at the top of the hierarchy and transferred downwards, thus forming a direct cascade of loading. Failures are initiated at the lowest level of the hierarchy, and gradually propagate upwards, thereby forming an inverse cascade of failures, which is followed by healing. The interaction of direct and inverse cascades establishes the dynamics of the system: loading triggers the failures, and failures redistribute and release the load. In its applications to seismicity, the model's hierarchical structure represents a fault network, loading imitates the effect of tectonic forces, and failures imitate earthquakes.

Our BDE model is similar with the model discussed in [16], a ternary tree where each element is connected to and interacts with its six nearest neighbors: the parent, two siblings, and three children (see fig. 7). At each epoch a given element may be either intact or failed (broken), and either loaded or unloaded. The state of an element *e* at a moment *n* is thus defined by two Boolean functions $s_e(n)$ ={*0(intact) or 1(failed)*} and $l_e(n)$ ={*0(unloaded) or 1(loaded)*}. An element of the system may switch from one state to another under an impact from its nearest neighbors and external sources.

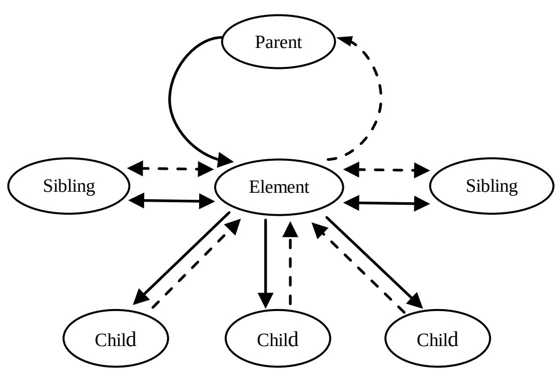

Fig.7. A BDE ternary tree model for seismicity

The dynamics of the system is controlled by the time delays between the given impact and switching to another state. At the start, $t = 0$, all elements are in the state (0, 0), intact and unloaded. Most of the changes in the state of an element occur in the following cycle: (0, 0) → (0, 1) → (1, 1) → (1, 0) … . The duration of each particular delay, from one switch of an element's state to the next, is determined from four basic delays, depending on the state of the element as well as of its nearest neighbors during the preceding time interval. The two primary delays are the loading time Δ_L necessary for an unloaded element to become loaded under the impact of its parent, and the healing time Δ_H necessary for a broken element to recover. Failures are initiated randomly within the elements at the lowest level. The other two basic delays are Δ_F , between the increase in weakness and switching to the failed state and Δ_D, between failure and switching to the unloaded state.

The model is forced and dissipative, if we associate the loading with an energy influx. The energy dissipates only at the lowest level, where it is transferred downwards, out of the model. In any part of the model not including the lowest level energy conservation holds, but only after averaging over sufficiently large time intervals. On small intervals it may not hold, due to the discrete time delays involved in energy transfer. The output of the model is a catalog of failures of the elements of an earthquake, similar to the simplest routine catalogs of observed earthquakes:

$$C = (t_k, m_k, h_k),\ k = 1, 2, \ldots;\ t_k \le t_{k+1}. \qquad (3.2.)$$

where t_k is the starting time of the rupture; m_k is the magnitude, a logarithmic measure of energy released by the earthquake and h_k is the vector that comprises the coordinates of the hypocenter (i.e. the point of the area where the rupture started).

The quantitative description of model earthquake sequences is given by two measures: the density $\rho(n)$ of the elements that are in a failed state at the moment *n* and the irregularity $G(I)$ of energy release over the time interval *I*.

If we consider $v_i(n)$ is the fraction of failed elements at the *i*-th level of the hierarchy at the moment *n* and *m* the depth of the tree, then

$$\rho(n) = [v_1(n) + \ldots + v_m(n)]/m \qquad (3.3)$$

and can be denoted by $\rho(I)$ – the measure averaged over a time interval *I*.

At its turn, *G* can be calculated as the average sum of the energy developed in a set of nonoverlapping intervals N_I, such that $|I|=\delta\ N_I$ where |.| denotes the length of the interval. Nonetheless, *G* has a transparent intuitive interpretation: it equals unity for a catalog consisting of a single event (burst of energy) and it is zero for a marked Poisson process (uniform energy release). Generally, it takes values between 0 and 1 depending on the irregularity of the observed energy release.

The model produces synthetic sequences that can be divided into three seismic regimes, denoted *H*, *I* and *L*. Regime *H* corresponds to *high* and nearly periodic seismicity, when the fractures within each cycle always reach the top level. Regime I exhibits *intermittent* seismicity (the seismicity reaches the top level for some but not all cycles). Regime *L* is characterized by *low* seismicity (no cycle reaches the top level and seismic activity is much more constant).

This BDE model was utilized in a case study on the seismic events in the Vrancea region of Romania [18]. The aim of this study was to decide on the basis of the behavior of the seismic activity prior to *t*, whether a strong earthquake will or will not occur in a specified region during a subsequent interval *(t; t+Δt)*. We used an algorithm derived from the intermediate-range earthquake prediction algorithm M8 of Keilis-Borok and Kossobokov [19] which allows to combine BDE model with a model build on the principles of the Extreme Values Theory (EVT). EVT provides a solid probabilistic foundation [20] for studying the distribution of extreme events in many fields of applications. Probabilistic EVT theory is based on asymptotic arguments for sequences of independent and identically distributed (i.i.d.) random variables; it provides information about the distribution of the maximum value of such an i.i.d. sample as the sample size increases.

The data set on seismic activity in the Vrancea region was taken from the RomPlus earthquake catalogue compiled at the National Institute of Earth Physics (Caldarusani, Romania).

Only strong earthquakes, with magnitude $M \geq M_0$, were considered extreme events of interest Three values have been chosen for the magnitude threshold of strong earthquakes (M_0): 6.0, 6.5 and 7.0 and we have specified a threshold $\Delta M = 0.5$ for all these values because the M8 algorithm focuses specifically on earthquakes in a range $M_0 \leq M < M_0 + \Delta M$ with given ΔM. The combined model allows to test an earthquake prediction method designed by the retrospective analysis of the dynamics associated with seismic activity starting from 1960 (i.e. the last 50 years).

VII. Conclusion

Disaster management is characterized by large volumes of real-time events and complex domain models which require a combination of data fusion, event correlation and semantic reasoning in order to identify and assess the current context and recommend actions. Such application domains include preparedness, disaster recovery and risk reduction management. Due to the highly distributed and multidisciplinary nature of these applications, MAS is a convenient development model. From a system design perspective, the data fusion, event correlation and situation management technologies offer significant scalability for real-time event processing and state analysis. In addition, the large scale of these application domains suggests that multiple agent platforms will have to cooperate.

One can conclude that because of the complexity of information management tasks, the proposed system is based on the integration of different types of intelligent agents, i.e. a hybrid architecture under real-time constraints. We have demonstrated the model integration technique using the concepts of relational theory. We have argued the need for model integration in the area of disaster management and usefulness of model integration techniques. We have also elaborated the usefulness of intelligent agent technology for model selection and stated the actions that an agent can perform while selecting the model from the model base.

Advanced real-time techniques based on modeling, sensor analysis, and intelligent agents integrated with traditional procedural and statistical methods can recognize, filter, and correlate events and data collected by various sensors and sources. These techniques support the capability to provide automated feedback to correct the problems including useful advice to a human to take actions and prevent ongoing attacks. That means that the ISDM can be considered a cyber-physical system which ensures a Cognition-Adaptive Human–Computer Interface for any type of mission-critical systems, in particular for disasters management.

The key question for the description, understanding and prediction of extreme events is if one can extrapolate knowledge on the numerous small ones phenomena to characterize the few large ones. This approach allows one to jump from the description of the many to the prediction of the few. Such systems are better known than others, and can be modeled by using fairly sophisticated tools, like sets of differential equations and other modeling frameworks, whether deterministic, stochastic or both. In our research we choose to explore “partial BDEs” in which the number of Boolean variables is quite large. These systems stand in the same relation to “ordinary BDEs,” explored so far. It would appear that BDEs are well suited for the exploration of poorly understood phenomena and extreme events occurrence in the natural world. Moreover, the robustness of fairly regular solutions in a wide class of BDEs, for many sets of delays and a variety of initial states, suggests interesting applications to certain issues in massively parallel computations.

Our research based on cognitive load measurement (the BDI approach) has yielded promising outcome and validated the feasibility of ISDM. We proposed a novel architecture of an intelligent system for disaster and extreme events management. The proposed architecture is based on multidisciplinary paradigm which includes information management, network communications, process control, computer science, artificial intelligence, modern control theory, statistics, management science, risk analysis. No single approach can resolve the growth and increased sophistication of such a system. We need to apply several paradigms to meet the objectives of information management for the modern organization of the 21st century. Intelligent agent technology offer additional facilities. The system has to be adaptive and capable of discovering and building new knowledge for the information domain. Future work should seek a systematic proof-of-concept that integrates all modules to support any situation of disaster management. At the moment, we are working on the improvement of the software architecture and the development of other key modules.

References

[1] R. J. Craddock, *Crisis Management Models and Timelines*, Thales Research and Technology, 2006

[2] S. Asghar, D. Alahakoon, and L. Churilov, “A Comprehensive Conceptual Model for Disaster Management”, *Journal of Humanitarian Assistance*, 1360 (0222), pp. 1-15, 2006

[3] G. Jakobson, N. Parameswaran, J. Buford, P. Ray, and L. Lewis: “Situation-Aware Multi-Agent System for Disaster Relief Operations Management”, *Proceedings of the 3rd International ISCRAM Conference*, pp.313-324, 2006

[4] J. F. Buford, G. Jakobson, and L. Lewis, “Multi-Agent Situation Management For Supporting Large-Scale Disaster Relief Operations”, *International Journal of Intelligent Control and Systems*, Vol.11, No. 4, pp. 284-295, 2006

[5] G. Jakobson, J. Buford, and L. Lewis, “A Framework of Cognitive Situation Modeling and Recognition.” *Proceedings of the 3rd Workshop on Situation Management*, 2006

[6] M. Bratman, *Intension, Plans, and Practical Reason*, Harvard University Press, 1987.

[7] M. d'Inverno, M. Luck, M. Georgeff, D. Kinny, and M. Wooldridge, “The dMARS Architecture: A Specification Of the Distributed Multi-Agent Reasoning System”, *Journal of Autonomous Agents and Multi-Agent Systems*, 9(1-2):5-53, 2004

[8] S. Ossowski, J. Z. Hernández, C. A. Iglesias, and A. Fernández,: “Engineering Agent Systems for Decision Support”, *Proceedings of the 3rd Int. Conf. on Engineering societies in the agents world ESAW'02,* 2002

[9] G. P. Alexandris and N. G. Bardis, “Comparing accuracy and solution times in coverage models”, *Proceedings of the 18th International Conference on Systems, Latest Trends on Systems - Volume II,* July 2014, pp.408-413.

[10] A. Desmet and E. Gelenbe, “Graph and Analytical Models for Emergency Evacuation”, *Future Internet, 5*(1), 2013, pp. 46-55

[11] A. M. Meystel and J. M. Albus, *Intelligent systems architecture, design, and control*, John Wiley & Sons, 2002

[12] G. Wu, D. Lu, F. Xia and L. Yao, “A FaultTolerant Emergency-Aware Access Control Scheme for Cyber-Physical Systems”, *Information Technology and Control*, vol.40, iss. 1, Dec. 2011, pp. 29-44.

[13] H. Öktem, R. Pearson, and K. Egiazarian, “An adjustable aperiodic model class of genomic interactions using continuous time Boolean networks (Boolean delay equations)”, *Chaos*, 13, 2003, pp. 1167–1174

[14] M. Ghil, and A. P. Mullhaupt, “Boolean delay equations. II: Periodic and aperiodic solutions”. *J. Stat. Phys.*, 41, 1985, pp.125-173

[15] C. J. Allegre, J. L. Le Mouel, and A. Provost, “Scaling rules in rock fracture and possible implications for earthquake prediction”, *Nature*, 297, 1982, pp.47-49.

[16] M. Ghil, I. Zaliapin and B. Coluzzi, ”Boolean delay equations: A simple way of looking at complex systems”, *Physica D*, 237, pp.2967–2986, 2008

[17] I. Zaliapin, V. Keilis-Borok and M. Ghil, “A Boolean delay equation model of colliding cascades. Part II: Prediction of critical transitions”, J. *Stat. Phys.*, 111, pp. 839-861, 2008

[18] S. Arghir, R. Dobrescu, and F. Ionescu “Using boolean delay equations for predicting fractal processes”, *Proc. of the 18th Int. Conf. on Control Systems and Computer Science*, CSCS18, 2011,vol. 2, pp. 913–915

[19] V. I. Keilis-Borok and V. Kossobokov, “Premonitory activation of earthquake flow: algorithm M8”, *Phys. Earth Planet. Inter.*, 61, pp.73–83, 1990.

[20] L. De Haan and A. Ferreira, *Extreme Value Theory: An Introduction*, Springer, New York, 2006