

Multi-modeling and co-simulation-based mobile ubiquitous protocols and services development and assessment

Tom Leclerc, Julien Siebert, Vincent Chevrier, Laurent Ciarletta, Olivier Festor

▶ **To cite this version:**

Tom Leclerc, Julien Siebert, Vincent Chevrier, Laurent Ciarletta, Olivier Festor. Multi-modeling and co-simulation-based mobile ubiquitous protocols and services development and assessment. 7th International ICST Conference on Mobile and Ubiquitous Systems - Mobiquitous 2010, Dec 2010, Sydney, Australia. inria-00534575

HAL Id: inria-00534575
https://inria.hal.science/inria-00534575

Submitted on 10 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Modeling and Co-Simulation-based Mobile Ubiquitous Protocols and Services Development and Assessment

Tom Leclerc, Julien Siebert, Vincent Chevrier, Laurent Ciarletta, and Olivier Festor

MADYNES & MAIA - INRIA Lorraine, Nancy Université, France
{first_name.name}@loria.fr

Abstract. Mobile and Ubiquitous Computing is about interconnected computing resources embedded in our daily lives and providing contextual services to users. The real influence between user behavior and ubiquitous communication protocols performance and operation needs to be taken into account at the protocol design stage. Therefore, we provide a generic multi-modeling approach that allows us to couple a user behavior model with a network model. To allow both assessment and benchmarking of ubiquitous solutions, we define formal reference scenarios based on the selection of a set of environmental conditions (contexts). We illustrate the use of the framework through its application to the study of mutual influences of mobility models and ad hoc network protocols.

1 Introduction

1.1 Context and motivation

Ubiquitous or Pervasive Computing is about interconnected often dynamic and mobile computing resources embedded in our daily lives and providing services to users in a changing context and environment.

Several scientific domains (network, AI, physics, sociology, ...) are involved in the field of Ubiquitous Computing together with their own vocabulary, habits, needs and culture. To deal with the interacting complex models of ubiquitous computing, no single universal model exists. Experience in ubiquitous systems demonstrates that advanced research in such a complex topic cannot be pursued by only broadening an initial domain with unavoidably partial knowledge from others. A typical example is the design of mobile services where the user carries devices that contribute to the delivery of data to other users. In this case, the behavior of the users like (e.g. their mobility patterns) in a crowd highly impact the overall operation of the service and need thus to be considered early in the service design.

We therefore propose a methodology and a novel distributed framework to design, implement and assess "mobiquitous" communication related technologies. Our solution is build on two key elements : model interaction and multi-simulation engine.

First, our approach enables the combined use of reference models and simulators coming from different specific domains (figure 1). Through a simple interface implemented for each simulator, the presented framework eases interaction among both models and simulators. This significantly improves the initial design of the EXiST (EXperImental Simulation Tool [4]) co-simulator by both providing decentralization support and a better formalization.

1.2 Case study

In the domain of dynamic networks addressed here, wireless technologies, ad hoc or mesh routing protocols, or ubiquitous services are often studied (designed, experimented, assessed) using network simulators. Indeed, real world experimentations with a representing set of devices is excessively time and money consuming, especially in the case of ad hoc networks or large scale peer-to-peer environments. It is even scientifically of little relevance since reproducing a scenario / an experiment is not possible due to the ever changing experimental conditions. Therefore, a lot of models and simulators have been developed in the field of ubiquitous computing over time [10, 9]. They aim at simulating the network layers in more or less details and indeed most of them are not designed for doing more, like advanced node dynamics for example, or users goals. In fact, in ubiquitous computing, one key element of the equation is "the human" and more specifically his behavior.

As a case study, in this paper we focus on mobility in MANETs (Mobile Ad-hoc NETworks), as an example to demonstrate our approach. MANETs are wirelessly connected devices connecting spontaneously without any preexisting infrastructure. In MANET simulation, nodes move according to a mobility model. Most mobility models are computed by merely considering the user as a random walker without goal or decision process, and without any knowledge of how the network actually behaves. Unfortunately this is what is generally considered sufficient to give the system its "dynamic" characteristic, and therefore used to prove the validity and demonstrate the performance of protocols which later fail when deployed in the real world. Our approach circumvents this limitation.

As a proof of concept, we combine two existing simulators: a mobility simulator (based upon a multiagent model) and a network simulator. By doing this, we combine sociological research achieved in urban simulation community with network research. Our experiments in Section 5) show the possibilities that the framework offers and also the importance of the mutual influences between the network and user behavior. We believe that the originality of our approach is to allow to close the loop between the users behaviors and their mobile ubiquitous environment.

The remainder of this paper is organized as follows. Section 2 motivates the usage of multi-modeling and co-simulation. In Section 3 we present our conceptual framework and a prototype implementation. We focus in Section 4 on mobility modeling and present the multiagent paradigm applied to the modeling

of users behaviour. Experiments and results are described in Section 5. Section 6 summarizes the contribution, ongoing and future work.

2 Multi-modeling and co-simulation motivation

Our approach is built on the use of multi-modeling and co-simulation in order to take into account both users behaviors and network performance within an integrated study. The framework offers a way for protocol and service designers to get a "bigger picture" early in the design phase.

As stated in the introduction, we argue that the study and the design of mobile ubiquitous applications cannot be achieved efficiently by taking into account only one point of view. By point of view, we mean the physical medium aspects, the network aspects (protocols, services, messages, topology ...), the users behavior aspect (mobility, sharing resources ...), etc. (figure 1). Depending on the study and questions asked, omitting some of these points of view may lead to non-significant simulation results. For example the authors of [2] show the impact of taking into account different physical medium models for the wireless communication. Moreover as many models and simulators already exists and have been validated reusing them is the best approach.

We propose to use multi-modeling and co-simulation in order to represent all the different aspects or point of view needed for the simulation to be more significant. We rely upon a meta-model and a framework called AA4MM which allows us to couple different existing models and simulators in order to build a more complex and more accurate simulation. These simulations are used, on one hand, to evaluate protocols and services against different usage scenarios and, on the other hand, to design new protocols and services by taking into account some global usage scenarios (the user behavior and the environment parameters). By co-simulation or multi-simulation, we mean the ability to combine multiple simulators and/or real implementations (prototype, software and/or hardware) at the same time.

The main advantage of this approach is to achieve a good separation of concerns. Computer network scientist and designers only focus on the network aspect (protocol and services definitions, network parameters: radius, bandwidth, latencies ...) and cognitive and human scientists focus on the user behavior modeling (mobility, user needs, (ir)rationality). The whole simulation efficiency is our main limitation. Reusing existing simulators - that may not have been designed for distributed simulations - may be less efficient than a single multi-model implemented in a natively distributed simulator. However we consider that the advantages brought by the separation of concerns are conceptually more important and that the simulation efficiency is a technical question that can be targeted later.

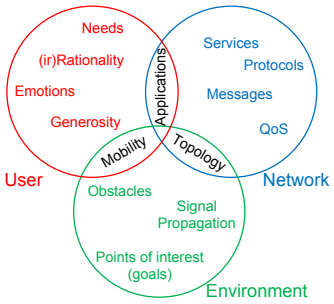


Fig. 1. Abstraction levels

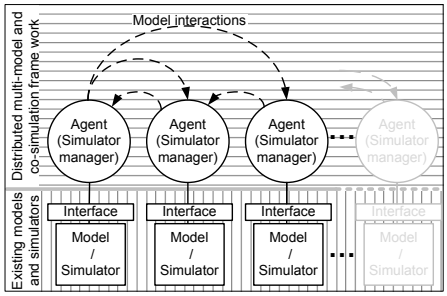


Fig. 2. Framework model interactions

3 The AA4MM meta-model and platform

We develop a multi-modeling platform called AA4MM (Agent and Artefact for Multiple Models [12]). Its main goal is re-usability and interoperability of different simulators like HLA (High Level Architecture), EXiST or MSI (the Multi Simulation Interface) but its software architecture is completely decentralized and based upon multiagent paradigm. This paradigm allows us to take into account solutions developed in complex systems modeling and multiagent community such as automated parametrization or simulation control [5, 3]. In this paper we only highlight how existing simulators interacts withing the framework.

Figure 2 describes the composition of the AA4MM framework. Each simulator is controlled by a simulator manager (formally an agent) which is an autonomous entity. All these manager agents cooperate in order to run the whole simulation and to take care about the interaction problematics. To make different simulators interact in the AA4MM platform, the following steps are required:

- define a simulator interface (one for each simulator): implement 6 basic functions as described in table 1 directly from the source code or from the api, or more laborious by extracting an api if only binaries available; this is the only modification to make.
- create the specific AA4MM entities:
  - for each simulator create an entity (called an agent) in order to manage the simulator (input/output data flows, model execution and simulation time management).
  - for each link between the simulators create an entity in charge of the data flow exchange (called an artifact).

It is relatively simple to add a model and its simulator within the interfacing (Table 1). The interactions problematic (e.g. simulators synchronization and model compatibilities) are managed by the AA4MM platform itself by creating entities external to the original tools : the agents and the artifacts. This presents some advantages: it is easy to reuse existing models and simulators without knowing anything of distributed simulations. The few modifications brought to

the models and simulators allow us to use them either as a standalone application or inside the multi-simulation.

	Fonctions	Description	Example
1	Initialization	Initialize, passing parameters	
2	Model execution	Execute 1 step of the model	Execute 1 simulation step, 1 simulation event or a given time interval
3	Get simulation time	Obtain current simulation time	
4	Data input	Provide data to the simulator	Input information from another simulator
5	Data output	Retrieve data from the simulator	Output information going to another simulator or for logging purposes
6	(Optional) Finalize simulation	Finalize simulation after last step	Retrieve logs from / execute logging scripts

Table 1. Interface to define for a simulator to work within the AA4MM platform

Artifacts created can be seen as a distributed tuple space where agents exchange simulation data timestamped with a specific validity interval [13]. The whole platform is sustained by a distributed simulation algorithm (a Chandy-Misra-Bryant algorithm variant) that allows the agents to manage the whole simulation process in a distributed way, whatever the execution policy of their simulators (discrete event, step by step, continuous time).

Our framework uses a series of XML configuration files that allow for the simple description and tweaking of the different simulators involved and of the global simulation.

4 Case study: MANETs and users behaviors

4.1 Mobility modeling: a quick survey

There are many ways to model the different types of mobility. Classical mobility models are well documented and can be classified, as surveyed in [1], in 4 categories: random models (e.g. Random Waypoint), models with temporal (e.g. Gauss-Markov), spatial (e.g. Reference Point Group) or geographical (e.g. obstacle mobility) dependencies.

There is no formal model combining some of those classical models. And most critically, none ever considered any feedback from the network to the user behavior (e.g. impact of perceived QoS). Our work allows both by proposing to use the agent paradigm as a unique tool for modeling the largest and various sort of mobility.

4.2 The multiagent paradigm

The multiagent paradigm is a way to model sets of autonomous interacting entities within an environment. It is a well known paradigm used in human sciences, ecology or in robotics. It describes the systems into, at least, these different components: agents, environment, interactions. The agents are autonomous and proactive entities, situated in an environment. They only have a partial (local) view of it and decide which action to take dealing with their own perceptions and reasoning.

MABS (Multiagent Based Simulation) offers us the right level of description when we want to model users' behavior, goals and actions. Instead of using a global equation to model users' trajectories, we can, via the agent based model, re-create the way users move. It means that we can directly model behaviors such as "if an obstacle is present in front of you, then avoid it" or "reach a goal, stay nearby during five minutes and then go".

More generally, with this approach, we can model more complex behaviors such as willingness to use and share a service depending on the bandwidth consumed or the generosity of a user ; or the reaction to unpredictable events.

Mobility has already been studied and modeled via the multiagent paradigm. Here an agent can describe human, animal or robots. In [11], Craig Reynolds worked on bird flocks modeling where each agent tries to stay inside the flocks only by computing a small set of forces (Boids). Individual-based pedestrian modeling is also used in urban simulations [7, 14]. This paradigm is also used to model crowd scenes in movies (as battlefields in Lord of the Ring) and implemented in video animation software such as MASSIVE[1].

4.3 User model description

Our agent based mobility model is inspired from urban research and pedestrian modeling [5, 7], but can also model classic mobility behaviors (e.g. random waypoint). Each mobile node (a user) is represented by an agent (named a_i). The agent behavior can be seen as a combination of simple behaviors resulting in a complex one. For example random waypoint is implemented as the simple following rule: Each every time period each agent changes its direction. More complex behaviors such as obstacle avoidance, flocking or goal attraction are modeled as a function, a sum of forces, resulting in a node movement. Each force/behavior describes an interaction of the agent with its environment and the other agents. The agent has a limited perception (figure 3): these interactions are effective only on the neighborhood of the agent. In our case, the movements of the agents are computed by applying laws of mechanics: namely point kinematics. These models are easily extensible, easy to implement and can express a large set of behaviors by weighting each force. The examples below (figures 4,5 and 6) depict force oriented behaviors.

[1] http://www.massivesoftware.com

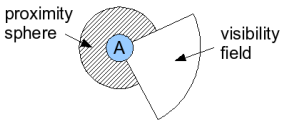


Fig. 3. Perceptions of an agent (a user)

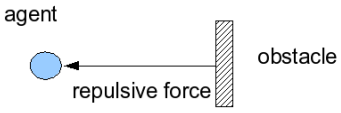


Fig. 4. Repulsive force for obstacle avoidance model

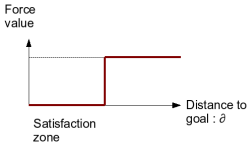


Fig. 5. Attractive force to the goal

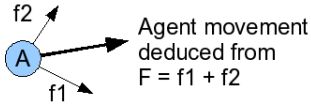


Fig. 6. Movement computation

4.4 Modeling network aware users

Integrating network aspects into the agent decision process is achieved easily and straightforward. Indeed, once the agent perceived the network information (e.g. connectivity presence/loss, quality of services) a simple rule defines its reaction. For example in Section 5 we describe users that slow their speed or stop moving when they perceive good connectivity.

4.5 Synthesis

This model respects the constraints cited in [1]: temporal dependency, spatial dependency and geographical dependency. Describing sophisticated movements is straightforward: for example from our two simple movements we have nodes avoiding obstacles and following multiple succeeding goals. Moreover, we can easily model mobility of groups of people just by adding a force that attracts agents that go in the same direction (as shown in Section 5).

We've develop a set of mobility models, from simple random waypoint or restricted random waypoint to advanced particles engine, flocking or explorer behavior, that are fully parametrized. Using MABS to simulate basic behaviors such as random waypoints seems probably overkill at first.However, since this modeling approach is individual-based, we can easily tune each behavior and describe heterogeneous ones. Indeed, the highest level of granularity can be reached by implementing a different model of behavior per agent. Thus, we can describe, for example, different kind and mixes of populations. Finally, with our approach, a user can dynamically switch from one behavior to another.

5 Experiments and results

As a proof of concept of our vision and framework we coupled a users behaviors simulator that we developed (MASDYNE: MultiAgent Simulator of DYnamic

Network usErs) with a MANET simulator (JANE: The Java Ad hoc Network Emulator [6]).

The goals of the following experiments are on the hand, to show the simplicity of a realistic usage scenario design and implementation, and on the other hand, to show the effects of having interactions between the user behavior model and the network model.

5.1 Building realistic usage scenarios

The goal of the first experiment is to obtain a mobility scenario (figure 7): A group of students visiting a museum. This scenario fits with our goal to test and deploy in the future an ad hoc network within a museum (ANR SARAH project).

This mobility model is based upon force oriented behaviors: the user behavior, the interaction between the users and the environment are represented by simple forces. We use, for this scenario, four simple force oriented behaviors: *goal force, avoid walls, repulsive force, attraction force.* This provides us the following scenario: Agent 1, a tour guide, follows goals unknown to the students. The other agents, the students, follow agent 1. The combination of these simple behaviors, done by summing the forces, results in a complex and more realistic behavior.

- All Agents have a Goal: e.g. the students follow Agent 1, if visible.
- All Agents avoid walls: Repulsive force from the walls.
- Agents have a repulsive force from each other (comfort zone).
- Agents are attracted by other agents that go in the same direction.

Figure 7 show this usage scenario in different environments (e.g. corridor, crossroad, doors, museum). Parameterization of the model is done according to [8]. We observe that in different environments the student group is clearly following the Tour guide even when walls are involved.

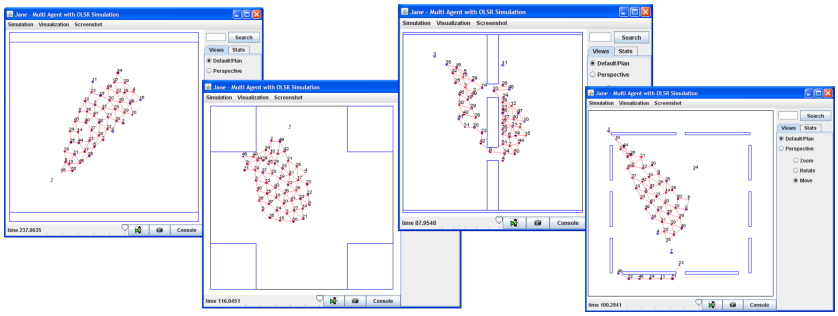


Fig. 7. Museum visit example in multiple environments

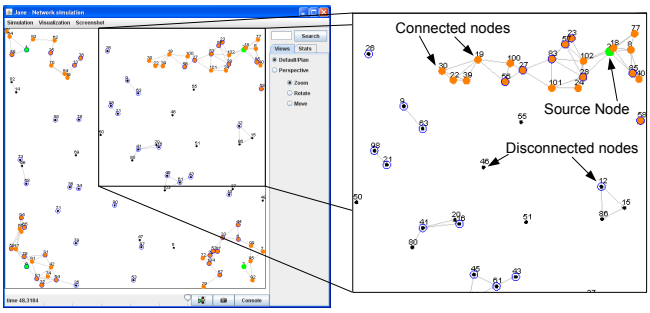


Fig. 8. 4 source nodes: green nodes, 100 moving nodes: black when disconnected, orange when connected to a source

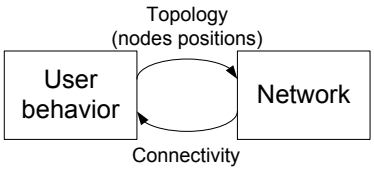


Fig. 9. Interactions between MASDYNE and JANE

5.2 Network and user behavior mutual influences

Even the simplest network protocol such as a broadcast obviously would perform very well if the students follow seriously the rules. However, what happens if not all the students follow the rules and react for example to network events (e.g. network connectivity). The AA4MM framework allows us to respond to this kind of questions. To show the effects of these mutual influence, the second experiment is a scenario with network feedback (figure 8). The aim of this experiment is not so much about being realistic but more about showing the possibilities offered by our approach.

Experimental protocol: The usage scenario is the following: 4 source nodes (access points) are placed in every corner of a place. 100 nodes/users want to connect to a source. To keep it simple to explain and to avoid errors or bias induced by the algorithm of a protocol, we used a basic flooding algorithm that simply rebroadcasts every non seen message. Messages already seen are silently discarded.

The user behavior is the following. At beginning the user moves randomly. He is aware of its connection status: connected or not connected. Then we propose 3 basic behaviors: continue to move randomly (user is not aware of network feedback or doesn't care), slow down speed (user continues walking but only slowly), stop (user sits down to enjoy the connection). Figure 9 depicts the interactions between both simulators.

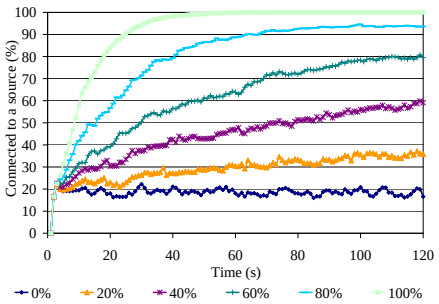


Fig. 10. Percentage of nodes stopping when connected, remaining nodes move randomly

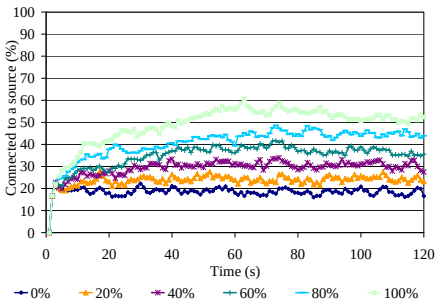


Fig. 11. Percentage of nodes that divide their speed by 6 when connected, remaining nodes move randomly

We measured the evolution of the percentage of connected nodes. For each experiment we use 2 behaviors. We vary the percentage of agents having each behavior. For example in figure 10 the curve marked 40% means that 40% of the agents implement the stop behavior while the remaining 60% implement the random behavior. Each experiment was done using 50 distribution seeds.

Results: We observe that the more the users slow down or stop the better is the connectivity rate. The stopping nodes create a sort of backbone for the other nodes while the backbone created by slowing nodes is only temporary until nodes move out of range.

In figure 10 we increased the percentage of nodes stopping while the remaining ones continue moving randomly. With 100% stopping nodes, after 60 seconds all nodes reached an access point. With already 60% of nodes stopping, 80% of the nodes are connected. In figure 11, the nodes divide their speed by 6 when connected. Again we increased the percentage of nodes slowing while remaining nodes move randomly. Performances compared to stopping nodes are worse. In figure 12 and 13 every node reacts on connection. We varied the percentage of stopping nodes while the remaining nodes divide their speed. We observe that the results significantly differ when using random waypoint model or more complex usage scenario.

5.3 Synthesis

This experimental work show that our approach has: 1) the ability to take mutual influences of users behaviors and network performances into account; 2) the ability to design usage scenarios with heterogeneous users behaviors; 3) the ability to benchmark a network protocol against a wide range of usage scenarios.

In order to consider this work from a higher standpoint, we don't assume that the users behaviors will always be predictable. But, with this approach, we are able to predict that if only a percentage of the users behave like we predict, then the network performances will be better or worse.

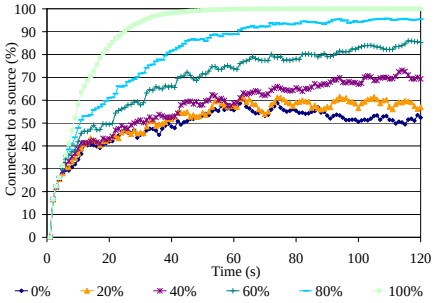


Fig. 12. Percentage of nodes stopping when connected, remaining nodes divide their speed by 6 when connected

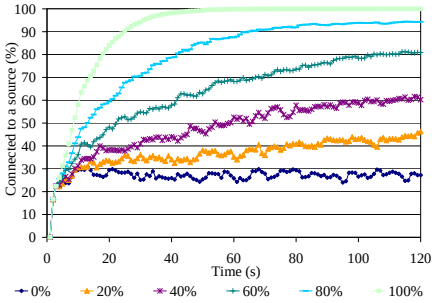


Fig. 13. Percentage of nodes stopping when connected, remaining nodes divide their speed by 3 when connected

6 Conclusion

When using classical modeling approaches, it is not straightforward to take into account the users behaviours and their interactions with the network performances. We presented the conceptual framework and a prototype implementation. With our multi-modeling approach, different existing models can be easily coupled in a loose and generic way. We focus in this paper on mobility modeling which is a key point in evaluating wireless technologies and services, and described a couple of experiments.

We presented the multiagent paradigm applied to the modeling of users behaviour. We argued and shown that while it still provides the usual mobility models, it is very simple to design, fine-tune, redesign those models or even design completetly new ones. Multiagent allows the description of heterogeneous behaviours. The new mobility models can take into account networks or more generally environment inputs, basically having a closed-loop system where something closer to the "human behavior and real-life" is considered.

Our approach offers a basis for valid comparison of wireless technologies and services but it can be extended to any dynamic environment, such as P2P networks for example. It is very well-suited for every situation where there are interactions between the users, the networking and the physical environments.

Our experiments demonstrate that closing the loop leads to new ways of evaluating technologies. Even a basic protocol, such as our flooding example, can have strong performances if the users follow their directives. Some can be difficult to enforce (complete stop when detecting a connection), but other could be reasonable in a real world (slowing down).

In the short term, we plan to show on the opposite the disruptive effect of non conforming behavior, and to extend our experiments to more advanced protocols and scenarios. In parallel, we continue to work on the theoretic and practical aspects of the AA4MM simulation framework. Our platform will be extended by defining and implementing more standard and novel mobility models (node/users behaviors), and reference environments.

In the longer term, the framework will be enriched with a set of mobility models, a set of environments models. We also plan to have a real setup (a typical existing room or building or city modeled in 3D from real data for example). These sets can serve as references that could be used to assess the performances and applicability of a solution, and validate it in certain contexts. This could be a good basis to provide the ubiquitous computing community with a benchmarking evaluation toolkit.

References

1. F. Bai and A. Helmy. *A SURVEY OF MOBILITY MODELS. in Wireless Adhoc Networks.* University of Southern California,U.S.A., 2004.
2. E. Ben Hamida, G. Chelius, and J.-M. Gorce. Impact of the Physical Layer Modeling on the Accuracy and Scalability of Wireless Network Simulation. *Simulation*, 85:574–588, 09 2009.
3. S. Bonneaud, P. Redou, and P. Chevaillier. Pattern oriented agent-based multimodeling of exploited ecosystems. In *6th EUROSIM congress on modelling and simulation*, september 9-13 2007.
4. L. Ciarletta, V. Iordanov, and A. Dima. Using Intelligent Agents to assess Pervasive Computing Technologies. In *IAWTIC 2001*, page 10 p, Las Vegas, USA, 2001.
5. N. Gaud, S. Galland, F. Gechter, V. Hilaire, and A. Koukam. Holonic Multilevel Simulation of Complex Systems. Application to real-time pedestrian simulation in virtual urban environment. *Simulation Modelling Practice And Theory (SIMPAT)*, 16(10):1659–1676, Nov. 2008.
6. D. Gorgen, H. Frey, and C. Hiedels. Jane - the java ad hoc network development environment. In *ANSS '07*, pages 163–176, USA, 2007. IEEE Computer Society.
7. D. Helbing, L. Buzna, A. Johansson, and T. Werner. Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions. *Transportation Science*, 39(1):1–24, 2005.
8. D. Helbing, P. Molnar, I. J. Farkas, and K. Bolay. Self-organizing pedestrian movement. *Environment and Planning B: Planning and Design*, 2001.
9. S. Kurkowski, T. Camp, and M. Colagrosso. Manet simulation studies: The incredibles. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9:50–61, 2005.
10. S. Naicken, A. Basu, B. Livingston, and S. Rodhetbhai. A survey of peer-to-peer network simulators. *Proceedings of The Seventh Annual Postgraduate Symposium, Liverpool, UK*, 2006.
11. C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21:25–34, 1987.
12. J. Siebert, L. Ciarletta, and V. Chevrier. Agents and Artefacts for Multiple Models coordination. Objective and decentralized coordination of simulators. In *SAC 2010 25th Symposium on Applied Computing*, Lausanne Suisse, 2010. ACM.
13. J. Siebert, J. Rehm, V. Chevrier, L. Ciarletta, and D. Mery. Aa4mm coordination model: event-b specification, rr-7081. Technical report, INRIA, 2009.
14. K. Teknomo, Y. Takeyama, and H. Inamura. Review on microscopic pedestrian simulation model. In *Proceedings Japan Society of Civil Engineering Conference*, 2000.