Journal of Ambient Intelligence and Smart Environments 1 (2016) 1–5
IOS Press

AmI environments simulations approach integrating social and network aspects: A case study

Álvaro Sánchez-Picot [a,*], Diego Martín [a], Borja Bordel [a] and Diego Sánchez-de-Rivera [a]
[a] *ETSI Telecomunicación, Technical University of Madrid, Av. Complutense 30, 28040, Madrid, Spain*
E-mails: alvaro.spicot@gmail.com, diego.martin.de.andres@upm.es, bbordel@dit.upm.es, diegosanchez@dit.upm.es

Abstract. Each day more and more new technological devices appear in our world. We want to have everything connected so that we can monitor different aspects inside a city such as the traffic, or inside a building such the position of the people. Even nearly everyone now owns a mobile phone that carries with himself all the time and that enables them to receive information of that type in real time. The problem with such infrastructures is how expensive it is to install and maintain an ambient intelligence environment, so a tool to help in this task would be really appreciated. In this paper we propose the creation of such tool as an AmI simulator that integrates both a network and a social simulator. The authors performed a case study where four experts and twenty-one learners were involved in order to analyze the qualitative characteristics of an AmI simulator, studied whether a simulator can draw more conclusions than the execution of a social and network simulator separately; and finally studied the user satisfaction with an AmI simulator.

Keywords: Simulation, Social simulation, Network simulation, Ambient Intelligence, Case study

1. Introduction

We are heading towards a technologically connected world where more and more devices are being installed in the cities, in the buildings and inside our homes. Some of these devices are nothing new, such as smart televisions, air conditioning units, security cameras or smart fridges. In most cases they are devices we had before but that now are connected to the Internet or interconnected with other devices. Others are relatively new such as ambient lights or microphones to enable the interaction with a computer to control the house. Nowadays we want to have much more information on what happens in our home and our surroundings than several years ago, and thanks to the mobile phones nearly everybody owns, we can easily access this information anywhere and in real time. We also want the environment to act proactively depending on the events that happen, for example, that lights turn on automatically when the nightfall comes and turn off when there is nobody present. Another example is opening and closing the blinds depending on the outside light, the time of the day or the desired temperature inside the building. All this corresponds to what is known as an Ambient Intelligence (AmI) environment, this means a system sensitive and responsive to the presence of people and environmental factors [1].

The idea of an AmI environment is that all the different devices cooperate in order to obtain a desired result and to monitor the environment checking everything works as expected [2]. The intelligence behind all these devices resides in a computational system, such as a smart application, that manages the data that all the sensors generate and process it to check what events are happening in the environment. After the analysis, using some predefined rules or some in-

*Corresponding author. E-mail: alvaro.spicot@gmail.com.

structions from a person, the actuators react to a system command to do certain tasks if required. An example could be a rise in the temperature received from a sensor that after the system processes the information then sends a command to the air conditioning in order to turn it on, until the temperature lowers enough and then turns it off. An AmI environment is a complex one where there can be a huge amount of different devices recollecting different type of information and the system can control several different actuators in order to affect the environment using some predefined intelligence and learning from previous data it has gathered [3].

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 sensors, the actuators, the required communication devices and the installation of these, but also the time involved in selecting the optimal position of these devices and the test necessary to check that everything works as expected. In this research paper we propose a tool that helps in this task. This tool is a simulator that enables the study of a design in an AmI environment to check that everything works as expected using different test cases, some of them nearly impossible to test in a real environment, such as the evacuation of a building. The problem we find is that there are simulators that perform part of the work, but there is not a tool that covers all the cases present in an AmI environment. I.e. there are social simulators able to simulate the behavior and movement of the people inside a building that covers the social aspect of AmI environments [4][5], and there are network simulators capable of simulating communications and devices of a network that covers the communication aspect of AmI environments [6][7]; however, there is no simulator that integrates both simulations. This is a problem because it is very hard 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 elements in the AmI environment. Other use cases of an AmI simulator could be the selection of a proper technology (Bluetooth, Zigbee or Thread for example) when deploying devices in an AmI environment. Another use case could be an analysis of the coverage of WiFi in a conference building, where one wants to know where to place the access points based on the areas more crowded and the movement of the people.

This paper is the continuation of our previous work [8] that explains the integration of an existing NS (network simulator) and an existing SS (social simulator) to create an AmI environment simulator or a co-simulator we have named *Hydra*. In that paper we presented the architecture and the models necessary to integrate both simulators and in this paper we expand the idea justifying the necessity of an AmI simulator comparing the different features of the existing social and network simulators and based on those features we present the model and the architecture that supports our AmI simulator and finally showing the status of the tool and a validation. The main advancements in the tool from our previous work is the inclusion of 3 new scenarios that have been used in the validation of this paper.

The authors carried out a case study where experts in social and network simulations were involved with the analysis and study about the characteristics of simulations and obtaining conclusions during the execution of three different scenarios using firstly a social and a network simulator separately and then using *Hydra*, the tool proposed in this research work. In order to approach this research objectives the authors propose the following research questions that will guide our work through the paper.

- RQ1: Does the network and social co-simulation offer more features than the simple execution of each separately?
- RQ2: Does the network and social co-simulation help in the analysis and drawn of conclusions?
- RQ3: What is the perception in the use of the co-simulator?

This paper is structured as follows: in section 2 we present the related work and the requirements of an AmI simulator; in section 3 we show the proposal overview explaining the architecture and the model; in section 4 we present the validation done and section 5 explains the results obtained; finally sections 6 and 7 present the conclusions of this work and some ideas for future work.

2. Related work

In this section we present the related work of the key elements in the paper related to the different simulations required in an AmI simulation.

2.1. Social 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 (called the agent) and its own behavior, and the interaction between agents will result in the overall behavior. We will focus in the later one as its way of working is more adapted to an AmI environment.

There are different agent based Social Simulators (SS) such as MASON[9], Repast [10], Swarm[11], each one with its own characteristics and usually focused 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, what usually means that the position of all the agents is updated every step.

The SS specializes in the behavior of the human and it can simulate other elements found in an AmI environment, such as sensors or actuators, but it will not be able to do a very detailed simulation of those devices.

2.2. Network simulation

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 [7] taking into account the delay, the jitter or packets that can be lost . 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 NS3 [12] or OMNet++[13] and proprietary such as OPNET[14] or NETSIM [6]. These simulators are event driven, so that they have a queue where they store the future events that are going to happen in the network. An example of an event can be a package leaving or arriving at a router or a new device that connects to the network. The first event in the queue is processed what can also generate new events that are added in order to the queue, and then the new first event is processed. This process continues until a specific condition is met or the user aborts the simulation. Once the simulation ends they generate a log that contains all these events, useful for a later analysis of the network.

NS are very good at simulating the events in a network in an AmI environment and could also simulate the other elements in this environment, such as the people, using specific algorithms for their movement, but they cannot do a very realistic simulation that could include their behavior, such a SS would.

2.3. Proposal overview

In our previous work [8] we have proposed an AmI simulator named Hydra that integrates both a network simulator (NS3) and a social simulator (Mason). The creation of an AmI simulator starting from scratch is beyond our possibilities due to the huge amount of time and people involved that this would require.

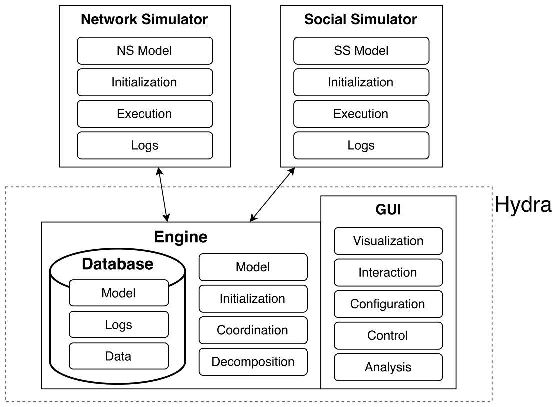

Fig. 1. General Architecture

The core of Hydra is an engine that coordinates both simulators and a graphical user interface that enables the user to visually configure and run the simulation. This can be seen in figure 1. When the simulation starts, the engine informs both simulators to start their own version of the simulation sending only the data they require. Once both NS3 and Mason have set up the simulation, the engine starts informing one simulator to run a fraction of the simulation and then stop. The engine keeps alternating between both simulators until the simulation reaches a user defined time. Finally the engine informs both simulators to end their simulation and generate all the relevant data that will be later presented to the user for a post analysis of the simulation. This process can be seen in figure 2.

A more in depth information of how Hydra works, the models used, the interconnection between both the NS3 and Mason, and the different messages send between the engine and the simulators can be found in our previous work [8].

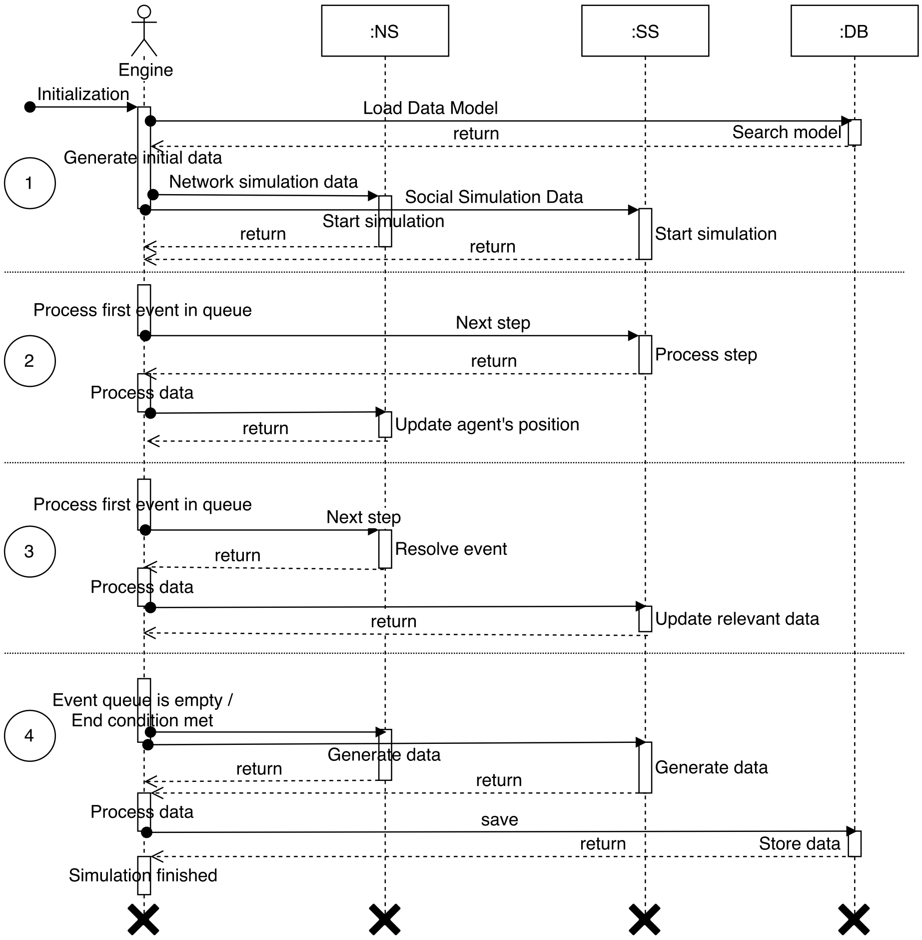

Fig. 2. Data Model

Since our previous work we have improved the tool defining and programming 3 new scenarios detailed in section 4. These scenarios are the evacuation of a building, the Wi-Fi coverage inside a building and the end of classes in a university building. In addition to this update of the tool, this paper improves our previous work analyzing the requirements of an AmI simulation (extracting and categorizing features from existing Network and Social simulators and comparing them with our tool) and finally doing a validation of the tool with both learners and experts in simulation and AmI where we try to answer the research questions proposed in section 1.

3. Requirements of an AmI simulator

In this section we are going to analyze the AmI environment to define the features that characterize a simulator for AmI environments. Because we understand that AmI environments integrate the social simulation with the simulation of the communications we are going to analyze the features of the AmI environments as a combination of social and network simulators with the objective of creating a prototype of AmI simulator using features from existing network simulators and social simulators complementing them with features none of them is able to offer.

For our co-simulator we have chosen *MASON* [9] as the Social Simulator and *NS3* [12] for our Network Simulator [10] [6]. The reasons to choose these simulators over others are because they are free, widely extended, currently supported and we have experience working with both of them. The features presented in this section come mainly from these simulators.

3.1. Features

In this subsection we are going to list the different features of the actual social and network simulators analyzing what features does each simulator support [15] [16] [17] [18] [19]. These features can be divided in model, execution and analysis.

We have identified the following **model features** associated with the social and network simulators. These features group everything related to the modeling of the different elements in a simulation such as people, devices or other elements in the environment; as well as other features that govern these elements or the environment such as the behavior, the mobility or the physics. All these features are related with everything prior to the execution of the simulation. Their explanation is as follows:

- **Environment element models:** Social simulators have different models for elements in an AmI environment such as walls or furniture. Network simulators can also model these elements and others that can affect the behavior of the devices such as temperature or humidity [20]. Network simulators also take into account these elements in the signal propagation models.
- **Mobility models:** Social simulators have different mobility models based on the grid and usually for both the 2D and the 3D environments. Network simulators also have access to mobility models from the random movement to the predefined one.
- **People behavior models:** Social simulators have different models to how the people behave in the environment. These behavior includes the movement and the interaction with other agents or with the environment.
- **Group behavior models:** Social simulators have models for the behavior of a group so that people's decisions are different when they are alone and when they are surrounded by others.
- **Human interaction models:** Social simulators have models that control the interaction between different people affecting their behavior.
- **Physics modeling:** Some social simulators need models to manage the physics in the environment in order to calculate collisions between people and between people and the environment. These might be useful in highly populated areas and when studying evacuations or other similar scenarios.
- **Network protocol models:** Network Simulators contain all different protocol models both wired such as IP, TCP or UDP and wireless such as Wi-Fi, Bluetooth or Wimax. Social simulators have nothing related with network protocols and our AmI simulator, using the network simulator models also have those.
- **Network device models:** Network Simulator have models for common network devices such as routers but have the tools to create more complex devices using simpler modules.

We have identified the following **execution features** associated with the social and network simulators. These features contain everything related with the proper execution of the simulation such as synchronization, modifications or integration with other systems; as well as with its initialization such as the creation and configuration of the scene. Their explanation is as follows:

- **Preconfigured scenes:** The social simulators have different scenes preconfigured to work with such as for example an ant colony or a flock of birds in the *MASON*. Network simulators also have preconfigured scenes usually to check how different network protocols work.
- **Parameters configuration in scenes:** Social simulators enable certain variability in the scenes enabling the modification of specific parameters, without the need of recompiling the code, such as the number of elements in the simulation or a variable defining the movement of these elements. Similarly Network simulators enable the modification of some variables such as the number of devices or the size of the packets sent.
- **Scene design tool:** The scene in a Social simulator is created programatically where the people and the environment are created via code. Similarly all the devices and connections between them in a network simulator are programmed in the code. It is complicated to have a complete visual tool that can help in the creation of the scenarios because the elements an the relations can be very complex, but a basic tool should at least enable to draw the environment and place some of the elements in it.
- **Real-time execution:** Social simulators enable the execution in real-time being able to pause and resume the simulation or select the execution speed. Network simulators usually execute the whole simulation at once, but as they can be inte-

grated with real-time systems, they can be modified to enable real-time execution and the possibility of pause and resume the simulation, but this is not usually a default behavior.
- **Batch execution:** In simulation one execution is not usually relevant but rather what matters is the execution of several simulations and the after analysis of the data generated. Social simulators enable the execution of batch simulations via command line with the possibility to modify some of the parameters during the execution. Similarly Network simulators enable the execution of several simulations via command line each one generating its own report for a later analysis.
- **Real-time modification:** Social simulators enable the modification of some very specific parameters, but usually most of the parameters can only be modified before the beginning of the simulation. As network simulators are not designed with the idea of a real-time execution they don't enable by default the possibility of the modification of some of the parameters, but as they are integrated with real-time systems, they can be easily modified to enable these changes.
- **Integration with Real-time systems:** Social simulators are not usually designed with the integration of a real-time system. Otherwise, Network simulators, are designed with the possibility to be integrated in a real-time environment so that some of the data comes from real entities rather than from the simulation.
- **Discrete-event simulator:** Network simulators are discrete-event where each event represents something happening in the network such as a packet being sent, a disconnection of a device or a reboot of another. The execution of an event may create new future events (for example when a packet arrives to router it might be sent to another one). Social simulators are not discrete-event, but rather are synchronous.
- **Synchronous simulator:** Social simulators are synchronous where all the agents in the simulation are updated every certain amount of time (a second for example). Network simulators are asynchronous, they are discrete-event.

In the features analyzed we observe some important one missing. Specifically we consider a tool to help design the scenes a good addition to the simulator. Currently the scenes need to be defined in the code in both simulators but a visual tool could with the possibility to add several predefined elements to the scene.

We have identified the following **analysis features** associated with the social and network simulators. These features correspond with all the data analysis the simulation has generated and can be seen in different ways such as visually or in text. Their explanation is as follows:

- **Real-time analysis:** Social simulators enable the possibility of an analysis in real-time: visually one gets access to the evolution of the position of different elements during the simulation and the inspection of these elements for further information. Network simulators enable a post simulation analysis of the data generated but not a real-time one.
- **Post execution analysis:** Social simulators enable a post execution analysis usually checking the logs generated during the simulation. Network simulators generate a file after the simulation ends that enable a port-visual analysis of the execution of the simulation.
- **Visual analysis:** Social simulators enable a visual analysis during the execution of the simulation but usually not afterwards. Network simulators enable a post execution visual analysis of the simulation.
- **Text logs:** Both Network and Social simulators enable the access of the logs generated during the simulation. *NS3* can generate pcap trace files, that are files that contain the packages sent in the Network and can be analyzed using external network protocol analyzers such as *Wireshark* [21].

In the analysis features listed above we observe an important one missing. The final objective behind a simulation is to run several (tens or hundreds) simulations and then analyze the data generated to extract some conclusions. Both simulation have the possibility to run batch simulations but require external tools to analyze the huge quantity of data generated. Some simple tool that enables a quick analysis of the data could result very useful, some examples are the analysis of a variable during all the simulations calculating its mean and median drawing some diagrams with the values or the possibility to follow the different paths a person has taken.

3.2. *Features in Hydra*

In this section we are going to check the different features of the actual social and network simulators presented in the previous subsection and analyze how does our AmI Simulator, Hydra, supports those features and some other not supported by either simulator. As explained in the previous subsection these features are divided in three categories: model, execution and analysis.

Table 1
Simulators model features

Model Feature	**SS**	**NS**	**Hydra**
Environment modeling	✓	✓	✓
Mobility modeling	✓	✓	✓
Individual behavior modeling	✓	×	✓
Group behavior modeling	✓	×	✓
Human interaction modeling	✓	×	✓
Physics modeling	✓	×	✓
Network protocol modeling	×	✓	✓
Network device modeling	×	✓	✓

Hydra inherits the different models from both the network and the social simulators. Everything related to the people and its behavior is taken from the social simulator, this includes the individual behavior model, the group behavior model, the human behavior model and the physics model. Everything related to the network is taken from the Network simulator, this includes the Network protocol model and the network device model. The environment model is inherited from both simulators while the mobility model, despite both simulators have it, is taken from the mason model because it is usually the people who will move around in the environment.

All the conclusions from the model features presented here can be seen in table 1 where the features supported by each simulator are checked. It might seem obvious that the AmI simulator should have at least all the features from both simulators but this shouldn't necessary be true.

Our Ami simulator have several different scenarios preconfigured. We define a scenario as an environment, usually composed of several rooms, the people inside this environment, their behavior, a possible special event that can happen such as the start of a fire and all the rest of the elements presented in an AmI environment such as sensors and actuators. The first preconfigured scenario is a very basic test with people with mobile devices moving around a router and connecting and disconnecting form the Wi-Fi; the second one is a section of a University with different sensors and actuators and explained in more detail in section 4.

Table 2
Simulators execution features

Execution Feature	**SS**	**NS**	**Hydra**
Preconfigured scenes	✓	✓	✓
Parameters configuration in scenes	✓	✓	✓
Scene design tool	×	×	✓
Real-time execution	✓	×	✓
Batch execution	✓	×	✓
Real-time modification	×	✓	✓
Integration with Real-time systems	×	✓	✓
Discrete-event simulator	×	✓	✓
Synchronous simulator	✓	×	×

The scenes also have some parameters configurable to tune the simulation as required similarly to other simulators as explained in the previous subsection. It also enables real-time execution of a visual simulation with the capability to pause and resume and, once paused the elements of the simulation can be changed. There is also the possibility to run a batch simulation. The AmI simulator runs as a discrete-event simulator where the events are queued, event the periodic ones generated by the social simulator.

We are currently working with the integration of the AmI simulator in a real-time environment so that it can influence real actuators and can get data from real sensors and we are also designing the tool to visually configure a scene, as commented in the previous subsection.

All the conclusions from the execution features presented here can be seen in table 2 where the features supported by each simulator are checked.

Table 3
Simulators analysis features

Analysis Feature	**SS**	**NS**	**Hydra**
Real-time analysis	✓	×	✓
Post execution analysis	✓	✓	✓
Visual analysis	✓	✓	✓
Text logs	✓	✓	✓
Batch data analysis	×	×	✓

Our AmI simulator enables the inspection of the different elements during the visual simulation in real time. An element in the screen can be selected and a overlay appears that shows information related to that element. Once a simulation ends a new screen appears that lets the user check the logs generated during the simulation enabling him to search and filter for a keyword. Also as commented in the previous section we are working on a tool to enable the analysis of multiple simulations at once.

All the conclusions from the analysis features presented here can be seen in table 3 where the features supported by each simulator are checked.

4. Validation

In order to define a case study that guides this work, we reintroduce the proposed research questions:

1. RQ1: Does the network and social co-simulation offer more features than the simple execution of each one separately?
2. RQ2: Does the network and social co-simulation help in the analysis and drawn of conclusions?
3. RQ3: What is the perception in the use of the co-simulator?

We have carried out a case study in order to address these questions. During the experimentation step where four experts evaluated the use of *Hydra* and the execution of *Mason* and *NS3* separately analyzing the use of these simulators in three specific scenarios. In addition, twenty one students from an "Ambient intelligence" course filled in a survey about the satisfaction using Hydra.

With this approach we analyzed a set of events triggered as a consequence of users' actions (both experts and students) with the *Hydra* co-simulator in a real-life setting in order to identify the factors that influence the simulations. The researchers that carried out the case study had little control of the participants using *Hydra* because they had only trained the participants in the use of *Mason* and *NS3* and solved their problems during its use. This approach is appropriate to replicate the experiment in similar contexts.

4.1. Context

The case study was designed, planned, guided, monitored and evaluated by the authors of this work (hereafter, experts) who have more than five years of experience in network and social simulation, cloud computing, etc.

To answer the first research question, experts qualitatively analyzed the characteristics of Hydra. In this study they modeled three AmI scenarios in *Hydra*:

– Scenario 1: The evacuation of a building. (This scenario is based on our previous work published in a research paper where an AmI evacuation system based on dynamic routes is proposed and then evaluated [22]).
– Scenario 2: The Wi-Fi coverage of a building and the movement of people around the building searching for connectivity with their mobile phones.
– Scenario 3: In a university school, the exit from class and the saturation of the corridors in the building at that time of the day.

To answer the second research question the experts analyzed the number of conclusions drawn from the execution of simulations of the three scenarios described above. We define a conclusion as something relevant extracted from the simulation in the expert's opinion; for example in the scenario 2 some conclusions could be the high concentration of people in some rooms or another conclusion could be the lack of Wi-Fi coverage in some areas. Two types of simulations were conducted: the first one executing the simulators separately (*Mason* and *NS3*) and the second one using *Hydra*.

The third research question was answered with the participation of twenty one people (hereafter, learners) at the Technical University of Madrid that carried out the case study proposed in this research work. They are graduates, mostly in telecommunications engineering, and Master students. They were collaborating with the GISAI group at Technical University of Madrid in the area of a subject of the Master related with Ambient Intelligence in order to test *Hydra*. These learners had to model some AmI scenarios using the approach proposed in this paper. They were surveyed by the experts in order to evaluate the satisfaction with the use of *Hydra*.

4.2. Plan

The case study was splitted in two experimentation tracks: the first track involved the experts and the second one the learners (see figure 3). Each track had very similar phases but with different objectives. These phases were:

Training phase: The first phase consisted in training the learners in the principles of social and network simulations and its benefits; they also were trained in the use of the *Hydra* tool and its capabilities. The training sessions consisted of 2 hour of theory at the beginning of the experimental validation with personal interviews to solve specific problems found.

Simulation execution: This phase was carried out by both groups, experts and learners but with different objectives for each one. Both groups modeled three different AmI scenarios. Experts used Hydra and then Mason with NS3 to model the three scenarios, while learners only used Hydra. The difference between the groups consists on the experts discussing the differences between using Hydra against Mason and NS3; while trainees only study the subjective perception of use and satisfaction of using only Hydra.

Evaluation phase: At this phase all the scenarios modeled and its conclusions were assessed and evaluated by the experts as well as the surveys performed by the learners.

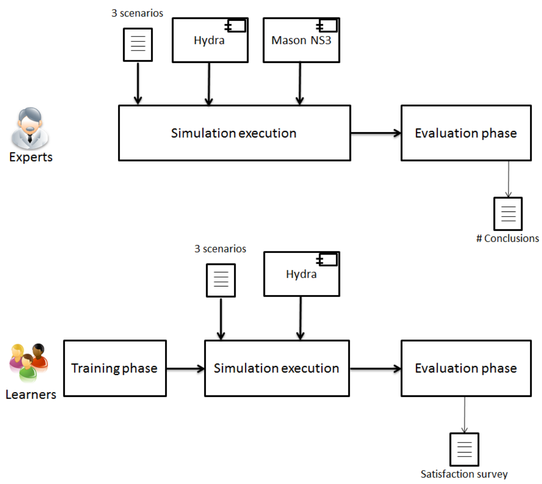

Fig. 3. Case study plan with inputs and outputs

4.3. Data gathering

All data was obtained during the evaluation phase presented in the previous section. On the track performed by experts an analysis of the qualitative characteristics of the proposed approach in this article was analyzed in order to answer the first research question; checking whether our proposal really meets the requirements presented in the subsection 3. In order to answer the second research question the experts also analyzed the number of conclusions obtained after executing three simulations (one per each scenario proposed) using both sets of simulators: *Mason* with *NS3* and then *Hydra*. In this evaluation, if the same conclusion was obtained by analyzing the results of Mason and NS3, it only counted as a single conclusion. In order to avoid learning among the experts, scenarios and conclusions; each scenario was simulated by two experts using *Hydra* and two different experts using *Mason* and *NS3* separately.

For this research work a conclusion is considered as a judgment obtained after the evaluation of the results (or partial results) proposed by the execution of a simulation; and a judgment is considered as a relevant fact for the scenario executed in a simulator.

On the track performed solely by learners, they were surveyed about the use of Hydra and its personal subjective satisfaction with its use; they were also asked about they subjective perception about the execution steps of Hydra presented in our previous work [8]. They were surveyed using a five-level Likert scale for both questions.

5. Results

This section presents and discusses the results obtained in the case study and will be presented according to the research questions proposed in this paper in section 1. In the first subsection an analysis performed by the experts about the qualitative features of the tool presented in this research work is presented. The next section shows the conclusions obtained by the execution the two simulators (*Mason* and *NS3*) separately and the approach proposed in this research in order to test whether our proposal discovers more relevant conclusions for the execution of AmI scenarios. And finally, a study of users' satisfaction using the Hydra tool is presented showing the overall assessment of the tool and a evaluation of the steps proposed by our approach for simulating AmI scenarios.

5.1. Qualitative analysis of the characteristics of Hydra

Table 4 shows the characteristics analyzed by the experts that used the tool. This section examines whether the desirable characteristics of AmI scenarios simulator have been satisfied by the Hydra tool or not.

The experts agree that all the models are currently included in the simulator. Some of them require an extension, such as the mobility model that currently has very few different possibilities or the group behavior that also requires an improvement.

The execution features are nearly complete with the exception of the scene design tool as has already been

Table 4
AmI simulator features fulfilled

Feature	AmI Simulator
Environment modeling	✓
Mobility modeling	✓
Individual behavior modeling	✓
Group behavior modeling	✓
Human interaction modeling	✓
Physics modeling	✓
Network protocol modeling	✓
Network device modeling	✓
Preconfigured scenes	✓
Parameters configuration in scenes	✓
Scene design tool	×
Real-time execution	✓
Batch execution	×
Real-time modification	✓
Integration with Real-time systems	×
Discrete-event simulator	✓
Real-time analysis	✓
Post execution analysis	✓
Visual analysis	✓
Text logs	✓
Batch data analysis	×

commented in the subsection 3 and is still being developed, The batch execution is also a work in progress as the efforts are currently in the visual simulation and the integration with a real time environment that still requires some adjustment.

The analysis features are also fulfilled, with the exception of the batch data analysis. Despite this, some of the analysis tools still require much more development, specially the post execution analysis that nowadays lacks many features.

5.2. *Number of conclusions drawn using Hydra*

This section presents an analysis about the number of conclusions obtained after the simulation of the three scenarios using Mason and NS3 on the one hand and Hydra on the other hand. An expert simulated each scenario using Hydra or Mason and NS3, but never both; in order to avoid learning. As explained in section 4 duplicates found both while simulating in Mason and NS3 are merged and we also trust in the expert's judgment for the relevance of the conclusions.

Table 5 presents the number of conclusions drawn by each expert and scenario. It can be seen that the number of conclusions is slightly higher when Hydra is used to model and simulate AmI scenarios. The total number of conclusions drawn using Hydra was 36, while using Mason and NS3 26 conclusions were obtained (see Figure 4) In this case the difference is only ten conclusions, which may seem few; but the important thing is that using an AmI simulator as Hydra there are some conclusions that can be drawn that are impossible to extract otherwise.

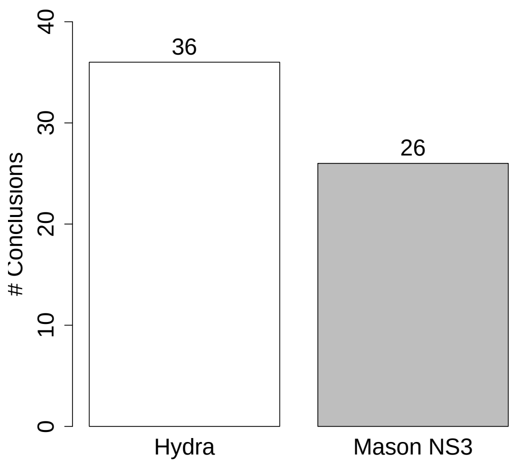

Fig. 4. Total of conclusions drawn

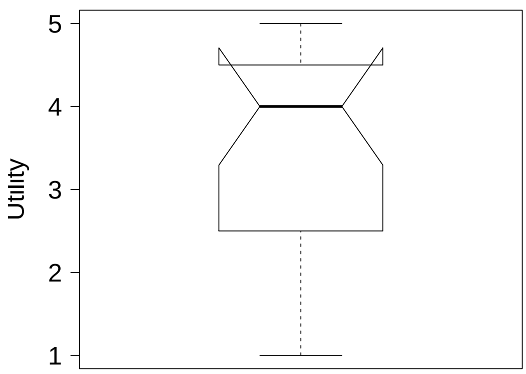

Fig. 5. Overall evaluation about the utility of a co-simulator

These results suggest that the use of a co-simulator as Hydra can provide more relevant conclusions for the

Table 5
Conclusions extracted by experts.

# Expert	Scenario 1		Scenario 2		Scenario 3	
	Hydra	Mason&NS3	Hydra	Mason&NS3	Hydra	Mason&NS3
Expert 1	×	3	5	×	×	5
Expert 2	×	4	×	4	6	×
Expert 3	6	×	6	×	7	×
Expert 4	6	×	×	4	×	6

simulation of AmI environments than simply using a social simulator and a network simulator.

5.3. Analysis of the users perception with an AmI environment simulator

The twenty one learners who participated in the case study evaluated their satisfaction with the Hydra simulator, and were asked about their subjective perception of the utility and the need for an AmI environment simulator (see figure 5). The mean of evaluations is 2.94 and the standard deviation is 1.07; this means that assessments have been generally very high and learners have evaluated it as very useful tool.

Anyway there are several low scores for the utility, meaning that learners have evaluated the tool as unhelpful, this may be due to the difficulty of understanding the need for a simulator of AmI environments; since, in an AmI environment, it is very difficult to differentiate between the social and the network sides and the advantages of a co-simulator.

Learners were also surveyed about the four steps proposed by Hydra to perform a simulation (explained in our previous work [8]), this survey was conducted in order to study what steps can be improved in Hydra. Table 6 and figure 6 presents the results obtained.

Table 6
RQ3 evaluation summary.

Step	Evaluation	
	Mean	SD
Step 1	2.43	0.75
Step 2	3.62	0.92
Step 3	3.76	0.7
Step 4	1.95	0.59
Total:	2.94	1.07

As we can see, steps 2 and 3 were the most highly valued by learners while steps 1 and 4 were the worst rated, especially the fourth step. This is because in step 1 there are still very few preprogrammed scenes and there is currently not a way to create new ones. Step 2 seem to have enough configuration options and similarly step 3 is solid enough with the ability to inspect elements and change the values. The analysis tool in step 4 shows very few options, so much more is expected for a proper analysis.

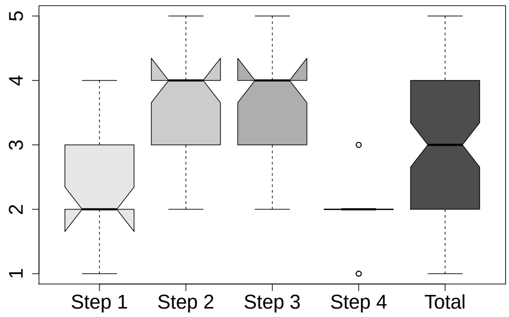

Fig. 6. Evaluation of the steps proposed by Hydra to perform a simulation.

6. Conclusions and future work

In this paper we present a tool that integrates a social and a network simulator to create an AmI simulator with the idea to get a proper tool that can analyze these environments. In order to create it we have checked the different features in network and social simulators, specially in *Mason* and in *NS3* that are our reference simulators. We have also identified some features that could be a great addition to an Ami simulator. The current status of our simulator has been analyzed by some experts that clearly identified some of the features missing because they are still being developed.

In this paper we also present the architecture and data model of the tool showing screen-shots with the current state of the tool. And finally present a validation of the tool with experts comparing the simulations in the AmI simulator with those of the Social and Network simulators and checking the satisfaction of some users of the tool.

This section will be also guided by the three research questions stated in this work.

Qualitative analysis of the characteristics of Hydra The results clearly state that most of the features are currently supported by the tool, but as it is currently being developed, some of the features are still lacking.

Number of conclusions drawn using Hydra The results obtained with the case study performed in this research work indicate that using a co-simulator for AmI environments can draw more conclusions than using a social and network simulator separately. Although it may apparently seem that not many conclusions are attained, the important thing is the quality of those conclusions, since them are impossible to obtain without the help of an AmI co-simulador.

Analysis of the users perceptions with an AmI environment A survey was used to gather information about the learners perception about the utility of the tool. The results were satisfactory when the learners were asked about the utility of Hydra. They were asked also about the steps used by Hydra to run a simulation, in this case they were highly satisfied with step 2, the configuration of the scene, and step 3, the visualization of the simulation. However steps 1, the selection of the scene, and 4, the analysis of the simulation, were especially valued negatively because their development is still in progress such as a visual post-analysis of the simulation, .

More features could be added to the tool such as the interconnection of Hydra with a real environment or the identification of critical events for which a prosumer tool could help [23].

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] Aarts, E. & Wichert, R. (2009). *Ambient intelligence*. Springer.

[2] Cook, D.J., Augusto, J.C. & Jakkula, V.R. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing*, **5**(4), 277–298.

[3] Preuveneers, D. & Novais, P. (2012). A survey of software engineering best practices for the development of smart applications in ambient intelligence. *Journal of Ambient Intelligence and Smart Environments*, **4**(3), 149–162. doi:10.3233/AIS-2012-0150. https://content.iospress.com/articles/journal-of-ambient-intelligence-and-smart-environments/ais150.

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

[5] Sokolowski, J.A. & Banks, C.M. (2011). *Principles of modeling and simulation: a multidisciplinary approach*. John Wiley & Sons.

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

[7] Breslau, L., Estrin, D., Yu, H., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., et al. (2000). Advances in network simulation. *Computer*, 59–67.

[8] Sánchez-Picot, Á., Martín, D., Bordel, B., Alcarria, R. & Sánchez de Rivera, D. (2016). Towards a simulation of AmI environments integrating social and network simulations. *1st International Workshop on Ambient Intelligence for Large Premises (AmILP)*.

[9] MASON Social Simulator. https://cs.gmu.edu/~eclab/projects/mason/.

[10] Allan, R.J. (2009). Survey of agent based modelling and simulation tools. Technical report.

[11] Swarm Social Simulator. https://github.com/swarmsim.

[12] NS3 Network Simulator. https://www.nsnam.org/.

[13] Omnet++ Network Simulator. https://omnetpp.org.

[14] OPNET Network Simulator. https://www.riverbed.com/es/products/steelcentral/opnet.html.

[15] Luke, S., Cioffi-Revilla, C., Panait, L. & Sullivan, K. MASON: A New Multi-Agent Simulation Toolkit.

[16] Luke, S., Balan, G.C., Panait, L., Cioffi-Revilla, C. & Paus, S. (2003). MASON: A Java multi-agent simulation library. In *Proceedings of Agent 2003 Conference on Challenges in Social Simulation* (Vol. 9, p.9).

[17] Luke, S. (2005). MASON: A Multiagent Simulation Environment. *Simulation*, **81**, 517–527. doi:10.1177/0037549705058073.

[18] Henderson, T.R., Roy, S., Floyd, S. & Riley, G.F. (2006). ns-3 project goals. In *Proceeding from the 2006 workshop on ns-2: the IP network simulator* (p.13). ACM.

[19] Issariyakul, T. & Hossain, E. (2011). *Introduction to network simulator NS2*. Springer Science & Business Media.

[20] Climent, S., Capella, J.V., Blanc, S., Perles, A. & Serrano, J.J. (2013). A Proposal for modeling real hardware, weather and marine conditions for underwater sensor networks. *Sensors*, **13**(6), 7454–7471.

[21] Combs, G., et al. (2007). Wireshark. *Web page: http://www.wireshark.org/*, 12–02.

[22] Morales, A., Alcarria, R., Martin, D. & Robles, T. (2014). Enhancing Evacuation Plans with a Situation Awareness System Based on End-User Knowledge Provision. *Sensors*, **14**(6), 11153. doi:10.3390/s140611153. http://www.mdpi.com/1424-8220/14/6/11153.

[23] Martín, D., Alcarria, R., Robles, T. & Sánchez-Picot, A. (2016). Prosumer Framework for Knowledge Management Based on Prosumer Service Patterns. *International Journal of Software Engineering and Knowledge Engineering*, **26**(07), 1145–1173. doi:10.1142/S0218194016500406.