Designing emergency management services by ontology driven social simulation*

Geovanny Poveda†, Emilio Serrano, and Mercedes Garijo
Universidad Politécnica de Madrid, 28040 Madrid, Spain
{gpoveda, emilioserra, mga}@dit.upm.es

Abstract

There are numerous attempts to assess emergency plans by simulations, and excellent contributions have been proposed. 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. However, there is no "most complete" tool, and most of the time, it is not possible to reuse and combine components of different frameworks. As a result, without a true interest in sharing and interconnecting works, researchers in this domain are constantly damned to reinvent the wheel. To improve this situation, this paper proposes a general model for designing emergency management services in indoor environments. The core of this model is EinSim, an ontology which gathers the most relevant concepts for the design of these services and which, in the spirit of the Liked Data methods, links a number of external and well-referenced data. The paper supports the hypothesis that semantic web technologies are a powerful tool to reuse, extend, and combine different simulation components. Moreover, the use of semantic reasoners allows providing developers with automatic verification in the emergency management service development.

Keywords: Service Creation Strategies, Emergency Plans, Ambient Intelligence Simulation, Agent-based Social Simulation, Ontology

1 Introduction

Ambient intelligence (AmI) is a paradigm in the information technology which represents a user-centric methodology for sensing, prototyping and validating complex solutions in evolving real life context [49]. AmI focuses on adapting to people's needs and particular situations through a digital environment composed of domotic devices and embedded interfaces [43]. AmI enables people to be assisted with intelligent and intuitive features on complex real life scenarios. One of the most promising areas for AmI is to provide coordination mechanisms in emergency situations. AmI enables emergency committees to improve the collaboration and coordination strategies of emergency plans, since it provides features for adapting and assisting people in emergency situations by means of interoperability mechanisms between devices and specialized emergency components.

Living labs (LL) comes a crucial approach to evaluate and validate AmI systems. A living lab is an space for experimentation and creation of AmI applications with real users in imitated real environments [17]. The benefits of using LL for validating emergency plans based on AmI approach are very valuable. LL enable users to make more-strategic decisions during an emergency situation since it uses context-aware capabilities for identifying contextual complex attributes and describing the interaction

IT CoNvergence PRActice (INPRA), volume: 3, number: 1 (March), pp. 17-32

*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).

†School of Telecommunications Engineering (ETSIT), UPM. Complutense Avenue 30, 28040, Madrid, Spain, Tel: +34914595700

between people and domotic devices. Although the use of living labs is always desirable, they have the same problems as drills [42]. They do not provide mechanisms to reproduce social and physiological behaviours, as well as, they not include a complete specification of some emergency components such as disaster conditions (fire spread, fire intensity, fire spread, etcetera).

Agent Based Social Simulation (ABSS) has become a useful approach for modelling many natural and human systems that include social features [19]. ABSSs have allowed emergency experts and researchers to improve the performance and efficiency of emergency plans since it is possible to model artificial emergency environments populated with multiples autonomous simulated entities, which carry out a set of operations on behalf of people, devices and different emergency components. Such entities act autonomously by employing some emergency knowledge or representation of its goals. By using ABSS it has been possible to: (1) reduce cost of deploying emergency plans; (2) control variable and parameters related to emergent conditions; (3) control the emergency experiment time, and (4) characterize human behaviours and its social interactions. Currently, emergency ABSS models are usually complex and hard to completely understand. Many of them are designed for representing specific emergency scenarios, which can not be extended or adapted to other cases. There is a lack of standard explicit emergency content for enabling interoperable and reusable emergency simulation models.

The use of an ontology driven simulation approach (ODS) is a useful approach for contributing to solve the current ad-hoc restrictions and subjective interpretations in the emergency simulation process. The inclusion of a formal agreed emergency definitions enable emergency designers to use a common standardized set of agreed concepts for extending, adapting and improving emergency features and functions from different emergency systems and components. Moreover, it provides the basis for building interoperable emergency simulated models. ODS enables simulation systems to be connected with specialized external emergency systems and different simulation components. Furthermore, ODS provides the basis for an automated adaptation and extension of structural changes on the simulation models. In order to provide an ontology driven simulation approach which contributes to improve the aforementioned shortcomings, a semantic emergency model has been proposed. Basically, we have created a model that allow users the implementation of semantic features in emergency simulation models.

Our model has been defined as a subsystem on top of the semantic architecture framework proposed by Serrano et al. [44]. 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. Moreover, this work presents an extension of Poveda et al. workshop paper [33] with, among others, a more detailed related works revision, and a further ontology explanation and a case study.

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 ABSS frameworks and emergency domain simulator framework. Section 3 introduces the proposed design model by describing the most important components. Next, section 4 introduces the knowledge base and the EinSim ontology, which is the core of the design model presented. Section 5 illustrates the advantages of the contribution presented by a case study. Finally, section 6 concludes and gives possible future research directions.

2 Related works

This section describes a number of possible frameworks to evaluate emergency plans using simulations in four main categories: ABSS frameworks, game engines, emergency domain simulators and AmI simulators.

An interesting wiki [1] 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 [6], MASON [47] and Repast [14], 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. In this vein, Dawson et al.[16] use NetLogo to study emergency plans for flood incidents.

The use of general game engines is also very common when there is a need for modelling a real environment with agents moving through it. This is an essential requirement for emergency simulators, which is usually solved in these platforms, every time better and more automatically, by using navigation meshes and pathfinding methods. Barbosa et al. [23] propose the use of virtual simulation to evaluate emergency operating procedures in the nuclear industry through reusing a game engine platform. In this work, the real environment may be virtually modelled, and people are able to virtually navigate and interact with each other. Szymanezyk et al. [15] adapt popular video-game technology for an agent-based crowd simulation framework in an airport terminal. There is a wiki [3] that details the main properties of a large number of these engines. Strangely enough, this wiki also lists exactly 81 game engines, the same number of ABSS frameworks mentioned above. Some of the most popular engines are Unity [9] (proprietary and written in C++), GamePlay3D [4] (under the Apache license and written in C++) and JMonkey [5] (under the BSD license and written in Java). Although these engines are incredibly useful for our goal, they lack important elements, such as a model of simulated sensors or abstract behaviours for the users modelled. Some works overcome these limitation by considering real users as players to study their response in a simulated emergency [40]. Moreover, the parametrization of simulations built by these engines is still complex; thus, they are only employed for closed simulations, i.e., those in which changing the scenario or the elements requires rebuilding the engine.

A number of domain-specific simulators address the emergency domain in indoor environments [26]. Some examples are EVACNET4 [20], WAYOUT [45], STEPS [55] and SimWalk [41]. These works give excellent tools to model emergencies, but: (1) they do not contemplate the inclusion of AmI features to assess an intelligent plan, such as a sensor model and actuators to guide users intelligently; (2) they tend to ignore the multi-agent paradigm, which, as stated in the Introduction, is an intuitive manner of studying complex adaptive systems; and, more importantly; (3) the code is often not available, and thus, researchers cannot extend, reuse or simply learn from it.

Campillo-Sanchez et al. [13] review six different simulators devoted to evaluating general AmI systems, including UbiWise [10], TATUS [32] and UbiReal [31]. Although they enable developers to design realistic environments, the user role has to be played by a real user interacting with the simulator. Therefore, it is not possible to conduct batches of experiments to obtain statistically significant results. Moreover, the authors do not give the source code, thereby hindering researchers from reusing modules of these works. This is also the case of the simulators proposed by Tang et al. [48] and Diewald et al. [18], which, although they are significant contributions for the AmI field, do not provide researchers with open implementations. Hence, to the best of the authors' knowledge, there is no open simulator that focuses on AmI general evaluation besides the one extended, released and presented in this paper.

These works demonstrate that there are numerous attempts to assess emergency plans by simulations, and excellent contributions have been proposed. However, there is no "most complete" tool, and most of the time, it is not possible to reuse and combine components of different frameworks. Further evidence of this can be found by consulting the EU Research Projects on the CORDIS (Community Research and Development Information Service) website [2], where a search with the keywords "emergency" and "simulation" returns over 8,000 results of funded projects. As a result, without a true interest in sharing and interconnecting works, researchers in this domain are constantly damned to reinvent the wheel.

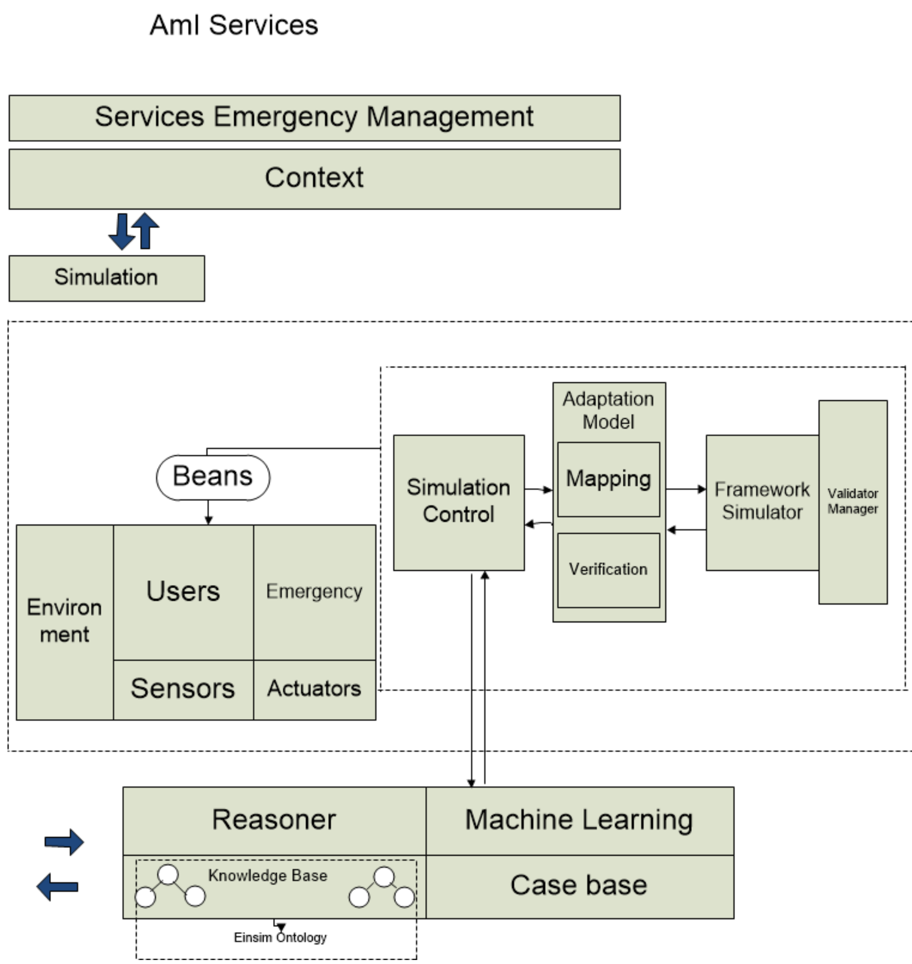

Figure 1: Proposed design model.

3 Design model for ontology driven social simulations

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 [22]. The proposed design model is presented in figure 1. The basis of the stated design model includes a modelling knowledge process, which is based on an ontology called EinSim, which is given with this paper.

The presented design model allows the implementation of agents in the emergency simulation model for creating and validating emergency simulations as well as emergency services. The design model has three main layers: (1) semantic layer at the bottom; (2) a simulation layer; and, (3) a component for deploying emergency management services. These layers are explained in the following sections.

3.1 Semantic layer

This layer is responsible for providing semantic features on the proposed architecture. The core element of this layer is an ontological emergency model which enables reasoning and machine learning features. Machine learning component is responsible for providing learning algorithms for training emergency information collected from the simulation layer, which is explained below. Reasoner component is responsible for providing reasoning and inference mechanisms for checking consistency, relations and augmented inferences on emergency models. This layer enables designers and developers to use emergency semantic models without the need to use features from the simulation component. This is because it is based on the broker pattern [54]. Thus, external specialized systems or AmI services can use those features without instance the simulator layer.

3.2 Simulation layer

This layer is responsible for providing the components and elements for creating simulated emergency scenarios as well as for providing mechanisms for controlling and adapting semantic features on the emergency agent model. This layer is composed of a emergency simulation bean model (ESBM), a simulation control component (SC) and adaptation model (AM). ESBM component is responsible for providing an emergency component bean model repository, which enables users to reuse emergency software emergency definitions without start from scratch. Simulation control is the core element of the simulation layer. It is a component whose function is to provide mechanisms for defining the simulated emergency models as semantic metadata. That is, it provides a function for including semantic representations into the the emergency simulation bean model. Adaptation model is responsible for providing mechanisms to map the emergency semantic meta-data to the agent code.

3.3 Service component

The last layer is the service emergency management component, which is responsible for exposing emergency semantic capabilities and simulation features as a service. The layer includes a context component, whose function is to identify the context of the external requests and provides specific services according it. When the context layer is properly designed, the emergency management system does not distinguish between the simulation and the reality. Services have been designed to be consumed by using the RESTFUL architecture [39]. Every simulated entity and concepts from the ontological model are described by using a unique URI resource.

4 Knowledge base: Einsim Ontology

As instantiation of proposed semantic component, which was explained in section 3.1, we have created an ontology which attempts to cover some of the most important concepts around the emergency simulation process. Design and modelling phases of our ontology have followed the guidelines and procedures of a certain number of sources and guides, among those: (1) the analysis of related works to life cycle of simulation emergencies [34]; (2) analysis of commercial and academic emergency simulators frameworks such as UbikSim [12], Drillsim [30], Evacnet [51] and Simwalk [53]; and, (3) Analysis from related emergency ontologies such as: Ontology Fire [46], Dires Ontology [27] and Semantic Sensor Network [24]. As a result of aforementioned process, it has been possible to define an ontology emergency scheme for indoor evacuations. This ontology is called EinSim and is available on-line [21]. This ontology aims to clip all the phases of simulation emergency process together. EinSim ontology has been designed to maintain the integrity with semantic trends and standards to keep the ontology simple and put impact on its usability. This section details the ontology and its use.

Implementation of the proposed ontology have included achievements of technologies such as Web Ontology Language (OWL) [7] and Resource Description Framework Schema (RDF) [8]. The use of OWL for representing emergency knowledge model is an appropriated approach to handle concepts and methods in a formal and standardised way. By using OWL's logical base it is possible to use a set of

Subject Area	Ontology Name	Representation Language	Downloadable	Documentation
Disaster	Dires	OWL	Yes	[34]
Disaster	HXL	RDF	Yes	[35]
Disaster	Ecology-Fire	OWL	Yes	[36]
Infrastructure	SNS	OWL	Yes	[38]
People	SocOntPersons	OWL	N/A	[11]
Egress	Dmuser	OWL	N/A	[29]

Table 1: Ontology name spaces used for EinSim ontology.

useful features for the simulated design process. Among those, reasoning techniques can be applied to check the models consistency (check integrity and ensuring an unambiguous and consistent knowledge representation) as well as to perform knowledge deduction.

4.1 EinSim namespaces

As mentioned above, the core of the design model is an ontology which provides developers with a knowledge base for developing emergency management services. We have modelled an entire emergency scheme with a single name space (EinSim) and we have gradually started introducing other emergency related concepts by using external namespaces. The ontology design process has adopted an approach in which specifications of generic emergency concepts are defined by using the EinSim namespace. Specialized indoor emergency features are included by using namespaces from different specialized emergency knowledge representations. Table 1 shows some of the most important ontology namespaces used in the proposed ontological emergency model. The use of external namespaces has been focused mainly on the use of the specialized concepts, which are related to: (1) user characterization, (2) egress strategies and, (3) physical environment characterizations.

- *User profile and role characterization*: Some of the most important physiological response and emergency human behaviours in indoor spaces have been included. We intend to include some of the most relevant human behaviours features around the simulated emergency life cycle. Those features include some interesting descriptions such as: (1) role (emergency leader, coordinator, evacuee); (2) profile (disabled people, adult and child); as well as panic, stress, confusion and calm features. By including those features, it is possible to represent the dynamics of human behaviour and the way in which it could be affected by the environment and context. For including the stated human behaviours and physiological features we have used the SocOntPersons namespace [11].

- *Egress strategy characterization*: We have included some of the most important egress strategies used by people during emergency situations in indoor spaces. We intend to include the most relevant concepts for adapting egress strategies features around the simulated emergency life cycle. Those features include some interesting definitions about egress algorithms such as: (1) shortest path, (2) follow the leader, (3) follow indications, and, (4) emergency adaptive. By including this kind of concepts, we intend to improve the reliability of egress strategies in simulated emergency plan, since it is possible to include elements and components for adapting the egress strategies according to different emergency conditions and user profile features. Moreover, egress strategies can be adapted according to the evolution of different simulated disaster "events" such as fire and congestion. For including the stated egress strategy features we have used the Dmuser namespace [29].

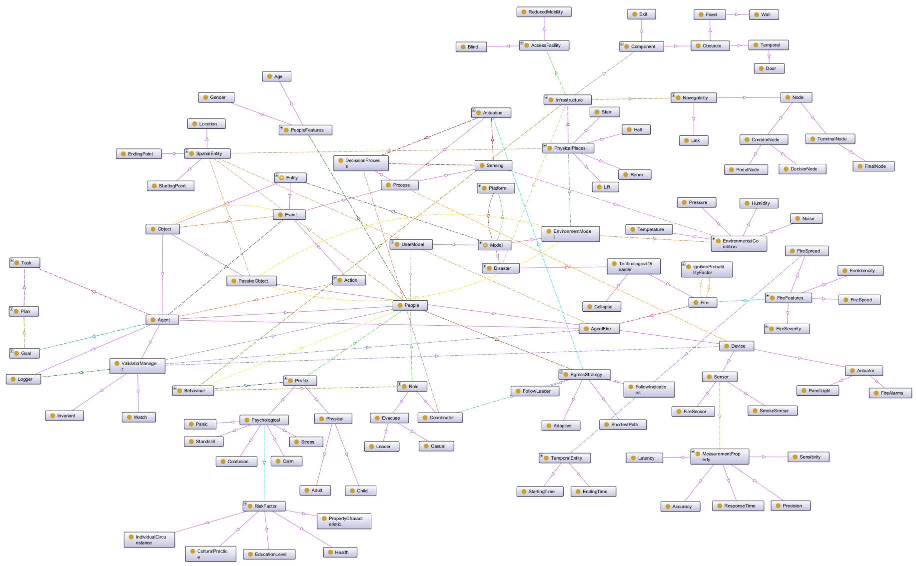

Figure 2: EinSim Ontology class design overview (also available on-line at http://goo.gl/ClTF74).

- *Physical environment characterization*: We have included some of the most important descriptions about physical infrastructure and environmental conditions in emergency situations in indoor spaces. By considering physical descriptions, we intend to include some useful features related to space navigability. Descriptions about indoor space navigation must support the creation of more consistent emergency plans, since it is possible to involve infrastructure components and its descriptions inside the egress algorithm definitions. By including environmental conditions descriptions it could be possible to deploy egress strategies based on the information provided by domotic devices such as sensors and actuators. For including the physical environment features we have used two namespaces: SNS [38] and Ecology-Fire [36].

Thus, we have included some important namespaces for building the basis of a "standardized", "specialized", "stable" and "connected" emergency model. By including such namespaces, we intend to offer an emergency model able not only to include knowledge and cognition features, but also, able to offer a set of descriptions to establish the interoperability between simulation models and specialized external heterogeneous emergency systems.

4.2 EinSim classes

As we mentioned before, EinSim ontology describes some of the most important concepts around the simulated emergency life cycle, in particular concepts for the creation and validation of adaptable emergency plans in indoor spaces. By following the principles of Gruber et al. [25], it was possible to identify the basis of the current ontological emergency model. EinSim ontology offers a model in which context conditions, user profiles and navigation features are included as basis elements of adaptable emergency plans. Currently, EinSim ontology has defined thirty three high hierarchy classes and ninety three low hierarchy classes. Figure 2 shows a partial view of classes, attributes and relations from the current emergency model definition. Below, we will introduce some of the most important ontological emergency concepts.

- *Environmental Model*: It represents the physical environment of simulation and its features. This class enables users to model the space in which agents and other simulated entities such as sensors and actuators are deployed. The environmental model involves an infrastructure composed of "PhysicalPlace" (rooms, stairs, halls, etcetera) and "EnvironmentalCondition" (pressure, noise, humidity, temperature). In order to support the adoption of smart egress strategies, a set of special subclasses have been included in the environmental model. Among those, a set of descriptions related to physical infrastructure and its components, as well as the navigability descriptions. Environment class also includes subclasses for characterizing the different domotic devices involved in emergency situations (sensors and actuators) as well as its corresponding measurement properties (sensitivity, precision, accuracy, etcetera). Domotic devices have been included by using the SNS namespace [38].

- *Disaster Model*: It represents the disaster model of simulation and its features. This class enables users to represent different disaster emergency conditions such as: fire, congestion and its relation with the user and environmental model. Fire features represent the main concept around the simulated disaster model. However, another interesting emergency descriptions such as collapse, congestion and gas leak are included in the ontological model This model involves a set of descriptions related to fire features, such as: fire spread, fire intensity, fire speed, and fire severity. The relation between the disaster model and environmental model has been reached by using the "agentFire" class. This concept allows the disaster model to connect the fire features with the sensors and actuators capabilities. Fire features have been included by using two namespaces: Dires [34] and Ecology-Fire [36].

- *User Model*: It represents the user model of simulation and its features. This class enables users to represent different user emergency behaviours descriptions such as: role, profile, user egress strategy and, physiological behaviours through the "People" and "Adaptive" classes. The relation between disaster model, environmental model and user model has been reached by using the "Control" class, as explained below. People features have been included by using the SocOntPersons name space [11].

- *Simulation Control*: The proposed ontology also includes a simulation control class, which enables users to model the classes for enabling the integration, interoperability and knowledge sharing between different emergency components. This class is responsible for enabling the interoperability between all simulated models (disaster model, environmental model and user model) and specialized external heterogeneous emergency systems. This concept provides a set of subclasses for standardizing the interchange of message between external systems and the ontological emergency model.

4.3 EinSim properties

In order to establish the relation between the different emergency concepts defined in section 4.2, we have defined a set of object properties [37]. Figure 3 shows some of the most important proposed relations. Relations have been categorized according to emergency model definitions explained in section 4.2.

4.3.1 Environmental Model

- *hasEnvironmentalCondition*: This property relates a "PhysicalPlace" instance (lift, hall, room) with a set of environmental conditions such as: noise, temperature, humidity and pressure. It is not a functional property since a physical space can be involved in different conditions at the same time. This property involves an inverse and transitive object restriction relation, which means a "Sensing" concept can only be instanced if there are environmental conditions.

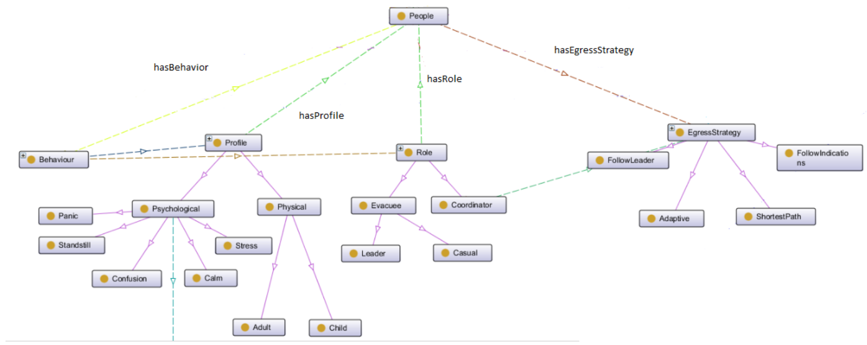

Figure 3: EinSim Object properties design overview (also available on-line at http://goo.gl/76jtEm).

- *hasPhysicalPlace*: This property relates an "EnvironmentalModel" instance with the different "PhysicalPlace" concepts such as: room, lift and stairs. This relation is a functional property since a "PhysicalPlace" class can only represent an element at the time. In other words, a space does not involve two places at the same time. This object property involves three external object properties as restrictions: hasInfrastructure, hasNavigability and hasInfrastructure. The use of this object property intents to create an order around the "EnvironmentalModel" definition, since a "PhysicalPlace" description is successfully instanced when infrastructure and navigability features are previously created.

4.3.2 User Model

- *hasRole*: This property relates a "People" instance with the different activities made by users over the simulation process (leader, evacuee, coordinator). This object property can be successfully used when two conditions are satisfied as restrictions: "hasDisaster" and "roleInvolvesProfile". That is, people would have a role when disaster and profile instances have been previously defined for the simulation process.

- *hasEgressStrategy*: This property relates a "People" instance with the different egress strategies used by people during an emergency situation (follow leader, shortest path, predefine route, etcetera). This property is not considered a functional property since it involves the aforementioned "Adaptive" class for using different egress strategies over the simulation runtime. In other words, it is possible to instance two egress strategies at the same time. For instance, simulated people can become affected by the "FollowtheLeader" and "ShortestPath" strategy at the same time. This object property can be successfully used when the "hasDisaster" condition is satisfied.

- *hasProfile*: This property relates a "People" instance with the different physical and psychological profile descriptions (panic, calm, streets, child, adult). This relation is not considered a functional property since it is possible to assign different profiles for the simulated people over the simulation runtime: child with streets, adult with panic, adult calm are some examples. This object property can be successfully used when the "hasDisaster" condition is satisfied.

4.3.3 Disaster Model

- *hasFireFeatures*: This property relates a "Fire" instance with the different fire features descriptions (fire spread, fire severity, fire intensity, fire speed). This relation is not considered a functional property since a "Fire" instance involves different fire features at the same time. This object

```
<owl:Restriction >
<owl:onProperty rdf:resource='(someproperty)'/>
 (one value or cardinality constraints)
</owl:Restriction >
```

Figure 4: Generic form of restrictions.

property can be successfully used when the "IgnitionProbabilityFactor" condition is satisfied. That is, a "Fire" class can be instanced only if there is a 'IgnitionProbabilityFactor" instantiation.

4.4 Properties restrictions

A property restriction is a special kind of class description which describes an anonymous class of all individuals that satisfy a restriction [28]. OWL distinguishes two kinds of property restrictions: value constraints and cardinality constraints. A value constraint puts constraints on the range of the property when applied to this particular class description. A cardinality constraint puts constraints on the number of values a property can take, in the context of this particular class description. Property restrictions have the form shown in figure 4.

One of the most important reasons to use property restrictions in emergency models, is because it makes possible to support the validation and checking of restrictions and dependences between concepts and objects properties. The use of property restrictions able mechanisms for assisting emergency users over the different phases of the simulated emergency process. Property restrictions aim to minimize the definition of "unwell-formed " simulated emergency models, since they allow users to include all possible conceptual basis for running efficient simulated emergency models. In our research, two approaches for evaluating restrictions has been included: over simulation runtime (OSR) and before simulation runtime (BSR). OSR approach uses property restrictions as a part of a process that analyse the simulation state of elements over the simulation runtime. In order to show how the properties restrictions can be involved over the emergency simulation process, below, an emergency restriction is explained.

Figure 5 shows how a property restriction is involved for applying a special order over the emergency simulation design process. By using the "owl:onProperty" restriction, the owl:Restriction specifies a "dependence" with an object property called "hasUser". This restriction means before instantiating the "UserModel" class, it is necessary to define the elements and components of the simulated physical space "PhysicalPlace".

5 EinSim and EscapeSim use

In order to show how the stated ontological emergency model and property restrictions can be used for supporting the design and implementation of simulated emergency plans, a software utility has been designed and implemented. This software provides an utility for guiding users over the simulation design process. It is a component belonging to the simulation layer, which was explained in section 3. This component uses the Jena API [52] for invoking reasoning and inference features from the semantic layer explained in section 3.1. The proposed utility is a wizard which enables users to be assisted step by step over the different phases of the simulated emergency process. It allows users to avoid the design of uncompleted and incoherent simulated emergency models, since it forces users to follow an order based on the ontological properties restrictions explained in section 4.4. Thus, users can get valuable features from the simulated emergency process since they can generate more "stable" and "coherent" simulated emergency models.

```
<owl:Class rdf:about="&einsim;UserModel">
        <rdfs:label xml:lang="en">UserModel</rdfs:label>
        <rdfs:subClassOf rdf:resource="&einsim;Model"/>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="&einsim;hasUser"/>
                <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
                    3</owl:minCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:subClassOf>
            <owl:Restriction>
                <owl:onProperty rdf:resource="&einsim;hasPhysicalPlace"/>
                <owl:minCardinality rdf:datatype="&xsd;nonNegativeInteger">
                    1</owl:minCardinality>
            </owl:Restriction>
        </rdfs:subClassOf>
        <rdfs:isDefinedBy rdf:datatype="&xsd;string">http://minsky.gsi.dit.
            upm.es/~gpoveda/einsim/0.1.5/einsim.owl#</rdfs:isDefinedBy>
        <rdfs:comment xml:lang="en">An instance of this class enables user
            features.</rdfs:comment>
    </owl:Class>
```

Figure 5: User model restriction.

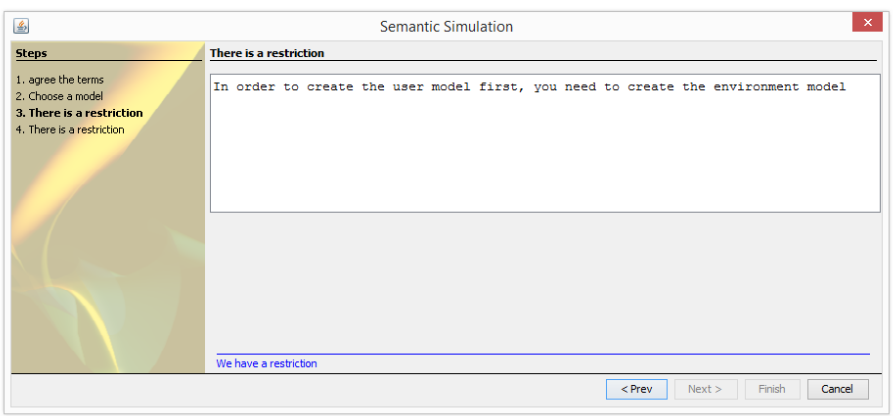

Figure 6: Wizard user restriction (also available on-line at http://goo.gl/YtjcOf).

Just for showing how the proposed wizard utility uses the semantic mechanism explained in section 3.1, let us consider the case in which a user decides to instance the “UserModel”. When user chooses this option, there is a mechanism which identifies and validates the consistence of the instanced classes as well as its relations. The proposed semantic utility finds there are a set of dependences between the “UserModel” and the “EnvironmentalModel”. In particular, there is a restriction which means before defining the “UserModel” it is necessary to specify the elements and components of the simulated physical space. Figure 6 shows an example in which user is redirected to use the “create / modify the environment model” option.

As mentioned in section 4.4, property restrictions can be used for validating and checking restric-

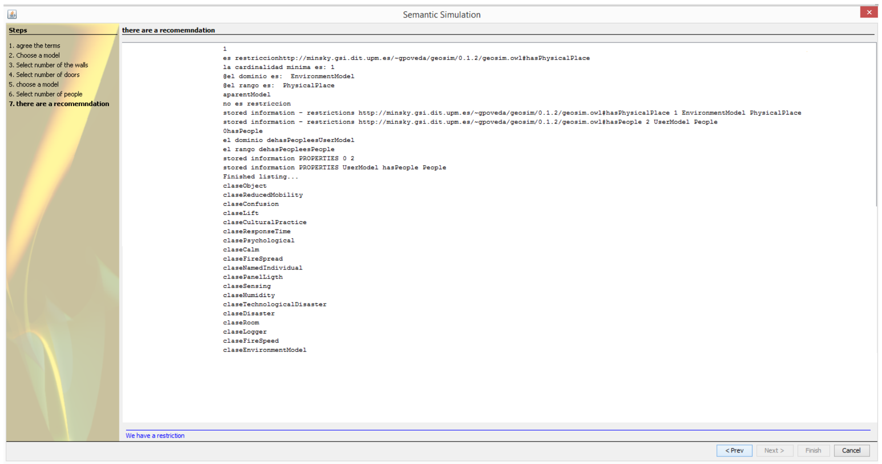

Figure 7: Wizard sensor coherence fault (also available on-line at http://goo.gl/A4NOIa).

tions over the simulation runtime. To show how restrictions can contribute to support the evaluation of coherence faults [1], let us consider the case in which a user is specifying the features related to the "EnvironmentalModel", particularly, by defining the number of sensors in the simulated scenario. By using a set of rules based on the Jena API [52], the proposed semantic utility is able to check the defined emergency model restrictions and dependences. Thus, just in the case in which user is specifying the number of sensors to be involved, the wizard utility suggests the user to specific a sensor number value between 6 and 7. This is because there is a rule which tries to check the consistency of object properties related to the environmental model definition. This rule involves two important features. One of them is responsible for verifying the object property restrictions. In this case, there is a restriction which means before creating the number of actuators, it is necessary to create the sensors. Another one refers to emergent concepts, there is a rule in which percentage of actuators will not be greater than 65% of created sensors. Figure 7 shows a case study where the Jena API was combined with EscapeSim. Figure 8 shows the environment simulated for a floor of the Research Centre for Smart Buildings and Energy Efficiency (CeDInt) [50].

6 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 which lays foundations for knowledge management based on simulated emergency scenarios in indoor spaces. Modelling and definitions of the proposed ontological model not only provide a vocabulary for avoiding the use of ambiguous interpretations of terms in the simulated emergency domain, but also, enable mechanisms for supporting the validation and checking of restrictions and dependences between concepts and its relations. Moreover, ontological model provides mechanisms for enabling the interoperability between simulated models and specialized external heterogeneous emergency systems. 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. In terms of future work, we plan to experiment with interlinking ontological

[1] refers to a process for validating emergency restrictions

Figure 8: Simulation scenario in EscapeSim.

emergency data with other specialized systems and research on possibilities of using the model defined in section 3.3 for providing a checking and validation service.

References

[1] ABSS frameworks list. http://en.wikipedia.org/wiki/ABM_Software_Comparison.

[2] CORDIS, EU research projects. http://cordis.europa.eu/projects/home_en.html.

[3] Game engines list. http://en.wikipedia.org/wiki/List_of_game_engines.

[4] Gameplay, free 3d game engine. http://gameplay3d.org/.

[5] jmonkeyengine 3.0, java opengl game enginejmonkeyengine 3.0. http://jmonkeyengine.org/.

[6] NetLogo 5.0.4 user manual. http://ccl.northwestern.edu/netlogo/docs/.

[7] Owl website. http://www.w3.org/2001/sw/wiki/OWLi.

[8] RDF website. http://www.w3.org/TR/rdf-schema/.

[9] Unity - game engine. http://unity3d.com/.

[10] J. J. Barton and V. Vijayaraghavan. Ubiwise, a ubiquitous wireless infrastructure simulation environment. Technical Report HPL-2002-303, Mobile and Media Systems Laboratory, Hewlett-Packard Company, 2002.

[11] M. H. Bickhard. The social ontology of persons. *Social interaction and the development of knowledge*, 2004.

[12] J. A. Botía, P. Campillo, F. Campuzano, and E. Serrano. UbikSim website. https://github.com/emilioserra/UbikSim/wiki.

[13] P. Campillo-Sanchez, E. Serrano, and J. A. Botía. Testing context-aware services based on smartphones by agent based social simulation. *Journal of Ambient Intelligence and Smart Environments - Intelligent agents in Ambient Intelligence and smart environments*, 5(3):311–330, May 2013.

[14] N. Collier. Repast: An extensible framework for agent simulation. *Natural Resources and Environmental Issues*, 8(4):16–21, March 2000.

[15] N. Collier. Agent-based crowd simulation in airports using games technology. *Transactions on Computational Collective Intelligence VIII*, 7430(0):192–213, 2012.

[16] R. J. Dawson, R. Peppe, and M. Wang. An agent-based model for risk-based flood incident management. *Natural Hazards*, 59(1):167–189, October 2011.

[17] B. De Ruyter and E. Aarts. Ambient intelligence: visualizing the future. In *Proc. of the Advanced Visual Interfaces (AVI'04), Gallipoli, Italy*, pages 203–208. ACM Press, May 2004.

[18] S. Diewald, L. Roalter, A. Möller, and M. Kranz. Towards a holistic approach for mobile application development in intelligent environments. In *Proc. of the 10th International Conference on Mobile and Ubiquitous Multimedia (MUM'11), Beijing, China*, pages 73–80. ACM Press, December 2011.

[19] P. Fishwick. *Simulation Model Design and Execution: Building Digital Worlds*. Prentice Hall, 1995.

[20] R. Francis and P. Saunders. Evacnet: Prototype network optimization models for building evacuation. Technical Report NBSIR-79-1738, National Bureau of Standards, Washington, DC. National Engineering Lab., 1979.

[21] M. Garijo, G. Poveda, and E. Serrano. Einsim website. `http://minsky.gsi.dit.upm.es/~gpoveda/einsim/`.

[22] M. Garijo, G. Poveda, and E. Serrano. EscapeSim website. `https://github.com/gsi-upm/EscapeSim/wiki`.

[23] L. B. S. Gatto, A. C. A. Mól, I. J. L. dos Santos, C. A. F. Jorge, and A. P. Legey. Virtual simulation of a nuclear power plant's control room as a tool for ergonomic evaluation. *Progress in Nuclear Energy*, 64(0):8–15, April 2013.

[24] W. S. S. N. I. Group. Sensor Ontology website. `http://www.w3.org/2005/Incubator/ssn/ssnx/ssn`.

[25] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies*, 43(5):907–928, November 1995.

[26] E. D. Kuligowski, R. D. Peacock, and B. L. Hoskins. A review of building evacuation models, 2nd edition. Technical Report 1680, National Institute of Standards and Technology, 2010.

[27] S. Liu. Dires ontology website. `http://www.disaster20.eu/dires`.

[28] G. Meditskos and N. Bassiliades. Towards an object-oriented reasoning system for owl. In *Proc. of the International Workshop on OWL Experiences and Directions (OWLED'05), Galway, Ireland*, pages 1–11, November 2005.

[29] G. Meditskos and N. Bassiliades. Exploring ontology potential in emergency management. In *Proc. of the Gi4DM conference—geomatics for disaster management (GI4DM'10), Torino, Italy*, pages 1–6, February 2010.

[30] S. Mehrotra, C. Davison, and D. Massaguer. DrillSimSite website. `http://www.ics.uci.edu/~projects/drillsim/`.

[31] H. Nishikawa, S. Yamamoto, M. Tamai, K. Nishigaki, T. Kitani, N. Shibata, K. Yasumoto, and M. Ito. Ubireal: Realistic smartspace simulator for systematic testing. In *Proc. of the 8th International Conference on Ubiquitous Computing (UBICOM'06), Orange County, California, LNCS*, volume 4206, pages 459–476. Springer-Verlag, September 2006.

[32] E. O'Neill, M. Klepal, D. Lewis, T. O'Donnell, D. O'Sullivan, and D. Pesch. A testbed for evaluating human interaction with ubiquitous computing environments. In *Proc. of the 1st International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM'05), Trento, Italy*, pages 60–69. IEEE, February 2005.

[33] G. Poveda, E. Serrano, and M. Garijo. Creating and validating emergency management services by social simulation and semantic web technologies. In *Proc. of the 8th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI'14), Belfast, UK, LNCS*, volume 8867, pages 460–467. Springer-Verlag, December 2014.

[34] D. . Project. Dires Ontology website. `http://www.disaster20.eu/dires`.

[35] H. R. Project. HXL Ontology website. `http://hxl.humanitarianresponse.info/ns/index.html`.

[36] P. Project. Wiki Protege website. `http://cerrado.linkeddata.es/ecology/fire/`.

[37] P. Project. Wiki Protege website. `http://protegewiki.stanford.edu/wiki/Pr4_UG_mi_Glossary`.

[38] J. C. Puras and C. A. Iglesias. Sharing ontology by web services: Implementation of a semantic network service (sns) in the context of the german environmental information network (gein). In *Proc. of the Workshop on Semantic Web and Databases (SWDB'03)*, pages 189–201, 2003.

[39] J. C. Puras and C. A. Iglesias. Disasters2.0: Application of web2.0 technologies in emergency situations. In

Proc. of the 6th International Conference on Information Systems (ISCRAM'09), Gothenburg, Sweden, pages 1–15, May 2009.

[40] U. Rüppel and K. Schatz. Designing a bim-based serious game for fire safety evacuation simulations. *Advanced Engineering Informatics*, 25(4):600–611, October 2011.

[41] T. Sakaki, F. Toriumi, K. Shinoda, K. Kazama, S. Kurihara, I. Noda, and Y. Matsuo. Disaster detection and location estimation of suffered areas using social media. In *Proc. of the Annual Conference of the Japan Society of Artificial Intelligence (JSAI'12), Miyazaki, Japan, LNCS*, pages 4C1–R–6–8. Springer-Verlag, June 2012.

[42] E. Serrano and J. Botia. Validating ambient intelligence based ubiquitous computing systems by means of artificial societies. *Information Sciences*, 222(0):3–24, February 2013.

[43] E. Serrano, P. Moncada, M. Garijo, and C. A. Iglesias. Evaluating social choice techniques into intelligent environments by agent based social simulation. *Information Sciences*, 286(0):102–124, December 2014.

[44] E. Serrano, G. Poveda, and M. Garijo. Towards a holistic framework for the evaluation of emergency plans in indoor environments. *Sensors*, 14(3):4513–4535, March 2014.

[45] V. Shestopal and S. Grubits. Evacuation model for merging traffic flows in multi-room and multi-storey buildings. In *Proc. of the 4th International Symposium (IS'11), Gaithersburg, MD, USA*, pages 625–632. International Association for Fire Safety Science, June 1994.

[46] A. Souza. Fire Ontology website. `http://cerrado.linkeddata.es/ecology/fire/`.

[47] L. Tang, X. Zhou, C. Becker, Z. Yu, and G. Schiele. Mason: A new multi-agent simulation toolkit. In *Proc. of the Eighth Annual Swarm Users/Researchers Conference (SWARMFEST'04), Ann Arbor, Michigan, USA*, pages 1–8. IEEE, May 2004.

[48] L. Tang, X. Zhou, C. Becker, Z. Yu, and G. Schiele. Situation-based design: a rapid approach for pervasive application development. In *Proc. of the 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing (UIC/ATC'12), Fukuoka, Japan*, pages 128–135. IEEE, September 2012.

[49] D. I. Tapia, J. A. Fraile, S. Rodríguez, R. S. Alonso, and J. M. Corchado. Integrating hardware agents into an enhanced multi-agent architecture for ambient intelligence systems. *Information Sciences*, 222(0):47–65, February 2013.

[50] C. R. Team. Cedint website. `http://www.cedint.upm.es/en`.

[51] E. Team. Evacnetsite website. `http://www.iklimnet.com/hotelfires/fire_egress_software_10.html`.

[52] J. Team. Jena api website. `https://jena.apache.org/`.

[53] S. Team. Simwalk website. `http://www.simwalk.com/`.

[54] N. Yoshioka, A. Ohsuga, and S. Honiden. Mobeet: A multi-agent framework for ubiquitous information systems. In *Proc. of the 6th International Bi-Conference Workshop on Agent Oriented Information System (AOIS'04), Riga, Latvia, LNCS*, volume 3508, pages 19–35. Springer-Verlag, June 2005.

[55] B. Zhang, S. V. Ukkusuri, and W. K. V. Chan. Agent-based discrete-event hybrid space modeling approach for transportation evacuation simulation. In *Proc. of the 2011 Winter Simulation Conference (WSC'11), Phoenix, AZ, USA*, pages 199–209. IEEE, December 2011.

Author Biography

Geovanny Poveda Is a PhD candidate at Technical University of Madrid (UPM). He Obtained the M.Sc. Degree in Computer Science from the Technical University of Madrid in 2011. His research interests include: semantic web, intelligent systems, multi agent systems and specially knowledge representation. Mr Poveda has participated in different international workshops and conference, as well as in different national and international research projects. He has collaborated as reviewer for the World Multi conference on Systemic, Cybernetics and Informatics. He is currently working as research assistant at Intelligent System Group of the the Telematic Engineering Department (DIT) at UPM.

Emilio Serrano Is a lecturer at the Department of Artificial Intelligence of the Technical University of Madrid (UPM). He obtained the M.Sc. Degree in Computer Science from the University of Murcia (UMU) in 2006. After a stay as a research associate at the University of Edinburgh for a year; he received the Ph.D. degree with European mention from the UMU in 2011 and the Extraordinary Ph.D. Award in 2012. Dr. Serrano has co-authored over 40 publications in international conferences and journals. He has also participated as a researcher in a number of European and National funding programmes. His main research interests include: advanced machine learning; social network analysis; agent based social simulations; ambient intelligence; and, specially, the interaction among these fields.

Mercedes Garijo is Associate Professor at the Universidad Politécnica de Madrid, Spain, where she received her Ph.D. (1982) in Telecommunication Engineering. She teaches at the School of Telecommunication Engineering (undergraduate, postgraduate and doctorate levels) in computer science and communications engineering topics. Her research interests are in telematic services engineering with intelligent technologies. She has been technical responsible in several research and development projects funded by EC and Spanish Government.