Federated Agent-based Modeling and Simulation Approach to Study Interdependencies in IT Critical Infrastructures

Emiliano Casalicchio, Emanuele Galli, and Salvatore Tucci
Dep. of Informatica Sistemi e Produzione
University of Roma "Tor Vergata"
{emiliano.casalicchio, tucci}@uniroma2.it, galli@disp.uniroma2.it

Abstract

Agent-based modeling and simulation (ABMS) is one of the more promising simulation techniques to study the interdependencies in critical infrastructures. Moreover, federated simulation has two relevant properties, simulation models reuse and expertise sharing, that could be exploited in a multi-sectorial field, such as critical infrastructure protection. In this paper we propose a new methodology which exploits the benefit of both ABMS and Federated simulation, to study interdependencies in critical infrastructures. First of all we discus advantages of federated agent-based modeling and difficulties in implementing a Federated ABMS framework. To demonstrate the relevance of our solution we propose an example driven approach that poses the attention on critical information infrastructure. We have also implemented a Federated ABMS framework, which federate Repast, an agent-based simulation engine and OMNeT++ an IT systems and communication networks modeling and simulation environment. A selection of simulation results shown how Federated ABMS could shed light on system interdependencies and how it helps in quantifying them.

Keywords: Agent-based modeling and simulation, federated simulation, critical infrastructure, interdependencies analysis.

1. Introduction

A critical infrastructure (CI) is a physical system that, if disrupted, can seriously affect the national security and the economic and social welfare of a nation. Examples of critical infrastructures include telecommunications, electric power systems, natural gas and oil, banking and finance, transportation, water supply systems, government and emergency services [1].

Within any critical infrastructure there are dependencies among its components and some of them involve humans. Moreover, critical infrastructure are intertwined and heavily dependent on each other [16]. Therefore, in case of disruptions, what happens to one infrastructure can directly and indirectly affect other infrastructures.

To protect the critical infrastructures, it is necessary to study the complex system behavior and the interaction processes among humans and the system components, when they are stressed or attacked, and to understand the emergent phenomena originated by the individual behavior of the infrastructure components. Therefore, as demonstrated by previous work on this subject (e.g., [7, 15, 16]), a challenge is to provide formalisms, methodologies, and tools to model the entire complex system composed of humans and critical infrastructures.

To address this challenging issue, in this paper we propose a new modeling and simulation formalism that exploits the power of both agent-based modeling and federated simulation, and which is referred to as Federated Agent Based Modeling and Simulation (FedABMS).

Agent-based Modeling and Simulation is a promising modeling and simulation technique used to study critical infrastructure interdependencies. ABMS uses a bottom-up approach to model the whole system starting from its individual parts: an agent-based model, is obtained interconnecting agents, i.d. independent systems that autonomously elaborate information and resources in order to define its outputs, outputs that become inputs for other agents, and so on. An agent is an individual entity with location, capabilities, and memory [11].

There are different motivations that have driven us to choose ABMS to study the interdependencies in critical infrastructures: i) this approach allows us to simulate a complex system composed by many subsystems; ii) by exploiting the ABMS features, we can embed the model of the complex system into agents and model such system at different levels of abstraction; iii) the ABMS offers not only a valid support to conduct a what-if analysis, but it also allows to investigate different aspects of the system dynamic, in particular to study the impact of an unexpected perturba-

1550-6525/07 $25.00 © 2007 IEEE
DOI 10.1109/DS-RT.2007.11

tion on the interconnected infrastructures and thus to predict the infrastructure responses to this perturbation.

Federated simulation is a simulation technique which: i) allows to reuse existing simulation model, thus reducing the cost to develop a complex system model; ii) it allows to distribute the execution of the simulation model over a set of nodes (locally or geographically distributed), thus to increase computational power, resource availability and fault tolerance.

The concept of federated simulation is not new and widely used and investigated in military and defense sector [12, 8, 10] (for example the High Level Architecture, actually IEEE std.1516, is the result of a research sponsored by the US Defense Modeling and Simulation Office). Also researcher apply distributed simulation, for example in computer networks and distributed systems design and performance evaluation[14, 4]. The sector which less uses federated simulation is the industry: Boer et al. realize an extensive survey on such topic [2, 3].

Federated ABMS exploits the ABMS capabilities, to model the whole complex system as a set of interacting agents, and the Federated simulation capabilities, to reuse existing models which will help in modeling agents behavior at different level of abstraction. In ABMS the simulation model of a complex system is typically built "from scratch" and the detailed agent behavior is embedded into a unique simulation model (see figure 1 (a)). Of course existing models could be reused, but the integration process requires a tedious re-engineering phase, that could be more or less expensive, depending on the simulation framework used to build the agent-based simulator and the technology used to develop the non-ABMS models. In FedABMS the idea is to use agent based modeling to provide an high level model of the system, representing the infrastructure, the active entity and the environment as interacting agents [5]. The detailed model of each subsystem, represented by an agent, is provided by existing models and simulation software (see figure 1 (b)). For example, in the figure, the detailed behaviour for the agent A1 is implemented by a simulation model running at siteA. Using the federated simulation terminology, the ABMS model and the specific simulation model are the federates, which altogether compose a federation.

The novelty of our approach is twofold: i) we use ABMS capability to give a natural high level description of the infrastructures model, and ii) we exploit federated simulation capabilities to detail the infrastructure models integrating existing simulation models. The proposed methodology explains how to determine federates and how takes advantages of agent-based modeling and simulation features.

The paper is organized as in the following. In section 2 we introduce the concept of Federated ABMS, its advantages, and the methodology to create a federated agent-based model. In section 3 we present a case study that facilitate us the presentation of the proposed approach. In section 4 we deal with design and implementation problems of a Federated Agents-Based Modeling and Simulation framework. A selection of the simulation results, showing the effectiveness of the proposed approach, is presented in section 5. Remarks and future works conclude the paper.

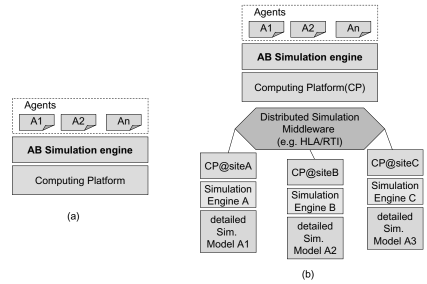

Figure 1. ABMS (a) v.s. FedABMS (b)

2. Federated agent-based modeling and simulation

The rational for FedABMS is to exploit the advantages of both ABMS and Federated simulation.

In ABMS a complex system is modeled as a set of interconnected, cooperative and or competitive interacting agents. An agent is an entity with a location, capabilities and memory. The entity location define where it is in a physical space (geographic region or abstract space, such as Internet). What the entity can perform is defined by its capabilities: perception, behaviors, intelligent reaction, cooperation and autonomy. Finally, the experience history (for example, overuse or aging) and data defining the entity state represent agent's memory. Agents allow to embed the model of a complex system into their behavior. As previously mentioned there are many advantages in using ABMS to study Critical Infrastructures. It provides a natural description of a complex system composed of behavioral entities. By exploiting the ABMS features, we can embed the model of the complex system into agents. ABMS offers a valid support to conduct a what-if analysis and it allows to investigate different aspects of the system dynamic. For example, some specific questions that can be answered are: what is the failure propagation path from the affected infrastructure to the interdependent ones and its propagation time? What are the direct and side effects of a failure? Can the system react to at certain type of failure and how much time does the reaction take? What is the impact of a miss behavior in a recovery plan?

Federated simulation allows simulation model and software reuse, and also it allows to distribute the simulation

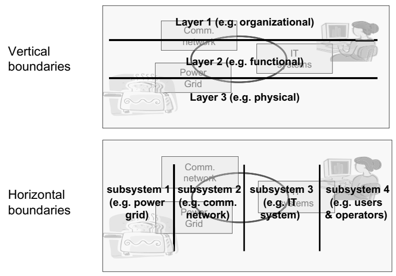

Figure 2. Difference between horizontal and vertical boundaries in federated modeling

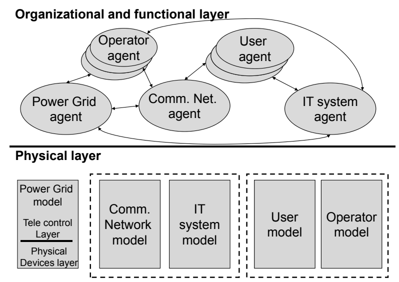

Figure 3. Federates Agent-based model of CI

across multiple nodes. These features reflect in development cost reduction and simulation performances improvement

One of the main issue in designing federated simulation model is "*how to chose the boundaries among federates?*". The partitioning of a complex system model into federates could be done in two direction: defining vertical boundaries and/or defining horizontal boundaries. Let us explain with an example the meaning of horizontal and vertical partitioning. Suppose that we would model a complex system composed of the communication network, the power grid, an IT system, operators and users.

Vertical partitioning models the whole complex system as a set of layers (see fig.2), for example: the organizational layers, which models all the interaction rules among subsystems; the functional layer, which models all the functionalities exposed by each sub-system; and the physical layer, which models the details of each sub-system, for example the architecture of the IT systems, all the links and the devices of the communication network, the devices and transport lines of the power grid, and so on. Each layer is represented by a different model implemented and executed by a specific simulation framework. The different models (federates) are glued together in a federation.

Horizontal partitioning views the whole complex system as a set of sub-system: each sub-system is modeled and simulated using specialized simulation environments. In our example (see fig.2) the power grid, the communication network, the IT system, and so on, are independent models (federates) glue together in a federation.

Obviously the vertical and horizontal partitioning is a recursive modeling concept: models resulting from an horizontal partitioning could be vertically partitioned and vice-versa.

2.1. Federated ABMS methodology

Federated ABMS requires three main steps: 1) system model partitioning; 2) agent-based modeling; 3) agents model refinement.

Let us discuss how FedeABMS could be applied to critical infrastructures. First of all we define a vertical boundary to separate the organizational and functional model of the whole system, from the physical model of each sub-system. Then, at the physical layer, we draw horizontal boundaries among the sub-system (see fig.3).

The ABMS take key role in the modeling of organizational and functional layer. Specialized models and simulation frameworks are used to model the physical layer.

In our example (see fig.3) the power grid, the communication network, the IT system, users and operators are modeled as agents. The agent's capabilities and behavior are used to model interaction rules, and system functionalities. A user could interact with an IT system requesting a particular web page, or he/she interacts with the power grid requesting a specified amount of electricity. The communication network provide functionalities to route messages, and the IT system provide functionalities to store and to visualize documents or to publish a web site, and so on.

The agent-based model is federated together the specific models for the infrastructures and for the actors. An event-driven simulation environment, such as OMNeT++ (or NS2), could be used to model both the details of the communication network, and of the IT system. For example, the IT system mentioned above is composed of resources such as disk, CPU, memory, network interface card. To implement its high level functionalities (modeled in the agent-based model), the IT system use its resources. A detailed simulation model of the IT system could be defined using, for example, a queue network. The Power grid could be modeled using specific framework such as the Load-flow electrical grid simulator (e-Agora) or, as in the EPOCHS [9], using PSCAD/EMTCD to simulate electromagnetic transients and PSLF to simulate electromechanical transients. Also the behavior of operators and users could be detailed using specialized models.

Applying the recursive property of federated agent-based modeling, the power grid could be again modeled as two interacting sub-systems (as shown in the figure): the telecontrol system, modeled with an IT modeling and simulation framework; and the power supply system, modeled with a power grid modeling and simulation framework.

3. The case study

The proposed FederatedABM&S approach is general enough to be applied to any critical infrastructure case study. However, for clarity and easy of presentation, we describe how to apply our methodology through an example-driven approach, which considers as a critical infrastructure application the Information System for Civic Emergency Management (IS4CEM) [5], which, in case of some natural disaster or special event, provides information about the health care centers availability, the transportation network availability, and the event evolution.

We chose such case study because the ubiquity and pervasiveness of information technology in any sector, from health care to water supply.

There is a generic service requestor (SR) that accesses the IS4CEM, which is hosted and published by a service provider through a wired network. The SR can be a citizen asking for help (in the following, wounded) or a succorer and she/he uses a set of critical infrastructures to access the IS4CEM. Specifically, the SR uses a wired or a wireless connection to the communication network; she/he can use the power grid to power its connecting device; finally, she/he may use the transportation system to provide first aid to other service requestors or to reach a Health Care Center (HCC) to obtain assistance.

The IS4CEM is an information system that runs on a service provider platform and provides various information to the citizens. It also operates as a broker for first aid requests by selecting the most appropriate HCC to be reached (e.g., by taking into account the transportation system and HCCs status). The IS4CEM is accessed through the communication network and uses the information system of the transportation network to get information about its availability. The IS4CEM does not directly use the transportation network; however, since the transportation network may be used to provide assistance to the IS4CEM, the latter depends on the transportation network. Finally, the IS4CEM relies on the power grid for its functioning.

Each HCC (specifically, its information system) uses the IS4CEM to obtain availability information on the other HCCs and to get information about emergencies. Each HCC uses the communication network and the power grid.

The transportation network is used by the service requestor and by all citizens. Its information system dialogs with the IS4CEM to update the transportation network status. Finally, the environment influences all the system components.

Details on how to model such system using AMB&S

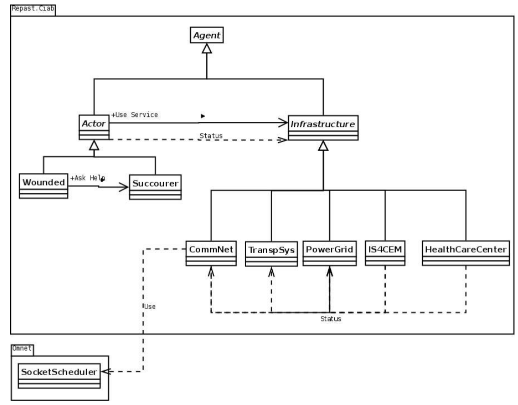

Figure 4. Class diagram of IS4CEM simulator

.

could be found in our previous work [5]. In the rest of the paper we focus our attention on the design and implementation of the federated agent-based model.

4. Design and Implementation of a Federated ABMS framework

The implemented framework integrates an agent based simulation environment, Repast [13], and a discrete event simulation environment for computer networks modeling and simulation, OMNeT++ [17]. Because the main goal of the paper is to demonstrate the applicability and the advantages of the Federated ABM&S approach, rather then to realize a general purpose federated simulation framework, we have decided to built our how implementation. The implemented framework satisfies our requirement and guarantees the basic support to federated simulation, that is event and time synchronization and message exchange.

There are other main reasons that lead us to discard the use of HLA-RTI standard. First of all, there are only two open-source HLA-RTI projects: Open HLA (http://sourceforge.net/projects/ohla/) and Littlebluefrog (http://wiki.littlebluefroglabs.com) which have not yet released stable versions and have incomplete documentation. Moreover it's hard to find open-source HLA-compliant simulators, for example HLA-Repast and HLA-OMNeT projects are not yet in a mature phase. For these reasons and because the prohibitive cost of HLA-RTI implementation and commercial simulator HLA compliant, seems to be an excessive effort using HLA-RTI standard.

We have used Repast to design the model of the organizational and functional layer (see fig. 4). An agent is developed by a Java class, where the behavior are modeled by methods while memory and characteristics by attributes.

OMNeT++ is used to model the "physical" layer[1] of the

[1]Here we are not talking about the network physical layer of the

communication network and of the IT system components of the different infrastructures (that is the servers and communication lines at the power grid, at the health care centers and at the IS4CEM). An OMNeT++ model is obtained composing components and modules. Components and modules are programmed in C++, then assembled into larger components and models using a high-level language (NED).

To federate the two simulators we have developed an interface (on both simulators) that allows time and event synchronization and message exchange (see fig. 4). Repast implements such interface with the CommNet class. All the agents (infrastructures and actors) communicate using the CommNet class methods, being unaware of the detailed model implemented in OMNeT++. In OMNeT++ the interface is provided by the SocketScheduler class, which manages both time and event synchronization and message exchange. The SocketScheduler class re-implements and extends the basic OMNeT++ scheduling functionalities. On the contrary, Repast do not need any change in the scheduler, being the coordinator and time manager for the simulation.

In our scenario we assume that both voice and data rely on the Internet protocols. Each agent has been mapped to a LAN/WLAN OMNeT++ node, which specializes the physical model of the IT system used to connect the infrastructure or the service requestor to the communication network. For example, the IS4CEM server is modeled in the details of the disk, CPU, network interface card components, using a queue network. A general network topology composed of routers and communication links with different capacities is used to model the Internet. However, OMNeT++ allow to specify any kind of network topology.

If an agent wants to communicate with an other one, it calls the method *sendMessage* of CommNet agent, which will prepare a well formatted message and send it to OMNeT++. OMNeT++ will simulate the delivery of the message on the physical network model. When the message will reach the destination, OMNeT++ sends an info message to the CommNet agent notifying that the message has been processed and that it will reach the destination at a specific time. The CommNet agent use the transmission time calculated by OMNeT++ to schedule the event “message received”. When the simulation time in Repast is equal to the delivery time of the message, the CommNet calls the method *receiveMessage* of the receiver agent, thus to effectively delivers the message to the receiver agent. Moreover the CommNet agent has a map of all components in OMNeT++. In this way the agents can control and set the status of every component. This solution allow to, for example, turn on/off a network device when there is a failure in the PowerGrid region which supplies such node, or when an hardware or software failure arises.

ISO/OSI protocol stack but we are talking about the physical layer of the federatedABM&S model, that is the layer which specialize the models of different infrastructures and actors.

4.1. Design and implementation issues

As already mentioned there are three main issues that must be considered in the design and implementation of a federated simulation environment: simulation time management, event synchronization and data exchange. Inspired by the solution proposed in literature, we have define a policy to manage the simulation time; an algorithm for event synchronization and a model for data exchange.

Simulation time management. There are two well known approaches to face time management [6]: *optimistic* and *conservative*. In our framework we have decide to use the conservative approach (the preferred one in federated simulation). We deal with two simulators using the discrete model. In particular, Repast is a time-stepped simulator while OMNeT++ is an event-driven simulator.

Instead of defining a separated server for time management (as in RTI), we assume that Repast manages the simulation time. Before starting the simulation, Repast communicates to OMNeT++ the value for a tick (in seconds); OMNeT++ assumes this value as common Lower Bound on Time Stamp (LBTS).

Repast notifies OMNeT++ when it can process the next event: events processing starts when OMNeT++ receives a notification from Repast and terminate after LBTS seconds, then Repast takes again the control of the simulation.

The synchronization algorithm. We have implemented a distributed synchronization algorithm to manage events synchronization.

Repast, before any simulation step, checks if OMNeT++ simulation is stopped and if there is some agent which requests to send a message. If an agent has requested to send a message, Repast communicates such need to OMNeT++, and it checks if there are messages from OMNeT++. A new message from OMNeT++ indicates the delivery time of a previously sent message; in this case a Repast event has to be scheduled.

Each time OMNeT++ processes a new event, it checks if the next event will occur in a future tick. If no new events will occur in the future, OMNeT++ continues the simulation, otherwise it notifies Repast (sending a synchronization message) to process a new step and it sets the new value for time synchronization. After that, OMNeT++ enters a pause state, until a wake up message by Repast.

The data exchange model. During a simulation OMNeT++ and Repast exchange state information and synchronization messages. The simulators communicate using sockets, that are available both in C/C++ (for OMNeT) and Java (for Repast). A message is a string with a specific semantic. During the initialization phase, both

Repast and OMNet++ create a server that opens a passive socket which will accept the connection requests from the respective clients. If Repast (OMNeT++) wants to send a message, it send a connection request to the OMNeT++ (Repast) server, then it sends the message and close the connection.

There are three type of messages: synchronization messages, communication messages and information messages.

A synchronization message, *SyncMsg*, is used in two way: by Repast to inform OMNeT++ that can complete all waiting events and schedule the next event; by OMNeT++ to request Repast for advancing of one event according to the synchrozation algorithm previously described.

Actors and Infrastructures at the Organizational and Functional Layer communicate exchanging communication messages (*NetPkt*) over the network. Each NetPkt has a specific size determining the delivery time of the message on the real network model and the processing time at the server side model.

Information packets (*InfoMsg*) are introduced to enable the run-time modification of network and IT systems characteristics. In the specific an agent could set a new weight for a link (use by routing algorithms) and could disable/enable a node or a link (i.e. in case of a failure/recovery).

5. Interdependencies Analysis

Starting from the case study presented in section 3, we have studied four different scenarios that consider different crisis which effect the communication network and the transportation system. The scenarios are defined as in the following.

Reference scenario. We suppose that the wounded agents generate a burst of help request at the beginning of the simulation. In this scenario we do not consider any failures on the communication network nodes and any traffic jam. This is the reference scenario and the other add failures and/or congestions on the communication network and transportation systems.

CommNet scenario. This scenario adds faults and congestion only at the communication network nodes and links. We do not model the details of a failure, that could be hardware and/or software or due to absence of power supply. The average time to failure of a node or link is 5 minutes and the recovery time is 10 minutes. Nodes or links subject to failure are chosen randomly.

TranspSys scenario. This scenario adds only traffic congestion (the communication network work properly). The round trip time of a congested route range from 14 to 60 minutes. The routes which will experiment a congestion are chosen randomly and the inter-congestion time is 30 minutes. When a route is congested, it still in this state forever. Route congestion could be caused by failure in the power grid or by many other unexpected events.

Parameter	Value
Num. of HCCs	3
Num. of soccourer agents	10
Avg. route round trip time (rtt)	5 min.
Avg. route rtt (during congestions)	7-30 min.
Route inter-congestion time	30 min.
Num. of wounded	10 - 50
Node/link mean time to failure	10 min.
Node/link mean recovery time	5 min.

Table 1. The main simulation parameters

CommNet+TranspSys scenario. This is the more complex scenario which adds both communication network nodes/links and transportation network weaknesses, as defined in CommNet and TranspSys scenarios.

In all scenarios we fix the number of Health Care Centers, the number of soccurer agents, and the average route round trip time. Each node of the power grid supply a subset of the communication network nodes, HCCs and routes (traffic light). The main simulation parameters are summarized in table 1.

We suppose also that the soccourer agents are uniformly distributed among the HCCs, that is we have about 3 soccourer agents for each HCC. We remember that each wounded agent could be reached from at least one HCC. The transportation network is modeled as a reachability matrix $TN = \{r_{i,j}\}^{n \times m}$, where $r_{i,j} = 0$ means that it is impossible (or prohibitive) to reach the wounded agent j from HCC i, otherwise $r_{i,j} = k$ means that the soccourer agent from HCC i will pick up the wounded agent j using route k. n and m denote respectively the number of HCCs and wounded agents.

In all scenarios the goal is to rescue as much wounded agents as possible, as fast as possible. From this study is possible to know how much wounded agents are rescued and in how much time. The simulation results could help in making decisions that will improve the performances of a first aid service.

Each wounded agent is characterized by an average time to live (ttl), that is the remaining time to live after that the help request is generated. We suppose that: 40% of wounded agents have a $ttl = 12h$ (LOW priority requests); 30% have a $ttl = 6h$ (MEDIUM priority); 20% have a $ttl = 30min$ (HIGH priority) and 10% have a $ttl = 15min$ (VERY HIGH priority). The request priority is used by the HCCs and by the IS4CEM to schedule the aid.

To analyze the scenarios under different stress conditions we vary the number of wounded agents from 10 to 50. After the first request, each wounded agent could generate a new one if she/he is not picked up in the time estimated by the IS4CEM, on the basis of the state information received from

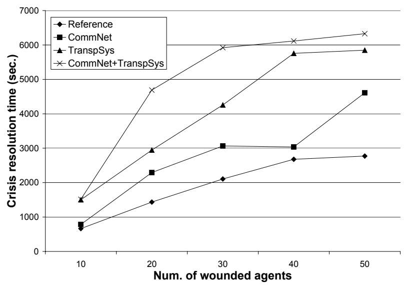

Figure 5. Crisis resolution time.

HCCs, soccourer agents and the transportation network.

We suppose that the crisis is resolved when all the wounded agents are rescued or are died (because not picked up in time or not picked up at all).

We can instrument the simulator to measure any kind of information, from high level, as the number of dead wounded agent, to low level, as the number of lost packets in the communication network or the IT system nodes utilization. While the low level information could be used to get knowledge of the individual infrastructure state, high level information could be useful to determine emergent phenomena or could be used by a crisis management team, that performs what-if analysis, to organize the aid. Therefore we have decided to use the following metrics (and related indexes).

Time needed to resolve the crisis (or crisis resolution time - T_c). In particular we measure the absolute value of the crisis resolution time and, for each scenario, the downgrade of the crisis resolution time (δT_c) respect the reference scenario. δT_c is defined as follow: $\delta T_c = \frac{T_c^i - T_c^0}{T_c^i}$, where T_c^0 is the crisis resolution time for basic scenario and T_c^i is the crisis resolution time for the other scenarios.

Time to rescue a wounded agent (or wounded agent pick up time - T_r). What we measure is the average time that a soccourer agent needs to rescue a wounded agent. Also for this metric we measure the downgrade δT_r respect the reference scenario. $\delta T_r = \frac{T_r^i - T_r^0}{T_r^i}$, where T_r^0 is the wounded pick up time for basic scenario and T_r^i is the wounded pick up time for the other scenarios.

Number of rescued wounded agents, that is the number of live wounded agents at the end of the crisis. In particular we measure the percentage of wounded agents that still live after the crisis.

Number of dead wounded agents. In particular we measure the percentage of wounded agents dead during the crisis. We classify who died without any aid and who died after a soccourer agent arrives.

The trend of the crisis resolution time is reported in figure 5. As expected, T_c degrades when the number of wounded agents increase and also when weaknesses in the different infrastructures are introduced. We point out that the communication network infrastructure is more flexible and adaptive than the transportation system. Indeed the communication network is initially capable to manage the fault of different nodes adapting the packet routing, degrading its performance only when a significant number of link is disrupted, or when the load generated by the wounded agents is too high to be managed in presence of reduced capacity. In our scenarios, faults in the communication network results only in a 27% degrade of crisis resolution time, while the congestion of the transportation system has a more severe impact. Indeed in the TransSys scenario $\delta T_c = 52.7\%$ (see fig.6). We observe an emergent phenomena (unexpected result), when we combine the CommNet and TranspSys scenarios. The measured T_c is lower then the sum of the crisis resolution time measured in the CommNet and TranspSys scenarios, indeed the downgrade is 60.5%.

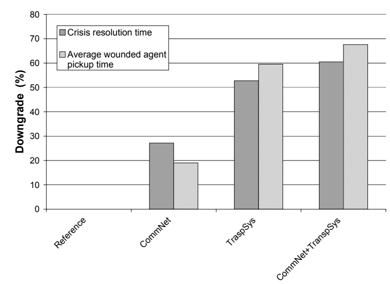

Figure 6. Downgrade of T_c (δT_c) and T_r (δT_r).

Also, the average wounded agent pick up time degrades as weaknesses are introduced in the system (see fig. 6). Failures in the communication network produce a downgrade of the 19%, while congestions in the transportation system is more critical, degrading T_r of the 59.53%. We obtain $\delta T_r > \delta T_c$ because the wounded agents that do not receive any aid. This metric also catches the emergent phenomena previously observed that is, the combination of weaknesses in the communication network and in the transportation system does not results in a linear combination of the downgrade, indeed we observe $\delta T_r = 67\%$.

Weaknesses in the communication network, as defined in our model, do not impact the percentage of wounded agents that die during the crisis. In figure 7 we observe the 18% of wounded agents died in the reference scenario and 20% died in the CommNet scenario. A strong impact is indeed observed when the transportation system is congested. In this case the half of wounded agents died (48%), 15% because do not receive any aid and 33% because picked up too late. Adding faults in the communication network do not increase too much the number of wounded agent died

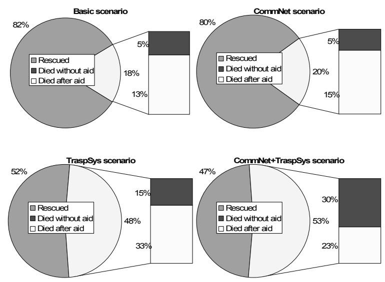

Figure 7. Percentage of the rescued and dead wounded agents.

(53%) but heavily increase the number of wounded agents that do not receive any aid (30%). While the congestion of the transportation system slows down the soccourer agents, the wounded agents generate more aid requests that congested the communication network, which capacity is already reduced by fault on link and routers. Then the soccourer agents do not receive help requests in time, or not at all.

6. Concluding remarks

In this work we have presented a modeling and simulation methodology which exploits the capabilities of both agent-based modeling and federated simulation. The proposed approach, based on recursive model partitioning, is scalable enough to model any kind of complex system composed of different type of subsystem, guaranteeing the reuse of already existing simulation models and reducing the development time. Other benefits introduced by the federated simulation are the possibility to use remote resources, to distribute the load of a simulation and to stimulate knowledge sharing, that is fundamental in complex system modeling.

Despite its simplicity, the case study considered is sufficient to show how emergent phenomena could be observed and how simulation helps in quantifying them. For example, we have observed that combining two scenarios which introduce crisis on different infrastructure do not produce foreseeable results. Simulation helps in quantifying them.

Acknowledgment. This work is partially supported by the CRESCO Project under the contract num. ENEA/2007/1303/FIM-INFO-AFU. The authors would like acknowledge Eng. Alessandro Lentini and all the participants of the SPIII.5 CRESCO project for the the precious discussion and brainstorming had during the project's meetings.

References

[1] I. Abele-Wigert and M. Dunn. *International CIIP Handbook*. Center for Security Studies, ETH Zurich, 2006.

[2] C. A. Boer, A. de Bruin, and A. Verbraeck. Distributed simulation in industry – a survey: part 1 – the cots vendors. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 1053–1060. Winter Simulation Conference, 2006.

[3] C. A. Boer, A. de Bruin, and A. Verbraeck. Distributed simulation in industry – a survey: part 2 – experts on distributed simulation. In *WSC '06: Proceedings of the 38th conference on Winter simulation*, pages 1061–1068. Winter Simulation Conference, 2006.

[4] L. Bononi, M. Bracuto, G. D'Angelo, and L. Donatiello. A new adaptive middleware for parallel and distributed simulation of dynamically interacting systems. In *DS-RT '04: Proceedings of the Eighth IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'04)*, pages 178–187, Washington, DC, USA, 2004. IEEE Computer Society.

[5] V. Cardellini, E. Casalicchio, and E. Galli. Agent-based modeling of interdependencies in critical infrastructures through uml. In *Proc. of 2007 Agent-Directed Simulation Symposium (ADS'07) of the Spring Simulation Multiconference, Norfolk, VA*, March 2007.

[6] R. M. Fujimoto. *Parallel and Distribution Simulation Systems*. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[7] O. Gursesli and A. Desrochers. Modeling Infrastructure Interdependencies using Petri Nets. In *Proc. of Int'l Conf. on Systems, Man and Cybernetics*, Oct. 2003.

[8] R. Hofer and M. Loper. Dis today [distributed interactive simulation]. In *Proceedings of the IEEE*, volume 83, pages 1124–1137, August 1995.

[9] K. Hopkinson, X. Wang, R. Giovanini, J. Thorp, K. Birman, and D. Coury. EPOCHS: a Platform for Agent-based Electric Power and Communication Simulation Built from Commercial Off-the-shelf Components. *IEEE Trans. Power Systems*, 21(2):548–558, may 2006.

[10] F. Kuhl, R. Weatherly, and J. Dahmann. *Creating computer simulation systems: an introduction to the high level architecture*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[11] A. Mikler, M. Monticino, B. Callicott, and S. Khalil. Agent based modeling of human and natural systems and their interactions. In *In Proceedings of 7th Annual Swarm Researchers/Users Conference (SwarmFest 2003), Notre Dame, IN.*, 2003.

[12] D. Miller and J. Thorpe. Simnet: the advent of simulator networking. In *Proceedings of the IEEE*, volume 83, pages 1114–1123, August 1995.

[13] M. North, N. Collier, and J. Vos. Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit. *ACM Trans. Model. Comput. Simul.*, 16(1):1–25, 2006.

[14] G. F. Riley, M. H. Ammar, R. M. Fujimoto, A. Park, K. Perumalla, and D. Xu. A federated approach to distributed network simulation. *ACM Trans. Model. Comput. Simul.*, 14(2):116–148, 2004.

[15] S. Rinaldi. Modeling and Simulating Critical Infrastructures and their Interdependencies. In *Proc. of 37th Annual Hawaii Int'l Conf. on System Sciences*, pages 5–8, Jan. 2004.

[16] S. Rinaldi, J. Peerenboom, and T. Kelly. Identifying, Understanding, and Analyzing Critical Infrastructure Interdependencies. *IEEE Control Systems*, 21(6):11–25, Dec. 2001.

[17] A. Varga. The omnet++ discrete event simulation system. In *Proc. of European Simulation Multiconference (ESM2001)*, Prague, Czech Republic, June 2001.