1st International Workshop on Ambient Intelligence for Large Premises (AmILP 2016)

co-located with ECAI 2016

August 29, 2016

The Hague, The Netherlands

Rubén Fuentes-Fernández and Marin Lujak

(editors)

Preface

This volume contains the papers presented at AmILP 2016: First International Workshop on Ambient Intelligence for Large Premises held on August 28, 2016 in The Hague.

Ambient Intelligence (AmI) is intended to provide users with systems tightly integrated with their everyday environment and activities. The goal is minimizing the need of explicit actions by users, through the continuous and distributed gathering of information and actuation devices. With the advances in the field, AmI is pursuing growingly ambitious goals in terms of the size of its smart spaces, the number of served users, and the level of adaptation to them. This workshop is focused on the particular challenges and potential solutions that appear when AmI moves to Large Premises (LP). In this context new requirements appear to understand big groups of people moving in premises that fall beyond the classical closed and controlled environments of most AmI systems. The ways of interaction, the expected services, and the behaviour of people acquire a new dimension and variability. Systems need to adapt to these crowds using large numbers of multiple and heterogeneous resources, in distributed and frequently unfriendly environments that cause changes in the system topology.

AmILP 2016 aims at providing a forum for discussing recent advances in engineering complex AmI systems acting in large premises. The research emerging in this domain faces to multidisciplinary issues, both technical and social. The papers to be presented here offer an interesting overview of some of these issues.

We are thankful for the support to organize this event to the MOSI-AGIL-CM (grant S2013/ICE-3019) project supported by the Autonomous Region of Madrid and co-funded by EU Structural Funds FSE and FEDER.

Also, thank you very much to the EasyChair team for all their work.

July 31, 2016
Madrid

The event chairs,
Rubén Fuentes-Fernández,
Universidad Complutense de Madrid,
Spain
Marin Lujak, Universidad Rey Juan
Carlos, Spain

Program Committee

Luis Antunes	GUESS/LabMAg/Univ. Lisboa
Joao Balsa	GUESS/Univ. Lisboa
Carole Bernon	IRIT / UPS
Ghassan Beydoun	University of Wollongong
Holger Billhardt	Universidad Rey Juan Carlos
Ulf Blanke	
Massimo Cossentino	
Tiziana D'Orazio	Institute of Intelligent Systems for Automation, National Research Council
John Drury	University of Sussex
Juergen Dunkel	FH Hannover - University for Applied Sciences and Arts
Rubén Fuentes-Fernández	Universidad Complutense de Madrid
Juan Carlos Gonzalez-Moreno	Universidad de Vigo
Stella Heras	Universitat Politècnica de València
Marin Lujak	University Rey Juan Carlos
Ambra Molesini	Alma Mater Studiourum - Universtà di Bologna
Paulo Novais	University of Minho
Giuseppe Vizzari	University of Milano-Bicocca

Schedule

Session 1 - Applications: 9:10-11:00	
Alberto Fernández-Isabel and Rubén Fuentes-Fernández	User Modelling Languages for AmI: A Case Study on Road Traffic
Nardine Osman and Carles Sierra	Reputation in the Academic World
Session 2 - Fundamentals and engineering: 11:00-12:30	
Jorge Gómez-Sanz, Rafael Pax and Millán Arroyo	Towards the Simulation of Large Environments
Álvaro Sánchez-Picot, Diego Martín, Borja Bordel, Ramón Pablo Alcarria Garrido, Diego Sánchez de Rivera and Tomás Robles	Towards a simulation of AmI environments integrating social and network simulations
Session 3 - Crisis management: 14:00-15:30	
Holger Billhardt, Juergen Dunkel, Marin Lujak, Alberto Fernández, Ramón Hermoso and Sascha Ossowski	An Architecture for Situation-aware Evacuation Guidance in Smart Buildings
Antonio M. Diaz, Ganggao Zhu, Álvaro Carrera, Carlos A. Iglesias and Oscar Araque	SmartSim: Improving visualization on social simulation for an emergency evacuation scenario
Marin Lujak and Stefano Giordani	Evacuation Centrality: Agile Evacuation Routing in Unpredictable Hazardous Conditions
Session 4 - Joint session AmILP and ALAW workshops - Augmented Worlds for AmI: 16:00-17:30	
Chairs	Summary talk on workshops AmiLP and ALAW
Alessandro Ricci (Università di Bologna Italy)	First thoughts on the use Augmented Worlds for AmI
Panelists to be confirmed	Joint panel

Table of Contents

An Architecture for Situation-aware Evacuation Guidance in Smart Buildings

Holger Billhardt[1], **Jürgen Dunkel**[2], **Marin Lujak**[1], **Alberto Fernández**[1], **Ramón Hermoso**[3], and **Sascha Ossowski**[1]

Abstract. Smart Cities require reliable means for managing installations that offer essential services to the citizens. In this paper we focus on the problem of evacuation of smart buildings in case of emergencies. In particular, we present an abstract architecture for situation-aware evacuation guidance systems in smart buildings, describe its key modules in detail, and provide some concrete examples of its structure and dynamics.

1 INTRODUCTION

As cities in the 21st century are growing both in size and population, it is necessary to have reliable means to manage installations that offer essential services to the citizens (e.g., airports, train stations, sports centres, museums, and so on). Although there are already experts who design and manage such facilities, there is a lack of operational tools and knowledge to explore their functional limitations in a principled manner, to identify potentially dangerous situations (a crisis is always identified when it is too late), and to support decision-making in case of emergencies.

Recommendations or guidelines about what to consider and how to react do exist, but they can hardly be challenged or debated upon as they are often based on specific cases and experiences rather than strong general arguments. In practice, frequently it is up to human decision-makers to design and monitor an appropriate and timely course of action in response to a specific emergency.

Recently, it was proposed that, by bringing together works from the fields of Agent-Based Social Simulation (ABSS), Ambient Intelligence (AmI), and Agreement Technologies (AT), advanced methods and tools can be developed to address the aforementioned problem [1]. In particular, it has been suggested to use ABSS as a means for realistically modelling human crowds in large installations (taking into account both individual and herd behaviours, as well as their interplay); AmI techniques are adequate to model and simulate physical devices in smart spaces that capture relevant features of the situation (sensors) and provide decision–makers with the means to act upon it (actuators); while AT are used to explore intelligent strategies for managing such advanced installations as large-scale open distributed social systems.

[1] CETINIA, University Rey Juan Carlos, Spain, email: {holger.billhardt, alberto.fernandez, marin.lujak, sascha.ossowski}@urjc.es
[2] Computer Science department, Hochschule Hannover, Germany, email: juergen.dunkel@hs-hannover.de
[3] Computer Science department, University of Zaragoza, Spain, email: rhermoso@unizar.es

In this paper, we focus on the problem of evacuation of installations of the aforementioned type in case of emergencies. In particular, we focus on smart buildings equipped with information processing, sensing and actuation facilities. In [2], for instance, a recommender system has been put forward that arranges personalized visits through a museum, based on user profiles and visitor location data provided by in-door localization techniques. Such situation-aware recommender systems con be considered as a special type of that take the current Context-aware Recommender Systems (CARS) that are discussed in detail in [3].

The present work aims at exploiting infrastructures of this type also for evacuation purposes.

The objective of an evacuation is to relocate evacuees from hazardous to safe areas or the areas where the life-threatening risk is minimal while providing them with safe routes. Present building evacuation approaches are mostly static and preassigned. Frequently, no coordination is available except for predefined evacuation maps. Still, due to the lack of the overall evacuation network information, there might be casualties caused by a too slow evacuation on hazardous routes. *Real-time* route guidance systems, which dynamically determine evacuation routes in inner spaces based on the imminent or ongoing emergency, can help reducing those risks. A dynamic, *context-sensitive* notion of route *safety* is a key factor for such recommendations, in particular as herding and stampeding behaviours may occur at potential bottlenecks depending, among other factors, on the amount of people who intend to pass through them. Furthermore, smart devices allow guidance to be *personalized*, taking into account, for instance, the specific circumstance of the elderly, disabled persons, or families. In such settings, an adequate notion of *fairness* of evacuation route recommendations is of utmost importance to assure the trustworthiness of the system from the standpoint of its users [4]: the guidance should not only achieve good overall performance of the evacuation process, but must also generate proposals for each of its users that each of them perceive as efficient. Finally, large groups of people may need to be evacuated so *scalability* plays a key role.

Therefore, we concentrate on real-time situation-aware evacuation guidance in smart buildings such that we keep track of the related fairness considerations among the paths assigned to individuals based on their mobility limitations, initial positions, respecting individual´s privacy, and other evacuation requirements.

Section 2 describes in detail the particular problem that we are addressing, extracts requirements for the architecture, and provides a brief overview of the devices, methods and tools, mainly from the fields of AmI and AT, that we will use to address them. Section 3 outlines our abstract architecture, describes the structure and dynamics of its key modules in further detail, and provides some concrete examples to this respect. We conclude the paper with Section 4, describing lessons learnt and future lines of work.

2 EVACUATION GUIDANCE IN EMERGENCY SITUATIONS

A pedestrian route recommender system for smart spaces that recommends the safest routes to pedestrians and simultaneously optimizes conflicting objectives of finding the social optimum and minimizing individual path travel times in steady state conditions while considering people flow and fairness was presented in [4].

The system considers the influence of stress on human reactions to the recommended routes and iteratively ponders user response to the suggested routes influenced by stress-related irrational behaviours until system acceptable routes are found. Moreover, the influence of affiliate ties and self-concerned individuals among evacuees was studied in [5]. Here, Lujak et al. model self-concerned and social group behaviour via individual and team reasoning. The recommended routes take in consideration the affiliate ties to guarantee evacuee's compliance with the routes.

If real-time infrastructure information is available to evacuees and they can negotiate their routes, it becomes possible to provide a selection of safe fair routes considering individual safety requirements. Therefore, we assume that the building and evacuees are monitored by a strategically positioned network of sensors. The monitoring permits us both to recognize the evacuees' behavior in respect to the suggested route and time window as to perceive the congestion and safety conditions of the infrastructure. Furthermore, we assume that the people flow demand (i.e., evacuation requests) is known at the beginning of the time window of evacuation. This can be achieved based on the number of persons detected by the sensor network in the building.

The aim of the architecture is, thus, to safely evacuate all the evacuees' demand on (temporally) efficient and safe routes and if not possible, then evacuate as many people as possible within the allotted time period. To this aim, we should find optimal paths toward safe exits that maximize evacuees' safety and minimize their evacuation cost considering critical crowd density and flow and thus avoiding the crowdedness conditions that might result in panic. The path cost can reflect different factors, such as the evacuation time or cost incurred because an evacuee is too close to a hazard (e.g., fire, smoke).

In the case of contingencies, the architecture should reroute evacuees towards safe exits and, thus, propose evacuation routes that are adaptive to unpredictable safety drops in the evacuation network.

As a continuation of the works mentioned previously that mathematically model the safe evacuation problem and propose a scalable and robust optimization method applicable in real world, in this paper we propose an architecture that uses necessary sensory, localization, semantics, and processing technologies that can provide real time situation awareness and evacuee guidance based on individual requirements.

2.1 Technologies

2.1.1 Indoor location infrastructure

A. Localization with landmarks

A prerequisite for intelligent routing guidance is a detailed knowledge about the current localization of all persons in the building: First, the routing algorithm must know about the occupancy of each space in a building for calculating an appropriate route. Secondly, the precise position of each person is necessary for providing her with individualized routing recommendations taking her specific constraints into account.

There are various technological approaches to localize persons in buildings:

- WIFI: The intensity of a WiFi signal can be measured (RSSI – received signal indication) to derive the distances to several access points, which allows calculating a person's position via trilateration. Unfortunately, WiFi doesn´t yield good accuracy: the distance between a mobile phone and a WiFi access point is often rather large and may not be precisely estimated on base of the RSSI, because the signal strength changes significantly with environmental conditions.
- RFID (Radio Frequency Identification) technology can also be used for indoor positioning. Persons equipped with passive RFID tags can be detected by RFID readers that are spread in the building. RFID technology has several drawbacks: First, it is rather expensive to equip a building with an adequate number of RFID readers. That means that the number of RFID readers is relatively small and localization must also apply triangulation based on distance measures, which causes the same drawback as the one described above for WiFi. Secondly, it might be difficult to provide each person with a personal RFID tag.
- iBeacon technology has recently been introduced to support indoor navigation [6]. An iBeacon device uses Bluetooth LE to send in a configurable frequency a unique ID that can be read by any smartphone. Therefore, an iBeacon infrastructure is set up easily: Beacons are cheap enough to distribute many of them, so that they can form a much denser network in the building. Furthermore, no specific beacon readers are necessary, because usual smartphones are capable of reading and processing beacon signals.

Table 1: Characteristics of indoor location technologies

	#Sender	*#Reader*	*Accuracy*
WiFi	few senders per floor	1 reader per person	low
RFID	1 sender per person	1 reader per room	medium
Beacon	many senders per room	1 reader per person	high

Table 1 summarizes the characteristics of the different technologies that are applicable for indoor localization. It states the superior accuracy of iBeacon technology: there are as many readers as users, and each building section can be equipped with so many beacons that a dense net of landmarks is given. Furthermore, some of our former projects proved that iBeacons provide sufficient localization accuracy [7][8]. Therefore, we applied beacon technology in our scenario, i.e. all sections of the buildings contain a sufficient number of iBeacons that cover completely the space in the building.

B. User smartphones:
The personal smartphones of the users play two different roles: they serve as readers of the iBeacon signals and they can exploit their built-in sensors to derive more details about the current situation of its particular user.

- *Beacon reader for localization*: In smartphone operating systems such as iOS and Android, the capability of reading iBeacon signals is already integrated. In ranging mode, a smartphone estimates the proximity to an iBeacon according to the three proximity ranges:
    - IMMEDIATE: [0, 0.5m]
    - NEAR: [0.5m, 2m]
    - FAR: > 2m

  Each room is equipped with several iBeacons with non-overlapping ranges. As soon as a user approaches an iBeacon within the predefined range (e.g. NEAR) the smartphone triggers an event carrying the iBeacon ID. Then the smartphone knows that it is near that iBeacon and can forward this information to a server that coordinates emergency situations. An iBeacon ID is hierarchical structured, (i) a UUID specifies the particular institution (such as a university), (ii) a major ID could correspond to a certain building and (iii) a minor ID to a certain room.
- *User activity recognition*: The built-in sensors of a smartphone can be exploited to derive the current activity of its particular user. There exist several works on how to use phone-based sensors for performing activity recognition. For instance, the authors in [9] applied different machine learning techniques, such as decision trees, logistic regression and neural networks to classify accelerometer data as certain activities. In our scenario, the current behavior of the users is crucial to detect panic situations, e.g. the situation that most persons in a room are running.

Furthermore, the smartphones serve as an individualized communication channel to each user to provide personalized routing guidance.

C. Further Sensors and Infrastructure
Further sensors are necessary for achieving situation awareness in the emergency recommender system. In particular, these sensors can be used to detected unexpected events in the building. For instance, smoke and temperature sensors could be used for fire detection. The signals of these sensors could be collected and analysed on a centralized emergency management system. This server also provides a central hub for the data of all user smartphones for calculating the global situation in a building such as room occupancy and general user behavior.

Furthermore, building operators can specify current incidents that could be detected automatically.

2.1.2 Complex Event Processing (CEP)

A key issue in emergency recommender systems is detailed knowledge about the current situation in the building. In our scenario, an appropriate and individualized guidance for all people in the building requires the information about:

- the smart space network structure, and dimensions
- the current position of each person and the occupancies of all sections in the building
- the situations that can provoke panic
- the space safety for each constituent part of the smart space network that can be jeopardized by, e.g., fire or build-up smoke, or panic related herding and stampeding behaviors.

Apparently, such situational knowledge cannot be predefined, but must be inferred by exploiting live data. Usually, live-data is provided by sensors, which monitor their environment and produce a continuous stream of data. In our scenario, we use smartphone sensors and further sensors that are permanently installed in the environment, such as iBeacons, temperature and smoke sensors. Each set of sensor data they emit corresponds to a particular event in the environment.

Situational knowledge can be considered as dynamic knowledge with a high change frequency. In emergencies, these streams of events must be evaluated in real-time to achieve situation awareness.

Considering a solitary event is usually of no significance, because it represents just a single incident in the physical world. For instance, it is of no importance if a single person is running, but if all persons in a room are running it could indicate a panic situation.

Complex event processing (CEP) is a software technology to extract the information value from event streams [10], [11]. CEP analyses continuous streams of incoming events in order to identify the presence of complex sequences of events, so called event patterns. The main goal of CEP is to extract a domain-specific meaning out of the observed streams of simple fine-grained and uncorrelated events. Instead, according to the key idea of CEP, a set of fine-grained simple events must be correlated to a single complex event with a significant meaning [10]. For instance, a panic event can be inferred, if the smartphones of nearly all visitors in certain area emit a running event.

Event stream processing systems manage the most recent set of events in- memory and employ sliding windows and temporal operators to specify temporal relations between the events in the stream (each event has a timestamp). The core concept of CEP is a declarative event processing language (EPL) to express event processing rules. An event processing rule contains two parts: a condition part describing the requirements for firing the rule and an action part that is performed if the condition matches. The condition is defined by an event pattern using several operators and further constraints.

In the following, we use a simplified pseudo language for expressing event processing rules, which is easier to understand than an EPL of a productive CEP system. This pseudo language supports the following operators:

Operators

∧, ∨	Boolean operator for events or constraints.
NOT	Negation of a constraint
->	Sequence of events (e1 -> e2 meaning e1 occurred before e2).
Timer	Timer(time) defines a time to wait Timer.at(daytime) is a specific (optionally periodic) point of time
.within	defines a time window in which the event has to occur.

An event processing engine analyses the stream of incoming events and executes the matching rules. Luckham introduced the concept of event processing agents (EPA) [10]. An EPA is an individual CEP component with its own rule engine and rule base. Several EPAs can be connected to an event processing network (EPN) that constitutes a software architecture for event processing. Event processing agents communicate with each other by exchanging events.

3 IN-DOOR EMERGENCY MANAGEMENT ARCHITECTURE

In this section we present an abstract architecture and describe the different components comprising it. Then, we give some details and examples of the CEP and Route recommender modules.

3.1 Abstract Architecture

We propose a solution concept of an evacuation guidance system architecture that combines different CEP modules in order to provide situation awareness for an evacuation route recommendation algorithm. An overview of this architecture is given in Figure 1.

The general operation dynamics of the system is based on two modes: standard mode and evacuation mode. In standard mode, the system continuously monitors the current state of the building, trying to detect a possible emergency scenario. If such a situation is detected (e.g., an emergency event is detected through complex event processing), the system alerts some human operator who can activate an evacuation process and the system enters in evacuation mode. In this mode, the situation of the building is still monitored and an evacuation route recommendation algorithm is executed, which provides individualized route guidance to the people that are currently in the building.

The system consists of two main parts: User Agents (UA) and Emergency Manager (EM), as well as a set of Sensors that are located at different points in the infrastructure.

User Agent (UA)

The user agent manages and stores all the information that is related to a particular user (a person that is currently located in the building under consideration). The UA is executed as an app on the smartphone of each user. Here, we assume that people that enter the building have either downloaded and run such an app on their smartphones, or they have been provided with some Smartphone like device that runs the app when they entered the building.

The UA contains three parts: a preference module, a user situation awareness module and a recommendation interface. The *preferences and constraints module* allows the user to specify certain preferences or constraints regarding evacuation scenarios; e.g. certain handicaps that imply to a restricted mobility of the person (wheelchair, blind, etc.). This information is entered during the configuration of the UA and is stored locally in form of RDF[2] data. RDF is a standard data model for knowledge representation commonly used on the semantic Web.

The *user situation awareness* module exploits sensor data (from the smart phone and beacons installed in the building) and reasons about the behaviour and location of the user (through local CEP processes). This derived information is passed to the situation module in the EM. In order to assure privacy, the amount of information provided to external components is different in standard and in evacuation mode. In standard mode, only certain basic data about the user's situation are forwarded to the EM (e.g., location, running events). In case of the activation of an evacuation

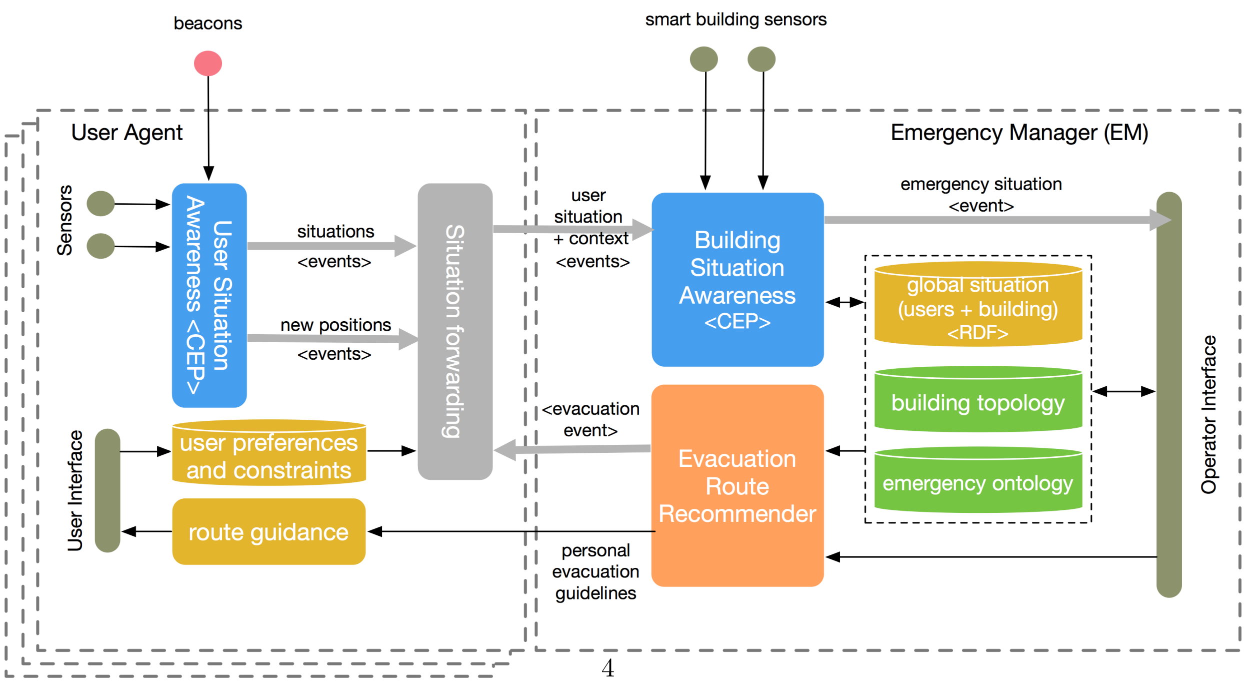


Figure 1.Overall architecture of the evacuation guidance system

(e.g., the EM broadcasts an evacuation event to all user agents), more detailed events are detected and also the preferences and constraints regarding user mobility are passed to the EM. That is, we consider that an emergency situation prevails upon privacy issues.

Finally, the evacuation mode will also trigger the recommendation interface. This interface provides the user with personalized navigation guidelines for evacuation, helping her to leave the building in the way it was calculated for her by the evacuation route recommender.

Emergency Manager (EM)

The emergency manager is the central part of the system. A *building situation awareness* module combines and analyses the events provided from the individual user agents with data from smart building sensors and generates information about the global situation of the building. This information is stored in the data model as RDF data. In this process CEP is used to filter irrelevant information and to generate higher level events. Especially in the case of the user events, individual data is aggregated to detect events regarding groups of users as well as identifying the density of the distribution of users in the building.

When the building situation awareness module detects an emergency situation, an alert is sent to the operator interface. This interface allows, on one hand, to monitor the situation of the building and, on the other hand, to trigger an evacuation process and to execute control actions in such a process (e.g., specifying blockage of parts of the building). If an evacuation process is initiated, the system enters evacuation mode and the *evacuation route recommender* [4] is executed. The module sends an evacuation event to all user agents informing them about the situation. Then it starts to calculate individual evacuation routes for all users. In this process, the algorithm uses three types of data:

- Data regarding the *building topology*: Static information about physical elements in a building (e.g. rooms, corridors, floors, doors, etc.) and relation among them (e.g. room A is 10 m^2, is next to room B and both are in floor F). In general, we use the term section to refer to physical elements. Topology knowledge is represented in such a way that is sufficient to describe the building network by a digraph with weights and tags on the constituent nodes and connecting edges. A node refers to some physical area (e.g., a room, a hall, a segment of a large corridor or floor, or some other open space). An edge connects two adjacent nodes and, thus, represents a way to move from one node to another. An edge represents, e.g., a passage, walkway, corridor, staircase, and alike. Nodes and edges are described through their type, surface, area, inclination, etc.
- *Emergency ontology*: This static ontology contains general knowledge about emergency and evacuation scenarios, e.g., facts that people with strong affiliate ties should always be evacuated together (for instance, families with children and persons with disability and their assistants), the appropriateness of certain routes for people with limited mobility in emergency situations, The influence of certain events like fire and smoke on the security level of an edge or node for evacuation purposes, etc.
- *Global situation*: Contains the current situation of the building itself as well as regarding the people that are currently in the building. This information includes:
    - The distribution of people in the building (e.g., number of persons in each node and edge)
    - Momentary positions, evacuation preferences, and mobility constraints of each person.
    - Information on nodes and edges that are blocked for evacuation, and the reason for blockage. Possible reasons are fire, smoke and panic (that can be detected through the situation awareness module) and others (as specified by an operator).

During evacuation, the global situation of the building is dynamically updated in order to reflect the situation in each moment. In the same way, the guidance algorithm controls continuously the viability of the current evacuation strategy. If changes occur (e.g., new events are detected) that may violate that viability, then the evacuation route recommender recalculates new guidance data for each user.

In the following two subsections we describe in more detail the CEP component deployed in the user and building situation modules, and the principal functioning of the evacuation guidance algorithm.

3.2 CEP Components

Both agent types, User Agent (UA) and Emergency Manager (EM) analyse the incoming streams of events to understand the current situation. In this subsection, we will discuss in some detail the underlying event models and give some examples for appropriate rules for achieving situation awareness. To make the description more comprehensive, we will simplify the event model and the corresponding rules.

3.2.1 CEP in the User Agent

The UA exploits sensor data and infers (i) the location and (ii) the behavior of a single user. To explain the CEP component in more detail, we will assume that the UA monitors two types of explicit (or atomic) events to achieve this type of situation awareness:

- `beaconEvent(beaconID):` an iBeacon with a certain ID[3] has been detected
- `accelerationEvent(velocity):` the phone is moving with a certain velocity

(i) The `beaconEvents` collected by a particular phone are used to derive the current position of its owner. The following CEP rule creates `enteringSection` and `leavingSection` events, meaning that the user is entering, respectively leaving a certain space. These events can be considered as complex (or materialized) events. They carry the ID of the user and the related beacon ID.

```
CONDITION  beaconEvent AS b1 → beaconEvent AS b2
           ∧ b1.id <> b2.id
ACTION:    CREATE enteringSection(userID, b2)
           CREATE leavingSection(userID, b1)
```

[3] Note that the `beaconID` is structured and includes, among other information, the ID of a certain section or room.

The rule describes the situation that a new `beaconEvent b2` has been read in the phone, where the beacon ID has changed. (Here the beacon ID, more precisely its minor ID, corresponds with a section of a building)

(ii) Detecting a running user is another situation that must be forwarded to the Emergency Manager, because many running users can indicate a panic situation. An appropriate CEP rule checks if the average velocity of a user is higher than 5 km/h considering a time window of 5 seconds:

```
CONDITION accelerationEvent As a[win:time:10sec]
          ∧ average(a.velocity) > 5 km/h
 ACTION    create runningEvent(userID)
```

If the condition matches, then the rule creates a `runningEvent` that contains the ID of the corresponding user.

3.2.2 CEP in the Emergency Manager

The CEP component in the Emergency Manager is responsible for deriving the global situation in the building. For instance, it could receive and analyze the following *atomic events*: produced by the CEP rules running on the users' smartphones.

- `enteringSection (userID, sec):` a user with a certain ID has entered section sec.
- `leavingSection (userID, sec):` a user with a
- certain ID has left section sec.
- `runningEvent (userID):` a user with a certain ID is running.

Another kind of situational knowledge describes the *global* situation. A first type of rules is calculating the occupancy of different sections in the building. This data is used as input for a situation-aware routing recommendation algorithm.

The following CEP rule calculates the number of persons staying in a certain section by counting all entries and exits in that section during the last 15 minutes:

```
CONDITION:
   (enteringSection AS e ∨ leavingSection As l)
            [win:batch:15min]group_by(e.sec)
      ∧ e.sec = l.sec
      ∧ count(e) AS entries
      ∧ count(l) AS exits
ACTION CREATE occupancy(e.sec, entries – exits)
```

The second type of rules tries to infer a global behavior of the people currently staying in the building. For instance, the next rule intends to detect a panic situation in the building:

```
CONDITION: runningEvent AS r [win:time:1 min]
                                 group_by(r.sec)
             ∧ count(r) > r.sec.occupancy * 0.2
 ACTION:  CREATE panicEvent(r.sec)
```

It groups all `runningEvent` according to a time-spatial window. The grouping criterion is defined by the section, where the `runningEvent` have occurred, and a time interval of 1 minute. If more than 20% of the people staying in the room are running, a panic situation is indicated.

Note that also other situation could be detected by appropriate CEP rules. For instance, a blocked staircase could be inferred, if numerous persons could not continue their recommended evacuation path along the staircase.

Furthermore, there are other sensors in the smart building that can be exploited to derive certain building states. For instance, the data from temperature and smoke sensor can be used to detect a fire situation in a certain space of the building. There are appropriate CEP rules that derive such situations as well.

3.3 Evacuation Route Recommender Model

An evacuation route recommender model was presented in [4]. For the self-completeness of this work, we describe it briefly in the following. The model is made of the optimization and human factor module. Furthermore, the optimization module is made of the *Routes' safety optimization* component and the *Routes' travel time system optimization with fairness* component, Figure 2.

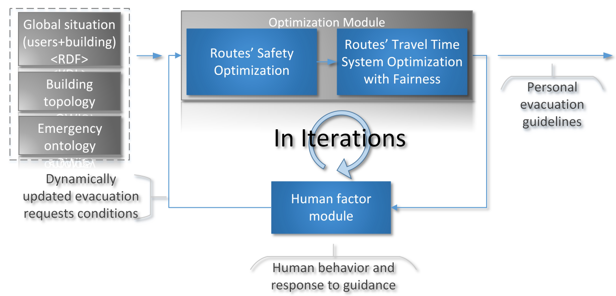


Figure 2. Evacuation route recomender model

Our objective is not only to find routes with satisfied minimal safety conditions since it may occur in hazardous situations that no such route exists. Thus, with the objective to increase the chances of survival, in the *routes' safety optimization*, we need to find routes that maximize Nash social welfare of the safety of the routes. We opt for this choice since it gives the best compromise between the optimization of the evacuees' utilitarian and egalitarian social welfare. Therefore, the safety optimization problem maximizing Nash product of the safeties of the constituent edges of evacuation paths is to be solved.

To facilitate scalability and robustness of the system in the evacuation of large premises, a distributed approach to this route safety optimization problem can be applied, as presented in [5].

Since we treat a highly computationally complex problem, the implementation of this distributed approach to our proposed architecture adds scalability by enabling the computation of the overall routing solution in parallel computation processes where each process is responsible of the computation of an evacuation route for a group of users with similar preferences and constraints in the same section of the building. The solution of the safety optimization model is a connected graph that assures the maximization of routes' safeties.

The basic idea of the module for the routes' travel time system optimization with fairness is as follows. The *route's travel time optimization with fairness* is divided into two layers. On the upper layer, Nash social welfare maximization problem with included envy-freeness and fairness constraints is decomposed to obtain a subproblem that can be optimized individually locally by the

processes described previously. The details on the optimization approach can be found in [12].
Moreover, based on the total demand expressed in terms of person flow per time unit, each process tries to achieve a sufficient number of shortest paths considering fairness for all its evacuees. The processes compute a sufficient number of shortest paths for their evacuees through, e.g., k-shortest path routing algorithm [13]. The prices of networks' edges are adjusted based on the overall processes' demand on the routes influencing congestion on the highly demanded arcs.

The prices are Lagrange multipliers that are calculated through a distributed dual-decomposition of the primal evacuation problem. On the other hand, each process calculates shortest paths to the set of safe exits with updated edges' prices, envy-freeness prices, consistency dual prices, and user demand distribution over routes' prices and thus decides upon the amount of users to be routed on each of the assigned routes.

After the route assignment is made for all evacuation requests on the first level of the optimization model, each process decides, on the second level, of its users' assignment to the routes assigned to it on the first level, based on relevant social welfare parameters that guarantee fairness of the assigned routes to its users through an iterative auction. While the negotiation for the assignment of the routes among different processes on the first level includes the communication among processes when they share the same arc(s), the negotiation through auctions on the second level is local between each process and its users and considers a fair assignment of the available routes based on the users' individual evacuation preferences and mobility constraints.

4 CONCLUSIONS

In this paper we have presented an abstract architecture for situation-aware evacuation guidance in smart building. The system provides an individual evacuation route recommendation to each user of a smart large installation. The proposal takes into account the current location and building state obtained through sensors and personal mobile devices, as well as human factors in emergencies.

We described the architecture and the main technologies proposed to implement it, namely, iBeacons and smartphones for obtaining live building information, CEP for efficiently event processing, and a distributed optimization algorithm for route recommendation.

Our proposal addresses the computational complexity of managing the huge amount of data that can be continuously generated in a large installation. On the one hand, users' smartphones process events perceived from the infrastructure and forward only relevant high level events to the emergency manager. On the other hand, we proposed a distributed evacuation route recommendation algorithm. Moreover, the decision of running the user agent on personal smartphones facilitates dealing with private information.

In the future we plan to test our architecture in a simulated scenario where we will evaluate the correctness of CEP rules and the route recommendation algorithm in different settings. Then, we will deploy a field test in a University building.

ACKNOWLEDGEMENTS

This work has been partially supported by the Autonomous Region of Madrid through grant "MOSI-AGIL-CM" (P2013/ICE-3019) co-funded by EU Structural Funds FSE and FEDER, "SURF" (TIN2015-65515-C4-4-R) funded by the Spanish Ministry of Economy and Competitiveness, and through the Excellence Research Group GES2ME (Ref. 30VCPIGI05) co-funded by URJC and Santander Bank.

REFERENCES

[1] MOSI-AGIL project: https://www.gsi.dit.upm.es/mosi/ (2016).
[2] R. Hermoso, J. Dunkel, and J. Krause, 'Situation Awareness for Push-Based Recommendations in Mobile Devices", *19th Intern. Conf. on Business Information Systems (BIS) conference*, (2016).
[3] G. Adomavicius and A. Tuzhilin. Context-aware recommender systems. In Recommender Systems Handbook, pages 217–253. Springer, 2011.
[4] M. Lujak and S. Ossowski, 'Intelligent People Flow Coordination in Smart Spaces'. *Post-Proceedings of the 13th European Conf. on Multi-Agent Systems / 3rd Int. Conf. on Agreement Technologies*, Rovatsos M. et al. (Eds.), LNAI vol. 9571, 34–49, (2016).
[5] M. Lujak, S. Giordani, S., and S. Ossowski, 'Distributed Safety Optimization in Evacuation of Large Smart Spaces'. *The 9th International Workshop on Agents in Traffic and Transportation (ATT 2016 @ IJCAI 2016)*, (2016).
[6] Apple Inc., *iBeacons* https://developer.apple.com/ibeacon/
[7] W. Zimmermann, *Indoor Navigation with iBeacon Technology*, Master thesis, Hannover University of Applied Sciences and Arts, 2016. [in German]
[8] M. Brown, *Introducing iBeacon Technology at SXSW 2015*. http://www.sxsw.com/news/2015/introducing-ibeacon-technology-sxsw-2015 (2015).
[9] J.R. Kwapisz, G.M. Weiss, and S.A. Moore, '2010: Activity Recognition using Cell Phone Accelerometers', *ACM SIGKDD Explorations*, **12** (2), 74–82, (2010)
[10] D. Luckham, *The Power of Events*. Addison-Wesley, 2002.
[11] O. Etzion and P. Niblett P. *Event Processing in Action*. Manning, 2010.
[12] M. Lujak, S. Giordani, and S. Ossowski, 'Route guidance: Bridging System and User Optimization in Traffic Assignment', *Neurocomputing*, **151** (1), 449-460, (2015).
[13] J.Y. Yen, 'Finding the k shortest loopless paths in a network', *Management Science*, **17** (11), 712-716, (1971).

SmartSim: Improving visualization on social simulation for an emergency evacuation scenario

Antonio M. Diaz and **Ganggao Zhu** and **Alvaro Carrera** and **Carlos A. Iglesias** and **Oscar Araque** [1]

Abstract.
The simulation of indoor evacuation is important for rescue and safety management, while a better visualization of simulation could help users to understand the evacuation plan better and to design the evacuation activities more effectively. The purpose of this paper is to show an indoor evacuation simulator with more realistic graphical user interface for both interacting and visualizing the simulation of evacuation plans. The proposed evacuation simulator combines a social simulation framework UbikSim and a character animation platform SmartBody. UbikSim is used as a back-end social simulation engine for evacuation scene management and evacuation simulation calculation such as computing agent positions and evacuation path. SmartBody is focused on various behaviours and capabilities of agents with digital 3D character in real time, which is used to visualize the locomotion, emotion and facial expressions of agents with more realistic animations in simulation. We develop a connector for SmartBody to control and visualize the simulation by communicating with UbikSim. The proposed evacuation simulator is validated in a real world university evacuation scenario with multiple simulation settings.

1 Introduction

Social simulation is a research field that applies computational methods to study issues in the social sciences. In social simulation, computers aim to imitate human reasoning activities by executing processes, mechanisms and behaviours that build the reality. This approach enables to investigate some complex models that cannot be investigated through mathematical models. Social simulation is considered as a third way of doing science, differing from both the deductive and inductive approach [1], in which simulating a phenomenon is akin to constructing artificial societies. Agent-Based Simulation (ABS) is a kind of social simulation that represents a simulation system as a society of agents that are designed to describe the behaviour of observed social entities such as individuals or groups [6]. Agent based social simulation is very useful to predict the behaviour of individual agents or crowds in complex environments, especially for simulating a dangerous environment and experimenting the possible results of some actions based on simple rules.

Various emergence cases can happen in a building such as fire, earthquake, water leak, and gas leak, to name a few. Crowd evacuations, such as disasters at massive parties, sport events and terrorist attacks can also lead to tragedies when performed without careful planning. In both type of emergencies, effective emergency evacuation is a key component of emergency response. Emergency evacuation preparation activities are required to be developed in advance because they ensure that people can get to safety in case of emergency. However, the evacuation demonstration in case of emergency is not always feasible because of ethical, practical and financial issues [8]. In order to define effective evacuation plans, understanding disasters and crowd emergency evacuation behaviours [3] conveniently with low cost, the ABS can be used to simulate the crowd behaviour and to analyze the effectiveness of evacuation plan. For instance, in a evacuation simulator, the building is modeled and populated by different numbers of agents representing various types of persons (e.g., handicapped persons, etc.) and common emergence situations such as blocked doors. Different agents behaviour according to predefined rules and the results of their actions are measured, hence the best of evacuation model can be selected according to simulation, without risking any real assets and situating human in dangerous situations.

UbikSim 2.0 [11] is such kind of agent-based social simulator to recreate the human behaviour inside a building. Specifically, UbikSim is used to model the map of the building where the emergency simulation takes place. Then, it simulates the virtual users (agents) under emergency and calculates the evacuation path for agents based on various criteria such as least crowd or nearest exit. However, UbikSim has limited features of graphical user interface in controlling and visualizing agents with abundant behaviours and various characters, where agents are represented as simple as equivalent figures in the map and there is no way to inspect visually the agents types (e.g. man or woman) or their emotions (e.g. fear or happiness). In order to enhance the visualization of UbikSim framework, we propose to incorporate SmartBody [12] to provide visualization of agents in an animation approach. More specifically, the agents in the map are represented as human-like 3D animations. The movements of agents can be demonstrated in a more realistic way and with more options such as walk, run or jump. Furthermore, agents are able to express emotions in their animated face and to response to events in an interactive life-like manner such as speak with gestures and face expressions. Moreover, the description of such behaviour is simplified by using Behavior Markup Language (BML) [9] because SmartBody is also a BML realization engine that transforms BML behaviour descriptions into real time animations. Consequently, the proposed system can provide a complete graphic rendering platform to bring various characters with predefined movement animations and behaviour sets together with a social simulation engine. In this way, we could add many different type of agents by simply adding their behaviour descriptions through BML settings. In addition, the system is also designed to be easily extended for future development.

[1] Intelligent Systems Group, Universidad Politécnica de Madrid, Spain, Spain, email: antoniom.diaz.dom@gmail.com, gzhu@dit.upm.es, a.carrera@dit.upm.es, cif@dit.upm.es, oscar.aiborra@alumnos.upm.es

To summarize, we would like to show the main contributions of this work:

- We propose and implement a novel agent-based evacuation simulator, named SmartSim[2], that combines an agent-based social simulator UbikSim with a character animation platform SmartBody.
- The proposed evacuation simulator has been validated and demonstrated in a realistic school building with different simulation scenarios.

The rest of this paper is organized as follows. In section 2, we present the background of this work. Section 3 gives the overview of the proposed evacuation simulator and discusses the implementation details. Finally, we describe the evaluation of the proposed simulator in a school evacuation scenario with different settings in Section 4 and close the paper in Section 5 by showing the main conclusions of this work and providing a possible view on future work.

2 Background

This section introduces some background knowledge of required components to develop the proposed evacuation simulator. We first review the idea and functionality of agent-based social simulator UbikSim in Section 2.1. Then, we introduce the character animation platform SmartBody and the behaviour description language BML in Section 2.2.

2.1 Agent-based Social Simulator UbikSim

Agent-based social simulation [5] is good at predicting the behaviour of agents in complex environments. Ubiksim 2.0 [11] is an implementation of an agent-based social simulator which has been developed by Universidad de Murcia and Universidad Politécnica de Madrid [3]. It is a framework that can be used to develop social simulation which emphasizes the construction of realistic indoor environments, the modeling of realistic human behaviours and the evaluation of Ubiquitous Computing and Ambient Intelligence systems. UbikSim is implemented in Java and employs a number of third-party libraries such as SweetHome3D and MASON. It consists of a console used to launch the simulation as well as a map in 3D or 2D where the position of all the agents involved in the simulation can be visualized.

Moreover, UbikSim tries to be a tool for using Multi-Agent Based Simulation (MABS) [4] in Ambient Intelligence (AmI) [10] which is a computerized environment that is sensitive to human and objects actions. MABS consists of modeling the environments with many artificial agents in order to observe the behaviours of agents, while it is possible to learn about their reactions. In case of evacuation simulation, effective activities can be derived from observing the behaviours of artificial agents and the outcomes of some simulated phenomena in evacuation. These behaviours cannot be observed in non-evacuation conditions. In contrast, other kinds of simulations model the entire environment as mathematical models where the individuals are viewed as a structure that can be characterized by a number of variables. Conventionally, it is not feasible to test a large number of users in AmI, whereas UbikSim enables the simulation of social behaviours from large group of users by applying the MABS approach to AmI environments.

As an example, Figure 1 illustrates a map used for evaluation based on UbikSim, including a demostration of an agent-based simulation.

[2] SmartSim Repository: https://github.com/gsi-upm/SmartSim
[3] Ubiksim Public Repository: https://github.com/emilioserra/UbikSim

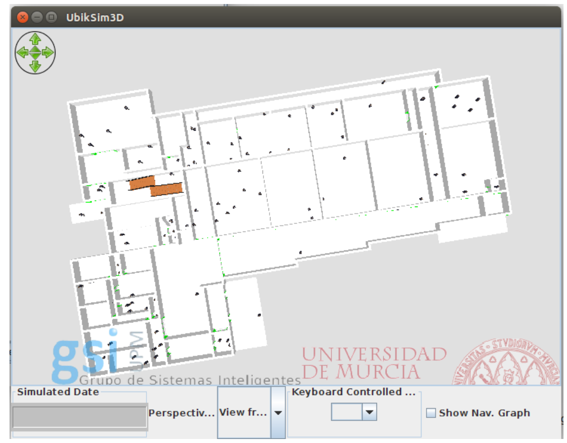


Figure 1. Example of Ubiksim framework for Agent-based Simulation

2.2 SmartBody and BML

SmartBody framework[4] is an open source character animation platform for animating Embodied Conversational Agents (ECAs) [12], which provides capabilities for digital 3D characters in real time such as the animations of locomotion, steering, object manipulation, lip syncing, gazing, non-verbal behaviour or re-targeting. SmartBody contains its own viewer and 3D renderer so that it can be run as a standalone system or incorporated into game or simulation engines. SmartBody is focused on proving various behaviours and interactive characters of artificial agents so we use it as graphical user interface of evacuation simulator, while UbikSim takes charge of scene management and simulation computation. In addition, the life-like behaviour requires the synchronized movement of multiple parts of the agents simulated body. For example, to realize the gaze behaviour requires coordination of eye, head, neck movements. Moreover, to support coherent interpretation of behaviour, the animation of gestures, eye flashes and speech audio must be synchronized in time with each other. SmartBody implements the behaviour realization engine that transforms BML behaviour descriptions into real time animations. As a consequence, we are able to have various predefined animations of agents with different types by describing their different behaviours with BML.

Listing 1. A BML Example

```
<bml id="bml1" characterId="Rachel">
  <required>
     <gaze id="gaze1" target="PERSON1"/>
     <speech id="speech1">
        <text>Welcome</text>
     </speech>
   </required>
   <head id="nod1" type="NOD"/>
</bml>
```

BML is an XML based description language for controlling the verbal and non verbal behaviour of ECAs [2]. BML is used to describe the physical realization of behaviours (e.g. speech and gesture) of the agents expressing them with movements that need to be realized by an agent. Those movements are single elements (e.g. gaze,

[4] SmartBody Web Site: http://smartbody.ict.usc.edu/

speech, head) and listed one after another, as exemplified in Listing 1.

3 SmartSim Simulator

The main goal of the proposed evacuation simulator is to use UbikSim as social simulation engine and to use SmartBody as graphical interface of the simulator. This section presents the implementation details of the proposed simulator system in Section 3.1 and also offers an overview of the simulation gateway in Section 3.2.

3.1 Architecture Overview

The SmartSim simulator system consists of a social simulation module (UbikSim), a simulation gateway module and a graphical visualization module (SmartBody). The overview of the proposed system has been illustrated in Figure 2. The idea is to connect the social simulation engine with the animation engine through the simulation gateway, in order to provide an integrated evacuation simulation system. We rely on the existing simulation engine, while we develop the simulation gateway and incorporate the animation engine into a complete graphical user interface for controlling, managing and visualizing the simulation.

The social simulation module is based on UbikSim and is used for managing agents, describing the emergency scenario, modeling the indoor evacuation environment and creating evacuation plan. The graphical visualization module is based on SmartBody and is used for visualizing the agent behaviour in life-like animation in simulation. To combine UbikSim and SmartBody, we implement a simulation gateway that helps to manage the social simulation configuration and to provide communication between UbikSim and SmartBody in real time while running the simulation. Moreover, a user friendly graphical user interface based on SmartBody has been implemented to utilized the simulation gateway so that end users can manage and visualize the simulation conveniently.

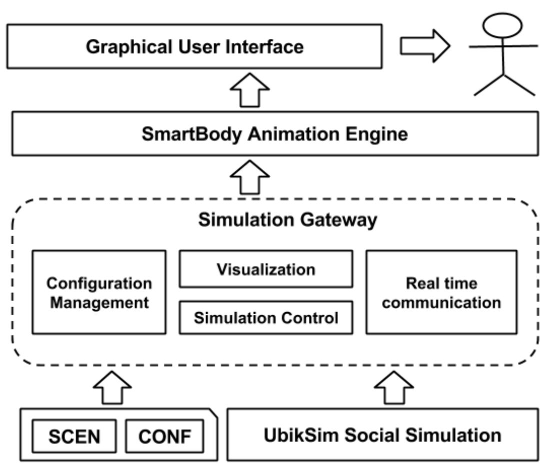


Figure 2. General Architecture of SmartSim simulator

In addition, UbikSim provides a scene editor that can pass the environment map to SmartBody. As SmartBody is not able to perform the simulation and calculate the paths that agents have to follow in order to evacuate the building. It relies on UbikSim to perform simulation computation and retrieves position data from UbikSim in real time. Within the map, and position data in run time, SmartBody presents a realistic 3D evacuation environment and enables users to control the simulation such as pausing or advancing.

UbikSim has many kinds of options, such as editing and creating artificial environment with an easy to use interface, configuring various number of agents. The communication between UbikSim and SmartBody is based on Representational State Transfer (RESTful) [7] architecture through web requests.

3.2 Simulation Gateway Implementation

The simulation gateway is composed by four different modules: simulation configuration module, graphical visualization module, real time communication module and simulation control module. The **simulation configuration module** parses user defined configuration of simulation such as agent numbers, emergency scenarios, initial positions and evacuation plans. Some relevant configuration options are illustrated in Table 1. Then, the configuration data are passed to UbikSim social simulator through web request API and to SmartBody through its Python API. According to the configuration data, social simulator initializes the simulation, creating the agents and setting their positions. The scenario resources are loaded to set the mark for emergency such as emergency position. The character resources such as skeleton and polygonal model are loaded for further usage of SmartBody.

Option	Description
amountAgents	The number of agents in our simulation.
amountLeaders	The number of leaders in our simulation.
ubikSimServer	The address of UbikSim server.
meshScenario	The scenario file for simulation.
modeSimulation	The possible simulation modes.

Table 1. Summary of SmartSim configuration options.

Based on configuration data, SmartBody creates the simulation scene (e.g. maps and agents) and starts the **graphical visualization module** that calls the graphic interface of SmartBody and a default camera to display the simulation. Moreover, the configuration module also loads the description resources for different character of agents from BML description resources, so the different type of animation can be rendered in simulation.

Option	Effect
output=web	Displays the web graphic interface.
control=pause	Executes the pause control.
control=play	Executes the play control.
control=stop	Executes the stop control.
control=frames	Starts the displayers in the server side.
position=people	Returns the agents positions.
position=map	Returns the map coordinates and obstacles.
position=emergency	Returns the emergency position and room.
position=(id,x,y)	Adds the agent to the position.

Table 2. Summary of UbikSim API actions.

The **real time communication module** retrieves agents' positions and paths from UbikSim and converts them into specific form of position and path for SmartBody. Consequently, the SmartBody can present the animation of agents that execute the evacuation plan. The UbikSim simulation run time Web API is illustrated in Table 2.

Furthermore, a **simulation control module** is implemented in SmartBody to control the simulation and make agents follow their path. It can be used to control every step of simulation and make agents *Pause*, *Advance*, and *Stop*. The actions of those commands are passed to UbikSim through the real-time communication module, so the gateway is able to coordinate the simulation in real time between UbikSim and SmartBody. After simulation finished, the simulation control module will record the simulation results containing the time that an agent spent to exit the building from its initial position and more relevant data for further analysis.

4 Use case scenarios

The implemented evacuation simulator system has been validated in a real use case scenario which is simulating evacuation activities. The indoor environment is selected as the building B of the School of Telecommunication Engineers (ETSIT) of the Universidad Politécnica de Madrid. A demonstration video of all the validation tasks can be found in YouTube[5]. The implementation of the simulator as well as validation case studies are published and available in a public Github repository[6]. We will first introduce how we create the validation map in Section 4.1 and then present three evacuation scenarios. Section 4.2 illustrates a single agent scenario where an agent escapes, leaving the building from any initial position of the building. Also, this section presents a more complicated case where multiple agents evacuate the building following a agent leader. Finally, we present a more realistic social simulation that different type of agents escape the building from different initial position and adopt different evacuation path.

4.1 Map Creation

The map of the building has been modeled in UbikSim and is illustrated in Figure 1. The generated map file has been exported to SmartBody in configuration. Note that any polygonal model generated with 3D modeling program such as Blender could fit the requirement of SmartBody. UbikSim editor is based on SweetHome3D which is a free indoor design application. We can draw the map of our scenario, arrange furniture on it and visualize the results in 3D. It is also easy to create a scenario as drawing the walls and rooms. Several objects libraries have been added and can be imported to the editor, which can add completion and detail of our scenario. We implemented an extension in UbikSim, so the created scene can be exported to SmartBody automatically. Figure 3 shows the 3D school map in SmartBody GUI that have been passed from UbikSim, using the map shown in Figure 1.

4.2 Single Agent Escaping

As mentioned previously, the simulation of agents is based on UbikSim, while SmartBody retrieves the paths and positions from it. By configuring scene in the UbikSim editor, we set the positions and numbers of agents, scenarios and location of the emergency. All these data is passed to SmartBody via the simulation gateway. UbikSim also retrieves the initial configuration from simulation gateway. The

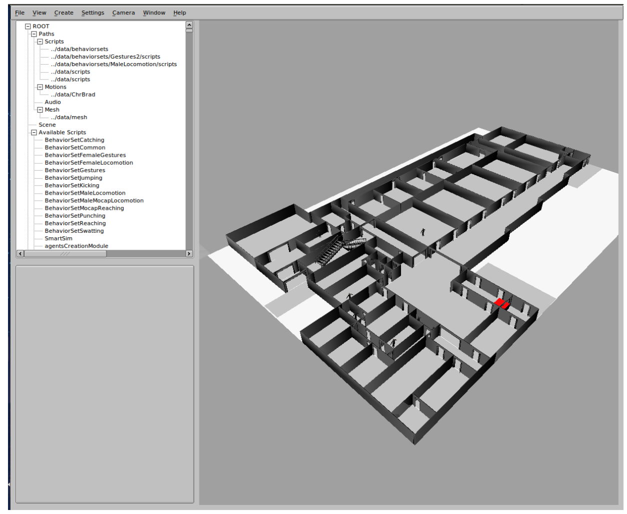


Figure 3. The map model loaded in SmartBody GUI automatically imported from UbikSim.

simulation result data are also generated by simulation gateway containing the exit time of each agent. We first validate the system in a scenario of evacuation of single agent from the building. Agent will escape the building following the path given by UbikSim. We demonstrate the emergency and the character escaping the building. The configuration of simulation is set as single agent without any character. The agent needs to exit the building from his initial position based on the predefined path. SmartBody is set up to show the evacuation of agent with animations, while simulation gateway will record the time the agent used to escape the building. This scenario is used to validate the system correctness.

The second scenario is the extension from the previous one by adding the number of agents and a simple evacuation model. The escaping in a crowd is a common phenomena in evacuation and is the main place that dangerous situation may appear. In the crowd simulation, we design a number of agents and one of these agents become the leader, while the other agents will follow the leader from their initial point to the exit. This scenario helps to extend the previous scenario with consideration of multiple agents.

It is a common phenomena in evacuation plans some crowds are leaded by a leader. The setting is similar to the previous case, while we also define the numbers of leaders and their following groups of agents. The non-leader agents will follow the path as their assigned leader. After simulation, the exit time of all the agents will be recorded. Figure 4 shows the animation of crowd escaping with leaders in SmartBody GUI. This scenario can help validate the performance of simulator with multiple agents and validate the correctness of evacuation plan execution.

4.3 Social Simulation with Characters and Emotions

Finally, we set a more realistic simulation scenario, where multiple agents with different type of characters escaping the building from different initial position following different paths. Figure 5 shows the screenshot of animation of agents starting from different location and execute different evacuation plans. It has been shown in Figure 5 that

[5] SmartSim Video: https://www.youtube.com/watch?v=8kGKD8Ofxuw
[6] SmartSim Repository: https://github.com/gsi-upm/SmartSim

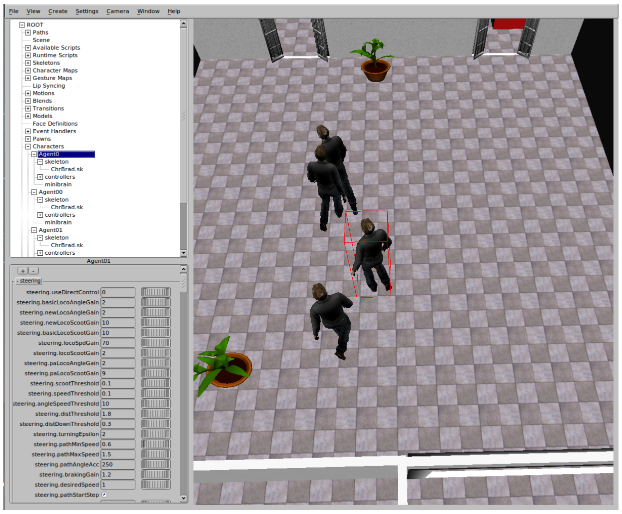


Figure 4. The Crowd Escaping

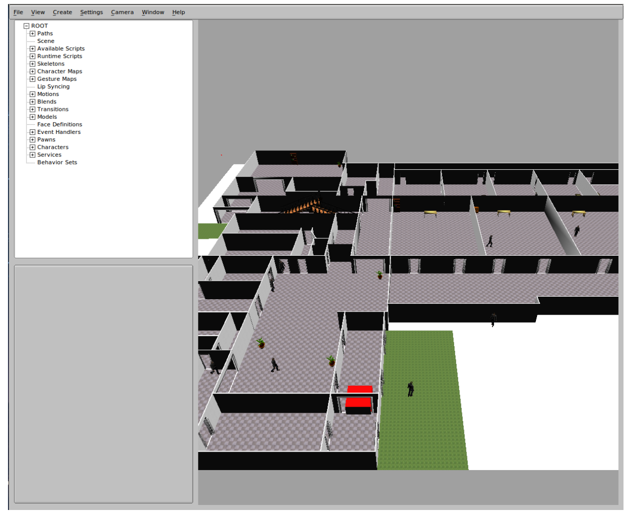


Figure 5. The agents escaping from different places

the simulator is able to present the social simulation of emergency evacuation correctly. Moreover, the visualization of the simulation become more realistic because there more kind of agents with different emotions. The previous scenario has leader and follower characters, while the agents can have different gender or ages. For example, Figure 6 illustrates a female agent named Rachel which is different from the previous male agents. The SmartBody and BML enable the animation of agents in a life-like way. By defining the behaviours in BML files, agents can have different motions and face expression to represent more human-like behaviours. For example, Figure 7 illustrates the agent expressing his happiness. This is achieved by configuring the face element in BML and realized by SmartBody. We believe that enabling the agents to express their feelings in face such as fear in facing an emergency and happiness after evacuation can make the visualization of simulation more realistic and help to make the evacuation plans better.

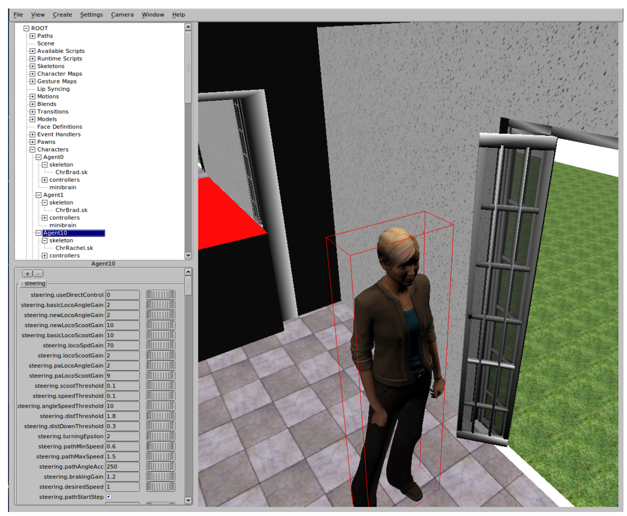


Figure 6. A 'Rachel' type character

5 Conclusions and Future Works

This paper presents an agent-based simulation system, named SmartSim, for evacuations based on Ubiksim, where the graphical interface has been enhanced with realistic animations and emotions in agents using SmartBody.

The interaction between UbikSim and SmartBody, which allowed end-users to interact with simulation systems conveniently and visualize the simulation more powerfully, is implemented in different modules written in Python. The system is designed as modular components that can ease the future implementation of various simulation purposes. The system has provided facilities for creating simulation scenarios easily based on simple configuration file and those scenarios can be exported to UbikSim and SmartBody automatically. The visualization of simulation is achieved by very detailed artificial agents in animations. Furthermore, agents are able to express emotions and various behaviours which make our simulator more realistic. End users are allowed to select the numbers of agents as well as their types with particular animation and behaviours.

Several research lines that can be considered as following work to continue and extend the features of this work. Firstly, a graphical interface for scene control might be useful to help users in avoiding mistakes in defining agent commands. Secondly, although SmartBody offers very good performance of visualizing agent's animations, it can be integrated with a graphical engine such as Unity to improve the quality of animation. Finally, apart from the current desktop version, we are planning to implement a mobile version or web-base version.

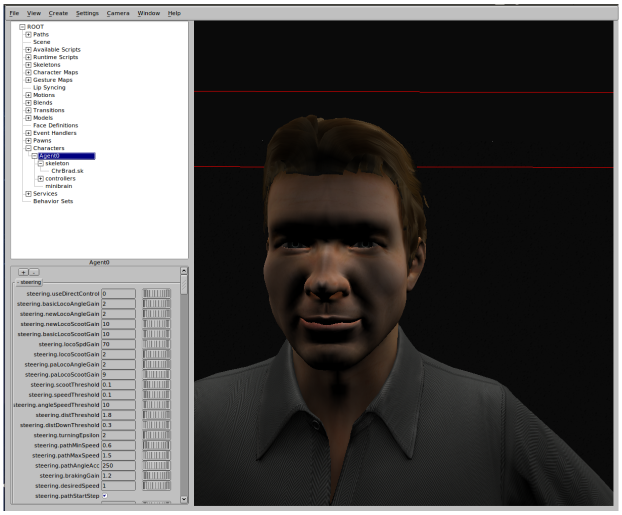


Figure 7. An agent expressing happiness

ACKNOWLEDGEMENTS

This work is supported by the Spanish Ministry of Economy and Competitiveness under the R&D projects SEMOLA (TEC2015-68284-R) and MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER).

REFERENCES

[1] Robert Axelrod, 'Advancing the art of simulation in the social sciences', *Japanese Journal for Management Information System*, **12**(3), (2003).
[2] Justine Cassell, *Embodied conversational agents*, MIT press, 2000.
[3] W Challenger, WC Clegg, and AM Robinson, 'Understanding crowd behaviours: Guidance and lessons identified', *UK Cabinet Office*, (2009).
[4] Paul Davidsson, 'Multi agent based simulation: beyond social simulation', in *Multi-Agent-Based Simulation*, 97–107, Springer, (2000).
[5] Paul Davidsson, 'Multi agent based simulation: Beyond socialsimulation', (2000).
[6] Paul Davidsson, 'Agent based social simulation: A computer science view', *Journal of artificial societies and social simulation*, **5**(1), (2002).
[7] Roy Thomas Fielding, *Architectural styles and the design of network-based software architectures*, Ph.D. dissertation, University of California, Irvine, 2000.
[8] Steve Gwynne, ER Galea, M Owen, Peter J Lawrence, and L Filippidis, 'A review of the methodologies used in the computer simulation of evacuation from the built environment', *Building and environment*, **34**(6), 741–749, (1999).
[9] Stefan Kopp, Brigitte Krenn, Stacy Marsella, Andrew N. Marshall, Catherine Pelachaud, Hannes Pirker, Kristinn R. Thrisson, and Hannes Vilhjlmsson4, 'Towards a common framework for multimodal generation: The behavior markup language', (2006).
[10] Emilio Serrano and Juan Botia, 'Validating ambient intelligence based ubiquitous computing systems by means of artificial societies', *Information Sciences*, **222**(0), 3 – 24, (2013). Including Special Section on New Trends in Ambient Intelligence and Bio-inspired Systems.
[11] Emilio Serrano, Geovanny Poveda, and Mercedes Garijo, 'Towards a holistic framework for the evaluation of emergency plans in indoor environments', *Sensors*, **14**(3), 4513–4535, (2014).
[12] Marcus Thiebaux, Stacy Marsella, Andrew N. Marshall, and Marcelo Kallmann, 'Smartbody: Behavior realization for embodied conversational agents', in *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '08, pp. 151–158, Richland, SC, (2008). International Foundation for Autonomous Agents and Multiagent Systems.

User Modelling Languages for AmI: A Case Study on Road Traffic

Alberto Fernández-Isabel and **Rubén Fuentes Fernández**[1]

Abstract. Traffic is an important phenomenon in modern societies. Its complexity and the difficulties to control the actual settings where it happens have made of simulation a key tool for its study. This approach requires suitable models to capture all its relevant aspects and their mutual influences. Among these aspects, people are the key one. However, there is a limited understanding of people attitudes and behaviours and their effect on traffic. Thus, simulation has here an important component of exploration of hypotheses. Our research contributes to this line of work with a set of general and extensible agent-based models about people in traffic. These models integrate existing research from Social Sciences and simulation. The agent paradigm supports the explicit specification of processes of information management, decision-making, action execution, and interaction both with people and the environment. Such approach facilitates model reuse, and linkage between different elements relevant for the studies. The paper illustrates the application of these models with a case study that shows how to integrate in them a well-known model for drivers attitudes.

1 INTRODUCTION

Life in modern societies is highly mediated by traffic. Every day, millions of persons move on foot and by private vehicles or public transport. These flows are organized according to certain social rules, but also depend on individual attitudes and behaviours, unexpected events happening in the environment, and their mutual influences. Given the difficulties to carry out these studies with real settings, models have emerged as a key tool to study traffic.

There are several approaches to model traffic [16]. Analytical models rely on a strong abstraction of the individual components described mainly with mathematical formulas [8]. They are useful to consider phenomena with large populations, but have limitations regarding the specification of procedural and non-linear behaviors, and heterogeneous populations. As an alternative, simulation facilitates the specification of these kinds of behavior and population, but it is not usually intuitive the correspondence between the actual system and its computational representation. Agent-Based Modelling (ABM) [8] addresses this problem using agents as its core modelling primitive. Agents are intentional abstractions conceptualized in terms of elements such as knowledge, goals or capabilities. They are able to interact with other agents and their environment. These features facilitate describing people behaviour with agents. However, making realistic models still demands a high effort to integrate different theories and give the needed information.

[1] University Complutense of Madrid, Spain, email: afernandezisabel, ruben@fdi.ucm.es

The work presented in this paper pursues reducing this effort by providing base models for people acting in traffic phenomena. These models are part of a wider effort to build a general framework for traffic simulation based on ABM, so they are designed looking for reusability with different theories and contexts. For this purpose, the basis of the models is a classification of people with three dimensions: their role in traffic, traits, and current state.

People role in traffic depends on their mean of transport and their relation with it People are classified in drivers, passengers of vehicles and pedestrians. All of them can be modelled at the individual level or as groups moving together.

People traits represent features of people that are permanent for the travels considered in the simulation. They include physical attributes of the body, such as age, gender or disabilities. There are also attitudes to capture personality and mental features. For instance, people can be more or less aggressive and have personal problems that increase their stress. Moreover, people get more traffic experience over time. This empowers them with additional knowledge and skills to face traffic situations through their learning.

The current state captures dynamic features that depend on the specific context and moment. For instance, they indicate if drivers attention is low because there are distracting passengers, or if traffic conditions bother the driver.

The suitability of these models is illustrated with a case study that specifies existent work on traffic simulation using the proposed primitives. It considers the simulation in [13] about drivers attitudes and their influence on group behaviour.

The rest of the paper is organized as follows. Section 2 introduces the agent modelling language used in our approach. Section 3 presents the models for people in traffic with that language, and grounds them in available research about traffic. Section 4 compares these models with those in [13] regarding the phenomenon called dominance at junctions. These results are further compared with related work in Section 5. Finally, Section 6 discusses some conclusions and future work.

2 AGENT-BASED MODELLING LANGUAGE

This work specifies their models using the modelling language of the INGENIAS methodology [14]. The key concept of INGENIAS is the agent.

An agent is an intentional entity that follows its own agenda characterized by goals. In order to achieve these goals, the agent is able to carry out certain tasks. An agent can trigger a task when it pursues an unsatisfied goal that the task is potentially able to fulfil, and all the elements required by the task are available. These elements are usually pieces of information known by the agent or events coming from the environment. As the result of the execution of the task, the agent acts on the environment and produces or modifies information.

Agents act on and perceive the environment through external applications. These are the sources of the events and tasks use their methods to affect the environment.

A final element relevant for the models presented in this paper is the AInherits relationships. This is an inheritance relationship that allows defining a type of agent as an extension or constraint of another type of agent. Thus, it highlights the common features of different types of agent and saves modelling effort.

3 TAXONOMY OF PEOPLE IN TRAFFIC

Traffic is the organized way of moving people using different means of transport. This people have as their main goal to arrive fast and safely to their destinations [7, 9]. They can achieve this goal through alternative sequences of actions as long as they meet some constraints. First, people use different means of transport, and can control them or be a passenger. This makes suitable only some routes and implies certain rules. Our work only considers those means sharing spaces in our cities and roads, e.g. on foot or by car. Second, the sequence also depends on the physical and mental characteristics of people and their current state. However, models cannot consider all the known people features and processes. This would be unsuitable regarding efficiency and abstraction, and even incorrect given our limited understanding of the phenomena. The taxonomy presented proposes a number of features based on literature, mainly coming from Social Sciences, widely accepted as relevant for traffic studies. Next subsections present in details all these aspects.

3.1 ROLE AND MEAN OF TRANSPORT

The behaviour of a person in traffic is first limited by his/her mean of transport. Although passengers influence traffic, e.g. distracting drivers, these models focus on people controlling their mean of transport.

The mean of transport requires certain processes to manage it, and also makes possible some processes. For instance, a person can know how to brake a car, but needs driving one to perform the action. At the same time, different means of transport obey different rules. These can be both explicit, e.g. traffic regulations, and implicit, e.g. drivers facilitating other drivers maneuvers.

The models represent this information with a hierarchy of agents (see Figure 1). The basic agent is person, which incorporates the goal of arriving to a destination following a certain route and perception of obstacles and signals. According to the mean of transport, a person is extended as a driver or a pedestrian. These agents have additional goals, information and tasks to move by car or on foot respectively. In the case of the driver, there is a related vehicle. The vehicle is represented as an external application with, for instance, methods to brake or to manage the steering wheel and events from the speed indicator.

3.2 TRAITS OF PEOPLE INVOLVED IN TRAFFIC

The way of behaving in traffic also depends on the personal traits of each person. A well-known example is the differences in accidents regarding age and gender [12]. These traits are static for each person, as they do not change during the travel, and thus in the simulation. There are three groups of traits: physical attributes, attitudes and traffic experience.

The group of physical attributes currently comprehends gender and age. The gender attribute classifies people into male or female. The age attribute uses four levels, young, adult-young, mature and elder. The age levels are different for drivers and pedestrians, as people can walk before they can drive and the required capabilities for both activities are different. These attributes mainly affect perception and reaction parameters, such as sight distance and time to maneuver.

Other group of traits is the attitudes. Models consider in it a traffic profile and relationship problems. The traffic profile is based on an extension of the selfish principle in [13], classifying drivers as aggressive, normal or moderate, and pedestrians as reckless, normal or prudent. This classification differentiates, for instance, between drivers who always drive below or at speed limits, or on the contrary usually break them. The relationship problems acknowledge this as a classical source of anxiety and distractions in traffic situations, making more likely suffering an accident or taking greater risks [18].

The last group of traits is the traffic experience. It classifies individuals regarding their traffic learning with values between 0 and 5, being 5 the maximum experience.

3.3 CURRENT STATE OF A PERSON

The traffic and personal conditions change during travels, and this affects people behaviour. For instance, drivers caught in a traffic jam can start relaxed, but their frustration and impatience will rise as they waste more time stuck, which can cause risky situations in their nearby environment. The models consider these dynamic features of behaviour with the attributes belonging to the current state of the person agent. Figure 2 shows these attributes classified in physical state and mood, depending on when they affect physical action and perception or thinking and attitudes respectively.

The physical state influences the perception of the environment. Individuals do not receive objective information from the environment, as this is really mediated by their own senses and depends on external conditions [15]. The personal conditions are represented with the values for this attribute, which are focused, drowsy, distracted and drugged. The influence of the external conditions is represented using the environment entity.

The environment entity has attributes for the weather conditions and type of environment [18]. The first one takes values between sunny, cloudy, rainy, heavy rain, windy, snowy, ice and foggy, while the second one is classified as familiar, unknown, difficult, affordable and straightforward. These attributes are linked to the physical state, pointing out that they affect its value.

The mood considers that external factors influence people mental state [7]. This state affects aspects such as decision making or level of attention to the environment. The specification of this attribute is further decomposed into the attributes impatience and self-confidence. The impatience represents the frustration of the person, perhaps because she/he is in a hurry or the traffic conditions are adverse. The self-confidence represents the assurance of individuals on their own knowledge, capabilities and skills. Both attributes take values between 0 and 5. Depending on the value, they can have a positive or negative effect on the person processes. For instance, a person with self-confidence 5 can make risky decisions that are inadequate for the perceived situation. In the case of pedestrians, the self-confidence also indicates how crowds influence individual trajectories [9].

As it happened with the physical state, the value of the self-confidence is affected by other attributes. A familiar type of environment and good weather conditions increase the self-confidence. Moreover, people frequently move in groups, and this companion al-

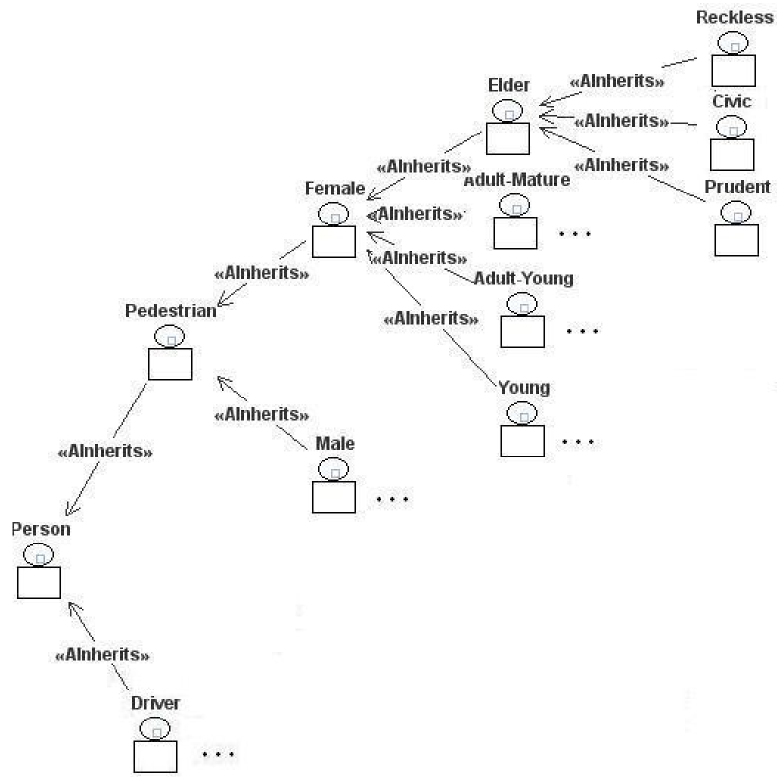


Figure 1. Inheritance of driver and pedestrian agents.

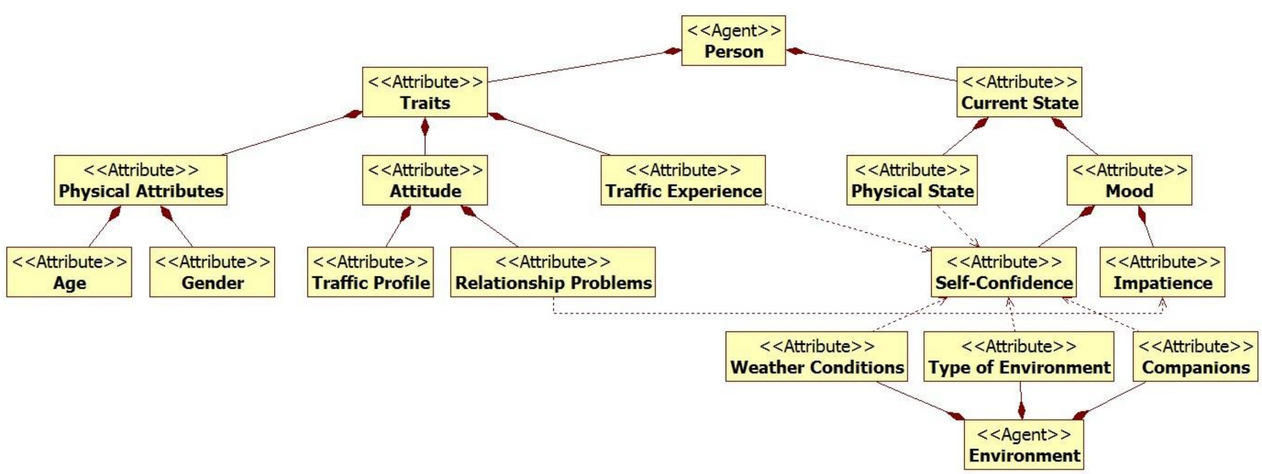


Figure 2. Relationships between the elements of the taxonomy. Dotted lines represent that an attribute affects the calculation of other.

ters the self-confidence with comments or actions. The companions attribute gathers this information. It considers the attitude of the companions with values in silence, little chatty, chatty and fun-loving, and the number of individuals.

Note that this presentation has pointed out several mutual relationships between attributes. For instance, a bad physical state worsens the perception of the road, reducing the self-confidence. In the models, tasks managing the internal state of agents implement these mutual influences.

4 CASE STUDY

The attribute traffic profile presented in this paper is based on the classification of drivers in [13]. This classification uses the selfish principle, which assumes that any driver has a certain level of selfishness when pursuing his or her goals. This level classifies drivers in moderate, normal and aggressive, determining their speed or proneness to make risky decisions. The main limitation of that work is that two drivers of the same group do not differ in their behaviour, which is not a realistic approach. This case study considers how the models proposed in our work cover the previous classification and facilitate its extension.

As previously mentioned, the traffic profile trivially supports the classification in [13]. Its effect over driving depends on the implementation of the tasks of the different agents. Note that since this is an attribute of the person agent, which is the base type of all our agents, all the agents in our models include that attribute.

The heterogeneity of behaviour for agents with the same traffic profile is achieved with several attributes. The impatience is particularly relevant in this context, as it captures the anxiety produced by the current traffic situation.

The original work in [13] also discusses the phenomenon appearing at junctions known as dominance. It happens when a driver or a group of them who are in a lane of a junction push their way, followed by other cars, and get to block the other lanes. This lane of cars will be the only one able to move forward as long as they do not free the junction. If drivers of two or more lanes of a junction exhibit this behaviour at the same time, they can produce a deadlock where nobody will go forward.

With the presented models, this kind of behavior can be the consequence of the traffic profile and certain attributes present in the current state of the person. An aggressive driver is more dominant at a junction than a moderate or normal one, and therefore the former tries to cross the junction with greater determination. When drivers have the same traffic profile, their current state is also crucial for the dominance. The attributes of current state more directly involved in this behavior are impatience and self-confidence. A high impatience makes the driver prone to make quick decisions, not always enough meditated. With a high self-confidence, the driver dares to perform maneuvers that in other circumstance she/he would not carry out. On the contrary, a low self-confidence leads the driver to doubt about maneuvers in complex settings (e.g. many cars around), causing that other more determined drivers cross before her/him. Furthermore, drivers that are more impatient push others, which increase the frustration of the later. This causes a widespread anxious mood in the junction [11], which makes it more hazardous.

The update of the impatience level requires that agents know their position and that of their neighbors. The own position can be included as a new attribute position of the driver agent. The positions of the other drivers are known through interactions of the driver with an external application that mediates its perception. Direct communication between driver agents is not suitable, as drivers are in their own vehicles. The perceived agents could depend on new physical attributes related to sight.

This approach enables that the effects of the driver attitude in the simulation can dynamically change. Such effects are modulated by attributes of the current state, which are influenced by traffic conditions, e.g. surrounding vehicles, speed or time waiting. Therefore, studies using these models provide behavior that is more realistic.

5 RELATED WORK

The current research must be framed within two main lines of work: studies on people and their behaviour regarding traffic, and traffic simulations. Both of them are sources of information to develop our models and validate them.

Studies on actual people provide information on the relevant attributes regarding traffic and the actual processes involved in it. These studies typically focus on obtaining data, statistics and relationships among some factors under scrutiny. For instance, they try to identify aggressive behaviour and the reasons of their appearance [17]. Some commonly considered attributes in these studies are gender and age, as in [4, 12], presence of passengers in a vehicle, the weekday and the most troubled hours [4], the driving experience of drivers [10], the physical state [18], and the mood [10]. There are also behavioral studies more focused on the driving processes. These studies monitor, for instance, physiological signals, gestures or speech to identify and/or predict decision-making, low-level maneuvers or drivers mood [19]. These studies propose attributes and processes that could be considered for simulation, but have some limitations for this purpose. They are difficult to use to validate simulation models or check the influence of new elements, as this would imply researchers to carry out new studies.

The previous knowledge has been used in a variety of simulations with different goals. Regarding the level of abstraction at which traffic phenomena are considered, these simulations can be classified in: macroscopic, mesoscopic and microscopic simulations. The first ones attempt to capture the general principles governing the system instead of individuals, in a way similarly to analytical models. They are typically used to represent large areas of terrain with large quantities of vehicles and traffic infrastructure conflicts [3, 16]. On the contrary, microscopic simulations present individual elements with higher complexity. Most of ABM in traffic belongs to this category [5]. The related computational costs make them suitable only to represent small areas of terrain with few individuals. Moreover, it is difficult to embed general rules of behaviour in them, as rules usually appear on each agent. Mesoscopic simulations are hybrid between the previous types. They try to solve their limitations locating each information or behavior at the most suitable level, either individuals or groups [2]. Our work belongs to this category. As it is based on INGENIAS [14], it supports modelling both individual agents and groups (not shown in this paper), as well as inheritance hierarchies involving those abstractions.

As a distinctive feature of our research, it works with simulations at the level of models. As shown in [6], this facilitates the automated generation of simulations for different target platforms from the specifications using model-driven techniques. Other works have this information embedded in their programming tools [1], reducing the possibilities of studying and reusing that knowledge.

6 CONCLUSION

This work has presented the models of a taxonomy of people regarding their participation in traffic phenomena. These models are intended to provide the basis for an extensible specification able to integrate available research and applicable to develop simulations.

The taxonomy is organized around three main dimensions: the mean of transport used by people, their traits and current state. The mean of transport currently only distinguishes between pedestrians and people travelling using some motor vehicle, and among the later between drivers and passengers. Traits provide information about static features of people, both physical (e.g. gender and age), attitudinal (i.e. traffic profile) and based on experience (i.e. traffic experience). The current state considers several dynamic attributes that depend on the current environment and traffic state, such as self-confidence and impatience. Among all these attributes, the discussions have been focused on the traffic profile, which distinguishes moderate, normal and aggressive people, and self-confidence and impatience as dynamic modifiers of that profile. The relationships between these attributes illustrate how people behavior can be linked to specific situations and combinations of attributes.

The case study has shown how to use the proposed models to specify the simulation in [13]. The drivers attitude from [13] is the traffic profile of our models, but our modes additionally fix some of the limitations pointed out in that work. In particular, they offer a simple way to introduce heterogeneity in the behavior of drivers in each attitudinal group using the current state. At the same time, the specifications in the case study were able to replicate other phenomena appearing in [13] such as dominance.

Working at the level of models facilitates comparing approaches and reusing information between different studies. It also reduces the costs of migration between different simulation platforms, as the relevant information is available at a higher level of abstraction than that of code. Moreover, it promotes the design of domain specific modeling languages for different needs in traffic studies.

The current models are part of an ongoing effort to build a general simulation platform for traffic. The current prototype integrates the previous information in multi-agent systems based on the A-Globe platform and using geographical information from Google . The development process is evolving to a fully model-driven approach in order to explore the actual benefits of these approaches for simulation. Regarding the models, those presented here still does not consider several relevant aspects of traffic. Some of those to be included are vehicles, companion agents or public transport. There is also need to consider extensions of the modelling language to facilitate the specification of, for instance, relationships between attributes and the influence of these on the execution of certain actions.

7 ACKNOWLEDGEMENTS

This work has been done in the context of the project "Social Ambient Assisting Living - Methods (SociAAL)" (grant TIN2011-28335-C02-01) supported by the Spanish Ministry for Economy and Competitiveness, the research programme MOSI-AGIL-CM (grant S2013/ICE-3019) supported by the Autonomous Region of Madrid and co-funded by EU Structural Funds FSE and FEDER, and the "Programa de Creación y Consolidación de Grupos de Investigación" (UCM-BSCH GR35/10-A).

REFERENCES

[1] Moshe Ben-Akiva, Michel Bierlaire, Haris Koutsopoulos, and Rabi Mishalani, 'Dynamit: a simulation-based system for traffic prediction', in *DACCORS short term forecasting workshop, The Netherlands*. Citeseer, (1998).

[2] Wilco Burghout, Haris Koutsopoulos, and Ingmar Andreasson, 'Hybrid mesoscopic-microscopic traffic simulation', *Transportation Research Record: Journal of the Transportation Research Board*, (1934), 218–255, (2005).

[3] Nurhan Cetin, Adrian Burri, and Kai Nagel, 'A large-scale agent-based traffic microsimulation based on queue model', in *IN PROCEEDINGS OF SWISS TRANSPORT RESEARCH CONFERENCE (STRC), MONTE VERITA, CH*. Citeseer, (2003).

[4] Sean T Doherty, Jean C Andrey, and Carolyn MacGregor, 'The situational risks of young drivers: The influence of passengers, time of day and day of week on accident rates', *Accident Analysis & Prevention*, **30**(1), 45–52, (1998).

[5] Patrick AM Ehlert and Leon JM Rothkrantz, 'Microscopic traffic simulation with reactive driving agents', in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pp. 860–865. IEEE, (2001).

[6] Rubén Fuentes-Fernández, José M Galán, Samer Hassan, Adolfo López-Paredes, and Juan Pavón, 'Application of model driven techniques for agent-based simulation', in *Advances in Practical Applications of Agents and Multiagent Systems*, 81–90, Springer, (2010).

[7] Ray Fuller, 'Towards a general theory of driver behaviour', *Accident Analysis & Prevention*, **37**(3), 461–472, (2005).

[8] Nigel Gilbert and Klaus Troitzsch, *Simulation for the social scientist*, McGraw-Hill Education (UK), 2005.

[9] Dirk Helbing and Peter Molnar, 'Social force model for pedestrian dynamics', *Physical review E*, **51**(5), 4282, (1995).

[10] Timo Lajunen, Dianne Parker, and Stephen G Stradling, 'Dimensions of driver anger, aggressive and highway code violations and their mediation by safety orientation in uk drivers', *Transportation Research Part F: Traffic Psychology and Behaviour*, **1**(2), 107–121, (1998).

[11] Timo Lajunen, Dianne Parker, and Heikki Summala, 'Does traffic congestion increase driver aggression?', *Transportation Research Part F: Traffic Psychology and Behaviour*, **2**(4), 225–236, (1999).

[12] Dawn L Massie, Kenneth L Campbell, and Allan F Williams, 'Traffic accident involvement rates by driver age and gender', *Accident Analysis & Prevention*, **27**(1), 73–87, (1995).

[13] Praveen Paruchuri, Alok Reddy Pullalarevu, and Kamalakar Karlapalem, 'Multi agent simulation of unorganized traffic', in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pp. 176–183. ACM, (2002).

[14] Juan Pavón, Jorge J Gómez-Sanz, and Rubén Fuentes, 'The ingenias methodology and tools', *Agent-oriented methodologies*, **9**, 236–276, (2005).

[15] Dario Salvucci, Erwin Boer, and Andrew Liu, 'Toward an integrated model of driver behavior in cognitive architecture', *Transportation Research Record: Journal of the Transportation Research Board*, (1779), 9–16, (2001).

[16] Jason Sewall, David Wilkie, Paul Merrell, and Ming C Lin, 'Continuum traffic simulation', in *Computer Graphics Forum*, volume 29, pp. 439–448. Wiley Online Library, (2010).

[17] David Shinar, 'Aggressive driving: the contribution of the drivers and the situation', *Transportation Research Part F: Traffic Psychology and Behaviour*, **1**(2), 137–160, (1998).

[18] Michael Sivak, 'Human factors and highway-accident causation: Some theoretical considerations', *Accident Analysis & Prevention*, **13**(2), 61–64, (1981).

[19] Shigeki Tezuka, Hitoshi Soma, and Katsuya Tanifuji, 'A study of driver behavior inference model at time of lane change using bayesian networks', in *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, pp. 2308–2313. IEEE, (2006).

Towards the Simulation of Large Environments

Jorge J. Gomez-Sanz and **Rafael Pax** and **Millán Arroyo** [1]

Abstract. The development of a smart environment working into large facilities is not a trivial matter. What kind of intelligence is needed and how this intelligence will interact with individuals is a critical issue that cannot be solved just by thinking about the problem. A combination of social and computer science methods is necessary to learn and model the interplay between the environment and the environment inhabitants. This paper contributes with an ongoing case study that exemplifies this kind of combination. The case study considers two faculty buildings and a behavior to be modified. The goal is to design a set of devices that sends signals to passing-by pedestrians in order to make them use more the staircases. Banners, videos, and directed intervention are used. The effect of each one is measured and such measurements are reproduced into computer simulations. This information is necessary in order to determine the duration, the intensity of the stimulus, and the response of the individuals. Opposite to most works, the measurements do not provide full information of what is going on in the large facility. As a consequence, algorithms and software to fill in the gaps consistently are needed. The paper describes the current state of the simulations and the difficulties in modeling with precision the results in a case study.

1 Introduction

If a large facility is expected to host an embedded system, as in a Internet of Things or Ambient Intelligence scenario, the definition of such system and its early validation is largely missing in the literature. Given a particular building or large space, a first question is whether the goal to make the visitors of the facility show a new behavior or to alter an existing one. This cannot be done in the traditional way, by consulting a few stakeholders. Interviewing and surveying the occupants of the environment seems more adequate.

How this information is captured and reused later on, it is still an issue. Documenting is out of question. However, the format of the documentation can be subject of discussion. Also, how this documentation is used and accessed in order to ensure the problem is completely understood.

The hypothesis of this work is twofold. First, that social sciences methods can be used in order to capture better the behavior of the humans inhabiting the large facility and what stimulus can trigger this behavioral change. Second, once there is a preliminary solution to the creation of new behaviors or modification of existing ones, the enactment of such behaviors can be something better documented if computer simulations are used.

Besides, documenting something as dynamic as the behavior of big groups of humans through simulations allows too to experiment with the expected effect of different stimulation procedures. This way, responsible of large facilities can have tools that enable them to discover how they want the facility to be altered before the actual smart environment is even built. In order for the simulation to be realistic, the simulation has to reproduce the interplay among users and between users and the environment. The later includes too the devices that are supposed to make the users behave differently. These devices do provide stimulus previously validated by experts as candidates to produce the desired kind of effects.

The paper is structured as follows. First, section 2 analyses if a particular behavior alteration/induction is really possible. The contribution of social sciences to the requirements gathering is made in 3. A guideline is proposed that combines interviews, surveys, but also field studies, as well as analyses of the captured information. Examples of the analysis phase is made in section 4, where a domain example is introduced. The design of simulations that aid to create the smart environment capable of enacting the new behaviors is made in section 5. The related work is introduced in section 6. Conclusions are presented into section 7.

2 Stimulus for Behavior Alteration

Literature shows that humans are sensible to external stimulus in subtle ways and that our behavior can be altered. The extent of the alteration may depend on the individual. Some may react notably while others hardly react. Nevertheless, the average person ought to notice this. The nature of the stimulus matters too. In certain conditions, such as evacuations, humans pay more attention to other humans rather than other artificial elements, such as banners.

Humans have sensibility towards the behavior of other humans. If an individual finds a group along the way, depending on its size, will either stop and look what is happening and stay or keep walking [11]. The larger the group, the greater the effect. This is explained as a mirroring behavior effect. If sufficient people stare at an arbitrary point, a passerby individual will unconsciously look at the same place [5]. Gaze copying happens mainly within 2 meters range and the response depends on the physical layout of the environment, the social context, and the sex of the individual.

When the stimulus come from artificial sources, the results are still promising. Sound and images can affect the behavior of pedestrians. Beyer et al. [2] introduce an experiment where an interactive large banner display affects the audience. Through visual stimulus, authors manage to attract approaching pedestrians and distribute them along the display. Miller [12] shows that noise can affect people's performance. A sleepy person may be aroused by noise, but it has also negative effects, like affecting the performance of complicated tasks, affect negatively the mood and disturb relaxation. Negative effects could be used to influence pedestrians. In this paper, it is assumed that, since it can annoy people, this could be used to clear out areas or to reduce the pedestrian traffic around some places where the noise comes from.

The context matters too. Foster [4] analyzes different domains in

[1] Universidad Complutense de Madrid, email:jjgomez@ucm.es

order to promote healthy habits. Each context is different. A shopping center and railway station involve different behaviors on behalf the population.

Also, sensibility towards stimulus changes depending on the context. In an airport, passengers pay special attention to information panels. A change in one panel may trigger movements of user groups, such as changing one boarding gate ten minutes before the boarding starts. Fun parks also influence the behavior of their visitors through information panels that tell expected waiting time for each attraction.

3 Guidelines for Developing Ambient Intelligence in Large Facilities

The system to be developed aims to interact with several inhabitants of a large space. These inhabitants may be transient ones or permanent inhabitants of the considered space. It is assumed that the people in this physical environment can be either a management staff, in charge of the facilities and aiding to the occupants of the facilities to fulfill the identified system goals; and the visitors, who are the clients of the facility. In general, the staff interacts with the visitors in order for helping them perform certain activities. In the physical space, it is assumed the staff is expected to modify the behavior of the clients in a way that clients perceive an benefit.

To identify what behavior modifications are possible and how to best convince inhabitants of the facility to commit to such behavior, a guideline affecting particular system development is introduced following:

- Analysis phase. The facility to be analyses is assumed to fulfill one or many purposes. The staff is expected to alter the behavior of the inhabitants in order to achieve certain behavior. This behavior is compatible with the purposes of the facility, and it is supposed to be regulated or activated through some environmental devices. There is a review of the meaningful behaviors, according to the literature, on the expected behaviors (domain or non-domain specific) for the chosen facility. A selection of stimulus is made based on the available resources (the budget of the modification, for instance). Also, field studies have to be planned to know more of the visitors and also to evaluate the effect of those stimulus over time. Effect of each stimulus is measured and annotated so that it can be reproduced later on. Each stimulus is expected to have a duration and an intensity.
- Design phase. The different stimulus and the expected reaction is modeled into a simulation that serves as reference. The simulation includes the physical space, the inhabitants of the space, the expected behaviors of those inhabitants according to the field studies, and the simulated devices that are going to provide the stimulus. The measurements made in the field studies are interpolated to guess the overall behavior of the whole population. Accordingly, the expected behavior is studied, taking into account the reaction to the stimulus. As a result, an expected orchestration of the stimulus is obtained.
- Deployment phase. The synchronization of the stimulus is deployed into real devices already working in the facility. The simulation is expected to have identified several critical observation points whose measurements indicate if the stimulus is working or not.

The role of human scientists is important in the development of this kind of systems. In this guideline, it is assumed that human scientists involve themselves mainly into the analysis stages. However, their collaboration is needed too along the design stages. Human sciences scientists, such as psychologists or social scientists, provide insight into the behavior of the users beyond common wisdom. Hence, they are needed in order to properly design the field experiments, to study the results, and to assess the validity of the simulations.

It is assumed that there is a simulation parameterization whose behavior is close to the observed behavior. Such simulation should be possible because the behavior of users into installations is not heterogeneous and tend to fit into standard behavior patterns, that we associate with activities of the daily living typical of the installation. The definition and parameterization of the simulation is considered following.

3.1 Specifying the crowd simulation

An important part of the simulation is the description of the physical space inhabitants, which is called here population description. For this goal, it is necessary to identify a set of possible actor behaviors, an enumeration of the number of instances of these behaviors, and a timestamp of when the behavior&actor instantiation happens.

Actor instances are created along the simulation and destructed when the behavior of the character finishes. It is assumed the designer determines a suitable place where this destruction happens. After all,actors cannot vanish from the scenario just anywhere. These actors instantiate a particular set of behaviors with particular parameters. The different parametrization determines individual variations of the behavior.

It is assumed that actors can belong to two distinguished groups: those responsible of operating the facilities and those visiting the facilities. The first are expected to perform different activities oriented towards coordinating the behavior of the second group within the facility. The second group are executing activities of the daily living related to the main purpose of the facility. It is not expected that one actor belonging to one group suddenly becomes an actor belonging to another. Even thought there maybe cases where this role switch makes sense, it is not considered in this paper. Within each group, there can be further decomposition of responsibilities, but it depends on each particular domain.

An actor behavior specification consists of a sequence of parameterized activities of the daily living plus an initial location. The amount of instances of each actor behavior specification determines the composition of the population.

Actors are not allowed to alter their behavior and they constantly perform the same sequence. The sequence terminates with the destruction of the character. This enforces designers to define precisely what actors are expected to do since their creation until completing their part in the simulation.

4 The case study

The crowd simulation has been applied to a scenario situated in two faculties. The goal is to alter the behavior of the inhabitants in order to make them choose an activity that requires additional effort over another activity that does not. The behavior to be altered is using the elevator, which ought to be replaced by using the staircases. The experiment is run into two different faculties, the Computer Science Faculty and the Political and sociological Sciences faculty.

The application of the methodology starts with a field study structured as follows. First, the managers of both faculties are interviewed to know more of the daily problems they have to face. This provides an insight on the students and other staff using the facility. It also helps to identify possible incompatibilities between the planned stimulus and the current activities. The chosen stimulus are:

- Human-to-human interaction. A person playing the role *facility operator* interacts with another playing the role *visitor* and tries to suggest the use of staircases is better.
- Banners. Banners are proposed containing information of interest to the visitor and that may aid in suggesting an alternative behavior. It is important to notice that there ought to be an evident profit for the visitor, otherwise the behavior modification will not occur. In this case, the banner is presented at figure 1. It suggests the visitor will gain health improvements, will arrive faster to the destination, and will save electricity. These facts, specially the savings in time during travels, has been proven to be true.
- Multimedia. A video shows a dramatization of a person that uses the elevator for everything even though can perfectly walk. The video is shown through a short distance beamer sufficiently visible and the equivalent of a 55' screen. The short distance beamer is projecting vertically and permits a less disturbing installation. The projection is made close enough to the elevator. Due to safety concerns, it was not allocated right next to the elevator.

Figure 1. Banner for motivating users to use the staircase. It written in spanish. The main title says *stair climbing and avoiding elevators* at the top. The alleged reason are *1. improving your health*, *2. You will get faster to your destination*, and *3. you will save energy*

A plan for measuring the effects of these stimulus was made. The plan consisted on a five week schedule. The first week (week A) there was no stimulus and it was used to collect a base line of staircase/elevator traffic stats; during the second week (week B) the banner stimulus was introduced; along the third week (week C1) the videos were added; and in the fourth week (week C2), the human-to-human interaction. Then, there were some days of no stimulus to let users decide whether they want to keep the new behavior or get back to the old one. Therefore, the fifth week (week D) is dedicated to measure the resilience of the stimulus.

Collected data was a set of pedestrian traffic into strategic checkpoints of the faculties. Measurements indicate whether visitors come or go, and whether they are using the elevator or the staircases. An account of persons per minute is provided. The resulting influence of the stimulus along the field experiment stages is included in table 1. The number of people arriving through the elevator remains mostly the same along stages. However, the number of people choosing not to use the elevator is reduced up to 4 points in phase C2. This is a variation of 13,65% over the original use of the elevator. The results are not shocking, but it should be taken into account that each stimulus lasted for one week, and not months.

Table 1. Variation of the traffic in elevators into two faculties

% use elevators	A	B	C1	C2	D
Total	23,1	21,9	21,4	20,3	22,2
Departures	29,3	28,2	26,2	25,3	26,4
Arrivals	14,4	14,4	15,2	14,0	16,2
#total=	9730	9797	9459	9165	9088
#departures=	5688	5371	5335	5109	5345
#arrivals=	4042	4426	4124	4056	3743

With the obtained traffic data, a simulation is arranged so as to reproduce the observed behaviors.

5 Reproducing the experiments

The result of the experiments is being transferred to computer simulations, to identify complexities and capture individual behavior as precisely as possible.

In the simulation, all actors are belonging to the visitor role. Their actions consists in entering the building, visiting a previously unknown number of rooms, and exiting the building. Hence, a parameterization of the problem includes an account of the rooms each actor visits and how long they stay there.

The computer simulation has to capture emergent behaviors. Rather than organizing dynamically the behavior of a whole population and letting a central node orchestrating everything, the approach is multi-agent based one, where individual behaviors of characters is coded. The individual behaviors is explained along the next paragraphs, but the goal is to attain the same, or close, traffic data to those obtained from the different experiments. Since the data from each phase is available, the simulation ought to capture the effect of the stimulus over the visitors. Henceforth, if the stimulus is a banner and the measured effect is a 25% variation, then the simulated traffic ought to show such change as well.

The total aggregation of the traffic ought to provide with numbers similar to those of table 1. Achieving this traffic data while coding individual behaviors is a hard task because of two reasons. First, there are several elements whose interplay affects the outcome of the simulation. Actors interact among themselves and with the environment, specially elevators and the physical layout of the environment, a building with several floors. Second, the gathered information is partial, since only a few pedestrian traffic check points were established in the field study from section 4. This means there were not cameras recording the full activity. As a consequence, there may be many populations of simulated actors whose movement along the facility matches the obtained measurements in the field experiment of section 4

The problem has been studied in [13] and the provisional solution is a greedy algorithm that produces a population of actors whose behavior matches to some extent the expected behavior of the whole population. A first attempt is presented in figures 2 and 3.

The behavior of each individual can be summarized as follows. Each character has a navigation path from the starting point to a particular location determined by the greedy algorithm [13] and going through some intermediate points that are part of the parameterization. Intermediate points may correspond to specific rooms the characters may or may not visit. Along the navigation, the character may find obstacles. Fixed obstacles are already avoided by the navigation algorithm. Mobile obstacles are avoided through maneuvers around the expected collision points. Afterwards, the navigation path is rechecked and resumed.

Figure 2 shows a part of the 3D simulation created with the greedy algorithm. In the simulation, to compare the simulated vs the real scenario, the simulation assumes there is a device in the area capable of counting people as they cross the section corresponding to the checkpoint. The counting is compared against the real measured traffic in the bottom part of the figure.

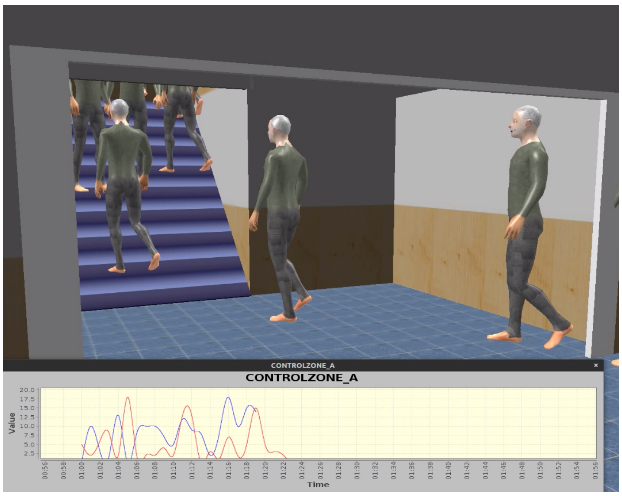


Figure 2. Simulation of pedestrian traffic along checkpoints and simulated traffic data gathering

There are many possible populations of actors whose movements have the same effect in terms of traffic through the checkpoints, at least, in theory. The greedy algorithm from [13] achieves the performance shown in figure 4. This figure focuses on the traffic data and compares the simulated to the real measured traffic along the experiment. The considered time window is different from 2. In figure in 3, there is a small variation in the obtained simulated traffic measurements. The main reason for such variations is the interplay of actors along their paths, which is not taken into account. Collisions and bottlenecks happen too, and they cause a different transit time. This is a positive sign the simulation is more complex than the simplified model the greedy algorithm uses.

Another source of complexity is the modeling of elevators, as shown in figure 3. The characters that occupy the interior of the elevator must coordinate to exit into each floor. Problems happen when one character situated at the back of the elevator wants to get out, but no one of those situated at the front wants to move. Again, this alters the traffic. To prevent this, the simulated actors have to be aware of what is the right use of an elevator.

The problem becomes more complex when the activities of the daily living is added to the considerations. The protocol of lectures in a classroom is simple: students come to the classroom; they sit down; a teacher comes and starts the lecture; more students may come during the lecture; the lecture finishes and then all, or a few, students leave the room. The uncertainty in the process, such as teachers finishing sooner or later, makes the evacuation of students from classrooms more smoother than it should be if all teachers coordinated

Figure 3. Elevator carrying people from one floor to the other

precisely the lectures to finish exactly at the same time. Such individual behaviors are relevant to be modeled too.

Also, the simulations may lead to inconsistent results. For instance, most of the resulting populations according to the algorithm [13] have in common that upper floors are mostly empty. Upper floors only have offices and not classrooms, what would explain this result. Then, it may be subject of discussion if a better occupation of the building was possible. If the space allocated in upper floors is the same as lower floors while the traffic is much lower, perhaps a higher number of offices could be arranged without compromising an eventual evacuation of the building.

Capturing complexity at the simulation allows to realize the software-in-the-loop approach. It is a goal of the project to include sensor/actuator devices in the simulation so that a designer can explore the effect of the stimulus of those devices on the population. The simulated devices would be operated using control software that was close to the simulated one. This approach has been essayed in [6] for gesture recognition devices design using 3D simulated environments generated with the AIDE environment [3].

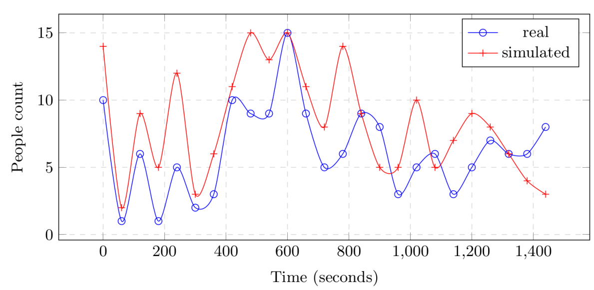


Figure 4. Measured traffic in the simulation compared to observed real traffic using a different time window from figure 2

6 Related work

There are works dealing with the design of smart systems, but they do not frequently consider human sciences and stimulus to plan the kind of system which is needed and what performance it will have. Harrison [7] claims the analysis of mutual and incidental user interaction has not been accounted and proceeds to apply fluid flow analysis to understand it. This kind of analysis is necessary, but, it does not replace a more conventional study and cannot assume a 100% response of the individuals every time. Other works focus on the devices expected to provide the stimulus at small scale, such as [14]. Thought authors stress the involvement of human scientists too, the behavior of people in small spaces cannot be compared to that of large spaces.

There are precedents too in reproducing observed data as simulations. In [8], video recordings were used to reproduced later on a crowd simulation of simulated actors. Behavior of the individuals were obtained from a multiple checkpoint observation that allowed. The project introduced in this paper, however,assumes incomplete information about activities and traffic. The less information is used, the less expensive a real installation would be. Following the same paradigm, Lerner et.al [9] propose the creation of an example database for evaluating simulated crowds based on videos of real crowds. Bera. et.al [1] also developed a behavior-learning algorithm for data-driven crowd simulation, capable of learn from mixed videos. Zong et.al [15] developed a framework for generating crowds for matching the patterns observed on video data,taking into consideration the behavior both at the microscopic level as at the macroscopic level. Finally, Yi Li et.al [10] developed a technique for populating large environments with virtual characters, cloning the trajectories of extracted crowd motion of real data sets to a large number of entities.

7 Conclusions

The paper has introduced a guideline with recommendations inspired into human sciences and the realization of field experiments. Such experiments are necessary to fine tune devices that aim to influence the behavior of the facility inhabitants. With this information, computer simulations have been created. These simulations reproduce the observed behavior and can be used to experiment with different setups and stimulus until a suitable combination is found. The next step is to devise the control software capable of synchronizing the stimulus over the population and then deploying such software to real devices having that capability.

Simulations can be used also to reason about what is happening inside the facility. In particular, the produced simulations show there are concerns in how the building is actually used. The identified traffic, and assuming most of the people that goes through a staircase/elevator do it only once, permits to infer that the upper floors of the building are less occupied than expected.

ACKNOWLEDGEMENTS

We acknowledge support from the project "Collaborative Ambient Assisted Living Design (ColoSAAL)" (TIN2014-57028-R) funded by Spanish Ministry for Economy and Competitiveness; and MOSI-AGIL-CM (S2013/ICE-3019) co-funded by Madrid Government, EU Structural Funds FSE, and FEDER.

REFERENCES

[1] Aniket Bera, Sujeong Kim, and Dinesh Manocha, 'Efficient trajectory extraction and parameter learning for data-driven crowd simulation', in *Proceedings of the 41st Graphics Interface Conference*, pp. 65–72. Canadian Information Processing Society, (2015).

[2] Gilbert Beyer, Vincent Binder, Nina Jäger, and Andreas Butz, 'The puppeteer display: attracting and actively shaping the audience with an interactive public banner display', in *Proceedings of the 2014 conference on Designing interactive systems*, pp. 935–944. ACM, (2014).

[3] Pablo Campillo-Sanchez and Jorge J. Gómez-Sanz, 'Agent based simulation for creating ambient assisted living solutions', in *Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection - 12th International Conference, PAAMS 2014, Salamanca, Spain, June 4-6, 2014. Proceedings*, pp. 319–322, (2014).

[4] Charles Foster and Melvyn Hillsdon, 'Changing the environment to promote health-enhancing physical activity', *Journal of sports sciences*, **22**(8), 755–769, (2004).

[5] Andrew C Gallup, Joseph J Hale, David JT Sumpter, Simon Garnier, Alex Kacelnik, John R Krebs, and Iain D Couzin, 'Visual attention and the acquisition of information in human crowds', *Proceedings of the National Academy of Sciences*, **109**(19), 7245–7250, (2012).

[6] Jorge J Gómez-Sanz, Marlon Cardenas, Rafael Pax, and Pablo Campillo, 'Building prototypes through 3d simulations', *Advances in Practical Applications of Scalable Multi-agent Systems. The PAAMS Collection*, 299, (2016).

[7] Michael D Harrison, Mieke Massink, and Diego Latella, 'Engineering crowd interaction within smart environments', in *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems*, pp. 117–122. ACM, (2009).

[8] Kang Hoon Lee, Myung Geol Choi, Qyoun Hong, and Jehee Lee, 'Group behavior from video: A data-driven approach to crowd simulation', in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '07, pp. 109–118, Aire-la-Ville, Switzerland, Switzerland, (2007). Eurographics Association.

[9] Alon Lerner, Yiorgos Chrysanthou, Ariel Shamir, and Daniel Cohen-Or, 'Data driven evaluation of crowds', in *Motion in Games*, 75–83, Springer, (2009).

[10] Yi Li, Marc Christie, Orianne Siret, Richard Kulpa, and Julien Pettré, 'Cloning crowd motions', in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 201–210. Eurographics Association, (2012).

[11] Stanley Milgram, Leonard Bickman, and Lawrence Berkowitz, 'Note on the drawing power of crowds of different size.', *Journal of personality and social psychology*, **13**(2), 79, (1969).

[12] James D Miller, 'Effects of noise on people', *The Journal of the Acoustical Society of America*, **56**(3), 729–764, (1974).

[13] Rafael Pax and Jorge J. Gómez-Sanz, 'A greedy algorithm for reproducing crowds', in *Trends in Practical Applications of Scalable Multi-Agent Systems, the PAAMS Collection, 14th International Conference, PAAMS 2016, Sevilla, Spain, June 1-3, 2016, Special Sessions.*, pp. 287–296, (2016).

[14] Norbert A Streitz, Carsten Röcker, Thorsten Prante, Daniel Van Alphen, Richard Stenzel, and Carsten Magerkurth, 'Designing smart artifacts for smart environments', *Computer*, **38**(3), 41–49, (2005).

[15] Jinghui Zhong, Wentong Cai, Linbo Luo, and Haiyan Yin, 'Learning behavior patterns from video: A data-driven framework for agent-based crowd modeling', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 801–809. International Foundation for Autonomous Agents and Multiagent Systems, (2015).

Evacuation Centrality: Agile Evacuation Routing in Unpredictable Hazardous Conditions

Marin Lujak [1] and **Stefano Giordani** [2]

Abstract. In this paper, we study the agility of evacuation routes in relation to dynamically changing unpredictable hazardous conditions in smart space networks. Infrastructure safety conditions may unpredictably change through time. Due to unpredictability, evacuees' safety can get jeopardized at any point of the evacuation route. Thus, it is not sufficient only to find the shortest evacuation route considering present safety conditions, but we should also consider other relevant characteristics that make the evacuation route sufficiently safe through time. With this aim, we propose a new node importance metric called evacuation centrality, inspired by betweenness centrality. The node evacuation centrality is a parameter that represents the importance of the node for evacuation considering the availability of alternative efficient routes from that node towards safe exits. Given a set of evacuees' positions and safe exits, we mathematically formulate the problem of finding shortest agile evacuation routes, where by agile route we mean the ability to efficiently and safely reroute from intermediate nodes in real time in case of unpredictable safety drops through maximizing the value of the evacuation centrality of the route's intermediate nodes. In addition, we propose a solution method for that problem and discuss its capability to react to the changes in safety circumstances along recommended routes.

1 Introduction

Emergencies or disasters occur unexpectedly and disrupt human activities and cause physical and/or environmental damage. They can strike anyone, anytime, and anywhere. Emergencies may be natural or manmade, small scale, as e.g., in a building due to an explosion or fire, or large scale, as e.g., in a city or a region because of a radiological accident, bombardment or dangerous weather system.

Emergency evacuation is the immediate and urgent movement of people away from the threat or actual occurrence of a hazard. In this situation, evacuees should be able to evacuate safely, rapidly, seamlessly, and in a coordinated way through an evacuation space while avoiding hazardous conditions.

Traditional evacuation approaches are based on the following procedure. In the case of an imminent or ongoing danger, evacuation is organized by a trained personnel that coordinates the evacuation process on critical evacuation points. In larger watercrafts, these are dedicated areas where evacuees must assemble in case of emergency (assembly stations). Each evacuee should reach his/her assembly station or exit by following the escape route shown on a plan which is usually positioned on a limited number of positions in the building, and the signs in the corridors and stairs that are attached on the floor or walls. If the primary escape route is blocked, there is usually a second escape route, which is marked on an evacuation plan. Moreover, if visibility is limited due to smoke, evacuees should follow the emergency lights situated close to floor level.

The routes in evacuation plans are predefined and static. In the case there is a blockage of these routes, evacuees are provided with no alternatives. Moreover, since the evacuation plans are randomly present in the evacuation infrastructure, evacuees might not get the necessary evacuation information. This may have hazardous consequences and may result in panic.

The concepts of rapidness and seamlessness, which are necessary in evacuation, are closely related to the concept of agility. Oxford dictionary (2016) describes the term agile as "the ability to move quickly and easily" and "the ability to think and understand quickly". It is a well known concept in many areas, such as, e.g., manufacturing, software development, and business organization, see, e.g., [2, 11, 12]. In terms of outcomes, agility is a means of a system to swiftly and easily handle continuous and unanticipated change by adapting its initial stable configuration and to effectively manage unpredictable external and internal changes, e.g., [11, 12]. Based on this conceptualization and paradigms of agile manufacturing and agile business systems, in this work we propose the concept of agility in evacuation routes and route recommendation systems.

Agility of an evacuation route assures to an evacuee a high real-time reactivity to safety changes possibly occurring along the route. It requires the ability to reroute from intermediate nodes to alternative routes towards safe exits. Agile route recommendation systems, hence, should be capable to run in real time in the cycle sense-analyze-decide-act. To achieve it, we need complete, accurate and up-to-the-minute situational awareness along the route. While in the open spaces, GPS and e.g., 3G and 4G communication can be used, in inner spaces, this requirement is filled by the interaction of ambient intelligence and smartphone technologies. Hence, an agile evacuation route recommendation system should respond quickly in inner and open spaces to sudden changes in evacuation safety conditions caused by a hazard, crowdedness or any other type of requirement or disruption. To the best of our knowledge, the literature on such route recommendation systems is very scarce (Section 2).

We can model evacuation agility of a route in terms of the characteristics of its intemediate nodes. For this scope, we examine relevant centrality measures in (Section 3) and in Section 4 propose new node importance metrics called evacuation centrality and evacuation betweenness centrality, both inspired by (node) betweenness centrality. The evacuation centrality is a parameter that represents the importance of a node for evacuation considering the availability of alternative efficient routes from that node towards safe exits.

[1] CETINIA, University King Juan Carlos, Madrid, Spain, Email: firstname.lastname@urjc.es

[2] Dept. of Enterprise Engineering, University of Rome "Tor Vergata", Rome, Italy, Email: stefano.giordani@uniroma2.it

Given a set of evacuees' positions and safe exits, in Section 5, we mathematically formulate the problem of finding shortest agile evacuation routes, where by agile we mean the ability to efficiently and safely reroute from intermediate nodes in real time in case of unpredictable safety drops. We achieve this through finding a route with a maximum value of the evacuation centrality of the route's intermediate nodes.

In Section 6, we propose a multi-agent system that finds the shortest agile evacuation routes. A possible rereouting in the proposed method is performed in regular time intervals by the algorithm's execution considering the evacuees' momentary positions. The evacuees are given only a necessary information of the next part of the route to pass, without saturating them with the unnecessary route information that may change through time. We conclude the paper in Section 7.

2 Related work

Building evacuation has been studied over the last decades from different perspectives such as, e.g., evacuees' behaviors, traffic control strategies, sheltering site selection, and route finding for displacement. For example, Pursals and Garzon in [10] considered the building evacuation problem and developed a model for selecting the proper routes for movement of people in a building during an emergency situation. Abdelghany et al. in [1] present a simulation - optimization modeling framework for the evacuation of large - scale pedestrian facilities with multiple exit gates. The framework integrates a genetic algorithm (GA) and a microscopic pedestrian simulation - assignment model. The GA searches for the optimal evacuation plan, while the simulation model guides the search through evaluating the quality of the generated evacuation plans. Evacuees are assumed to receive evacuation instructions in terms of the optimal exit gates and evacuation start times. The framework is applied to develop an optimal evacuation plan for a hypothetical crowded exhibition hall. A mixed-integer programming solver is used to derive routing plans for sample networks.

Conventional emergency evacuation plans often assign evacuees to fixed routes or destinations based mainly on geographic proximity. Such approaches can be inefficient if the roads are congested, blocked, or otherwise dangerous because of the emergency. Han and Yuan proposed in [4] the concept of *most desirable destination* for evacuees. This concept recognizes that municipalities responsible for large-scale evacuation have routinely assigned evacuees to routes and destinations based on limited experience and intuition rather than methodical optimization processes. Even with the implementation of dynamic traffic assignment, models that are based on fixed origin-destination tables are inefficient when a destination becomes difficult (or impossible) to access due to congestion or blockage. In [4], options that allow evacuees flexibility in selecting their exit routes and destinations are explored.

Destination assignment and route assignment to enable optimal evacuation operations are interrelated. To optimize the routing problem, one has to know the destinations; to optimize the destination assignment, one has to know the minimal travel time, and hence route assignment to all destinations.

To address the inherent complexity of the problem, Han et al. in [4] devised a framework for simultaneously optimizing evacuation-traffic destination and route assignment. Based on this framework, we can determine the optimal evacuation-destination and route assignment using a one-step optimization procedure.

In [8], we propose a pedestrian route recommender system for smart spaces in steady state conditions that recommends the safest routes to pedestrians and simultaneously optimizes conflicting objectives of finding the social optimum and minimizing individual path travel times while considering people flow and fairness, similarly to [6, 7]. Moreover, the system considers the influence of stress on human reactions to the recommended routes and iteratively ponders user response to the suggested routes influenced by stress-related irrational behaviors until system acceptable routes are found. However, in the case of a sudden safety drop on a part of the route, it might not be able to guarantee a safe evacuation of the safety jeopardized areas since in the route recommendation, it does not consider the unpredictability of safety conditions. In this case, it might thus result in evacuees' fatalities. Moreover, in [9], we consider the influence of affiliate ties among evacuees and their interaction with self-concerned individuals and model self-concerned and social group behavior via individual and team reasoning. The recommended evacuation routes take in consideration the affiliate ties to guarantee evacuee's compliance with the routes.

The aforementioned papers assume steady evacuation state and do not treat the safety dynamics in transitory evacuation safety conditions. Therefore, in this paper, we concentrate on evacuation routing in highly unpredictable dynamically changeable hazardous evacuation safety conditions. In this case, it is important to find the shortest safe routes for all evacuees considering other relevant characteristics that make the evacuation route sufficiently safe through time.

3 Centrality measures for evacuation routing

Generally, centrality measures indicate the most important nodes of a network. Some of the measures relevant to the computation of routes are node degree, eigenvector, and betweenness centrality. In the following, we first define a model of the area from which people need to evacuate and based on the introduced terminology and concepts, describe the relevance of these evacuation measures to evacuation routing and identify their flaws in this respect.

3.1 Evacuation network model

We represent a smart space evacuation network (building and/or urban district) by a directed graph $G = (N, A)$, where N is a set of n nodes representing rooms, offices, halls, and in general, any portion of a relatively small portion of space within a building or other structure separated by walls or partitions from other parts. In the case of larger spaces that can host a larger number of people, for simplicity, the same are divided into sections represented by nodes completely connected by arcs $a \in A$, where A is the set of m arcs $a = (i, j)$, $i, j \in N$ and $i \neq j$, representing corridors, walkways, doors, gateways, and passages connecting nodes i and j. To simplify the notation, we assume that there is at most one arc in each direction between any pair of nodes. Each arc $(i, j) \in A$ has an associated cost c_{ij}, which in our case is its traversal time.

We opt for a directed graph representation of the evacuation infrastructure since in the case of bi-directional corridors, roads, and passages, we can easily reduce the undirected graph to the directed graph by connecting two adjacent nodes with an arc in each direction, while in the case of unidirectional roads, representing the direction by an (undirected) edge is not possible.

Let $O \subseteq N$ and $D \subseteq N$ be the set of all evacuees' origins and safe exit destinations, respectively. We assume that there are n_O evacuees' origin nodes $o \in O$ disjoint from n_D safe exit destination nodes $d \in D$, where $n_O + n_D \leq n$.

For the definition of origin-destination demand, we introduce fictitious sink node $\hat{d} \in N$ that is adjacent to all the destination nodes (safe exits) by fictitious (dummy) arcs. In this way, we assume that graph G includes (together with actual nodes) also fictitious node $\hat{d}$ and its incoming dummy arcs. Then, let $w \in W$ be a generic evacuation request from node $o \in O$ to fictitious sink node $\hat{d}$, where W is the set of all evacuation requests.

Moreover, let R be a vector of cardinality n_O representing evacuation requests from origins O towards fictitious safe exit $\hat{d}$, where $R_{o\hat{d}} = R_w$ entry indicates the demand of evacuees in unit time period who request to leave origin node $o \in O$ to go to any of the safe exits $d \in D$ and, hence, to fictitious destination $\hat{d}$.

Then a simple path p is a finite sequence of adjacent distinct nodes connected by a sequence of arcs, each connecting two adjacent (different) nodes. Its total cost c_p is composed of the costs of the connecting arcs c_{ij}, where $(i, j) \in p$.

3.2 Aggregation of evacuation requests on arcs

While the evacuation requests $R_{o\hat{d}} = R_w$ are clearly defined for each node $o \in O$, we need to disambiguate the evacuation requests in the case there are evacuees present in the spaces represented by arcs $a \in A$.

For simplicity, there are two possible modelling approaches for this case: i) they are added to the closest node incident with that arc or ii) a node is added to the arc at the actual position of the evacuee(s), thus representing evacuee(s) request(s) at their actual position.

In the first case, we can use the Voronoi diagram principle where all the evacuees closest to a node are added to that node's evacuation request.

Since in the second case (where a node represents an evacuation request of one or more geographically close evacuees), a new, mobile type of nodes is introduced in graph G, the structure of the same is changing dynamically through time depending on the positions of each and every evacuee in the evacuation network. Moreover, in this case, the number of evacuee requests n_O is not related with the cardinality of the node set N, but with the number of evacuees in the building n_E, with $n_O \leq n_E$. The exact number of evacuation request clusters represented by evacuation nodes defined in this way is determined by the vicinity and geographical distribution of the evacuees. Geographically close evacuees may be considered by a single evacuation request node if the evacuees are within a limited travel time away from one another.

This modelling option results in the presence of static nodes N and arcs A representing the structure of the evacuation space, and the introduction of mobile evacuation request nodes O representing the evacuees' requests.

We might use different clustering techniques for the definition of the evacuees' groups and their distribution based on the similarity factor that should be dynamically adaptable to each case. However, due to the introduced additional complexity of this approach and for simplicity, we opt for the first approach where the demand is added to the closest incident node of an arc. Thus, the structure of evacuation requests changes dynamically through time and is created based on Voronoi diagram.

3.3 Degree centrality

The degree centrality $C_d(i)$ of node $i \in N$ is the number of arcs incident to the node. In directed graphs, we can either use the in-degree, the out-degree, or their combination as the degree centrality value. When we combine in-degrees and out-degrees, we are basically ignoring arc directions.

In general, nodes with a higher degree centrality tend to be used by more paths. However, connections of a node with the neighboring nodes that are a part of the shortest paths to safe exits are more important than others. Since the degree centrality does not guarantee the connectedness of a node to safe exits, it cannot be used in the computation of efficient evacuation routes.

3.4 Eigenvector centrality

A step forward the evacuation route's efficiency guarantee is the eigenvector centrality of a node, which depends both on the number and the *importance* of its adjacent nodes. In general, adjacency of a node to nodes that are themselves adjacent to more *important* nodes will give a node more *importance*.

While node degree centrality counts walks of (geodesic) unitary length from a node, the eigenvalue centrality takes into consideration walks of length infinity. It is the expected number of visits to a node $i \in N$ of an infinite random walk over graph $G = (N, A)$. It can only be calculated for connected undirected graphs and strongly connected digraphs. More formally, if we let $\mathbf{Ad} = (a_{i,j})$ be the adjacency matrix of graph $G = (N, A)$, eigenvector centrality $C_e(i)$ of node $i \in N$ is given by:

$$C_e(i) = \frac{1}{\lambda} \sum_{j \in N \setminus \{i\}} a_{i,j} C_e(j), \tag{1}$$

where $\lambda \neq 0$ is a constant. In matrix form, we have $\lambda \mathbf{C_e} = \mathbf{Ad} \cdot \mathbf{C_e}$. Hence the centrality vector $\mathbf{C_e}$ is the eigenvector of the adjacency matrix $\mathbf{Ad}$ associated with the eigenvalue λ. If we choose λ as the largest eigenvalue in absolute value of matrix $\mathbf{Ad}$, then as a result of Perron-Frobenius theorem, if matrix $\mathbf{Ad}$ is irreducible (i.e., the graph is (strongly) connected), then the eigenvector solution $\mathbf{C_e}$ is both unique and positive.

The nodes with a high eigenvector centrality values, then, will be traversed by more paths. Moreover, nodes with a high eigenvector centrality are network hubs and their presence is crucial in maintaining the paths among all network nodes. However, a high centrality value of a node does not guarantee the existence of efficient paths from that node towards safe exits. Additionally, a high eigenvector centrality value of a node might be a root to panic and a related herding problem [8] in the case of high people flows traversing the node. Therefore, eigenvector centrality does not characterize sufficiently the importance of the nodes for evacuation. Since we want to guarantee the efficiency of evacuation towards a limited set of safe exit nodes, as such, it can not be directly used as a parameter for evacuation optimization.

3.5 Betweenness centrality

Betweenness centrality is a concept that is closer to the efficiency of evacuation routes and is a departure point in our proposition of the evacuation centrality metrics. It is defined as the fraction of shortest geodesic paths between nodes different than $i \in N$ that i is a part of:

$$\sum_{o \in N} \sum_{d \in N} \frac{\sigma_{od}(i)}{\sigma_{od}}, \forall i \neq s \neq t \in N, \tag{2}$$

where $\sigma_{od}(i)$ is the number of shortest geodesic paths (the paths with the minimum number of arcs) between o and d and $i \in N$ is an intermediate node of the path. Moreover, σ_{od} is the total number of shortest geodesic paths between o and d.

Betweenness centrality is, therefore, an indicator of the frequency a node serves as the "bridge" on the shortest geodesic paths connecting any two other nodes. That is, we find the shortest geodesic path (or paths) between every pair of nodes, and calculate the fraction of these paths that node i lies on. If we imagine crowd flowing between nodes in the network and always taking the shortest possible geodesic path, then betweenness centrality measures the fraction of that crowd that will flow through i on its way to wherever it is going.

Even though this measure might be relevant to the use cases with constant arcs' costs, the issues with the usage of betweenness centrality in evacuation are related with the definition of distance and the origin-destination pairs. In particular, we are concerned with the shortest evacuation time and not the shortest geodesic distance. Moreover, we are not interested in all origin destination pairs, but only in a limited subset of evacuees' origins O and safe exits D. In the following, we deal with these two issues.

4 Proposed centrality measures

Based on the analysis of the centrality measures related with evacuation routing, in the following we propose two new centrality measures for evacuation routing: evacuation betweenness centrality, and evacuation centrality.

4.1 Evacuation betweenness centrality

If we substitute the geodesic distance with a path cost $c_p \geq 0$, most probably there will be only one shortest path for every pair of nodes. This is why here we present a modification of betweenness centrality that considers k_{od} distinct shortest paths for each (o, d) pair, with $o \in O$ and $d \in D$. We call this measure evacuation betweenness centrality. Here, k_{od} is the number of distinct shortest paths of cost (travel time) at most, e.g., $\gamma \cdot c^{od}_{\min}$, where $\gamma \geq 1$ is a maximum evacuation route cost tolerance factor and $c^{od}_{\min}$ the minimum path cost for (o, d) pair.

Definition 4.1. Evacuation betweenness centrality $C_{\epsilon b}(i)$ of node i is a parameter that represents the fraction of k_{od} shortest paths between all origin destination pairs (o,d) where $o \in O$ and $d \in D$, both different than $i \in N$ that i is a part of.

For the computation of evacuation betweenness centrality, we use

$$C_{\epsilon b}(i) = \sum_{o \in O} \sum_{d \in D} \frac{\sigma_{od}(i)}{\sigma_{od}}, \forall i \in N \backslash \Big(O \bigcup D\Big), \tag{3}$$

where $\sigma_{od} = k_{od}$ and $\sigma_{od}(i) \leq k_{od}$ is the cardinality of the subset of k_{od} shortest paths that pass through node i.

If node i has a high evacuation betweenness centrality $C_{\epsilon b}(i)$ defined in this way, it serves as a bridge to many other nodes on efficient paths towards their safe exits, and therefore is a hub or gateway towards safe exits. Nodes with a high evacuation betweenness centrality might be difficult for crowd coordination since these are intersections with crowd flow in possibly multiple directions. This fact increases the probability of the occurrence of herding. Therefore, special attention should be given to the crowd coordination in these nodes.

4.2 Evacuation centrality

When an unpredicted hazard occurs on a part of the evacuation route, the same gets unsafe and impassable. If, in the computation of an evacuation route, we do not consider this fact and the related possibility to reroute to other safe evacuation routes on its intermediate nodes, then, in case of contingency, rerouting towards safe areas might be impossible causing imminent evacuees' fatalities. Similar may occur in the case of a too high flow of evacuees that might saturate evacuation paths and cause panic. Therefore, for intermediate nodes of each evacuation route, we need to find a sufficient number of dissimilar simple shortest paths towards safe exits, preferably within the maximum time of evacuation given for a specific emergency case. In that respect, we define the evacuation centrality as follows.

Definition 4.2. Evacuation centrality $C_{\epsilon}(i)$ of node i is a parameter that represents the importance of node i for evacuation. The value of the evacuation centrality of the node is the number of available sufficiently dissimilar simple shortest paths from that node towards safe exits constrained by the paths' total cost (traversal time) c^{max}, i.e., $c_p \leq c^{max}$.

Here, c_{max} is the maximum evacuation time of the infrastructure based on the real emergency situation of the same.

5 Agile evacuation route problem

If real-time infrastructure information is available to evacuees and they can negotiate their routes, it becomes possible to provide a selection of optimized routes. Therefore, we assume that the evacuees are monitored by strategically positioned cameras and are communicated with via smart space displays, acoustic signs, smart-phones, etc. Monitoring permits us both to recognize the evacuees' behavior as to perceive their momentary position, people flow and density together with their safety conditions. Furthermore, we assume that the evacuee flow demand is defined by the presence of infrastructure occupants at their momentary positions whose evacuation destinations are defined as the closest locations at which evacuees are considered to be safe.

Our aim is, thus, to safely evacuate all the evacuation requests and if not possible, then, as many people as possible within the allotted time period. To this aim, we should find agile evacuation routes toward safe exits that consider evacuation centrality of the routes' intermediate nodes and other relevant characteristics that make the evacuation route sufficiently safe through time.

Let $x_a \geq 0$ be the flow of people in a unit time period on arc $a \in A$, which is limited from above by arc capacity $u_a \geq 0$ being the maximum arc flow. Moreover, ρ_a is the density of people on arc a. We recall that, in general, flow x_a on arc $a \in A$ depends on the density ρ_a.

Let $\overline{P}_w$ denote the set of available (simple) paths acceptable in terms of duration cost for each evacuation request $w \in W$ from origin $o_w \in O$ to fictitious sink $\hat{d}$ taking into account fairness considerations. Here, by fairness considerations, we mean giving the preference to the evacuees on the nodes $o \in O$ that are further away from the set of safe exits. Additionally, by acceptable in terms of duration cost, we mean the paths for evacuation request considering the upper bound in respect to the minimum duration among the paths for that evacuation request. Furthermore, let $\overline{P}_W$ be the set of all such paths. Then, all the path flows in $\bar{P}_W$ can be gathered in the global path flow vector $\mathbf{x}_W = (x^1, \ldots, x^r)$, where $r = |\overline{P}_W|$.

Moreover, we define feasible flow $\mathbf{x}_w$ as a subvector of flows of paths $p \in \bar{P}_w$ and x^p as a flow along path $p \in \overline{P}_w$. For describing the people flows over the whole road network in terms of path flows, we introduce $[|W| * |\overline{P}_W|]$ evacuation request-path incidence matrix Ψ with rows indexed by $w \in W$ and columns indexed by paths $p \in \bar{P}_W$. Furthermore, let Φ be the $[|A| * |\overline{P}_W|]$ arc-path incidence matrix.

5.1 Finding a node's maximum number of efficient evacuation paths

To determine evacuation centrality of each node, we need to determine the maximum number of minimum cost dissimilar simple paths from source node s to safe exits $i \in T$ subject to the condition that the cost of each path be not greater than a specified value c_{max}. Mathematical formulation of this (maximum network flow) problem is as follows:

(Z)

$$w = \max K \tag{4}$$

subject to

$$\sum_{p \in P} \big(\sum_{j:(i,j) \in A} \phi_{(i,j),p} x_p - \sum_{h:(h,i) \in A} \phi_{(h,i),p} x_p \big) = \begin{cases} K, & \text{if } i = s \\ 0, & \text{if } i \in N \backslash \{s, \hat{d}\} \\ -K, & \text{if } i = \hat{d} \end{cases} \tag{5}$$

$$\Phi \mathbf{x}^{\mathbf{P}} \leq \mathbf{1} \tag{6}$$

$$x_p \geq 0, \forall p \in P, \tag{7}$$

where P is a set of simple paths from source s to dummy node $\hat{d}$ of length $c_p \leq \gamma c_{\min}$.

In particular, $\gamma c_{\min} \leq c_{max}$, i.e., the upper bound on the path's length is limited by the maximum building's evacuation time c_{max}. Note that this formulation produces an unbounded linear program if there are negative cycles and under this condition the problem is in general NP-hard. However, in the case of evacuation network, all the arcs' costs (travel times) are greater or equal to zero, so we avoid this problem.

Finding the maximum number of arc-disjoint simple shortest paths might result in a very limited number of solutions since the number of arc-disjoint paths depends on the topology of the graph. It will be limited from above by the number of outgoing arcs from source s and the sum of the numbers of incoming dummy arcs to fictitious sink node $\hat{d}$. This is why we opt for finding a number of sufficiently dissimilar paths for each O-D pair. To this aim, a penalized objective function, which takes into consideration the violation of Constraint (6) becomes:

$$z(\lambda^{\mathbf{A}}) = \max K - \lambda^{\mathbf{A}^T} \mathbf{y}^{\mathbf{A}} \tag{8}$$

subject to

$$\sum_{p \in P} \big(\sum_{j:(i,j) \in A} \phi_{(i,j),p} x_p - \sum_{h:(h,i) \in A} \phi_{(h,i),p} x_p \big) = \begin{cases} K, & \text{if } i = s \\ 0, & \text{if } i \in N \backslash \{s, \hat{d}\} \\ -K, & \text{if } i = \hat{d} \end{cases} \tag{9}$$

$$\mathbf{y}^{\mathbf{A}} \geq \Phi \mathbf{x}^{\mathbf{P}} - \mathbf{1} \tag{10}$$

$$\mathbf{y}^{\mathbf{A}} \geq \mathbf{0}, \tag{11}$$

$$x_p \in \{0, 1\}, \forall p \in P, \tag{12}$$

where $\mathbf{y}^{\mathbf{A}}$ is a vector composed of non-negative variables related to a multiple usage of each arc $a \in A$ by paths $p \in \bar{P}$. $\bar{P}$ is the set of selected paths, that is, the paths for which $x_p = 1$. $\lambda^{\mathbf{A}}$ is a non-negative penalty vector of cardinality $|A|$ for using each arc $a \in A$ more than once. In this way, we penalize a multiple usage of arcs by multiple paths.

The model will return a maximum number of dissimilar paths (of length at most $c_{\max}$. This strategy is in line with our necessity to reroute more frequently in case of emergency. It is easy to demonstrate that $z(\lambda^{\mathbf{A}}) \geq w$ for any $\lambda^{\mathbf{A}} \geq \mathbf{0}$.

Since we are not interested in the structure (i.e., the constituent arcs) of dissimilar paths, but in the maximum number K of the same, we can approximate the computation by assuming that path variables are continuous and, by doing so, we substitute Constraint (12) with the following:

$$0 \leq x_p \leq 1, \forall p \in P. \tag{13}$$

Finally, for the best approximation, we resolve a dual of the former problem, i.e.,

$$g = \min \mathbf{z}(\lambda^{\mathbf{A}}) \tag{14}$$

subject to

$$\lambda^{\mathbf{A}} \geq \mathbf{0}. \tag{15}$$

Note that the value of g is an upper bound on the number of dissimilar paths and, therefore, also on the number of arc disjoint paths.

5.2 Ripley's K and L functions

To assure secure evacuation in case of dynamic unpredictable changes in the network's safety, it is not only important to provide evacuees with an evacuation route with a maximum combined value of the evacuation centralities of intermediate nodes. We should also consider arcs' costs (travel times) and their variance across the path, i.e., the dispersion of the nodes with alternative routes along the recommended route.

There exist different measures of spatial dispersion, such as, e.g., trace, determinant, and the largest eigenvalue of the covariance matrix or the average distance between nearest neighboring nodes. However, these measures do not take into consideration the homogeneity of the distribution of the nodes in the path. Note that a homogeneous set of nodes in a path is a set that is distributed such that approximately the same number of nodes occurs in any delimited part of the path.

Inspired by Ripley's K and L functions, which are closely related descriptive statistics for detecting deviations from spatial homogeneity, we propose a path $\hat{K}_p$ function. Given a path p, we define a path

$\hat{K}_p$ function as follows:

$$\hat{K}_p(t) = \Lambda^{-1} \sum_{i \in p} \sum_{j \neq i \in p} I(c_{ij} < t)/q, \tag{16}$$

where c_{ij} is the cost (travel time) of subpath of path p from node i to node j. Moreover, q is the number of constituent nodes of path p, t is the search radius, Λ is the average density of nodes (generally estimated as q/A, where A is the total cost of the path) and I is the indicator function (with the value 1 if its operand is true, 0 otherwise).

Edge effects arise in the vicinity of extreme nodes because the rest of the nodes are only positioned from one side of the path. This has as a consequence a biased node density in the vicinity of these nodes. However, ignoring edge effects also biases $\hat{K}_p(t)$, especially at large values of t. Although $\hat{K}_p(t)$ can be estimated for any t, it is common practice to consider only $t < \sqrt{c_p/2}$, where c_p is a total path cost. If the nodes are approximately homogeneous, $\hat{K}_p(t)$ should be approximately equal to $\hat{K}(t) = 2t$. If the individuals are clustered, $\hat{K}(t) > 2t$ [3].

A possible way to mitigate edge effects is adding a dummy arc (with a cost equal to the mean path's arcs' cost) between the origin and destination of the path to obtain a cycle. By doing so, we compute path $\hat{K}_p$ function on the resulting cycle instead of on the aforementioned path.

The issue with computing $\hat{K}_p(t)$ is that it is an NP hard problem since we need to perform an exhaustive check of all paths evaluating the value of $\hat{K}_p(t)$ for each one of them. Another issue with $\hat{K}_p(t)$ is that even though it is a parameter showing us the homogeneity of the arcs' costs, it does not discriminate between their high and low values, i.e., it equally values the arcs with high and low costs as long as they are homogeneously distributed throughout the path. However, in evacuation, we are not concerned about the arcs whose cost (travel time) is low, but the problem are the arcs with a too high travel time. In case of contingencies on an assigned route, this implies that the evacuees cannot reroute to other routes until they have not arrived to the arc's end.

This is why, we propose to find routes over arcs that are sufficiently cheap in terms of travel time and, therefore, introduce an upper bound on an admissible arc's cost c_{max}^{arc} such that:

$$c_a \cdot \phi_{ap} \cdot x_p \leq c_{max}^{arc} \ , \forall a \in A, \ p \in P. \tag{17}$$

However, the value of c_{max}^{arc} is related both to the structure of the evacuation network as to the evacuees' maximum allowed travel time on safety - jeopardized arcs. In this light, if all arcs are very large, then putting a too low value on c_{max}^{arc} and relaxing Constraint (17) will result in a too high cost in terms of relaxation penalties, while if c_{max}^{arc} is too high, then all arcs will be acceptable from this point of view.

We investigate the fundamental problem of finding agile evacuation routes traversing nodes of a graph with a maximum combined evacuation centrality and the constraint on maximum arc's cost. An agile evacuation route is found by maximizing the overall evacuation centrality of the intermediate nodes towards safe exit nodes as is shown in the following.

5.3 Maximizing evacuation centrality: the largest path problem

We search for a path with a maximum geometric mean value of the intermediate nodes' evacuation centrality. We opt for the geometric mean since it balances the utilitarian and egalitarian social welfare of the evacuation network.

This problem is NP hard [] and therefore, we need to convert it into the minimization problem. We do so by introducing a new variable that we call evacuation decentrality defined as

$$C_{d\epsilon}(i) = \frac{1}{C_\epsilon(i)}. \tag{18}$$

Then we want to find for each (o,d) pair $w \in W$ the shortest path that minimizes this value of its constituent nodes, and by minimizing it, we maximize the value of evacuation centrality.

With this aim, we define the evacuation centrality of an arc (i, j) by the evacuation centrality of its head node j. Therefore, we have:

$$m(w) = \min \sum_{p \in P} x_p \sqrt[|(i,j)\in p|]{\prod_{(i,j)\in p} C_{d\epsilon}(j)} = \tag{19}$$

$$\min \sum_{p \in P} x_p \log \sqrt[|(i,j)\in p|]{\prod_{(i,j)\in p} C_{d\epsilon}(j)} = \tag{20}$$

$$\min \sum_{p \in P} x_p \frac{1}{|(i,j) \in p|} \sum_{(i,j)\in p} \log C_{d\epsilon}(j) \tag{21}$$

Since we have a constraint on the maximum arcs' cost, we eliminate part on the number of arcs in the path $|(i, j) \in p|$ and therefore end up with the following equation.

$$m(w) = \min \sum_{p \in P} x_p \sum_{(i,j)\in p} \log C_{d\epsilon}(j) \tag{22}$$

subject to

$$\sum_{p \in P} x_p = 1 \tag{23}$$

$$c_a \cdot \phi_{ap} \cdot x_p \leq c_{max}^{arc} \ , \forall a \in A, \ p \in P \tag{24}$$

$$x_p = \{0, 1\} \ , \forall p \in P. \tag{25}$$

In this way, (22)-(25) is basically a shortest path problem formulated over paths with Constraint (24) on the highest arcs' cost. Using the path-arc incidence matrix, it can be reduced to the arc-based formulation. The transformation to the definition over arcs of this problem is as follows:

$$x^a \leq \frac{c_{max}^{arc}}{c_a} \ , \forall a \in A. \tag{26}$$

We need to dualize (26) since as it is, it is a maximum flow problem, and when dualized it becomes a shortest path problem.

Since (24) introduces further complexity to the problem since it might give non-integer solutions, we opt for a modelling approach that multiples the costs of the arcs that do not comply with (24) with a very high number M such that in the case there is no alternative arc for the computation of a shortest path, these origin-destination pairs include the arcs with such high arcs' values. On the contrary, if there are arcs available that comply with Constraint (24), they will firstly be taken into consideration for the shortest path calculation.

6 Proposed multi-agent system for agile evacuation route problem

The objective of the proposed evacuation route selection approach is to find agile evacuation routes for all pedestrians.

We assume the presence of a set of node agents N who communicate and compute pedestrian routes in a distributed manner similar to [7]. Pedestrians request their route from the smart space node agent closest to the origin of their travel (origin agent o). Based on the total demand for each time period expressed in terms of person flow per time unit, each origin agent o tries to achieve a sufficient number of shortest paths considering fairness for all its safe destinations d_o. Those destinations are requested by the persons starting the travel on o and the paths are computed through, e.g., modified Yen's loopless k-shortest path routing algorithm [5].

After the traffic assignment is made for O-D pairs, the latter decide on the pedestrians' assignment to the paths based on relevant social welfare parameters that guarantee equality through an iterative auction. The negotiation through auctions is local between each origin agent and the persons starting their travel at that origin, similar to [7].

Guidance g_a^k is considered a decision variable for each arc $a \in A$ and each path $k \in \bar{P}_w,\ w \in W$ instead of flow rate x_a as in traditional models. It enables pedestrians to follow a proposed path by following visual, tactile, acoustic or audio-haptic signals. Vector $\mathbf{g}_A^k = [g_1^k, \dots, g_{|E|}^k]$ specifies an egress decision at each passageway for routes $k \in \bar{P}_w, \forall w \in W$. These decisions, when filtered for each member of route $k \in \bar{P}_w$ for each O-D pair $w \in W$ depending on his/her affiliate ties, provide an individual's route plan.

7 Conclusions

In this work we studied people flow coordination problem in smart spaces. We proposed the terms of evacuation betweenness and evacuation centrality related with the node's importance for evacuation, and the term of agile evacuation routes.

Furthermore, inspired by Ripley's K and L functions, which are closely related descriptive statistics for detecting deviations from spatial homogeneity, we proposed a path $\hat{K}$ function to represent the homogeneity of the evacuation path's arc lengths. We formulated agile evacuation route problem and discussed its capability to adjust to possible contingencies through time.

In future work, we intend to further develop an architecture for agile evacuation route selection that recommends agile routes while considering maximum admissible arcs' cost. Furthermore, we plan to further consider the issues related with affiliate ties among evacuees and the influence of panic-related behaviors of stampeding and herding, and their impact on the overall evacuee flow.

ACKNOWLEDGEMENTS

This work has been partially supported by the Autonomous Region of Madrid through grant "MOSI-AGIL-CM" (P2013/ICE-3019) co-funded by EU Structural Funds FSE and FEDER, "SURF" (TIN2015-65515-C4-4-R) funded by the Spanish Ministry of Economy and Competitiveness, and by an STSM Grant from COST Action IC1303.

REFERENCES

[1] Ahmed Abdelghany, Khaled Abdelghany, Hani Mahmassani, and Wael Alhalabi, 'Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities', *European Journal of Operational Research*, **237**(3), 1105–1118, (2014).

[2] Kieran Conboy, 'Agility from first principles: Reconstructing the concept of agility in information systems development', *Information Systems Research*, **20**(3), 329–354, (2009).

[3] Noel Cressie, *Statistics for spatial data*, John Wiley & Sons, 2015.

[4] Lee D Han, Fang Yuan, Shih-Miao Chin, and Holing Hwang, 'Global optimization of emergency evacuation assignments', *Interfaces*, **36**(6), 502–513, (2006).

[5] Eugene L Lawler, 'A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem', *Management science*, **18**(7), 401–405, (1972).

[6] Marin Lujak, Stefano Giordani, and Sascha Ossowski, 'Fair route guidance: Bridging system and user optimization', in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1415–1422. IEEE, (2014).

[7] Marin Lujak, Stefano Giordani, and Sascha Ossowski, 'Route guidance: Bridging system and user optimization in traffic assignment', *Neurocomputing*, **151**, 449–460, (2015).

[8] Marin Lujak and Sascha Ossowski, 'Intelligent people flow coordination in smart spaces', in *Multi-Agent Systems and Agreement Technologies*, eds., Rovatsos M. et al., volume 9571 of *LNCS*, 34–49, Springer, (2016).

[9] Marin Lujak and Sascha Ossowski, 'On avoiding panic by pedestrian route recommendation in smart spaces', in *Proceedings of the 4th Int. Black Sea Conf. on Comm. and Networking*. IEEE, (2016, In Press).

[10] Salvador Casadesús Pursals and Federico Garriga Garzón, 'Optimal building evacuation time considering evacuation routes', *European Journal of Operational Research*, **192**(2), 692–699, (2009).

[11] Marcel van Oosterhout, Eric Waarts, and Jos van Hillegersberg, 'Change factors requiring agility and implications for it', *European Journal of Information Systems*, **15**(2), 132–145, (2006).

[12] Yahaya Y Yusuf, Mansoor Sarhadi, and Angappa Gunasekaran, 'Agile manufacturing:: The drivers, concepts and attributes', *International Journal of production economics*, **62**(1), 33–43, (1999).

Reputation in the Academic World

Nardine Osman and **Carles Sierra** [1]

Abstract. With open access gaining momentum, **open reviews** becomes a more persistent issue. Institutional and multidisciplinary open access repositories play a crucial role in knowledge transfer by enabling immediate accessibility to all kinds of research output. However, they still lack the quantitative assessment of the hosted research items that will facilitate the process of selecting the most relevant and distinguished content. This paper addresses this issue by proposing a computational model based on peer reviews for assessing the reputation of researchers and their research work. The model is developed as an overlay service to existing institutional or other repositories. We argue that by relying on peer opinions, we address some of the pitfalls of current approaches for calculating the reputation of authors and papers. We also introduce a much needed feature for review management, and that is calculating the reputation of reviews and reviewers.

1 MOTIVATION

There has been a strong move towards **open access** repositories in the last decade or so. Many funding agencies — such as the UK Research Councils, Canadian funding agencies, American funding agencies, the European Commission, as well as many universities — are promoting open access by requiring the results of their funded projects to be published in open access repositories. It is a way to ensure that the research they fund has the greatest possible research impact. Academics are also very much interested in open access repositories, as this helps them maximise their research impact. In fact, studies have confirmed that open access articles are more likely to be used and cited than those sitting behind subscription barriers [2]. As a result, a growing number of open access repositories are becoming extremely popular in different fields, such as PLoS ONE for Biology, arXiv for Physics, and so on.

With open access gaining momentum, **open reviews** becomes a more persistent issue. Institutional and multidisciplinary open access repositories play a crucial role in knowledge transfer by enabling immediate accessibility to all kinds of research output. However, they still lack the quantitative assessment of the hosted research items that will facilitate the process of selecting the most relevant and distinguished content. Common currently available metrics, such as number of visits and downloads, do not reflect the quality of a research product, which can only be assessed directly by peers offering their expert opinion together with quantitative ratings based on specific criteria. The articles published in the Frontiers book [5] highlight the need for open reviews.

To address this issue we develop an open peer review module, the Academic Reputation Model (ARM), as an overlay service to existing institutional or other repositories. Digital research works hosted in repositories using our module can be evaluated by an unlimited number of peers that offer not only a qualitative assessment in the form of text, but also quantitative measures to build the works reputation. Crucially, our open peer review module also includes a reviewer reputation system based on the assessment of reviews themselves, both by the community of users and by other peer reviewers. This allows for a sophisticated scaling of the importance of each review on the overall assessment of a research work, based on the reputation of the reviewer.

As a result of calculating the reputation of authors, reviewers, papers, and reviews, by relying on peer opinions, we argue that the model addresses some of the pitfalls of current approaches for calculating the reputation of authors and papers. It also introduces a much needed feature for review management, and that is calculating the reputation of reviews and reviewers. This is discussed further in the concluding remarks.

In what follows, we present the ARM reputation model and how it quantifies the reputation of papers, authors, reviewers, and reviews (Section 2), followed by some evaluation where we use simulations to evaluate the correctness of the proposed model (Section 3), before closing with some concluding remarks (Section 4).

2 ARM: ACADEMIC REPUTATION MODEL

2.1 Data and Notation

In order to compute reputation values for papers, authors, reviewers, and reviews we require a *Reputation Data Set*, which in practice should be extracted from existing paper repositories.

Definition 2.1 (Data)**.** A *Reputation data Set* is a tuple $\langle P, R, E, D, a, o, v\rangle$, where

- $P = \{p_i\}_{i\in\mathcal{P}}$ is a set of papers (e.g. DOIs).
- $R = \{r_j\}_{j\in\mathcal{R}}$ is a set of researcher names or identifiers (e.g. the ORCHID identifier).
- $E = \{e_i\}_{i\in\mathcal{E}} \cup \{\perp\}$ is a totally ordered evaluation space, where $e_i \in \mathbb{N}\setminus\{0\}$ and $e_i < e_j$ iff $i < j$ and $\perp$ stands for the absence of evaluation. We suggest the range [0,100], although any other range may be used, and the choice of range will not affect the performance.
- $D = \{d_k\}_{k\in\mathcal{K}}$ is a set of evaluation dimensions, such as *originality*, *technical soundness*, etc.
- $a : P \to 2^R$ is a function that gives the authors of a paper.
- $o : R \times P \times D \times Time \to E$, where $o(r, p, d, t) \in E$ is a function that gives the opinion of a reviewer, as a value in E, on a dimension d of a paper p at a given instant of time t.
- $v : R \times R \times P \times Time \to E$, where $v(r, r', p, t) = e$ is a function that gives the judgement of researcher r over the opin-

[1] Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain, email: {nardine, sierra}@iiia.csic.es

ion of researcher r', on paper p as a value $e \in E$.[2] Therefore, a judgement is a reviewer's opinion about another reviewer's opinion. Note that while opinions about a paper are made with respect to a given dimension in D, judgements are not related to dimensions. We assume a judgement is only made with respect to one dimension, which describes how good the review is *in general*.

We will not include the dimension (or the criteria being evaluated, such as originality, soundness, etc.) in the equations to simplify the notation. There are no interactions among dimensions so the set of equations apply to each of the dimensions under evaluation.

Also, we will also omit the reference to time in all the equations. Time is essential as all measures are dynamic and thus they evolve along time. We will make the simplifying assumption that all opinions and judgements are maintained in time, that is, they are not modified. Including time would not change the essence of the equations, it will simply make the computation complexity heavier.

Finally, if a data set allowed for papers, reviews, and/or judgements to have different versions, then our model simply considers the latest version only.

2.2 Reputation of a Paper

We say the reputation of a paper is a weighted aggregation of its reviews, where the weight is the reputation of the reviewer. (Section 2.4).

$$R_P(p) = \begin{cases} \dfrac{\displaystyle\sum_{\forall r \in rev(p)} R_R(r) \cdot o(r,p)}{\displaystyle\sum_{\forall r \in rev(p)} R_R(r)} & \text{if } |rev(p)| \geq k \\ \perp & \text{otherwise} \end{cases} \quad (1)$$

where $rev(p) = \{r \in R \mid o(r,p) \neq \perp\}$ denotes the reviewers of a given paper.

Note that when a paper receives less that k reviews, its reputation is defined as unknown, or $\perp$. We currently leave k as a parameter, though we suggest that $k > 1$, so that the reputation of a paper is not dependent on a single review. We also recommend small numbers for k, such as 2 or 3, because we believe it is usually difficult to obtain reviews. As such, new papers can quickly start building a reputation.

2.3 Reputation of an Author

We consider that a researcher's author reputation is an aggregation of the reputation of her papers. The aggregation is based on the concept that *the impact of a paper's reputation on its authors' reputation is inversely proportional to the total number of its authors*. In other words, if one researcher is the sole author of a paper, then this author is the only person responsible for this paper, and any (positive or negative) feedback about this paper is propagated as is to its sole author. However, if the researcher has co-authored the paper with several other researchers, then the impact (whether positive or negative) that this paper has on the researcher decreases with the increasing number of co-authors. We argue that collaborating with different researchers usually increases the quality of a research work since the combined expertise of more than one researcher is always better than the expertise of a single researcher. Nevertheless, the gain in a researcher's reputation decreases as the number of co-authors increase. Hence, our model might cause researchers to be more careful when selecting their collaborators, since they should aim at increasing the quality of the papers they produce in such a way that the gain for each author is still larger than the gain it could have received if it was to work on the same research problem on her own. As such, adding authors who do not contribute to the quality of the paper will also discouraged.

$$R_A(r) = \begin{cases} \dfrac{\displaystyle\sum_{\forall p \in pap(r)} \gamma(p)^{\gamma} \times R_P(p) + (1 - \gamma(p)^{\gamma}) \times 50}{|pap(r)|} & \text{if } pap(r) \neq \emptyset \\ \perp & \text{otherwise} \end{cases} \quad (2)$$

where $pap(r) = \{p \in P \mid r \in a(p) \wedge R_P(p) \neq \perp\}$ denotes the papers authored by a given researcher r, $\perp$ describes ignorance, $\gamma(p) = \dfrac{1}{|a(p)|}$ is the coefficient that takes into consideration the number of authors of a paper (recall that $a(p)$ denotes the authors of a paper p), and γ is a tuning factor that controls the rate of decrease of the $\gamma(p)$ coefficient. Also note the multiplication by 50, which describes ignorance, as 50 is the median of the chosen range $[0, 100]$. If another range was chosen, the median of that range would be used here. The choice of range and its median does not affect the performance of the model (i.e. the results of the simulation of Section 3 would remain the same).

2.4 Reputation of a Reviewer

Similar to the reputation of authors (Section 2.3), we consider that if a reviewer produces 'good' reviews, then the reviewer is considered to be a 'reputed' reviewer. Furthermore, we consider that the reputation of a reviewer is essentially an aggregation of the opinions over her reviews.[3]

We assume that the opinions on how good a review is can be obtained, in a first instance, by other reviewers that *also reviewed the same paper*. However, as this is a new feature to be introduced in open access repositories and conference and journal paper management systems, we believe collecting such information might take some time. An alternative that we consider here is that in the meantime we can use the 'similarity' between reviews as a measure of the reviewers opinions about reviews. In other words, the heuristic could be phrased as 'if my review is similar to yours then I may assume your judgement of my review would be good.'

We note $v^*(r_i, r_j, p) \in E$ for the 'extended judgement' of r_i over r_j's opinion on paper p, and define it as an aggregation of opinions and similarities as follows:

$$v^*(r_i, r_j, p) = \begin{cases} v(r_i, r_j, p) & \text{if } v(r_i, r_j, p) \neq \perp \\ Sim(\bar{o}(r_i, p), \bar{o}(r_j, p)) & \text{If } \bar{o}(r_i, p) \neq \perp \text{ and } \bar{o}(r_j, p) \neq \perp \\ \perp & \text{Otherwise} \end{cases} \quad (3)$$

where Sim stands for an appropriate similarity measure. We say the similarity between two opinions is the difference between the two: $Sim(\bar{o}(r_i, p), \bar{o}(r_j, p)) = 100 - |\bar{o}(r_i, p) - \bar{o}(r_j, p)|$.

[2] In tools like ConfMaster (www.confmaster.net) this information could be gathered by simply adding a private question to each paper review, answered with elements in E, one value in E for the judgement on each fellow reviewer's review.

[3] We assume a review can only be written by one reviewer, and as such, the number of co-authors of a review is not relevant as it was when calculating the reputation of authors.

Given this, we consider that the overall opinion of a researcher on the capacity of another researcher to make good reviews is calculated as follows. Consider the set of judgements of r_i over reviews made by r_j as: $V^*(r_i, r_j) = \{v^*(r_i, r_j, p) \mid v(r_i, r_j, p) \neq \perp \text{ and } p \in P\}$. This set might be empty. Then, we define the judgement of a reviewer over another one as a simple average:

$$R_R(r_i, r_j) = \begin{cases} \dfrac{\sum_{\forall v \in V^*(r_i,r_j)} v}{|V^*(r_i, r_j)|} & \text{if } V^*(r_i, r_j) \neq \emptyset \\ \perp & \text{otherwise} \end{cases} \quad (4)$$

Finally, the reputation of a reviewer r, $R_R(r)$, is an aggregation of judgements that her colleagues make about her capability to produce good reviews. We weight this with the reputation of the colleagues as a reviewer:

$$R_R(r) = \begin{cases} \dfrac{\sum_{\forall r_i \in R^*} R_R(r_i) \cdot R_R(r_i, r)}{\sum_{\forall r_i \in R^*} R_R(r_i)} & R^* \neq \emptyset \\ 50 & \text{otherwise} \end{cases} \quad (5)$$

where $R^* = \{r_i \in R \mid V^*(r_i, r) \neq \emptyset\}$. When no judgements have been made over r, we take the value 50 to represent ignorance (as 50 is the median of the chosen range $[0, 100]$ — again, we note that any the choice of range and its median does not affect the performance of the model; that is, the results of the simulation of Section 3 would remain the same).

Note that the reputation of a reviewer depends on the reputation of other reviewers. In other words, every time the reputation of one reviewer will change, it will trigger changing the reputation of other reviewers, which might lead to an infinite loop of modifying the reputation of reviewers. We address this by using an algorithm similar to the EigenTrust algorithm, as illustrated by Algorithm **??** of the Appendix. In fact, this algorithm may be considered as a variation of the EigenTrust algorithm, which will require some testing to confirm how fast it converges.

2.5 Reputation of a Review

The reputation of a review is similar to the one for papers but using judgements instead of opinions. We say the reputation of a review is a weighted aggregation of its judgements, where the weight is the reputation of the reviewer (Section 2.4).

$$R_O(r', p) = \begin{cases} \dfrac{\sum_{\forall r \in jud(r',p)} R_R(r) \cdot v^*(r, r', p)}{\sum_{\forall r \in jud(r',p)} R_R(r)} & \text{if } |jud(r', p)| \geq k \\ R_R(r') & \text{otherwise} \end{cases} \quad (6)$$

where $jud(r', p) = \{r \in R \mid v^*(r, r', p) \neq \perp\}$ denotes the set of judges of a particular review written by r' on a given paper p.

Note that when a review receives less that k judgements, its reputation will not depend on the judgements, but it will inherit the reputation of the author of the review (her reputation as a reviewer).

We currently leave k as a parameter, though we suggest that $k > 1$, so that the reputation of a review is not dependent on a single judge. Again, we recommend small numbers for k, such as 2 or 3, because we believe it will be difficult to obtain large numbers of judgements.

2.6 A Note on Dependencies

Figure 1 shows the dependencies between the different measures (reputation measures, opinions, and judgements). The decision of When to re-calculate those measures is then based on those dependencies. We provide a summary of this below. Note that measures in white are not calculated, but provided by the users. As such, we only discuss those in grey (grey rectangles represent reputation measures, whereas the grey oval represents the extended judgements).

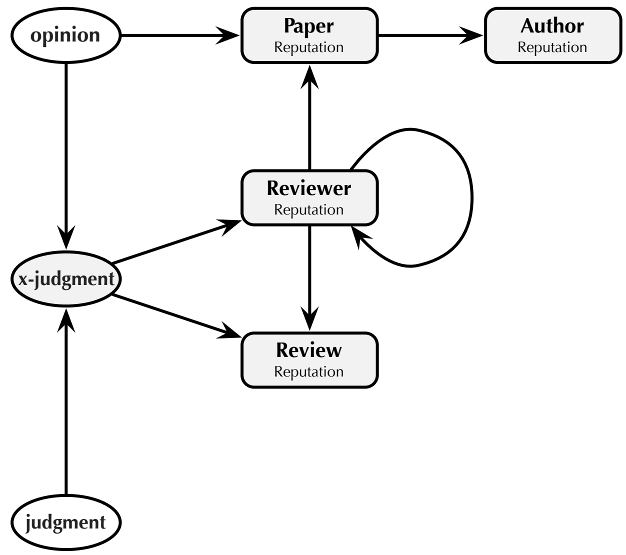


Figure 1: Dependencies

- **Author's Reputation.** The reputation of the author depends on the reputation of its papers (Equation 2). As such, every time the reputation of one of his papers changes, or every time a new paper is created, the reputation of the author must be recalculated.
- **Paper's Reputation.** The reputation of the paper depends on the opinions it receives, and the reputation of the reviewers giving those opinions (Equation 1). As such, every time a paper receives a new opinion, or every time the reputation of one of the reviewers changes, then the reputation of the paper must be recalculated .
- **Review's Reputation.** The reputation of a review depends on the extended judgements it receives, and the reputation of the reviewers giving those judgements (Equation 6). As such, every time a review receives a new extended judgements, or every time the reputation of one of the reviewers changes, then the reputation of the review must be recalculated.
- **Reviewer's Reputation.** The reputation of a reviewer depends on the extended judgements of other reviewers and their reputation (Equation 5). As such, the reputation of the reviewer should be modified every time there is a new extended judgement or the reputation of on of the reviewers changes. As the reputation of a reviewer depends on the reputation of reviewers, then we suggest to calculate the reputation of all reviewers repeatedly (in a manner similar to EigenTrust) in order to converge. If this will be computationally expensive, then this can be computed once a day, as opposed to triggered by extended judgements and the change in reviewers' reputation.
- **x-judgement.** The extended judgement is calculated either based on judgements (if available) or the similarity between opinions

(when judgements are not available) (Equation 3). As such, the extended judgement should be recalculated every time a new (direct) judgement is made, or every time a new opinion is added on a paper which already has opinions by other reviewers.

3 Evaluation through Simulation

3.1 Simulation

To evaluate the effectiveness of the proposed model, we have simulated a community of researchers, using NetLogo [8]. We clarify that the focus of this work is not implementing a simulation that models the real world, but a simulation that allows us to verify our model. As such, many assumptions that we make for this simulation, and will appear shortly, might not be precisely (or always) true in the real world (such as having the true quality of a paper inherit the quality of the best author).

In our simulation, a breed in NetLogo (or a node in the research community's graph) represents either a researcher, a paper, a review, or a judgement. The relations between breeds are: (1) *authors of*, that specifies which researchers are authors of a given paper, (2) *reviewers of*, that specifies which researchers are reviewers of a given paper, (3) *reviews of*, that specifies which reviews give opinions on a given paper, (4) *judgements of*, that specifies which judgements give opinions on a given review; and (5) *judges of*, that specifies which researchers have judged which other researcher.

Also, each researcher has four parameters that describe: (1) her reputation as an author, (2) her reputation as a reviewer, (3) her *true* research quality; and (4) her *true* reviewing quality. The first two are calculated by our ARM model, and they evolve over time. However, the last two describe the researcher's true quality with respect to writing papers as well as reviewing papers or other reviews, respectively. In other words, our simulation assumes true qualities exist, and that they are constant. In real life, there are no such measures. Furthermore, how good one is at writing papers or writing reviews or making judgements naturally evolves with time. Nevertheless, we chose to keep the simulation simple by sticking to constant true qualities, as the purpose of the simulation is simply to evaluate the correctness of our ARM model.

Similar to researchers, we say each paper has two parameters that describe it: (1) its reputation, which is calculated by our ARM model, and it evolves over time; and (2) its *true* quality. Again, we assume that a paper's true quality exists. How it is calculated is presented shortly.

Reviews also have two parameters: (1) the opinion provided by the review, which in real life is set by the researcher performing the review, while in our simulation it is calculated by the simulator, as illustrated shortly; and (2) the reputation of the review, which is calculated by our ARM model and it evolves over time.

Judgements, on the other hand, only have one parameter: the opinion provided by the judgement, which in real life is set by the researcher judging a review, while in our simulation it is calculated by the simulator, as illustrated shortly.

Simulation starts at time zero with no researchers in the community, and hence, no papers, no reviews, and no judgements. Then, with every tick of the simulation, a new paper is created, which may sometimes require the creation of new researchers (either as authors or reviewers). With the new paper, reviews and judgements are also created. How these elements are created is defined next by the simulator's parameters and methods, that drive and control this behaviour. We note that a tick of the simulation does not represent a fixed unit in calendar time, but the creation of one single paper.

The ultimate aim of the evaluation is to investigate how close are the calculated reputation values to the *true* values: the reputation of a researcher as an author, the reputation of a researcher as a reviewer, and the reputation of a paper.

The parameters and methods that drive and control the evolution of the community of researchers and the evolution of their research work are presented below.

1. *Number of authors.* Every time a new paper is created, the simulator assigns authors for this paper. How many authors are assigned is defined by the number of authors parameter ($\#_{co\text{-}authors}$), which is defined as a Poisson distribution. For every new paper, a random number is generated from this Poisson distribution. Who to assign is chosen randomly from the set of researchers, although sometimes, a new researcher is created and assigned to this paper (see the 'researchers birth rate' below). This ensures the number of researchers in the community grows with the number of papers.
2. *Number of reviewers.* Every time a new paper is created, the simulator also assigns reviewers for this paper. How many reviewers are assigned is defined by the number of reviewers parameter ($\#_{reviewers}$), which is defined as a Poisson distribution. For every new paper, a random number is generated from this Poisson distribution. As above, who to assign is chosen randomly from the set of researchers, although sometimes, a new researcher is created and assigned to this paper.
3. *Researchers birth rate.* As illustrated above, every paper requires authors and reviewers to be assigned to it. When assigning authors and reviewers, the simulation will decide whether to assign an already existing researcher (if any) or create a new researcher. This decision is controlled by the researchers birth rate parameter ($birth\_rate$), which specifies the probability of creating a new researcher.
4. *Researcher's true research quality.* The author's true quality is sampled from a beta distribution specified by the parameters α_A and β_A. We choose the beta distribution because it is a very versatile distribution which can be used to model several different shapes of probability distributions by playing with only two parameters, α and β.
5. *Researcher's true review quality.* The reviewer's true quality is sampled from a beta distribution specified by the parameters α_R and β_R. Again, the beta distribution is a very versatile distribution which can be used to model several different shapes of probability distributions by playing with only two parameters, as illustrated shortly by our experiments.
6. *Paper's true quality.* We assume that a paper's true quality is the true quality of its best author, that is, the author with the highest true research quality). We believe this assumption has some ground in real life. For instance, some behaviour (such as looking for future collaborators, selecting who to give a funding to, etc.) assumes researchers to be of a certain quality, and their research work to follow that quality respectively.
7. *Opinion of a Review.* The opinion presented by a review is specified as the paper's true quality plus some noise, where the noise depends on the reviewer's true quality. This noise is chosen randomly from the range $[-(100 - review\ quality)/2, +(100 - review\ quality)/2]$. In other words, the maximum noise that can be added for the worst reviewer (whose review quality is 0) is ± 50, and the least noise that can be added for the best reviewer (whose review quality is 100) is 0.
8. *Opinion of a Judgement.* The value (or opinion) of a judgement on a review is calculated as the similarity between the review's

value (opinion) and the judge's review value (opinion), where the similarity is defined by the metric distance as: $100 - |review - judge's\ review|$. Note that, for simplification, direct judgements have not been simulated, we only rely on indirect judgements.

3.2 Results

3.2.1 *Experiment 1: The impact of the community's quality of reviewers*

Given the above, we ran the simulator for 100 ticks (generating 100 papers). We ran the experiment over 6 different cases. In each, we had the following parameters fixed:

$\#_{co\text{-}authors} = 2$

$\#_{reviewers} = 3$

$birth\_rate = 3$

$\alpha_A = \beta_A = 1$

$k = 3$ (of Equations 1 and 6)

$\gamma = 1$ (of Equation 2)

The only parameters that changed where those defining the beta distribution of the reviewers' qualities. This experiment illustrated the impact of the community's quality of reviewers on the correctness of the ARM model.

The results of the simulation are presented by Figure 2. For each case, the distribution of the reviewers' true quality is illustrated to the right of the results. The results, in numbers, are also presented by Table 1. We notice that the least error is presented when the reviewers are all of relatively good quality, with the majority being great reviewers (Figure 2e). The errors start increasing as bad reviewers are added to the community (Figure 2c). They increase even further in both cases, when the quality of reviewers follows a uniform distribution (Figure 2a), as well as when the reviewers are equiprobably good or bad, with no average reviewers (Figure 2b). As soon as the majority of reviewers are of poor quality (Figure 2d), the errors increase even further, with the worst case being when good reviewers are absent from the community (Figure 2f). These results are not surprising. A paper's true quality is not something that can be measured, or even agreed upon. As such, the trust model depends on the opinions of other researchers. As a result, the better the reviewing quality of researchers, the more accurate the trust model will be, and vice versa.

The numbers of Table 1 illustrate how the error in the papers' reputation increases with the error in the reviewers' reputation, though at a smaller rate. One curious thing about these results is the constant error in the reputation of authors. The next experiment investigates this issue.

Last, but not least, we note that the error is usually stable. This is because every time a paper is created, all the reviews it receives and the judgements those reviews receive are created at the same simulation time-step. In other words, it is not the case that papers accumulate more reviews and judgements over time, for the error to decrease over time.

3.2.2 *Experiment 2: The impact of co-authorship*

In the second experiment, we investigate the impact of co-authorship on authors' reputation. We choose the two extreme cases from experiment 1, when there are only relatively good authors in the community ($\alpha = 5$ and $\beta_R = 1$), and when there are only relatively bad

	Error in Reviewers' Reputation	**Error in Papers' Reputation**	**Error in Authors' Reputation**
$\alpha_R = 5$ & $\beta_R = 1$	~ 11 %	~ 2 %	~ 22 %
$\alpha_R = 2$ & $\beta_R = 1$	~ 23 %	~ 5 %	~ 23 %
$\alpha_R = 1$ & $\beta_R = 1$	~ 30 %	~ 7 %	~ 23 %
$\alpha_R = 0.1$ & $\beta_R = 0.1$	~ 34 %	~ 5 %	~ 22 %
$\alpha_R = 1$ & $\beta_R = 2$	~ 44 %	~ 8 %	~ 23 %
$\alpha_R = 1$ & $\beta_R = 2$	~ 60 %	~ 9 %	~ 20 %

Table 1: The results of experiment 1, in numbers

authors in the community ($\alpha = 5$ and $\beta_R = 1$). For each of these cases, we then change the number of co-authors, investigating three cases: $\#_{co\text{-}authors} = \{0, 1, 2\}$. All other parameters remain set to those presented in experiment 1 above.

The results of this experiment are presented by Figure 3. The numbers are presented in Table 2. The results show that the error in the reviewers and papers reputation almost does not change for different numbers of co-authors. However, the error in the reputation of authors does. When there are no co-authors ($\#_{co\text{-}authors} = 0$), the error in authors' reputation is almost equal to the error in papers' reputation (Figures 3a and 3b). As soon as 1 co-author is added ($\#_{co\text{-}authors} = 0$), the error in authors' reputation increases (Figures 3c and 3d). When 2 co-authors are added ($\#_{co\text{-}authors} = 2$), the error in authors' reputation reaches the maximum, around 20–22% (Figures 3e and 3f). In fact, unreported results show that the error in authors' reputation is almost the same in all cases for $\#_{co\text{-}authors} \geq 2$.

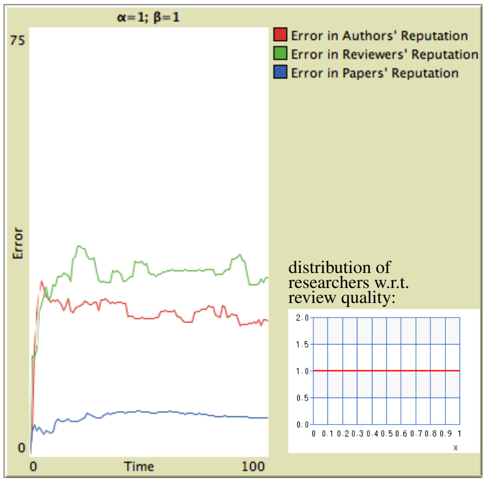
	Error in Reviewers' Reputation		**Error in Papers' Reputation**		**Error in Authors' Reputation**	
	α_R=5; $\beta_R=1$	α_R=1; $\beta_R=5$	α_R=5; $\beta_R=1$	α_R=1; $\beta_R=5$	α_R=5; $\beta_R=1$	α_R=1; $\beta_R=5$
$\#_{co\text{-}authors} = 0$	$\sim$11%	$\sim$60%	$\sim$2%	$\sim$9%	$\sim$22%	$\sim$20%
$\#_{co\text{-}authors} = 1$	$\sim$13%	$\sim$57%	$\sim$3%	$\sim$9%	$\sim$12%	$\sim$15%
$\#_{co\text{-}authors} = 2$	$\sim$13%	$\sim$54%	$\sim$3%	$\sim$9%	$\sim$2%	$\sim$7%

Table 2: The results of experiment 2, in numbers

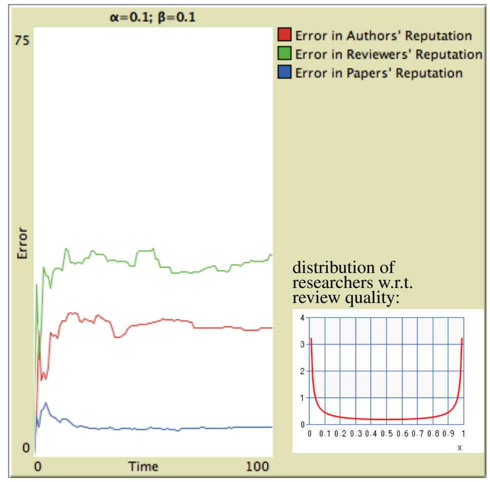
4 Conclusion

We have presented the ARM reputation model for the academic world. ARM helps calculate the reputation of researchers, both as authors and reviewers, and their research work. Additionally, ARM also calculates the reputation of reviews.

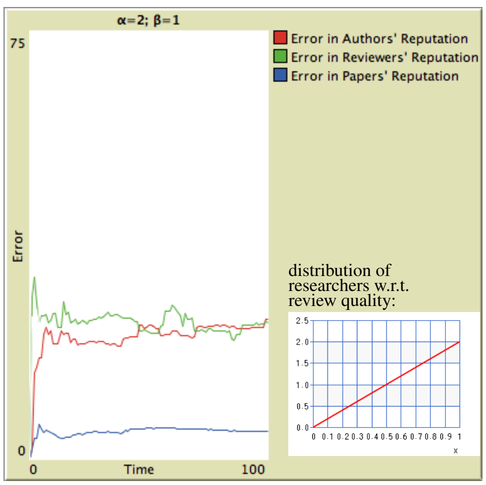
Concerning the reputation of authors, the most commonly used reputation measure is currently the h-index [4]. However, the h-index has its flaws. For instance, the h-index can be manipulated through self-citations [1, 3]. A study has also found the h-index as not providing a significantly more accurate measure of impact than the total number of citations [9]. ARM, on the other hand, bases the reputation of authors on the opinions that their papers receive from other members in their academic community. We believe this should be a more



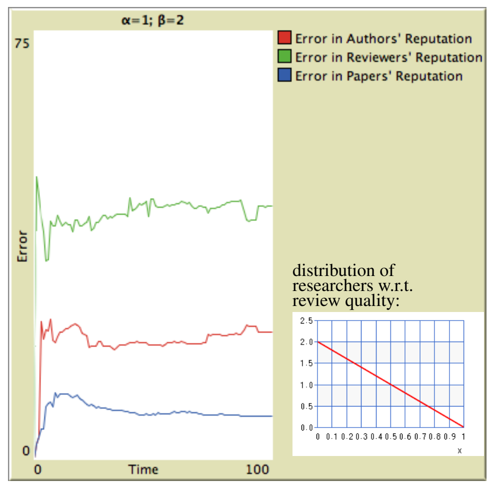
(a) $\alpha_R = 1$ and $\beta_R = 1$



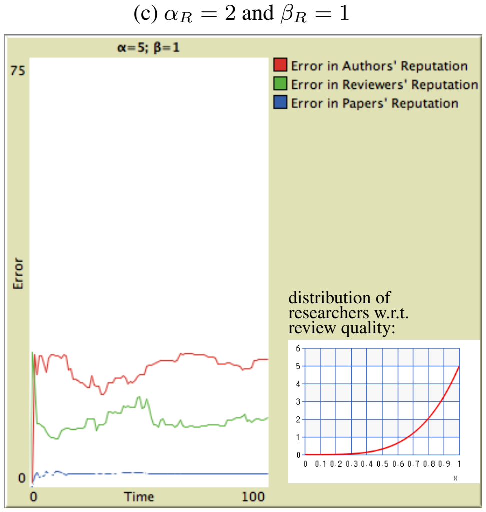
(b) $\alpha_R = 0.1$ and $\beta_R = 0.1$



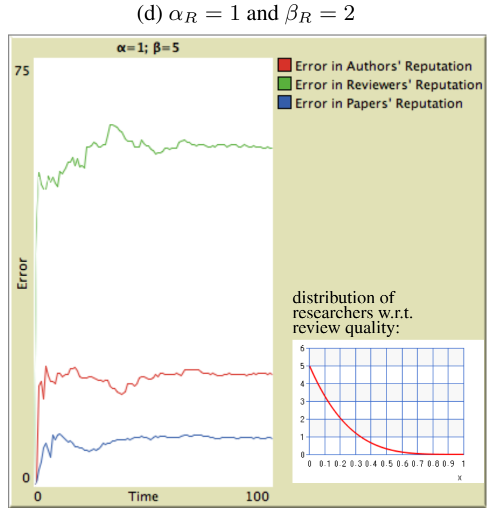
(c) $\alpha_R = 2$ and $\beta_R = 1$



(d) $\alpha_R = 1$ and $\beta_R = 2$

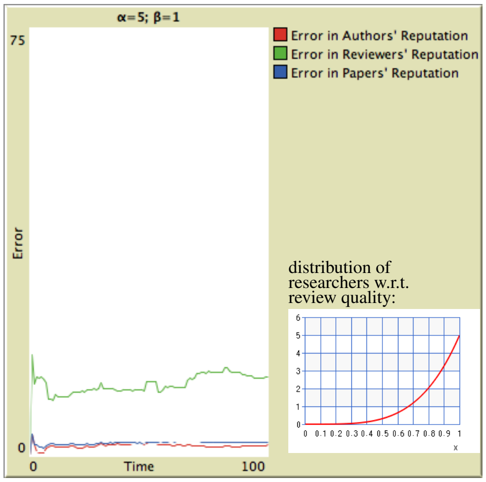


(e) $\alpha_R = 5$ and $\beta_R = 1$

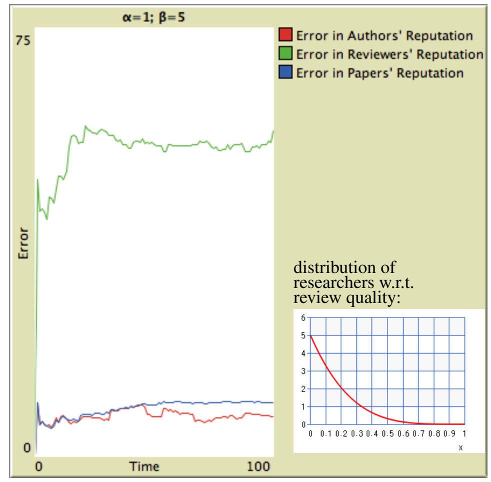


(f) $\alpha_R = 1$ and $\beta_R = 5$

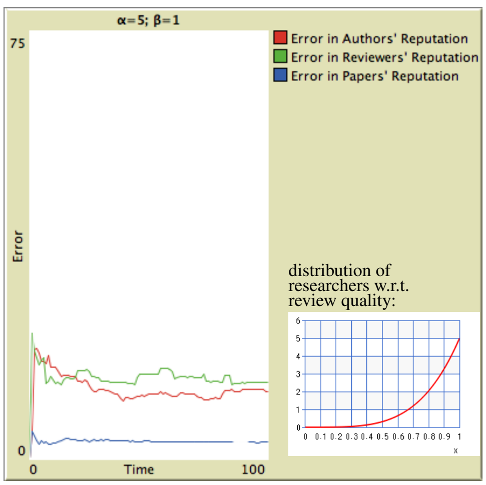
Figure 2: The impact of reviewers' quality on reputation measures. For each set of results, the distribution of the reviewers' true quality is presented to the right of the results.



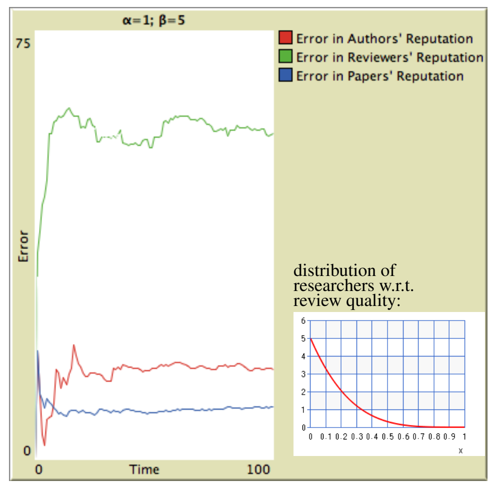
(a) $\alpha_R = 5$, $\beta_R = 1$, and $\#_{co\text{-}authors} = 0$



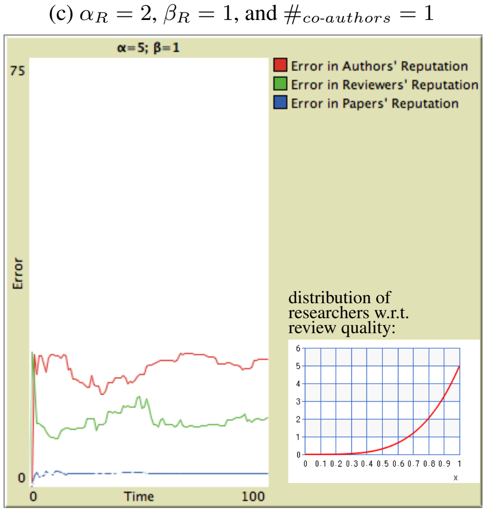
(b) $\alpha_R = 1$, $\beta_R = 5$, and $\#_{co\text{-}authors} = 0$



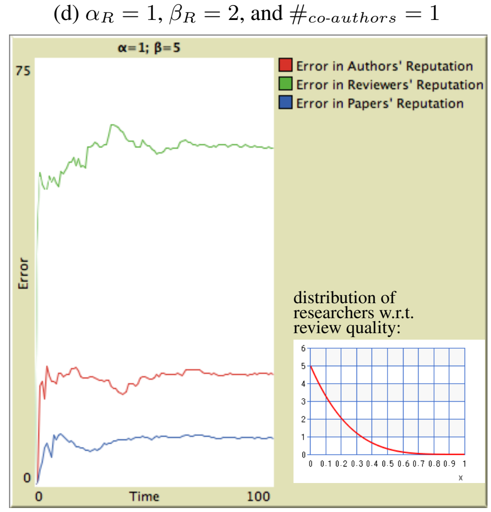
(c) $\alpha_R = 2$, $\beta_R = 1$, and $\#_{co\text{-}authors} = 1$



(d) $\alpha_R = 1$, $\beta_R = 2$, and $\#_{co\text{-}authors} = 1$



(e) $\alpha_R = 5$, $\beta_R = 1$, and $\#_{co\text{-}authors} = 2$



(f) $\alpha_R = 1$, $\beta_R = 5$, and $\#_{co\text{-}authors} = 2$

Figure 3: The impact of co-authorship on reputation of authors. For each set of results, the distribution of the reviewers' true quality is presented to the right of the results.

accurate approach, though future work should aim at comparing both approaches.

Concerning the reputation of papers, the most common measure currently used is the total number of citations a paper gets. Again, this measure can easily be manipulated through the self-citations. [7] presents an alternative approach based on the propagation of opinions in structural graphs. It allows papers to build reputation either from the direct reviews it receives, or inherit reputation from the place where the paper is published. In fact, a sophisticated propagation model is proposed to allow reputation to propagate upwards as well as downwards in structural graphs (e.g. from a section to a chapter to a book, and vice versa). Simulations presented in [6] illustrate the potential impact of this model. ARM does not have any notion of propagation. The model is strictly based on direct opinions (reviews and judgements), and when no opinions are present, ignorance is assumed (as in the default reputation of authors and papers).

Concerning the reputation of reviews and reviewers, to our knowledge, these reputation measures have not been addressed yet. Nevertheless, we believe these are important measures. Conference management systems are witnessing a massive increase in paper submissions, and in many disciplines, finding good reviewers is becoming a challenging task. Deciding what papers to accept/reject is sometimes a challenge for conference and workshop organisers. ARM is a reputation model that addresses this issue by helping recognise the good reviews/reviewers from the bad.

The obvious next steps for ARM is applying it to a real dataset. In fact, the model is currently being integrated with two Spanish repositories: DIGITAL.CSIC (https://digital.csic.es) and e-IEO (http://www.repositorio.ieo.es/e-ieo/). However, these repositories do not have any opinions or judgements yet, and as such, time is needed to start collecting this data. We are also working with the IJCAI 2017 conference (http://ijcai-17.org) in order to allow reviewers to review each other. We will collect the data of this conference, which will provide us with the reviews and judgements needed for evaluating our model. We will also continue to look through existing datasets.

Future work can investigate a number of additional issues. For instance, we plan to provide data on the convergence performance of the algorithm. One can also study the different types of attacks that could impact the proposed computational model. While similarity of reviews is now computed based on the similarity of the quantitative opinions, the similarity between qualitative opinions may also be used in future work by making use of natural language processing techniques. Also, while we argue that direct opinion can help the model avoid the pitfalls of the literature, it is also true that direct opinions are usually scarce. As such, if needed, other information sources for opinions may also be considered, such as citations. This information can be translated into opinions, and the equations of ARM should then change to give more weight to direct opinions than other information sources.

ACKNOWLEDGEMENTS

This work has been supported by CollectiveMind (a project funded by the Spanish Ministry of Economy & Competitiveness (MINECO), grant # TEC2013-49430-EXP), and Open Peer Review Module for Repositories (a project funded by OpenAIRE, which in turn is an EU funded project).

REFERENCES

[1] Christoph Bartneck and Servaas Kokkelmans, 'Detecting h-index manipulation through self-citation analysis', *Scientometrics*, **87**(1), 85–98, (2010).

[2] Gunther Eysenbach, 'Citation advantage of open access articles', *PLoS Biology*, **4**(5), e157, (05 2006).

[3] Emilio Ferrara and Alfonso E. Romero, 'Scientific impact evaluation and the effect of self-citations: Mitigating the bias by discounting the h-index', *Journal of the American Society for Information Science and Technology*, **64**(11), 2332–2339, (2013).

[4] J. E. Hirsch, 'An index to quantify an individual's scientific research output', *Proceedings of the National Academy of Sciences of the United States of America*, **102**(46), 16569–16572, (2005).

[5] Nikolaus Kriegeskorte and Diana Deca, eds. *Beyond open access: visions for open evaluation of scientific papers by post-publication peer review*, Frontiers in Computational Neuroscience. Frontiers E-books, November 2012.

[6] Nardine Osman, Jordi Sabater-Mir, Carles Sierra, and Jordi Madrenas-Ciurana, 'Simulating research behaviour', in *Proceedings of the 12th International Conference on Multi-Agent-Based Simulation*, MABS'11, pp. 15–30, Berlin, Heidelberg, (2012). Springer-Verlag.

[7] Nardine Osman, Carles Sierra, and Jordi Sabater-Mir, 'Propagation of opinions in structural graphs', in *Proceedings of the 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pp. 595–600, Amsterdam, The Netherlands, The Netherlands, (2010). IOS Press.

[8] Seth Tisue and Uri Wilensky, 'Netlogo: Design and implementation of a multi-agent modeling environment', in *In Proceedings of the Agent Conference*, pp. 161–184, (2004).

[9] Alexander Yong, 'Critique of hirschs citation index: A combinatorial fermi problem', *Notices of the American Mathematical Society*, **61**(11), 1040–1050, (2014).

Towards a simulation of AmI environments integrating social and network simulations

Álvaro Sánchez-Picot and **Diego Martín** and **Borja Bordel** and **Ramón Alcarria** and **Diego Sánchez de Rivera** and **Tomás Robles**[1]

Abstract.
We are heading towards a technological and hyper-connected world where every building is going to be full of sensors and actuators to monitor and interact with it, in what is known as an Ambient Intelligence (AmI) environment. The main problem when creating such environment is how expensive it can be, so a tool such a simulator could help to improve the way in which the devices are installed, testing with different configurations until you arrive to the optimal one. Also this simulator could help once the infrastructure is created to detect certain events before they happen, being able to apply a countermeasure. In this paper we propose the architecture to integrate a social and a network simulation in order to create a simulation for an AmI environment.

1 INTRODUCTION

We are heading towards a technologically connected world. More and more devices are installed in our homes and our environment. Some of these devices are not very new such as televisions, air conditioning units or security cameras but others are relatively new such as ambient lights, temperature sensors or microphones to talk with a computer. Currently we want to know much more about what happens in our home and our surroundings than several years ago and thanks to the mobile phones we can easily access this information anywhere and in real time. Nowadays we also want the environment to act proactively depending on what happens, for example, turn on the lights automatically when the nightfall comes and turn it off when there is nobody present or open and close blinds depending on the light outside or the desired temperature inside the building. This corresponds to what is known as Ambient Intelligence (AmI) environment, that is sensitive and responsive to the presence of people and environmental factors.

The idea of an AmI environment is that all its devices cooperate together in order to obtain a desired result. The intelligence behind all these devices resides in a computational system that manages the data of all the sensors an analyzes it to get an idea on what is happening in the environment. Then, using some predefined rules or some instructions from a person the actuators react to a system command to do certain tasks. For example, if the temperature is rising, the system receives a notification from the sensor, the system processes that information and send a command to the air conditioning in order to turn it on, until the temperature lowers and it can be turned off. An AmI environment is a complex one in that there can be lots of different devices recollecting information and the system can control lots of different actuators in order to affect the environment.

Deploying all the infrastructure to create an AmI environment in a building can be a very complex and expensive task, depending on the desired objective, not only for the cost of the devices and the required communication devices, but also the time to select the optimal position of these devices and the testing necessary to check that everything works as expected. In this research paper we propose a tool that helps in this task. That tool would be a simulator that enables the study of the optimal position for placing the devices assuring that the system works as expected. The problem is that we have simulators that perform part of the work but not a simulator that covers all the cases. I.e. there are social simulators able to simulate the behavior and movement of the people inside a building for the social part of AmI, and there are network simulators capable of simulating communications and devices of a network for the communication part of AmI; however, there is no simulator that integrates both simulations and use the outcome from the two simulations. This really is a problem because it's impossible to perform simulations of AmI environments and therefore its design, development and deployment will be very costly, and also many problems will arise that were not taken into account after deploying the AmI environment.

Other valuable functionality that is obtained from joining both simulators and interconnecting them to an AmI real time environment is that the simulator could analyze real time data and predict certain events that are going to happen and act in consequence, trying to avoid them to happen or minimizing the possible damage. The simulation could also use previous data to search for a pattern before certain event happens and use machine learning techniques.

This paper is the continuation of our previous work [8], where we expand the architecture, the models, add the prediction of events and present the current status of the simulation; we also present a tool that integrates both social and network simulators in order to obtain a simulator that covers all information and that is necessary in an AmI environment. In chapter 2 we present the related work this paper is based on. In chapter 3 we show the architecture. Chapter 4 explains several models created for the simulator containing a data model and a sequence model. In chapter 5 we talk about the use of the simulator in a real time environment to predict certain events. Finally in chapters 6 and 7 we present the conclusions and some future work.

2 RELATED WORK

This section describes the related work with the tool that we present in the paper. These includes AmI environments and also the simulations, specifically both the social simulation and the network simulation.

[1] Technical University of Madrid, Av. Complutense 30, 28040 Madrid Spain, Telecommunications School - ETSIT, email: alvaro.spicot@gmail.com

2.1 Ambient Intelligence

AmI is a discipline that makes our everyday environments sensitive to what is happening with the use of sensors, actuators, the network that interconnects all these devices and the server that orchestrates all these elements [2]. The main objective of AmI is the improvement of people's life that use the environment. In order to achieve this objective the information generated in the sensors is recollected and processed in the server and then certain orders are sent to the actuators, based on the information gathered from the sensors. The actuators in the end will influence the people that are present in the environment, ideally not being conscious of the technology. In AmI environments we expect several features [3]:

- Sensitive: The system needs to base its decisions on what is happening in the environment reacting to the people in them.
- Adaptive: The system also requires to adjust its behavior depending of the situation, considering the best possible behavior, and ideally anticipating to an event.
- Transparent: The people in an AmI environment should not be conscious of the technology that surrounds them. Thanks to the miniaturization of the technology this is easily achieved nowadays.
- Ubiquitous: An idea behind AmI is that it requires being present in as many places as possible, ideally everywhere. In this way there is more data recollected, and the more information, the best the system can react to a particular event.
- Intelligent: The system works using AI in order to achieve its goals. This is done recollecting the data from the sensors, processing it, and giving orders to the actuators in order to, in the end, influence the environment, specially the people.

AmI is mainly used in home environments controlling the elements of the house such as the air conditioning, the watering of the plants or the security but it can also be extended to larger places such as an office or a cinema to control those elements but also to prevent certain catastrophes such as a fire or, should it happen, manage the evacuation as best as possible guiding the people to the quickest and safer exit [5].

2.2 Simulation

Simulation is the process of designing a model of a real or imagined system and conducting experiments with that model to achieve certain goal [7]. Simulations of a very simple environment can be done with a mathematical model but one that is slightly complex, requires the execution of the simulation in a computer, as there are too many variables to take into account in the mathematical model.

There are many different types of simulations, each aimed for a particular field but in the field of AmI, as there two very important variables people and devices, we are going to focus in two simulations, the social simulation and the network simulation.

Social simulation studies the interaction among social entities taking into account their psychology and their behavior, both between people and with the people and the environment [4]. There are two main types of social simulation, system level simulation that analyzes the situation as a whole and agent-based simulation where we model a person (the agent) and its own behavior, and the interaction between agents will result in the overall behavior. We will focus in these last one as its way of working is more adapted to an AmI environment.

There are different agent based Social Simulators (SS) such as MASON, Repast [1], Swarm, each with its own characteristics and usually particularized for a certain case study. Some of them work with a 2D environment while others have a 3D one. All of them include some kind of physical engine to calculate the collisions between the agents and the environment. These simulators work using steps, so that all the information is updated every step (some seconds defined during the initialization).

The SS specializes in the behavior of the human and it can simulate other elements in an AmI environment such as sensors or actuators but it won't be able to get an very deep simulation of those devices.

In a network simulation, a program models the behavior of a network and each entity present in it, as well as the messages sent between them [2]. It can also simulate in detail the behavior of the entities such as routers or computers.

There are several Network Simulators (NS) nowadays both open-source such as NS or OMNet++ and proprietary such as OPNET or NETSIM [6]. All are event driven, meaning they calculate the next event in the network, where an event could be, for example, sending or receiving a packet or a new device that enters the range of a wireless network. After the simulation ends they generate a log that contains all these events, useful for a future analysis of the network.

NS are very good at simulating the network in an AmI environment and can simulate the other elements in this environment, mainly the people, using specific algorithms for their movement but they can't do a very realistic simulation, specially in their behavior, such a SS would.

3 ARCHITECTURE

In this section we present the proposed architecture that integrates both simulators. In order to achieve this we need to solve certain problems that might arise during the interconnection of both simulators. We have identified the following ones:

- Initialization: Each simulator requires specific information in order to start the simulation. Much of the information is shared between the simulators but probably a different format is necessary. Still some of the information is only required to one of the simulator, for example all related to the behavior of the people is only required by the SS while the NS only needs the position of the people, but nothing more.
- Synchronization: There is a very serious problem with synchronization while the NS is synchronous SS is asynchronous. I.e. the network simulator is based on events, updating the simulation when something happens, while the social simulation is based on steps, updating the position of all the agents every certain time. This requires a special synchronization between both simulators so that events are converted to time and everything can work.
- Visualization: Both simulators have their own visualization module but we need a common one to use with the integration so that the user can operate the whole AmI simulator from a single graphic interface. This visualization will have to manage the information from both simulators.
- Management: Different parameters can be managed before the simulation start as shown in figure 8, so we can set the behavior of the different elements as well as the characteristics of the environment. This helps to run several simulations with different parameters and then analyze the differences in the results.
- Decomposition: Both simulators require different parameters so, we need to keep track of the whole system but we need a way to particularize the information to each simulator as each simulator has its own way of processing the data.

- Results: Once the simulation finishes we need a mechanism to store the data generated so that we can analyze it in the future and compare the results from different simulations.

In order to solve these problems we propose the creation of an engine that will integrate both simulators including also an interface for the interaction with the user and a database to store the information generated. We call this engine Hydra. The general overview of this architecture can be seen in figure 1.

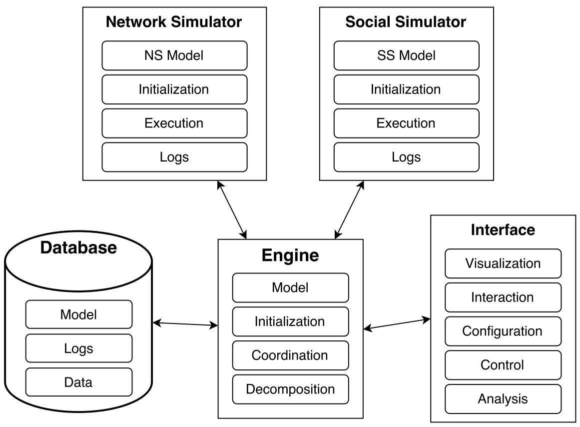


Figure 1. General Architecture

In this architecture we see both simulators communicating with Hydra. Hydra is going to integrate both simulators and is responsible of the following actions:

- The initialization of the simulation. Hydra has to send each simulator all the information it needs to start its own simulation. The user will be required to configure several parameters particular to each simulation specifying a condition that has to be met to end the simulation, such as a specific elapsed time or certain event.
- The synchronization of the simulations. After each step a simulator generates a new state of the elements in the simulation, and then Hydra needs to send the relevant information to the other simulator. For example, if after a step the SS updates the position of a person, this movement needs to be sent to the NS because it could imply the movement of the mobile phone this person is carrying and possibly it could enter or leave a wireless area.
- Ending the simulation. Once the ending condition of the simulation is reached as previously defined in the initialization or if the user manually ends it, Hydra recollects all the information that has been generated during the simulation in order to store it in the database so that it can be processed in the future. It also enables the user to view this information.

Hydra also works as the interface with the user allowing him to configure the initial parameters and to check the information generated once the simulation has finished. Hydra uses certain models presented in the next section and adapts them to each simulator following their requirements.

4 MODEL

In this section we present the different models associated with the architecture explained in the previous section. Firstly the data model used by the simulators which are executed by the engine. And finally several sequence diagrams that explain in detail the communications between both simulators in the different cases: when the simulation starts, when the SS needs to be executed, when the NS needs to be executed and finally when the simulation ends.

4.1 Data model

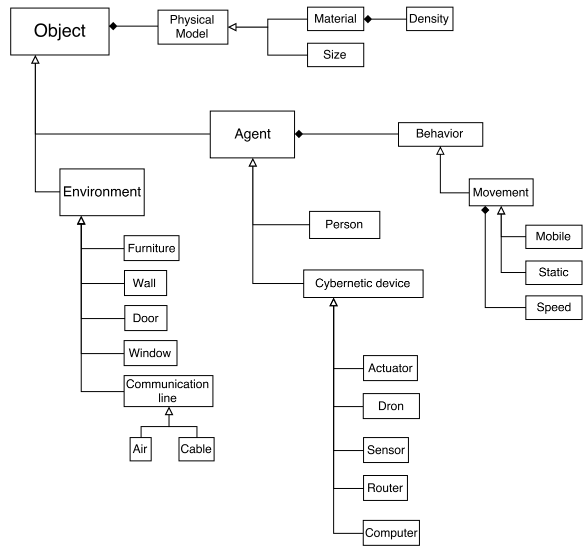


Figure 2. Network Simulator Model

In figure 2 we can see the data model used in the architecture previously presented by the NS.

This model is a general conceptualization of the different objects we can find in an AmI environment but oriented towards the network.

We have divided the object in two different fields. First there is the environment that includes all inert objects found in the defined space. Considering a closed space such as that of a building we can find in a room elements such as walls, doors and windows and inside these we can find different types of furniture as well. All these objects are general in an AmI environment but we can also find cyber-physical elements particularly important to the network such as communication lines.

Then we have the agents that represent all that requires certain intelligence in the simulation. It includes two subsequent groups, first the people that contains all the information relevant in the social simulation so that the NS will only be interested in their movement. Then there are also the cybernetic devices such as sensors, actuators, drones, etc.

All objects might possess certain relevant information in the physical model such as its size, weight or the material they have which might be interesting for the NS to check how the wireless communications propagate through the obstacles.

In figure 3 we can see the model specific to the SS. There are lots of elements shared with the NS model but some of them have disappeared as they are not relevant in a social simulation, such as communication lines, routers and computers. Instead there are some new elements such as everything related to the interaction between agents.

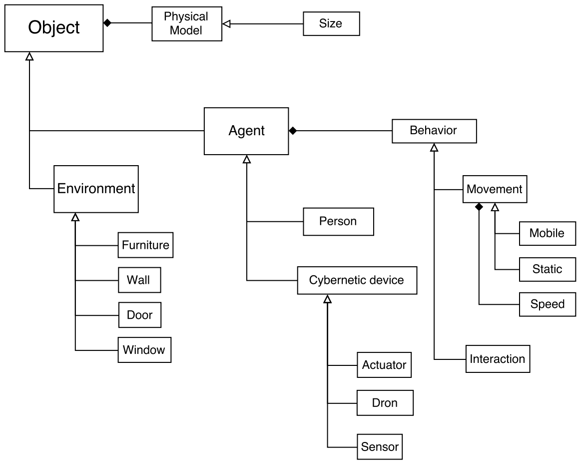


Figure 3. Social Simulator Model

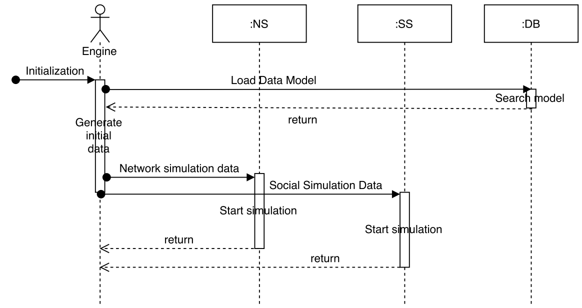


Figure 4. Initialization of the simulation

4.2 Sequence model

In this subsection we will see in detail how the communication between both simulators and Hydra is done and in particular the different tasks Hydra needs to execute in order to guarantee that the events are solved in the correct order and that both simulators have the information updated.

There are 4 sequence models that correspond to the four different states the simulation can be, and are later explained in detail. In order to understand these states we need to explain first how Hydra works with both simulators and some key aspect of them.

One of the main tasks of Hydra is to coordinate both simulators and keep track of the current state of the simulation. In order to achieve this, Hydra stores the close future events in a queue ordered by in simulation time, so that the first event in the queue is the next one to be executed. Each event contains information to which simulator does it belongs. Each step Hydra extracts the first event from the queue and informs the corresponding simulator to execute an step in its simulation. Once the simulation of the step finishes the simulator informs the engine, possibly with information about new events generated that are then added to the queue in order. Then Hydra can possibly send information to the other simulator so that it can update its state, and finally a new step starts. All these process is explained in detail later with the sequence models.

Once the user informs Hydra to start the simulation there are four possible situations. The first one is the proper initialization of the simulation where each simulator starts its own simulation The second and third one are during the simulation when different events are extracted from the queue and sent to the NS or the SS as corresponds. Finally the simulation ends when the queue is empty or when a certain condition predefined by the user is met and then the information related to all the simulation is generated, processed and stored in the database.

4.2.1 *Initialization of the simulation*

The initialization of the simulation happens once the user has configured the parameters of the simulation and starts the simulation, both visually or in batch. This process can be seen in figure 4.

First Hydra has to load the different models from the database, necessary in the selected scene. These models are then particularized with the configuration parameters selected by the user and by the scene so that the different elements of the simulation can be placed in its locations and behave as expected. The models are also particularized for each simulator as not both simulators are going to need the same information as explained in the data models in section 4.1.

Then the information from the models is sent to each simulator so that it can start its own version of the simulation. Each simulator will then configure the simulation with the parameters received from Hydra, and once the set up is done, they inform Hydra that the are ready to continue with the simulation when required. Each simulator also send to Hydra information about what are the next events. In the particular case of the SS the only next event is when the next step happens as defined by the user, but in the case of the NS these events can be new packets generated or systems booting up are any other possible event.

Once the first simulator finishes configuring its simulation, Hydra creates the queue where the events will be stored with the information it received from the simulator. Similarly, once the second simulator ends, Hydra will add the events to the queue.

Now the simulation is ready to start. The next step explains how Hydra processes the queue and continues with the simulation.

4.2.2 *Update of the social simulation*

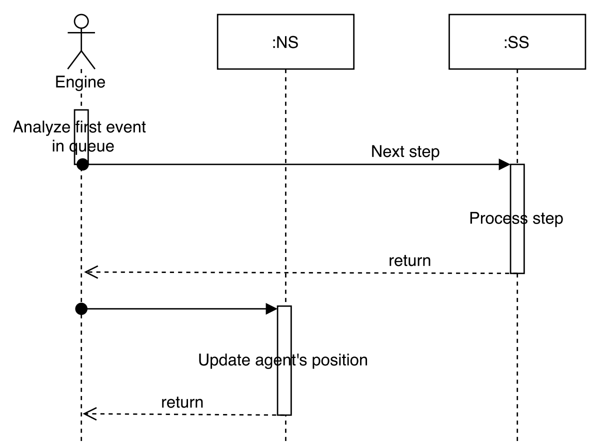


Figure 5. Update of the social simulation

Once the simulation is ready to start and then after each step is resolved, Hydra extracts the first element in the queue (so it is also the first event in chronological order), removing it from the queue and then processes it.

If the first element is one from the social simulator, Hydra sends a message to the SS informing it that it can simulate the next step. In figure 5 it is explained how does this process work.

Each update in the SS usually requires to update the position of all the agents in the simulation depending on the interaction between them and the surroundings. Once the simulation has been updated, the SS sends a message to Hydra to inform that the simulation of the step has finished but his message also contains information about the new positions of the elements in the simulation and any other information that may be relevant. This message also contains information about when the next event is going to happen in the time of the simulation.

Then the engine processes this information and adds the event of the next simulation to the queue in the chronological order that corresponds. Hydra also processes the updated positions from the agents and sends a message with the information that is relevant to the NS so that it can update its own simulation and it is synchronized with the social simulation. Not all the information might be relevant to the NS, for example, depending on the scenario the position of a person might not be interesting to the NS, but it is the position of its mobile phone the one that is important.

Once the NS confirms that it has updated the new positions of the agents, the step is completed and then Hydra checks if a condition to end the simulation has happened. This condition is defined by the user when configuring the scenario and could be, for example, a certain time of the simulation or a certain region that has to be empty or there could be even no ending condition, as for example when running a visual simulation. In this case, the user will have to manually stop the simulation when he desires.

Now Hydra will extract the next event in the queue and continue with the simulation.

4.2.3 *Update of the network simulation*

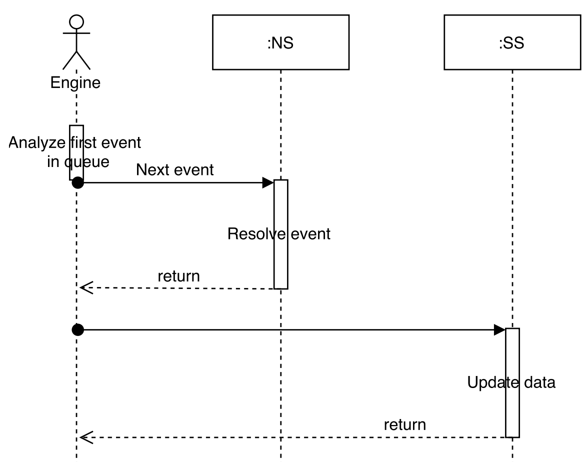


Figure 6. Update of the network simulation

Similarly to the previous case, once the simulation is ready to start or when a new step begins, Hydra extracts the first element in the queue (so it is also the first event in chronological order), removing it from the queue and then processes it.

If the first element is one from the network simulator, Hydra sends a message to the NS informing it that it can simulate the next event. In figure 6 it is explained how does this process work.

Each update in the NS requires to execute certain event such as a packet that arrives to a router and needs to be processed or a user that moves within range of a Wi-Fi. Once this event is resolved the NS informs Hydra that the update is complete. In the process of solving the event, new events might have been generated with a time-stamp in them. This events are sent to Hydra within the message informing the conclusion of the update.

When Hydra receives the message it adds the new events to the queue, if any, and then parses the information to send the one is relevant to the SS. Similarly to the previous case, not all the information will be relevant to the SS but some might. For example if a mobile phone has lost its signal the SS needs to know this information because the person could react to the event.

Once the SS confirms that it has updated the new information, the step is completed and then Hydra checks if a condition to end the simulation has happened as explained at the end of section 4.2.3.

Now Hydra will extract the next event in the queue and continue with the simulation.

4.2.4 *End of the simulation*

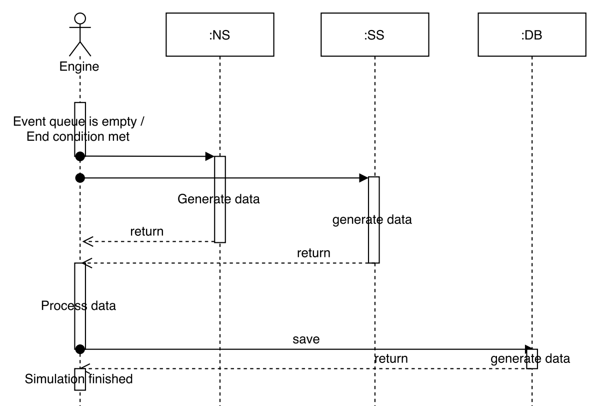


Figure 7. End of the simulation

Once the simulation reaches the end, Hydra has to do certain tasks to store all the information relevant to the whole simulation as explained in figure 7.

First Hydra informs both simulators to finish their simulation. Each simulator end the simulation but also process all the data that has been generated during the simulation and sends it to Hydra. In the case of the SS this information will be the evolution of the position of the elements in the simulation as well as certain parameters that might have changed. In the case of the NS this will be a file with all the packets that have been sent and its content as well as other parameters that might be relevant.

Once Hydra has the information from both simulators, it proceeds to store it in the database including all the parameters selected initially by the user as well as some other information as the date or the duration of the simulation.

Now the simulation has finished. If the user run a batch simulation, then a new simulation might start or if it was the last one then Hydra pauses and waits for new user input. If the user run a visual simulation, now there is a new management screen where he can analyze any simulation and compare them.

5 PREDICTION OF EVENTS

A simulation of an AmI environment should not only be used when designing the real AmI environment testing where the devices should be placed, but it could also be very helpful once the system is ready and the devices are installed in the environment. The idea is that the simulation can use the data from the devices in real time and use this information to predict future events that may cause the system to malfunction or something dangerous that could happen. A simulation is run with the data obtained in the present searching for certain patterns previously defined such as for example a great concentration of people in a small area. Should the simulation find this pattern the system tries to react in order to avoid it, for example indicating the people to abandon the area or, if necessary, informing a supervisor. Each time lapse a new simulation is run with the current data. This time lapse could be shorter or longer as required depending on how fast is going to change the data.

In order for this prediction to work the simulation should run fast enough so that when the data is processed, not enough time has been elapsed and the event has not yet happened so that some measures can be taken to prevent it.

The simulation can also use data from the past in order to predict these events searching for certain patterns that can cause them.

6 PROPOSAL OF VALIDATION

We are currently working in the validation of the architecture, creating Hydra to coordinate both simulators and an interface on each Simulation to do as intermediate between Hydra and the proper simulator.

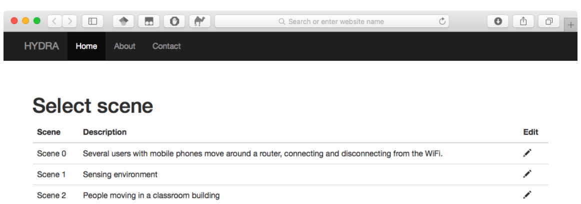


Figure 8. Scenario selection

Once Hydra and the simulators are started, the user can access the graphic interface using a browser. The first screen the user sees is a table with the scenarios as can be seen in figure 8. Currently three scenarios appear with a small description but ideally the user could create a new one or edit one that exists. In order to create a new scenario, a new screen would appear enabling the user to drag and drop different elements predefined to create the desired scenario and then add the configurable parameters.

Once the user selects a scenario, a new screen appears as can be seen in figure 9. The selected scenario is one with a router and different users with mobile phones walking around the router so that their

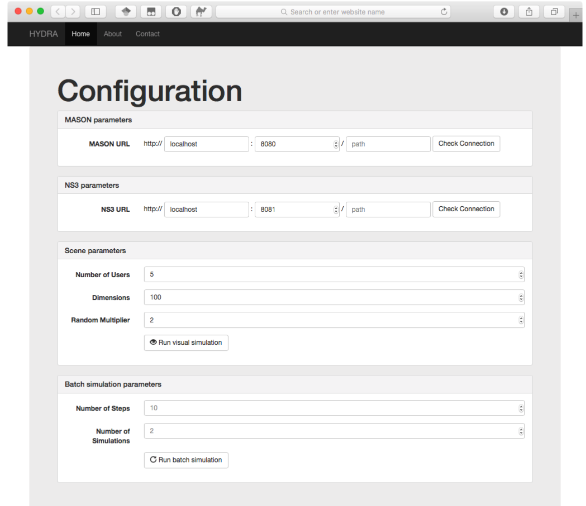


Figure 9. Configuration of the scenario

phones connect and disconnect from the network. The configuration screen enables the user to change the IP and port where the simulators are running as well as configure certain parameters proper to the simulation selected. In this case the user can choose how many people are in the simulation, the size of the area where the people can move and a parameter to define the movement of the people. Once the parameters are configured, the user can click to run a visual simulation what will take him to the next screen.

Alternatively, once the scene parameters are configured, the user can, rather than running a visual simulation, select to run a batch simulation. In this case the user can select how many simulations to run and each one, after how many steps are stopped. In this case the simulations will run in background and once finished the user will be taken to a screen where he can analyze the data generated in the simulations.

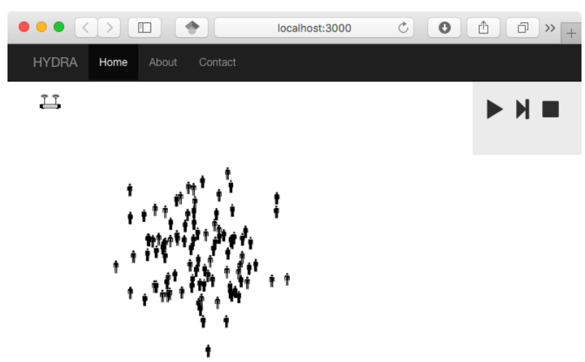


Figure 10. Running the simulation

Once a visual simulation starts, the user has control of the flow of the simulation, being able to run a step by step simulation or running the simulation as fast as possible. The user can pause the simulation and click in any of the elements in the screen to access its information. Once the user decides to stop the simulation, he is then taken to a screen where the data of the simulation appears.

7 CONCLUSIONS

In this paper we present an integration of a social simulation and a network simulator in order to get an enhanced AmI environment simulator that can precisely simulate the whole environment. A simulator for Ambient Intelligence environments is very useful due to they can be tested before being developed and deployed and checking if it's feasible; as a result these environments can be designed, developed and deployed more efficiently and effectively. Several difficulties have arisen during the development of this research work and we gave them solution with the proposal of an engine that integrates and coordinates and orchestrates both simulators. This engine is responsible of the initialization and coordination of both simulators keeping track of the different events that happen and the finalization of the simulation, storing the data generated. Apart from the engine there is also a visualization element that allows the user to follow the simulation as it advances and the inspection of the data generated, enabling him to check if everything worked as expected and comparing this data with one from a previous simulation. It also allows the user to configure the different parameters before starting the simulation and also configure an execution of a batch of simulations.

8 FUTURE WORK

The architecture presented in this paper enables several improvements. Here we comment some of them.

The most important task is the realization of a validation of the architecture. This would include a deployment of a simulated scenario containing several people and cyber-physical devices and the comparison of the data obtained in the real environment with that obtained in the simulation. We are currently working on this, but there is still a lot of work to do.

The scenarios we are working with are very basic and are created by hand. The user should be able to create its own scenario adding graphically the elements he wants, from those defined in the model, and configuring their parameters or being able to define the ones that can be configured later, just before the execution.

During the simulation the user should be able to modify the simulation on the run, so that he can experiment with new changes in the simulation, like moving certain agents, or adding or removing new ones. This would enrich the visual simulation so that it is not just a visual representation of the batch simulation.

Finally the screen that enables the user to analyze the simulation should be the most important one because this is why the user runs a simulation in the first instance. This screen currently shows the logs sent by each simulator, but it should present the information in a more visual way, enabling the user to see the simulation in each step as well as compare it with other simulations previously run. It should also enable the user to filter the parameters he is interested in.

Another idea for the simulator is the inclusion of different simulators in the engine. We have only included a social simulator and a network simulator but several others could be added depending on the scenario simulated. For example, a fire simulator, that precisely simulates the advancement of a fire inside a building, could be included in order to improve a simulation to test the evacuation time of a building. Other ideas could be a weather simulator a day and night simulator that can influence in the behavior of both the people and the cyber-physical devices in the simulation.

Finally, another possible future work could be the distribution of the simulation so that each component runs in one or several machines and the data is shared between all. This could be really important when the simulation works with the real environment firstly because the great quantity of data that it can process but also, should some of the machines stop working, the simulation could keep working with other machines continuing the work of the one that failed.

ACKNOWLEDGEMENTS

This work has been partially supported by the Autonomous Region of Madrid through program MOSI-AGIL-CM (grant P2013/ICE-3019, co-funded by EU Structural Funds FSE and FEDER) and has also received funding from the Ministry of Economy and Competitiveness through SEMOLA project (TEC2015-68284- R).

REFERENCES

[1] Robert John Allan, 'Survey of agent based modelling and simulation tools', Technical report, (2009).

[2] Lee Breslau, Deborah Estrin, Haobo Yu, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, et al., 'Advances in network simulation', *Computer*, (5), 59–67, (2000).

[3] Diane J Cook, Juan C Augusto, and Vikramaditya R Jakkula, 'Ambient intelligence: Technologies, applications, and opportunities', *Pervasive and Mobile Computing*, **5**(4), 277–298, (2009).

[4] Paul Davidsson, 'Agent based social simulation: A computer science view', *Journal of artificial societies and social simulation*, **5**(1), (2002).

[5] Augusto Morales, Ramon Alcarria, Diego Martin, and Tomas Robles, 'Enhancing evacuation plans with a situation awareness system based on end-user knowledge provision', *Sensors*, **14**(6), 11153–11178, (2014).

[6] Saba Siraj, A Gupta, and R Badgujar, 'Network simulation tools survey', *International Journal of Advanced Research in Computer and Communication Engineering*, **1**(4), 199–206, (2012).

[7] John A Sokolowski and Catherine M Banks, *Principles of modeling and simulation: a multidisciplinary approach*, John Wiley & Sons, 2011.

[8] Álvaro Sánchez-Picot, Diego Martín, Diego Sánchez de Rivera, Borja Bordel, and Tomás Robles, 'Modeling and simulation of interactions among people and devices in ambient intelligence environments', in *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 784–789. IEEE, (2016).

Author Index