Creating and validating emergency management services by social simulation and semantic web technologies

Geovanny Poveda, Emilio Serrano and Mercedes Garijo
{gpoveda,eserrano,mga}@dit.upm.es

Technical University of Madrid, Madrid 28040, Spain

Abstract. One of the most promising fields for Ambient Intelligence (AmI) is the implementation of intelligent emergency plans. By using AmI, it is possible to improve the collaboration and coordination strategy of response efforts in emergency situations. Despite AmI systems are generally evaluated by using Living labs, it is desirable to use simulations in the emergency case. Simulations have allowed emergency committees and emergency experts to improve the performance and efficiency of many emergency plans while decreasing the limitations of regular drills and AmI restrictions. However, despite their wide range of benefits, simulations are currently facing many problems. Among those, simulations are performed in an ad-hoc model and are usually "closed". To improve this situation, this paper proposes using semantic web technologies as a powerful tool to reuse, extend, and combine different simulation components.

Keywords: Service creation strategies, Emergency plans, Ambient Intelligence, Ambient Intelligence Simulation, Agent-based Social Simulation

1 Introduction

Ambient intelligence (AmI) is an emerging discipline in information technology in which people are empowered through a digital environment that primarily consists of sensors and devices connected through a network [1]. AmI offers a digital environment in which it is possible to support the cooperation of devices, services and people [2]. One of the most promising fields for AmI is implementation of intelligence emergency plans. By using AmI it is possible to improve the collaboration and coordination strategy of response efforts in emergency situations.

One of the most common ways to evaluate AmI systems is by using living labs (LL). LL refer an approach for representing a user-centric methodology for sensing, prototyping, validating and refining complex solutions in evolving real life contexts [3]. The benefits of using LL to validate emergency plans are highly valued. LL use an infrastructure that enables people to simulate the real environment of physical spaces such as houses or buildings with multiple IT devices connected and distributed across a network. However, in this approach some features are modelled without including a complete characterization of people and spaces. Living labs face some restrictions such as: (1) control the variables and parameters related to emergency, (2) control the time constant of the emergency.

Over the last years Agent Based Social Simulation (ABSS) have been used by many emergency experts and emergency committees. Currently, ABSS are usually "closed" and for specific services, i.e., they cannot be parameterized to adapt them to other cases beyond the studied case. Moreover, the experiments are not reproducible, i.e., the information given about how the authors have built the simulations is insufficient to repeat them. Many simulation models are often accompanied by underlying assumptions that are unknown to the researchers or cannot be explicitly characterized for a particular model.

It is necessary to provide a solution for the ad-hoc emergency modelling issue and subjective interpretations for improving the emergency plan strategies by including more realistic egress plans in the emergency simulations. Ontologies are useful across the simulation modelling and for knowledge sharing [4]. In order to provide ontology driven simulations and improve the aforementioned shortcomings, we propose a mechanism based on Semantic Web Technologies. More specifically, a model where users can build emergency scenarios based on contextual information and semantic relationships. Our model has been defined as a subsystem on top of the semantic architecture framework proposal by Serrano et al. [1]. In particular, we have followed the methodology for the evaluation of emergency plans and we have adapted the vision presented in such work by implementing a module to support the creation and validation of emergency simulations.

This paper is organized as follows. Section 2 presents an overview of the most common frameworks used for validating emergency plans. The overview includes a description for ABBS frameworks and emergency domain simulator framework. Section 3 introduces the proposed architecture by describing the most important components. Next, section 4 introduces the implementation of a semantic module used to validate the proposed architecture. Finally, section 5 summarizes the results and points out the possible future research directions.

2 Background

This section describes some of the most important frameworks used for validating emergency plans by using the simulation approach. We have included two main categories: (1) ABBS frameworks and (2) emergency domain simulator frameworks.

2.1 Agent-Based Social Simulation Frameworks

An interesting wiki [5] on platform comparisons presents the large number of frameworks available for general ABSS. This wiki currently lists 81 frameworks with information that is important for the purpose of this paper, such as the license. To the best of the authors' knowledge, none of these simulators gives abstract mechanisms for simulating emergency plans. In general, the most popular ABSS frameworks, such as NetLogo, MASON and Repast, do not offer tools to build a realistic environment model with the capacity to perform emergency-plan evaluations. Nevertheless, these platforms can supply interesting support for developing more abstract resources.

2.2 Emergency Simulation Frameworks

Currently there are several domain-specific frameworks to address the emergency simulation approach for indoor environments. Some interesting works such as [6], [7], [8], [9], [10] and [11] have shown the importance of studying emergency plans based on simulation approach. However, currently, their design and validation process need to consider new tools, components and platforms for reducing the development effort and facilitate the deployment of realistic simulations.

Some of them need to support knowledge sharing and reuse mechanisms. In [6], the authors propose an interesting approach for supporting the egress in crowd simulations. However, its architecture is based on an ad-hoc approach in which actions are performed by restricted subjects from a multi agent world. There is not AmI features that allow designers to create participatory simulations and augmented experiments for the crowd actions. In [8], the authors propose an interesting approach for supporting the emergency egress process. However, every function around the egress situation does not include user profile characterizations. In [7], the authors propose an interesting architecture for emergency response simulation which includes human, environment and validation features. However, it does not include mechanisms to formalize the emergency knowledge. There is not ontology modelling.

In [1], the authors have proposed a methodology for developing and using a holistic framework based on an ontological approach. The architecture shows a novel and useful approach for generating and validating emergency plans. The architecture has been designed by using a service-oriented approach in which a set of components are defined for offering a set of functionalities as emergency services. In this framework, ontologies are used to formalize the emergency knowledge domain and provide a common understanding that allows users to support the reasoning process and integration service.

3 Proposed Architecture

In an attempt to narrow down the aforementioned shortcomings, we have designed a semantic emergency model and we have implemented a semantic utility in order to validate the stated model. The semantic utility has been created for supporting the emergency simulation development process on a simulator framework named EscapeSim [12]. The proposed architecture is presented in Fig 1. The design phase of our architecture includes a modelling domain knowledge

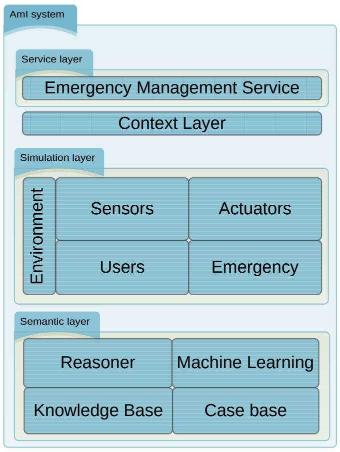

Fig. 1: Proposed Architecture

process, which is based on ontology called EinSim. This ontology aims to clip all the phases of simulation emergency process together. EinSim Ontology schema has been designed to maintain the integrity with semantic trends and standards to keep the ontology simple and put impact on its usability and easy to appliance to encourage other developers. For a detailed technical information about the EinSim Ontology schema please refer to ontology website [13].

3.1 Components

The presented is used as a basis for the the implementation of agents in the emergency simulation model. The proposed architecture allows users to design, create and validate emergency simulations and emergency services. The architecture has three layers: (1) semantic checking and discovery components at the bottom; (2) a simulation layer; and, (3) a component for deploying emergency management services.

Semantic checking and discovery components layer The base layer is composed of set of elements that enable users to use semantic reasoning features for checking model consistency, checking model relations and augmented inferences in the case of AmI services. The core element of this layer is a reasoner, which act as an intermediate entity between the domain knowledge and a machine learning component. Inference feature is provided by using a discovery mechanism which allows systems to infer relations on the knowledge base and "facts" on a case-based component. One important feature of this layer is the decoupling level, which able designers and developers to use the semantic checking and inference component without the need to instance the simulation component. Thus, it is possible that external platforms or AmI services can use those features without simulation features.

Simulation layer This layer is composed of set of the elements that enable designers to represent some of the most important features and conditions related to the emergency simulation. In order to define a standard for the definition of emergency simulation models, this layer has a repository in which are specified a common-base of emergency simulation models. The aim of this repository is to define a frame that allows users not to start the emergency simulation process from scratch. This layer is responsible for offering the API functions from the frameworks simulators. This layer is replaced with the reality in the final AmI system deployed. When the the context layer is properly designed, the emergency management system does not distinguish between the simulation and the reality.

Service component The last layer is the service emergency management component, which is responsible for exposing reasoning capabilities and simulation features functions. This layer includes a context component, whose function is to identify the context of the external request and provide specific services according it. Services have been designed to be consumed by using the RESTFUL architecture. This approach takes advantage of some principles of multi agent system paradigm and semantic web by providing a unique resource for every entity of the simulation model and for every node of the reasoning layer.

4 Implementation

As mentioned above, we have created a semantic utility for evaluating and validating the proposed architecture. We have designed a Java component which provides an auto-assisted design wizard utility for building emergency simulation scenarios by applying reasoning capabilities for checking model consistency and checking model relations. The Java component is composed of two modules: (1) *a simulation control module*, which is responsible for controlling the verification and validation process on simulation and (2) *an adaptation module*, which is responsible for checking the consistence of the simulation models. Another function of adaptation model is to generate the mappings between ontological concepts and objects from simulator framework. In order to facilitate the simulation design process, a model repository beans have been created. It is basically a component in which the common-base elements of the simulation process are stored.

Simulation workflow start by using the models stored on the repository: (1) user model, (2) environment model and (3) catastrophe model. When users instance the simulation control, it provides a function that enables users to be assisted for defining the features related to every model. Thus, designers and developers can create models according to the features that they need to use from the model beans. As a part of the simulation verification process, a function is deployed for checking the restrictions and inconsistency on the emergency simulation models. There is a function responsible for checking the relations and dependence between the simulation models. This function instances a reasoning mechanism in order to advise users for a possible mistakes, dependencies and suggestions. Thus, every time users choose elements from the model beans, a reasoning mechanism is enabled for checking the relations and restrictions. Proposed simulation workflow has followed some recommendations of managing distributed process proposed by Alcarria et al [14] and Chung et al [15].

Fig 2 gives an example of OWL class with restrictions. This OWL class is included in the public ontology given in this paper [13] and can be linked by the interested reader. Class restrictions are analyzed automatically by a semantic reasoner, such as Jena, when building a disaster model with the JavaBeans repository. For example, given the restriction started in line 5, instances of this OWL class (called individuals in OWL) have to include an object property called *hasPhysicalPlace*. This properties relate instances of two OWL classes. Thus, if there is not an environment model previously defined in the simulation, the reasoner does not allow the developer to define a specific disaster as an individual (or instance) of the OWL class described. After defining the property hasPhysicalPlace, the reasoner may be used to check that the property is in the correct domain and range according to the ontology definition and the specific developer instantiation. Specificity, this property [13] belongs to the *EnvironmentModel* domain and its range is *PhysicalPlaces*. As seen, the use of an ontology and a semantic reasoner enables developer to follow a methodological order on the model definitions.

```
1
2  <owl:Class rdf:about="&geosim2;Disaster">
3      <rdfs:label xml:lang="en">Disaster</rdfs:label>
4      <rdfs:subClassOf rdf:resource="&geosim2;Model"/>
5      <rdfs:subClassOf>
6          <owl:Restriction>
7              <owl:onProperty rdf:resource="&geosim2;
                   hasPhysicalPlace"/>
8              <owl:minCardinality rdf:datatype="&xsd;
                   nonNegativeInteger">1</owl:minCardinality>
9          </owl:Restriction>
10     </rdfs:subClassOf>
11     <rdfs:isDefinedBy rdf:datatype="&xsd;string">http://minsky
           .gsi.dit.upm.es/~gpoveda/geosim/0.1.3/ns.owl#</
           rdfs:isDefinedBy>
12     <rdfs:comment xml:lang="en">Description of a disaster
           modelled with EinSim ontology</rdfs:comment>
13 </owl:Class>
```

Fig. 2: OWL Restriction

Once restrictions and relations are checked, the simulation control module (explained at the beginning of this section) generates the simulation model (see simulation layer in Fig 1). This model is a file that contains the descriptions and relations about the users, environments and catastrophe models by means of RDF triples. RDF statements are generated by applying a set of rules that execute a mapping process between the values specified by user on the wizard utility and the ontological model. The next step inside the simulation workflow is to check if the simulation model is consistent with the ontology EinSim [13]. Basically, this activity involves analyzing every subject and predicate of the RDF relations. The aforementioned process is conducted automatically using the reasoner (Jena API in our case). Once the emergency simulation consistency is verified, the adaptation module (explained at the beginning of this section) uses a mapping process to perform the transformation of the ontological classes instances (defined RDF) into objects (Plain Old Java Object). Finally, simulation is running according to the EscapeSim [12] framework features.

Currently, UbikSim[1], a general AmI simulator, has been extended to implement different modules of the framework proposed in the EscapeSim [12] library.

Fig. 3: Simulation scenario in EscapeSim

5 Conclusion and future works

In this paper, we have presented an architecture for designing and validating emergency plans by using a novel semantic-based approach. This approach is based on an ontological domain knowledge that provides reasoning capabilities for checking models and discovery relations features. By applying this approach, the experience on the development and design process is enhanced by allowing: (1) to be assisted step by step with the construction of emergency simulation; (2) to extend and reuse simulation models concepts (user, environment and emergency) by using portable RDF emergency simulation resources; and, (3) to extend and reuse simulation components previously implemented which could be automatically suggested by semantic web technologies.

The paper gives an ontology for this purposes [13] and a partial implementation of the framework [12]. Although the complete implementation is a future work, the open and free source of EscapeSim already allows the interested user to create new environments (walls and rooms may be drawn, and then, the user can drag and drop other elements, such as sensors, users or furniture) and creating new user profiles with different emergency plans, Fig 3 shows an example of emergency simulation scenario on EscapeSim. A video is available on-line[2].

[1] `https://github.com/emilioserra/UbikSim/wiki`.
[2] `http://goo.gl/IsALQz`.

Acknowledgments

This research work is supported by the Spanish Ministry of Economy and Competitiveness under the R&D project CALISTA (TEC2012-32457) and the European Commission under the project SmartOpenData (FP7-ENV-2013 Project 6038).

References

1. Serrano, E., Poveda, G., Garijo, M.: Towards a holistic framework for the evaluation of emergency plans in indoor environments. Sensors **14**(3) (2014) 4513–4535
2. Alcarria, R., Robles, T., Morales, A., Diego, Aguilera, U.: Enabling flexible and continuous capability invocation in mobile prosumer environments. Sensors **12**(7) (June 2012) 8930–8954
3. Eriksson, M.: State-of-the-art and good practice in the field of living labs. (2006)
4. Morales, A., Alcarria, R., Martin, D., Robles, T.: Enhancing evacuation plans with a situation awareness system based on end-user knowledge provision. Sensors **14**(6) (2014) 11153–11178
5. : ABSS frameworks list. `http://en.wikipedia.org/wiki/ABM_Software_Comparison`
6. Lozano, M., Morillo, P., Lewis, D., Reiners, D., Cruz-Neira, C.: A distributed framework for scalable large-scale crowd simulation. In Shumaker, R., ed.: Virtual Reality. Volume 4563 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2007) 111–121
7. Jain, S., Mclean, C.R.: An architecture for integrated modeling and simulation for emergency response
8. Zoumpoulaki, A., Avradinis, N., Vosinakis, S.: A multi-agent simulation framework for emergency evacuations incorporating personality and emotions. In Konstantopoulos, S., Perantonis, S., Karkaletsis, V., Spyropoulos, C., Vouros, G., eds.: Artificial Intelligence: Theories, Models and Applications. Volume 6040 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 423–428
9. Yotsukura, S., Takahashi, T.: Framework of an emergency management system using different rescue simulators. Advanced Robotics **23**(9) (2009) 1233–1246
10. Araujo, R., Rocha, R., Campos, M., Boukerche, A.: Creating emergency management training simulations through ontologies integration. In: Computational Science and Engineering Workshops, 2008. CSEWORKSHOPS '08. 11th IEEE International Conference on. (July 2008) 373–378
11. Pan, X., Han, C., Law, K.H., claude Latombe, J.: A computational framework to simulate human and social behaviors for egress analysis
12. Garijo, M., Poveda, G., Serrano, E.: EscapeSim website. `https://github.com/gsi-upm/EscapeSim`
13. Garijo, M., Poveda, G., Serrano, E.: EinSim website. `http://minsky.gsi.dit.upm.es/~gpoveda/einsim/`
14. Alcarria, R., Robles, T., Morales, A., Cedeno, E.: Resolving coordination challenges in distributed mobile service executions. International Journal of Web and Grid Services **10**(2) (January 2014) 168–191
15. Chung, J., Gonzalez, G., Armuelles, I., Robles, T., Alcarria, R., Morales, A.: Experiences and challenges in deploying OpenFlow over real wireless mesh networks. Latin America Transactions, IEEE (Revista IEEE America Latina) **11**(3) (May 2013) 955–961