# Smart Cities and Resilience Plans: A Multi-Agent Based Simulation for Extreme Event Rescuing

# Karam Mustapha, Hamid Mcheick, and Sehl Mellouli

# Abstract The concept of smart Cities is one that relies on the use of new Information and Communication Technologies in order to improve services that cities provide to their citizens. The resilience of a city is one of the services that it can provide to its citizens. Resilience is defined as its capacity degree to continue working normally by serving citizens when Extreme Events (EEs) occur. This chapter will propose a new framework based on multi-agent systems to help cities build simulation scenarios for rescuing citizens in the case of an EE. The main contribution of the framework will provide a set of models, at different levels of abstraction, to reflect the organizational structure and policies within the simulation, which involves the integration of truly dynamic dimensions of this organization. The framework will also propose methods to go from one model to another (conceptual to simulation). This framework can be applied in different domains, such as smart cities, earthquakes, and building fires.

**Keywords:** Extreme Events, City Resilience, Agent Based Simulation, Multi-Agent Systems, Organization, Architecture, Modeling, Simulation.

**List of Abbreviations**

AA Agent Artefact

ABDiSE Agent-Based Disaster simulation Environment

ABS Agent Based Simulation

ACL Agent Communication Language

AUML Agent Unified Modeling Language

BDI Believe, Desire, Intention

CAOM Conceptual Agent Organizational Model

CROM Conceptual Role Organizational Model

D4S2 Dynamic Discrete Disaster Decision Simulation System

EE Extreme Events

FACL Form-based ACL

FIPA Foundation of Intelligent Physical Agents

GIS Geographical Information System

JADE Java Agent Development Environment

MAS Multi Agent System

MDA Model Driven Architecture

MDD Model Driven Development

MOON Multiagent-Oriented Office Network

ND Natural Disaster

OMT Object Modeling Template

OPAM Operational Agent Model

PIM Platform Independent Model

PSM Platform Specific Model

RTI Real Time Infrastructure

SAMoSAB Software Architecture for Modeling and Simulation Agent-Based

UEML Unified Enterprise Modeling Language

# Introduction

Smart Cities is a concept that relies on the use of new Information and Communication Technologies by cities in order to improve services that cities provide to their citizens. One of the services that cities have to provide to their citizens is their resilience. The resilience of a city is defined as its capacity degree to continue working normally by serving citizens when Extreme Events (EEs) occur. The management of Extreme Events (EEs) is becoming more complex since EEs are becoming more frequent and more powerful. These events may be caused by either nature like storms or earthquake, or humans like wars or airplane crashes. When happening, these events require the intervention of different teams to rescue people such as police, fire-workers, Non Government Organizations (NGOs) like Red Cross, etc. Meantime, these emergency teams have to collaborate and coordinate their activities to better rescue people. However, these teams have different skills, use different tools, adopt different strategies, and play different roles. All this heterogeneity adds a lot of complexities to the rescuing activity. To this end, these teams will need tools to help them making efficient interventions. Simulations are one of the means that can be used by these teams to predict their behaviour during an EE.

Over the past few decades, EEs such as droughts, floods, cyclones, earthquakes and volcanic eruptions have resulted in the mortality of approximately three million people and affected the lives of 800 million people worldwide. These have caused diseases as well as serious economic losses and homelessness. Therefore, modeling and simulating the rescue procedure may help to facilitate their management and limit their impact on the society. These simulations may improve the efficiency of the teams on the field that may lead to reducing losses and damage of goods, and saving lives. Multi-agent systems (MAS) are among the techniques used for modeling and simulating EE emergencies.

A MAS can model the behaviour of a set of entities expert, more or less organized by respecting the laws governing their relations [5]. Agents have a degree of autonomy and are immersed in an environment in which and with which they interact [4]. There are several areas where MAS can be applied; they can act as a modeling paradigm or as a solution for software implementation. Therefore, the application of MAS in this area could help managers to experiment all possible scenarios of a disaster and assist them in making decisions. This approach involves the simulation of systems in terms of models and their use.

Agent-Based Simulation (ABS) [24] has spread out into many areas, including sociology, biology, economics, physics, chemistry, ecology, industrial applications and EE. ABS have the ability to capture different dynamic models which usually consist of simple entities (called reactive agent if a simple behaviour is required) or more complex entities (called deliberative agent if decision-making and negotiation are needed). The global objective of this chapter is to provide a methodological framework that ranges from domain model analysis to running a simulation while considering the different entities that can make an intervention in the case of an EE. In [20], we proposed a specific agent-based methodological framework allowing, from modeling to simulation, the production of observables at different levels of details related to an EE rescuing activity. The proposed framework in this chapter is an extension of our previous work [20] with the objective to integrate dynamic organizational characteristics of an EE rescuing activity in the modeling and simulation procedures. It will also include the specification of the translation process from generic models to specific models, to ensure the transition between the proposed models.

This chapter is structured as follows: Section 2 defines the objectives of the research and related concerns modeling and simulation of EE. Section 3, introduces an organizational-oriented methodological framework, which is capable of taking into account the organizational aspects at both the conceptual and the operational abstraction levels. Section 4 describes the dynamic EE organization. Section 5 introduces a model driven architecture to transform the models proposed in the methodological framework. Section 6 details the agent-based software architecture in line with the proposed methodological framework, to simulate EE’s organizational aspects. Section 7 presents an illustrative example of the proposed software architecture through the modeling of a building fire. Finally, conclusions and recommendations for future work are summarized in Section 8.

# Smart cities and resilience plans

Smart city is an emerging strategy, based on information and communication technologies that cities are adopting to mitigate the problems generated by the urban population growth and rapid urbanization. This emerging concept is a type of urban development able to meet the needs of institutions, businesses and citizens, both economically, environmentally, and socially. We can described a smart city when we investments in human capital, traditional communication, energy social infrastructure and a high quality of life, with a wise organization of natural resources, and through participatory governance [[1]](#footnote-1).

There is an increase in frequency of use of the phrase smart city; there is not a clear and reliable understanding of the concept in the middle of academia. A limited number of studies investigated and began to systematically consider questions related to this new urban phenomenon of smart cities.

The concept is used all over the world with different nomenclatures, the context is still emerging, and the research work of defining and conceptualizing it is in evolution [45, 46]. Several work definitions have been put forward and adopted in academic use. This discord of definitions is resulting in calls for conceptual research in this regard [45].

* A city well performing in a forward-looking way in economy, people, mobility, environment, governance, and living, built on the smart combination of endowments and activities of self-decisive, independent and aware citizens. [47]
* A city that monitors and integrates conditions of all of its critical infrastructures, for example, roads, bridges, tunnels, rails, subways, airports, seaports, communications, water, power, even major buildings, can better optimize its resources, plan its preventive maintenance activities, and monitor security aspects while maximizing services to its citizens. [48]
* A city “connecting the different infrastructure, like physical, IT, social and the business to leverage the collective intelligence of the city” [49].
* A city striving to make itself “smarter” (more efficient, sustainable, equitable, and livable) [45].
* Use of Smart Computing technologies to make the critical infrastructure components and services of a city-which include city administration, education, healthcare, public safety, real estate, transportation, and utilities-more intelligent, interconnected, and efficient [50].

To become smart, existing cities should develop new efficient services in all areas such as:

* Intelligent transport and mobility one of the challenges is to integrate different modes of transport - car, cycle and walking - in one system that is efficient, accessible, affordable, safe and environmentally. This integration allows a reduced environmental, optimizing the use of urban space and offers a diverse range solutions of urban mobility that meet all their needs. In addition, the city will have to implement the latest technologies of transportation and electric mobility;
* Environmental sustainability: cities must act in two main areas: waste and energy. On waste, cities mission will be to reduce or avoid their waste and put in place effective systems for recovery and recycling of waste (process by which a material waste or a useless product is transformed into a new material or product quality or greater value).
* Responsible and intelligent urban habitat: the high value of property in city centers combined with the limited availability of land make the complex current urbanization. For example, buildings must be smarter to facilitate and improve the management of energy, or reduce consumption.

# Agents and organization oriented EE modelling and simulation

A multi-agent based system is a powerful modeling technique for simulating individual interactions in a dynamic system and is distinctive in its ability to simulate situations with unpredictable behaviour [16].

Previous researches have focused on modeling of the rescue during NDs. However current technological developments allow envisioning systems approach that includes modeling of all aspects of an extreme event, from its impacts on the resources, population to the required response by the involved agencies. ABS approach can help to model and simulate these aspects and it allows simulation designer to model different levels of representations, such as individuals and groups of individuals. Hence, agent-based modeling allows capturing the dynamic nature of the EE and facilitates the study of numerous resource coordination associated with the interaction of multiple teams [18].

#### Agent-Based EE Frameworks

Even if, agents are used in the simulation of EEs, few researchers have proposed a framework to support both the design and the implementation of the EE simulation. Two studies are presented hereafter:

* ABDiSE (Agent-Based Disaster Simulation Environment) is a framework that provides model elements and tools to support the modeling and the simulation of different types of natural disasters such as fires, floods and debris flows. This tool describes how agents move, attach, and interact with each other and with their environment [9];
* D4S2 (Dynamic Discrete Disaster Decision Simulation System) is a comprehensive decision support system to simulate the large-scale disaster responses. This model has a specialized architecture designed for decision makers who can be public safety service officials such as fire and police [25]. More precisely, the proposed architecture integrates several models such as: an agent-based simulation model, a geographical information system (GIS) data bases, and a rule based system for responders and optimization modules to create a hybrid system of agent-based and discrete simulation components.

#### Agent Oriented Frameworks

The organizational modeling in multi-agent systems is based on the management of a process metaphor that underrates the organizational structure [20]. A more general study of agent oriented software engineering methodologies, undertaken in order to find conceptual and operational solutions, has confirmed that organizational issues were added to the actor approach. This approach is the basis of a methodological framework for helping the domain experts to design their models in their own language, as well as transitional agent-based models which are used to produce the distributed simulation model on which experiments are conducted [15]. Methods like GAIA [26], CRIO [7], MOISE+ [10] or the Luis Antonio’s work [1], provide only a part of the solution for the required objectives. Most of these approaches use the notion of roles in order to promote the flexibility in the design process, even with different abstraction or hierarchical levels. As an abstract view of the distributed organization, roles can be combined and associated to the agents' specific architecture, from complex information processing units (i.e. with deliberating capacities) to simple programmable units (reactive agents or state-machine like automata).

#### EE Specific Models

There exist other EE models that do not use the agent approach as a method for modeling and simulating complex problems, such as emergency responses, evacuations, fires, traffic events, earthquakes and flooding. Among these approaches, we mention:

* Emergency Response Framework [11]: this framework allows the integration of modeling, simulation, and visualization tools for emergency response. The development and implementation of this framework should significantly improve the nation’s capability in the emergency response area.
* Buildings evacuation models: there are more than 26 models that have focused on simulating building evacuations. Many of these models are used to simulate evacuation procedure from different types of structures. Featured models include: EVACNET4, WAYOUT, STEPS, PedGo, PEDROUTE, Simulex, GridFlow, ASERI, FDS+Evac, Pathfinder, SimWalk, PEDFLOW, buildingEXODUS, Legion, SpaceSensor, Evacuation Planning Tool (EPT), MassMotion, PathFinder, Myriad II, ALLSAFE, CRISP, EGRESS, SGEM, Evac/FDS, Massegress, Hidac [14].
* Traffics model: there are three main approaches for the modeling and simulation of traffics: T
  + The macro simulation approach, also referred to as macroscopic [8];
  + The micro simulation approach also referred to as microscopy [2, 13];
  + The mesoscopic approach [3] that is widely used in the economic research and studying patterns of movement.

#### Limitations of the Presented Methods

From the above literature review, we found that the different presented models can be improved at different levels:

* The different presented methodologies do not take sufficiently into account the purely organizational aspects of an EE, i.e. explicitly including the structure and organizational dynamics, particularly those related to the behaviour of actors or agents, behaviours generally associated with multiple roles.
* The presented methodologies do not take into account observable and indicators specific to the organization of the natural disaster. Observables and indicators are data and information used in ongoing decision processes, which need to be highlighted in the simulation results.
* The presented models lack of aspects of evaluation. In evacuation modeling, validation refers to a systematic comparison of model predictions with reliable information [28]. Model predictions are dependent upon the data and codes of the evacuation model and the user of the evacuation model. The lack of suitable experimental data to feed the evacuation modeling causes a challenge.
* The fourth element to be improved is related to the presentation of occupants in the evacuation models. Accurate occupants’ representation based on comprehensive anthropometric data and human performance and behaviour should be used in evacuation modeling to provide additional level of validity to the models. Otherwise, building codes and standards should be reformed according to the dynamic changes of individuals’ ages and sizes.
* The interoperability between emergency response modeling and simulation applications is currently extremely limited such as for example the interoperability between different models such as fire model, evacuation model).
* The cost of transferring data between emergency response simulation software applications is often very high.
* The emergency response organizations usually do not have the technical expertise or the time for building simulation models [11].

Therefore, this study proposes a solution to overcome some limitations related to organizational dynamics, interoperability, and transferring data in order to allow modeling / simulation of more "corporate" management after an EE occura.

# A methodological framework for EE organizational aspects modeling and simulation

The proposed modeling approach is based on an incremental process to deal with the complexity in the modeling and implementation simulation process of EE. The proposed approach relies on a gradual increasing of the level of details in the models. The real system is represented by a domain model of EE (e.g. an UEML model -Unified Enterprise Modeling Language -www.ueml.org) to represent the organizational aspects. In our previous work, an organizational methodological framework for modeling a complex system was proposed, which was according to two main abstraction levels: a conceptual and an operational levels [20]. Using the domain model provided by the domain expert, a simulation model is built step by step. The conceptual level proposes concepts and models helping to grasp the complexity of the problem and its simulation objectives. Whereas the operational level involves the implementation of the simulation model which includes the software integration issues. The different models and the transition to agent-oriented modeling and simulation in our methodological framework are presented in Fig 1 (refer to [20] for further details).

The modeling of the organizational Concepts is engaged through a dialogue between the domain expert and an agent-knowledgeable modeller. Identifying the active entities and their organizations from the domain model produces an actor model. The modeller translates/abstracts the domain model into a Conceptual Organizational Model based on (hierarchical) levels, actors, roles and groups named Conceptual Role Organizational Model (CROM) (Fig. 2) (refer to [20] for further detail). This stage highlights the organizational structure of the EE as wells as the structural and dynamic relations between the entities composing the EE. Following this conceptual model, an agent-based model is produced on the basis of observables that the user needs to obtain from the simulation, building up the route toward the implementation of the simulation.

|  |
| --- |
| Fig 1.A Methodological Framework for ND  The software designer details the Conceptual Agent Organizational Model (CAOM) by associating a conceptual agent with a software agent architecture (e.g. BDI - Believe, Desire, Intention - [23]) and specifying their behaviorus (e.g. an UML - “Unified Modeling Language” - activity diagram for a reactive agent) and interactions (e.g. AUML - “Agent Unified Modeling Language” - sequence diagram [21]), resulting in an Operational Agent Model (OPAM). The implementation of these models in a simulation(s) environment results in an ABS system that can be executed. The observables that are related to the organizational structure of a real system are not described in the design model [15]. They are only mentioned in the multi-agent system model, i.e. one step before the implementation stage. It is necessary to describe them earlier in the modeling process (at a conceptual and operational level) as they may induce different modeling requirements.    Fig. 2. CROM Metamodel [refer to [20] for further detail] |

Our work can be summarized as the enrichment of the proposed methodological framework, which would include the specification of the translation process to ensure the transition between the CROM models, CAOM and OPAM (proposed in the methodological framework models) [20] using MDD (Model Driven Development [6] or MDA (Model Driven Architecture) [17]. Further, a decision making criteria has been proposed for the transformations, however more extensive experiments are required to define explicit rules which facilitate the work of modellers and computer scientists. As for the concepts and techniques of model-driven engineering, this could be implemented using tools such as ATL [13]. Another objective was the consideration of the organization in the simulation of EE which involves the integration truly dynamic dimension of the organization by making more explicit organization's image that offers [22] by combining A & A [19] and Moise + [10].

# Dynamic EE organization

In this section, we solve one of the problems presented in section 2.4, which involves the integration of truly dynamic dimensions of the organization in the methodological framework proposed in [20]. The dynamic organization is composed of levels that include one or several groups. Each group contains the actors of the CROM model (presented by the agent or a group of agents in the CAOM model) where each agent can play several roles and a role can be played by different agents. The organization, the group and the agent can generate observable (quantitative or qualitative). An observable is characterized by the activity it monitors (quality, cost), its quantitative or qualitative nature that requires defining its measuring units and the authorized values (whole or real number if quantity, list of values if qualitative) and finally its dated value. A role can provide services to other roles in the same group. Relationships can be developed between agents; every agent may receive an invitation to join a group such as police, fireman, first aid, evacuation or traffic. An agent can communicate with another agent when needed to own business process (collaboration). In this case, agents are invited to join a group to meet some order. During the communication process, all agents are competing with each other that may orient their behaviour, their business volume and their capability. Also, during the modeling and simulation of EE, all the agents are able to ensure the success of the final management plan and collaborate to resolve problems. Agents can join or leave a group when the requested tasks are completed. Then, groups can be automatically created in case of two agents intend to communicate in a new group and vice versa. In the case of a group that contains two agents that have completed their work, this group can be terminated. Therefore, in our methodological framework, the organization can be changed during the simulation including agents joining and leaving groups, changing the agents’ behaviours and rules, and removing/adding groups.

# Model transformation using model driven architecture: From CROM to CAOM

In our research we propose to use MDA (Model Driven Architecture) to transform models in order to ensure the interoperability between the proposed models and transferring the data between these models without any loss of information (Limits presented in the section 2.4). The MDA approach can be seen as a "pragmatic" framework for the implementation of the transformational approach. It advocates the development of software for model transformations. The basic idea is to separate the specification of the functions of a system for the implementation of these functions on a specific platform. Thus, during the development of a system, the desired functionality must be specified independently of the platform (i.e. business model). The platform must also be described by models. We choose a particular platform for the system and the Platform Independent Model (PIM) is converted into a Platform Specific Model (PSM). MDA proposes getting into abstraction by manipulating models and building applications by processing models. MDA offers different approaches to transform models.

The approach to transform the PIM to PSM is based on two steps: first mark the PIM, and second transform the PIM. A brand is a concept of a platform and it is used to mark a PIM to indicate how the element should be changed. These marks are defined in a model that describes the target platform. When a platform is chosen, the brands associated with it are then used to mark the elements of PIM to guide transformation to PSM. A marked PIM is then transformed to get the PSM.

|  |
| --- |
| Fig. 3.Transformation model: CROM to CAOM  In Fig. 3, we illustrate the transformations between CROM and CAOM models using MDA. Firstly we marked the PIM model (CROM), and then transform it to PSM or CAOM model. The principle transformation task is to decide about the role that should be included in the CAOM model. Roles can be combined into one or several agents, according to the kind of behaviour that is expected to be studied (simple or complex). The second task of this transformation to CAOM model is related to the transformation of the relationship between the CROM actors. Thus, CROM relationships are transposed into agents as interactions while keeping their classification; we used observables to filter the reactive agents and deliberative agents. Then, we transform the conceptual model (CAOM) to an operational model. The operational model provides a solution for implementing the conceptual model (CAOM). This step has led to the development of an operational model based on agents and that includes the choice of the agent architectures. For the representation of agents and their behaviour at the operational level, we propose a modeling approach that allows differentiation between reactive agents and deliberative agents.    Fig. 4. CAOM Metamodel [refer to [20] for further detail]  The transformation between different models (CROM, CAOM, OPAM) can be considered as a new communication tool between modellers and developers. The objective of CAOM is to specify the behaviour of each CROM actor. It involves “filtering” the CROM model to only retain actors that have some interest for the simulation along with the desired level of detail of their behaviours. Thus, it highlights the quantitative / qualitative observable data relevant to the objectives of system representation. The following meta-model in Fig. 4 shows these concepts that form the building blocks of a CAOM model. The above meta-model is obtained after the transformation of the CAOM. |

# An Agent-based Software Architecture

The simulation of the operational model, produced after several stages of models refinement, assumes the existence of a software infrastructure that supports heterogeneous simulation models. In addition, it should ensure the integrity of the distributed simulation (of two or more software environments) while providing the desired simulation data (observable). In this section, firstly we present what requirements rise up from these objectives, before introducing the general architecture of an agent and the organizational oriented simulator.

#### Architectural requirements

This section addresses simulations integration and interoperability issues, viewed as the management of data and event dependencies between simulators. Considering the complexity of such task, we combine different integration approach: FIPA (Foundation of Intelligent Physical Agents) specifications on agent-based software integration [29], HLA specification on distributed simulation integration [30], in order to redefine initial ad-hoc Actor simulation architecture [15].

FIPA proposes to agentify software services in order to separate the discovery and selection of services from the actual service call. Interaction protocols are defined to support the chain of actions that agents follow to track and execute the software distributed over an open environment. It is a general software integration approach that, however, does not deal with data sharing and time synchronization at a conceptual or software level.

HLA, an IEEE standard, is totally dedicated to the management of distributed simulations HLA does not propose a software implementation or consider the internal structure of Simulators (Federate). Its reckoning by the simulation community has resulted in numerous implementation and adaptation to different application domain, including Natural Disasters simulation [31]. A Distributed Simulation is seen as a Federation of Simulators, coordinated by a central unit - the RTI (Real Time Infrastructure) – exchanging data and instantiating an Object Modeling Template (OMT) in respect with simulation rules which maintain the integrity of the global simulation (data format, time synchronization, events causality chain…).

There are also, several research works that have been proposed to solve different problems of interoperability such as asynchronous of communication and agent-based communication language. Below are several architectures we studied:

* An Agent Platform for Reliable Asynchronous Distributed [32]: in this platform the authors introduce a distributed communication model based on autonomous software agents. Agents act as software components that can migrate from node to node.
* Agent-Based Middleware for Web Service Dynamic [33]: in this architecture the author aims to construct an agent-based middleware for Web service dynamic integration based on Peer-to-Peer networks to facilitate the integration of optimal quality of Web services for application integration.
* XML-based Mobile Agents [34]: the authors present a mobile agent system based on the use of XML.
* An Agent-Based Distributed Smart Machine [35]: the authors present a software agent based technology to enhance remote tool service system by developing related remote service ontology to manage its smart and distributive characteristics.
* An Agent XML based Information Integration Platform [36]: an Agent/XML based information integration platform is proposed in order to integrate process operation systems effectively,. The subsystems are encapsulated as Agents-based on XML. The encapsulation of different subsystems was implemented through Agent technology.
* A Cross-Platform Agent-based Implementation [37]: the authors present a cross-platform, networked implementation of control tasks based on software agents and streaming technologies.
* FACL (Form-based ACL) [38]: the authors describe a multi-agent framework and an Agent Communication Language (ACL) for the Mu1tiagent-Oriented Office Network (MOON) systems that are distributed systems of E-commerce. The multi-agent framework is a Java application framework and includes a form-based ACL (FACL) as a common protocol for passing application forms.

Looking at these different platforms, we found that there are several limitations: 1) these platforms support only a small number of execution and are limited to link with complex applications (Like JASON Platform [39,40]); 2) they are not open to no-agent software environments; 3) they cannot keep track of the execution; and 4) they have a low level of autonomy.

To overcome these limitations, our approach to EE simulation considers heterogeneity of agents and behaviour as the consequences of the domain which expert observable choices and not necessarily the nature of the EE entities. Therefore the simulation deals with heterogeneous complex behaviour which the simulation framework must integrate.

#### Software Architecture for Modeling and Simulation Agent-Based (SAMoSAB)

The objective of this section is to propose a software architecture that facilitates the production of these simulations by integrating the functional requirements and software related to the simulation of EE.

The different stages of the methodological framework presented in the previous section, lead to an operational modeling crystallized by OPAM model and characterized by:

* The refinement of the multi-agent organization split into two separate environments: one called cognitive (or deliberative) simulating decision-making process, and the other so-called reactive simulating single process or highly automated
* Specification of the agent’s behaviour: in languages appropriate to the granularity of the agents, it describes how agents should behave during the simulation without prejudice to the way they are actually implemented (programming language, simulation language, and environment).
* Specification of the interactions between agents: This results in the dynamic simulation. These interactions will present issues set out according to different agents that are involved in the same society or two different societies (reactive and cognitive).

As a first step toward generality, we have considered two simulation environments integrated through a mediator. The basic idea was to identify and isolate the simulation functions that ensure the simulators integration. As shown in Figure 5, the architecture is divided into the following component elements:

* **Simulators:** our generic architecture is open to different software platforms to simulate different behaviours (Simple or Complex Behavior).
* **Mediator:** the mediator realizes the transmission of information (message, signals, objects, data…) while keeping simulation specific constraints respected (for e.g. time synchronicity between both environments). The mediator role in the integration process is synthesized into five services presented in Table 1.

1. description

|  |  |
| --- | --- |
| **Services** | **Description** |
| **Agents**  **Management** | Classical agent life-cycle management, this module manages the birth and death of agents, etc. |
| **Communication Management** | A basic service in Multi-Agents Systems, this module manages the agent directories (address and capabilities), as well as the logical routing of messages or events. |
| **Organizational Model Management** | This module manages the organization dynamics: group creation, subscribing and unsubscribing to groups… |
| **Interoperability**  **management** | Responsible at the software level for interactions between simulations. It can rely on APIs to route physically message, events, data between the simulators, or clock synchronization signals. |
| **Time**  **Management** | Ensures time is managed coherently in the simulators. Depending on the time management strategy it controls the execution of the simulator (for example, pauses a simulator while response is computed in another simulator). |
| **Indicator Management** | The aim is to produce the indicators characterizing the observable defined in the conceptual modeling of the ND studied. |

* **Connector System:** This element is used to ensure the interoperability between agent and non-agent platforms, which are situated in distributed systems; these must be designed to satisfy the requirements: transport of information, description of dependency, interoperability between different type of simulators, and format of information. According to the properties of the distributed connectors, we provide a description of connectors in a distributed system: “Connectors [41] link distributed components by using some formatted information according to the dependency (relationship) between components.” In the light of this description, we propose a model “Connector Stack” which comprises three layers to design and analyze connectors. It consists mainly of a transport layer, dependence layer, and presentation layer. The Transport layer is situated in the base of the connector and is responsible for the basic delivery of messages in a network. It is also used to support the dependence or relationship between components. The Dependence layer describes the dependency of components in detail. To fulfill the goal, we can use object design pattern or architecture pattern to implement it. Most of the patterns can provide the loosely coupled dependency mechanism for system, such as the Publish-Subscribe. The presentation layer is responsible for building the transferred information depending on the type, format and amount of the information. We can use existing Markup language such as XML or a customized format defined by developers. “Connector Stack” provides a way to design and analyze connector by dividing the connector into separate layers. Each layer is an important part of connectors. They affect each other and are also independent of each other in some ways.
* **Database and libraries:** a number of different databases and libraries can be used to drive the simulation. The libraries will include the event scenarios and the related files required by the multiple simulators.
* **Data Files:** These files are important for understanding the entire concept of the integrated simulators. The files predict a mechanism for configuring the simulator modules and sharing data between them. We have used different types of files such as, XML, Excel and Oracle to encode the data. These files contain the executable data to be processed by the simulation. These files also contain the descriptive text that is intended only for human interpretation. We have also added the links between different types of data required for the interaction between different simulators.
* **Supervisor Simulation:** this element is responsible for synchronization of all the simulators. It configures the scenario through synchronized initialization of all the simulators from data contained. Also, coordinating the execution of the various simulators during a simulation run; and outputs simulation reports.
* **User:** this element provides capabilities to create, modify the scenario data files, manage the display screens for configuring the system, observe the simulation runs, and display results. It is also responsible for the generation of simulation results.
* **Visualization:** this element provides the graphic representation of the scenario and the events.

However, while keeping in mind such objectives, we have chosen to test our propositions by starting with two “specialized” simulation environments. SAMoSAB is thus presently composed of: i) the JASON platform; ii) the JADE platforms, iii) a mediator, and iv) connector.

The JASON platform is adapted to the development of BDI (i.e. deliberative) agents [39] [40]. It is an extended interpreter [42] of AgentSpeak [44] a BDI programming language allowing complex behaviour modeling. The JADE platform (Java Agent Development Framework) [43] is also a FIPA compliant Agent Oriented Software Engineering tool implemented in Java. It proposes a framework for agent management (agent directories, communication management). Agent internal structure is open and left mostly to the programmer initiative. The mediator, see [43], supports the simulations integration by proposing generic services the more independently as possible of the simulators architecture. JASON is used to implement and simulate decision-making processes, whereas JADE deals with simple agent behaviorus. Agents from both environments must interact; the mediator realizes the transmission of information (message, signals, objects, and data) while keeping simulation specific constraints respected (for e.g. time synchronicity between both environments). A Database is also included to capture the model parameters, record simulation data and results analysis. It is accessed by the simulators and the mediator. Figure 5 summarizes the general architecture of SAMoSAB.

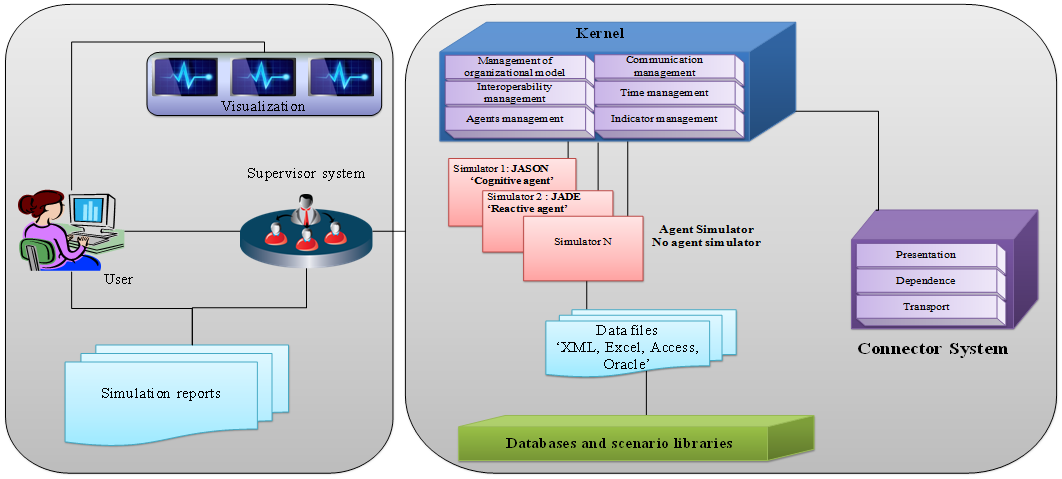


Fig .5. SAMoSAB architecture

#### Agent modeling and interoperability

Current SAMoSAB environment contains several type of agents: i) Deliberative agents, developed in JASON, that implement Natural Disaster decision-making processes i.e. ND entities whose behaviours produce complex observables; ii) Reactive agents, developed in JADE, implementing basic behaviours; and iii) Service agents, i.e. agents not directly concerned by the simulation models but supporting the simulation process.

Table 2 summarizes the different types of agents and their roles in SAMoSAB, some of them are provided by the JADE Platform.

1. Agents description

|  |  |
| --- | --- |
| **Agent** | **Description** |
| **AMS Agent Management System** | Manage agent life cycle, as well as “white pages” directory, i.e. the list of the agents name and their communication address. |
| ***DF* –**  **Directory Facilitator** | Provide a “Yellow Page” service. It records agent roles, capabilities and may answer requests for other agent directory needs. |
| ***ACC* – Agent Communication Channel** | Routes messages from one agent to another, independently of the platform of both agents. Implements for this purpose the IIOP protocol. |
| ***IAg -***  **Indicator Agent** | Is associated to an indicator: It provides computational facilities to produce the value of aggregated indicators. Thus it agentifies the observables identified in the conceptual models. Indicator agents are also categorized depending on the type of indicator they represent (Activity, Productivity, and Quality). |
| ***DSA***  **Data Source Agent** | DSA Centralizes the source of data in a group of agents, and is responsible for finding the agents that have the required information. DSA then regroups and sends these data to the right Indicator Agent, an IAg is needed for exploiting these values. |
| ***GMA***  **Group**  **Manager Agent** | Manages a group i.e. allows an agent to play a role in the group, as well as represent the agents in the group for specific requests. For e.g. If an IAg needs a particular type of data, the Group Manager will locate the agents producing that data. |

# An Illustrative SAMoSAB implementation

Figure 6 illustrates a software architecture supporting our methodological and “simulation-related” requirement. As exposed in the previous section, this SAMoSAB implementation contains Jade Agents, JASON agents, non-agent platform, and a mediator in charge of the interactions.

The “simulation model” presented in Figure 6, results from applying our methodological approach i.e. progressive translation of the CROM and CAOM models of the case study presented in [20]. It is composed of 2 groups describing a simplified ND organization structure. Communication between three platforms is done through messages. Therefore, a mediator layer (denoted Kernel) and connector system ensures the communication link between different platforms (“physical” interoperability is simulated in this case as both are FIPA compliant environment). Note that the mediator is presently developed as a group of specialized agents.

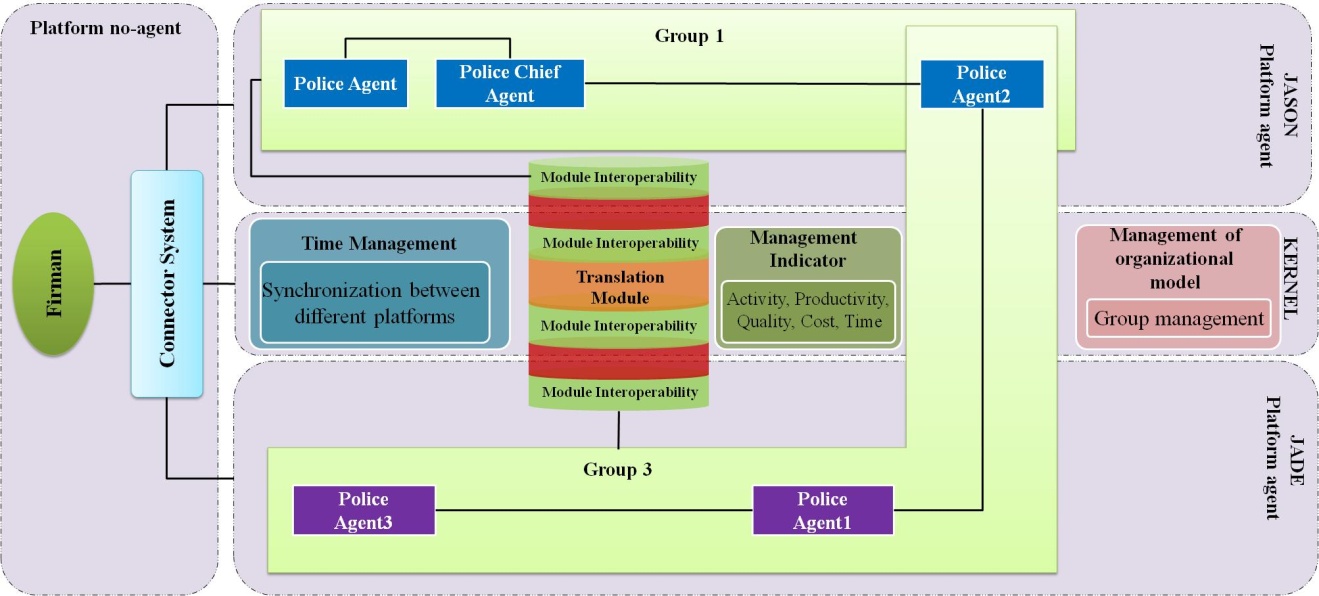


Fig. 6. SAMoSAB architecture illustration

# Conclusion and future work

Natural Disasters, as a subtype of EEs, have resulted in the mortality of three million people and affected the lives of 800 million people worldwide. These have caused diseases as well as serious economic losses and homelessness. The organizational structure and the policies are mostly neglected while simulating a real emergency activity.

In an agent-based ND simulation, we have presented an organizational oriented methodological framework, which permits modeling and simulation of ND organizational aspects. It allows observables of different level of detail while reproducing the ND behaviour according to desired observables. This methodological framework is structured according to a conceptual and an operational abstraction levels. At the conceptual level, the modeling is based on a Conceptual Role Organizational Model (CROM), which is then refined into a Conceptual Agent Organizational Model (CAOM). At the operational level, modeling is mainly based on the Operational Agent Model (OPAM).

This framework allows the study of the impact of a specific ND organizational structure and its related management policies on ND performance. Based on a ND expert modeling of a particular ND, an organization/role oriented (CROM) and an agent-oriented (CAOM) conceptual model help in designing a simulation model, which will reproduce the ND global and local behaviours. These conceptual models are defined independently of particular agent architecture or even on specific software architecture but propose transitional steps to guide their development.

In this chapter, we focused on the proposal of an open software architecture supporting the transformation of the conceptual model into an operational model by generalizing the previous “hard wired” architecture inspired by previous agent-based integration framework. This architecture can be seen as the interaction between different simulation platforms (Agent Platform and No Agent Platform). We showed how different types of agents - deliberative and reactive agents - can interact during simulation as well as the role of some service agents (group manager, indicator and DataSource Agent) supporting this simulation. Development is currently based on the interaction between the JADE platform (for the reactive agent) and the JASON environment (for the deliberative agent).

In our future work we account to work on several points: real data collection in order to have more accurate results Simulation) (example: data structure fire or other EE). These simulations are the first goal, a validation of the operationalization of the methodological framework for modeling and agent oriented simulation, taking into account explicitly the organizational aspects of natural disasters. These simulations should also allow us to validate the software architecture proposed, architecture for the implementation of the previous methodological framework and the execution of simulations. To illustrate the interest of our approach to modeling and simulation oriented agents for the management of EE, we also propose to explore different contextual scenario management of natural disasters, such as building fire, earthquake, etc. The different results may well show the interest of our tool in understanding the behavior of a EE.

# References

1. Antonio, L., D”Amours, S., Frayret, J.M., “A methodological framework for the analysis of agent-based supply chain planning simulations”, SpringSim '08: Proceedings of the 2008 spring simulation multiconference, Society for Computer Simulation International San Diego, CA, USA, (2008).
2. Ben-Akiva, M.E, Koutsopoulos, H. and Mukundan, A. A dynamic traffic model system for ATMS/ATIS operation. IVHS Journal.. 2. 1-19. MITTNS (1994).
3. De Palma, A. and Marchal, F. and Nesterov, Y.,. A modular system for dynamic traffic simulations, (1996).
4. Erceau, J. et Ferber, J. ‘L’Intelligence Artificielle Distribuée’, La recherche, vol. 22, pp. 750-758, (1991).
5. Ferber, J. Les Systèmes Multi-Agents, vers une intelligence collective, InterEditions, (1995).
6. France, R., Ghosh, S., Dinh-Trong, T. and Solberg., A., Model- Driven Development Using UML 2.0: Promises and Pitfalls. Computer, 39(2), Feb. (2006).
7. Gaud, N., Galland, S., Koukam, A.,Towards a Multilevel Simulation Approach based on Holonic Multi-agent. Published in the 10th International Conference on Computer Modeling and Simulation (EUROSIM/UKSiM‟08),pp.180–185, England, (2008).
8. Helbing, D. and Treiber,M.,Numerical simulation of macroscopic traﬃc equations,Computing in Scs & Engineering 1, (1999).
9. Hsu T. L., Liu J. W. S. An Agent-Based Disaster Simulation Environment. RITMAN Workshop 2012, Taipei, Taiwan (2012).
10. Hübner J. F, Sichman J.S., et Boissier O., (2007), Developing Organised Multi-Agent Systems Using the Moise+ Model: Programming Issues at the System and Agent Levels,Int. J. Accounting, Auditing and Performance Evaluation,1(3/4):370–395
11. Jain, S. and C.R. McLean, “A Framework for Modeling and Simulation of Emergency Response,” Proceedings of the 2003 Winter Simulation Conference, Dec. 7-10, New Orleans, Louisiana, 1068-1076,( 2003).
12. Jouault, F., Kurtev, I., Transforming models with ATL, in: Proceedings of the Model Transformations in Practice Workshop at MoDELS 2005, MontegoBay, Jamaica, (2005).
13. Kosonen,I. and Pursula, M.,A simulation tool for traffic control planning. IEEE Conference Publication Number 320. Third international Conference on Road Traffic Control. vol 320 pp 72-76, (1991).
14. Erica D. Kuligowski, Richard D. Peacock. A Review of Building Evacuation Models, Fire Research Division Building and Fire Research Laboratory, (2006).
15. Labarthe, O., Espinasse, B. ,Ferrarini, A., Montreuil B., Toward a Methodological Framework for Agent-Based Modeling and Simulation of Supply Chains in a Mass Customization Context, Simulation Modeling Practice and Theory International Journal (SIMPAT), vol. 15, n° 2, pp. 113-136, February (2007).
16. Lampert, R., Agent-based modeling as organizational and public policy simulators. Proceedings of the National Academy of Sciences of the United States of America, 99:7195-196,(2002).
17. MDA: Model Driven Architecture Guide Version, [www.omg.org/cgibin/doc?omg/03-06-01](http://www.omg.org/cgibin/doc?omg/03-06-01), Juin (2003).
18. Monteiro T., Anciaux D., Espinasse B., Ferrarini A., Labarthe O., Roy D.,The Interest of Agents for Supply Chain Simulation, in: Wiley-ISTE (Ed.), Simulation for Supply Chain Management', C. Thierry–A. Thomas–G. Bel, septembre (2008).
19. Montagna, S., Ricci, A., et Omicini, A., A&A for modeling and engineering simulations in Systems Biology, International Journal of Agent-Oriented Software Engineering- Vol. 2, No.2 pp. 222 – 245, (2008).
20. Mustapha, K., Tranvouez, E., Espinasse, B. & Ferrarini,An Organization-oriented Methodological Framework for Agent-Based Supply Chain Simulation.4th International Conference on Research Challenges in Information Science,France (2010)
21. Odell, J., Parunak, H.V.D.,Bauer,B.,Representing agent interaction protocols in UML,Proceedings of the First International Workshop on Agent-Oriented Software Engineering,CIANCARINI, P.& WOOLDRIDGE, M.,(2001).
22. Piunti, M., A. Ricci, Boissier, O. et Hübner, J.F. (2009) Accéder à une organisation multiagent par l’environnement des agents.
23. Rao A. S., M. P. Gorgeff. Modeling rational agents within BDI-Architecture.in J. Allen & al Ed., Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning. San Mateo, USA, Morgan Kaufmann, Pub, p. 473-484, (1991)
24. Russel, S. et Norvig, P.. Artificial Intelligence A Modern Approach. Pearson Education, Upper Saddle River, New Jersey, second edition (2003).
25. Wu, S., Shuman, L.J., Bidanda, B., Kelley, M., Sochats, K., and Balaban, C. 2007c. System implementation issues of Dynamic Discrete Disaster Decision Simulation System (D4S2) - Phase I. In the Proceedings of the 2007 Winter Simulation Conference.
26. Zambonelli, F., Jennings N., Wooldridge, M.. Developing multi-agent systems: the GAIA methodology. ACM Trans. on Software Engineering and Methodology, 12(3), (2003).
27. Okada N., City and Region Viewed as Vitae System for Integrated Disaster Risk Management, Annuals of Disas. Prev. Res. Inst., Kyoto Univ., No. 49 B, pp 131- 136, (2006).
28. Galea, E. & Gwynne, S..Principles and Practices of Evacuation Modeling. London,UK:CMS Press, (2005).
29. DMSO: “High Level Architecture”, (1998).
30. FIPA, FIPA Contract Net Interaction Protocol Specification, Foundation for Intelligent Physical Agents, www.fipa.org/specs/fipa00029/, (2002).
31. Ounnar, F., Archimède, B., Pujo, P., Charbonnaud, P., HLA Distributed Simulation Approaches for Supply Chain”, in: Hermès Science Europe Ltd (Ed.), ''Simulation for Supply Chain Management'', Hermès Science Europe Ltd, (2008).
32. Bellissard, L. and A.F. N. De Palma, M. Herrmann, S. Lacourte, An Agent Platform for Reliable Asynchronous Distributed Programming. 1999, IEEE: FRANCE.
33. Lin, A. and P. Maheshwari, Agent-Based Middleware for Web Service Dynamic Integration on Peer-to-Peer Networks, in AI 2005, LNAI 3809. 2005, Springer: Verlag Berlin Heidelberg. p. 405 – 414.
34. Steele, R., et al., XML-based Mobile Agents, in Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC’05). 2005, IEEE.
35. Kao, Y.-C. and M.-S. Chen, An Agent-Based Distributed Smart Machine Tool Service System, in 3CA 2010. 2010, IEEE.
36. Li, X., An Agent/XML based Information Integration Platform for Process Industry, in 2010 2nd International Conference on Computer Engineering and Technology. 2010, IEEE.
37. Cervera, E., A Cross-Platform Agent-based Implementation. 2005, IEEE.
38. Chusho, T. and K. Fujiwara, A Formbased Agent Communication Language for Enduser-Initiative Agent-Based Application Development. 2000, IEEE.
39. Bordini, R. and Hubner, J, An Overview of Jason. Association for Logic ProgrammingNewsletter9(3).DOI=http://www.cs.kuleuven.ac.be/~dtai/projects/ALP/newsletter/aug06/nav/articles/article5/fs.pdf, (2006).
40. Bordini, R., Hubner, J., Vieira, R,. Jason and the Golden Fleece of Agent-Oriented Programming. Multi-Agent Programming. pp. 3-37, (2005).
41. Mcheick H., and Qi Y.: Dependency of Components in MVC Distributed Architecture. 24th IEEE (ccece'2011). May 8-11, 2011, Ontario, Canada.
42. Vangheluwe, H., et al. An introduction to multi-paradigm modelling and simulation. School of Computer Science, McGill University, Montréal, Canada, (2002).
43. Serment J., Espinasse B, Tranvouez E., An Agent Integration Infrastructure for the Development of Environmental Decision Support Systems based on Simulation, AIS-CMS International modeling and simulation multiconference, Buenos Aires - Argentina, 2007 ISBN 978-2-9520712-6-0
44. Rao, A,. S.,AgentSpeak(L): BDI Agents speak out in a Logical Computable Language. In W. Van de Velde and J Perram, editors, Proceedings of the Seventh Workshop on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW'96), Jan. 22-25, Eindhoven, Netherlands, no. 1038 in LNAI, pp. 42-55, Springer-Verlag, London, U.K (1996).
45. Boulton, A., Brunn, S.D., & Devriendt, L. (2011). Cyberinfrastructures and “smart” world cities: Physical, human, and soft infrastructures. In Taylor, P., Derudder, B., Hoyler, M., & Witlox, F. (Eds.) International Handbook of Globalization and World Cities. Cheltenham, UK:Edward Elgar.
46. Hollands, R.G. (2008). Will the real smart city please stand up? City, 12(3), 303-320.
47. Giffinger, R., Fertner, C., Kramar, H., Kalasek, R., Pichler-Milanovié, N., and Meijers, E., (2007), *Smart cities: Ranking of European Medium-sized cities.*Vienna, Austria: Centre of Regional Science (SRF), Vienna University of Technology.
48. Hall, R. E. (2000). The vision of a smart city. In Proceedings of the 2nd International Life Extension Technology Workshop, Paris, France, September 28,
49. Harrison, C., Eckman, B., Hamilton, R., Hartswick, P., Kalagnanam, J., Paraszczak, J., & Williams, P. (2010). Foundations for Smarter Cities. IBM Journal of Research and Development, 54(4).
50. Weber, E. P., & Khademian, A. M. (2008). Wicked problems, knowledge challenges, and collaborative capacity builders in network settings. Public Administration Review, 68(2), 334-349.

1. http://en.wikipedia.org/wiki/Smart\_city [↑](#footnote-ref-1)