

Co-simulation Solutions using AA4MM-FMI applied to Smart Space Heating Models

Leilani Gilpin^{*}
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
Leilani.Gilpin@parc.com

Laurent Ciarletta
Université de Lorraine
LORIA, UMR 7506, INRIA
54506 Vandœuvre, France
laurent.ciarletta@loria.fr

Yannick Presse
INRIA Nancy - Grand-Est
615 rue du Jardin Botanique
54600 Villers, France
yannick.presse@inria.fr

Vincent Chevrier
Université de Lorraine
LORIA, UMR 7506, INRIA
54506 Vandœuvre, France
vincent.chevrier@inria.fr

Virginie Galtier
Supélec UMI 2958
2 rue Edouard Belin
57070 Metz, France
virginie.galtier@supelec.fr

ABSTRACT

Multi-modeling and co-simulation are one of the solutions for dealing with complex systems. In this paper, we propose to apply the AA4MM framework to the co-simulation of a smart space heating “complex” environment with 2 objectives. Our first contribution is the development of AA4MM-FMI (Agent and Artifact for Multiple Models coordination for FMI). AA4MM is a modeling and simulation framework that can be used to implement the “master” in a multiple FMU (Functional Mock-up Unit) co-simulation of the FMI (Functional Mock-up Interface) standard with a fully decentralized view and intrinsic coordination. Our second contribution is to apply and evaluate this solution by creating a hierarchy of smart space models that are exploiting house geometry information. Each individual room is represented as an independent FMU taking inputs of the room’s geometric information (surface area, volume, insulation, etc.), target temperature, and neighboring rooms. The neighboring rooms are connected to each other using the AA4MM-FMI framework. This project aims to couple an electrical heating-based simulation with networking event-based simulations to gain intuition for how house geometry affects efficient heating and network connectivity. Using this smart space house heating problem, we are able to test AA4MM-FMI (our novel “master”) for FMI and multi-simulation.

^{*}The following work was initiated during an internship at INRIA - Grand-Est during from June 2013 to October 2013 and is part of the MS4SG (Multi Simulation for Smart Grids) collaborative research project with EDF R&D.

Categories and Subject Descriptors

I.6.7 [Simulation and Modeling]: Simulation Support Systems—*Environments*

General Terms

DESIGN, STANDARDIZATION

Keywords

Co-simulation, Functional Mock-up Interface (FMI), High Level Architecture (HLA), simulation, smart grid

1. INTRODUCTION

Many technological advancements are occurring in the energy industry, most predominantly, in the smart grid. The smart grid incorporates information and communication technology to improve the efficiency, reliability, sustainability, cost, production, and distribution of electricity. However, because it incorporates many different and interrelated components (namely, electrical networks and communication networks), it can be considered as a complex system. Therefore, it is difficult to model and simulate such an environment effectively and realistically. As a first step, we are studying a heating system for a smart room: the challenge being to couple models and simulators for the physical world (heating/geometry), the home network and a real control application (running on an android device). In order to couple a heating simulation with a networking simulation effectively, we have created a smart space simulation using the FMI standard and AA4MM-FMI: a middleware component to run our co-simulation solution.

The Functional Mock-up Interface (FMI) [3, 4] defines a standard for the development of multi-model simulations, interoperability, exchange, and reuse. The FMI has been designed to handle the interoperation of models described by differential, algebraic, and discrete equations exported as FMUs (Functional Mock-up Units). Each FMU is distributed as a zip file with the “.fmu” extension. This zip file contains an XML file with definitions, a defined set of C functions for all the equations used by the model, and other optional data.

The FMI standard consists of two major parts: FMI for Model Exchange and FMI for Co-Simulation. FMI for Model Exchange is intended for use in large systems to break down components. FMI for Co-Simulation is intended to couple simulation tools and exchange data between models. The “master” controls the data exchange between subsystems where the FMU components are the slaves. In our work, we focus on FMI for Co-Simulation. The AA4MM-FMI is used as an implementation of the “master” in FMI with a decentralized view.

In distributed multi-simulation the High Level Architecture (HLA) [11] is another candidate in regards with interoperation of discrete event simulation models. Compared with FMI which leaves the implementation of the synchronisation function to the simulation developer, HLA has a highly sophisticated time management provided by its Run-Time Infrastructure (RTI). Yet this “co-simulation bus” is most often implemented in a centralized fashion. In AA4MM, the synchronization is intrinsically decentralized.

2. MODEL

Our model currently contains two main components, with the capability of incorporating a third component in the near future. The first component is a set of room heating FMUs. The individual room heating FMUs are connected together to form a housing unit using our second component: the AA4MM-FMI middleware.

The third component, which will be incorporated into the house heating co-simulation solution in the future, is a discrete-event based communication network to incorporate the individual’s comfort temperature preference.

2.1 AA4MM: a master for FMI

2.1.1 The AA4MM meta-model

We proposed a meta-model called AA4MM (Agent and Artifact for Multiple Models [6]) for multi-simulation. Its main goals are to target re-usability and interoperability of different simulators, and to explicitly target different kinds of model coupling.

AA4MM represents a simulated system as different interacting simulators that are coordinated by a decentralized algorithm[7]. The implementation of these principles is based upon the Agent and Artifact paradigm[2]: each simulator is managed by an agent and coordination corresponds to agents’ interactions.

A software version of AA4MM in multi-threaded JAVA was used for proofs of concept[6], to study the mutual influences between mobility models and routing algorithms in mobile ad-hoc networks [8] or to represent multi-level phenomenon[5].

Figures 1 and 2 illustrate how the AA4MM framework deals with the interaction between models, using agents.

Each simulator is controlled by a simulator manager (formally, an m-agent) which is an autonomous entity. It manages the simulator, and it is in charge of interactions of this simulator with the other ones. All these manager agents cooperate in order to run the whole simulation and to take care

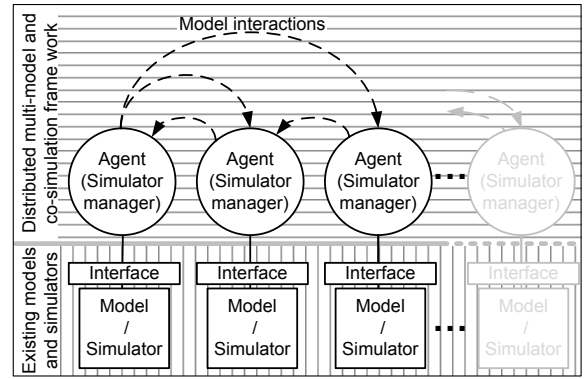


Figure 1: AA4MM overview

about the interaction problematics. We distinguish two kind of interactions *i*) between a m-agent and its simulator (this handled by an interface-artifact), and *ii*) between m-agents (handled by a model-artifact).

We defined a decentralized cooperation algorithm that ensures coherence and causality constraints.

2.1.2 AA4MM-FMI

AA4MM-FMI is middleware framework which allows for the coupling of FMUs to build a more complex simulation. AA4MM-FMI is based on AA4MM [8, 9, 10]. We present AA4MM-FMI as a way to redesign and implement the “master” in FMI with a decentralized view.

AA4MM is a multi-modeling platform for the interoperability of simulators like HLA. However, it differs from HLA in that its software architecture is decentralized. Each simulator is controlled by the specific AA4MM entities: the agent and the artifact. The basic framework of the AA4MM-FMI module is shown in Figure 3.

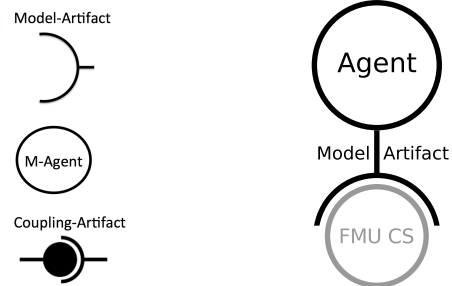


Figure 2: AA4MM agents notation

Figure 3: AA4MM-FMI framework

We use AA4MM-FMI to let the house heating FMUs interact with each other and exchange information about their current temperatures. The coupling of the framework for FMU exchange is shown in Figure 4. In order to let these simulators interact with each other, the following actions take place:

- The definition of a simulator interface (for each FMU), that implements the FMU’s basic functions.

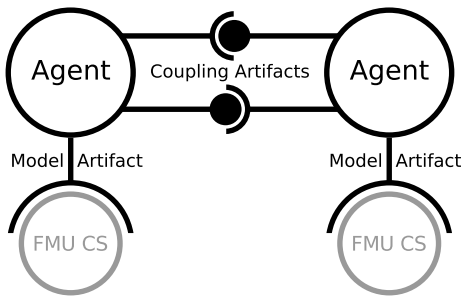


Figure 4: AA4MM-FMI coupling framework

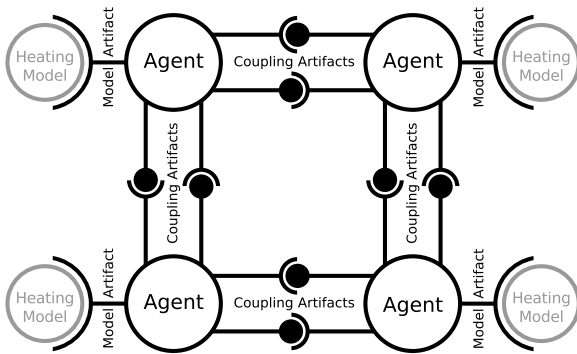


Figure 5: 4 room model using AA4MM-FMI

- The creation of the AA4MM entities. An agent is created for each FMU to manage the input/output data flows, execution and simulation time management. For each link between the FMU, a coupling artifact is created to control the data flow exchange.

For our house heating co-simulation, we use a 4 room model as shown in Figure 5.

2.2 NS-3 Smart Space Model

The NS-3 smart space model simulates a discrete-event communication network between real android devices and the heating regulator. In future work, this model will be exported to an FMU and incorporated in the smart space co-simulation, where the power consumption data will be exchanged. Currently, the network contains four nodes: a sensor, a regulator, a heating subsystem black box (this will be the Room Heating Model FMU in future work), and an android device which takes input from the user.

The NS-3 smart space network receives the current temperature from a sensor, which passes this information to a regulator. The regulator passes the power constraint to a heating system. Then, the regulator and android device exchange information about the desired temperature, or comfort temperature information. The user is able to change the comfort temperature using an Android application running on an Android tablet, which will then be passed to the regulator. The NS-3 communication network can be seen in Figure 6.

For simplicity of implementation, the communication protocol being used is UDP (User Datagram Protocol). We

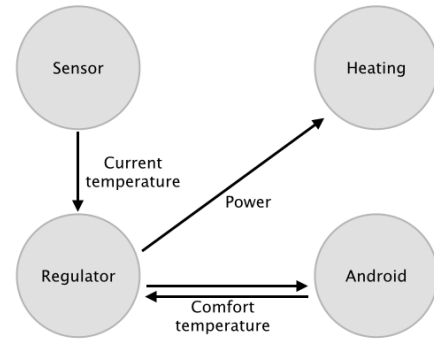


Figure 6: NS-3 communication network

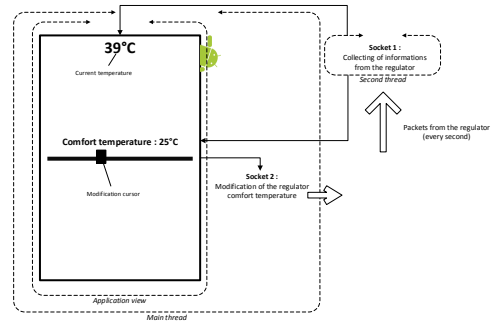


Figure 7: Real android application

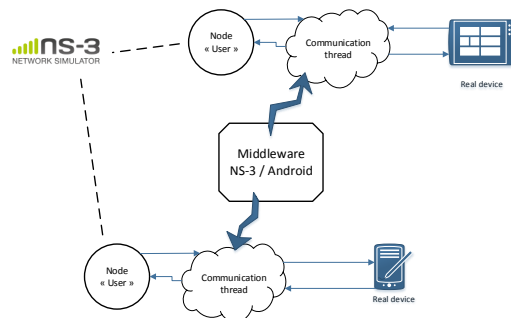


Figure 8: Co-simulation : interactions between real world applications/devices and network simulator

developed special relays/proxy between NS-3 and the outside/real world networked applications to handle hardware and software in the loop simulation. We plan to incorporate more complex protocols in future work.

2.3 Room Heating Model

Each individual room is represented as an independent FMU taking inputs of the room's geometric information (surface area, volume, insulation, etc.), target temperature, and neighboring rooms. The model diagram in Simulink is shown in Figure 11;

The neighboring rooms are connected to each other using the AA4MM-FMI framework. We use this room-linking-model to gain intuition for how house geometry affects efficient heating and network connectivity.

The House Heating FMU is exported from a Matlab model based on an existing house heating model provided by Simulink [1]. Our model calculates the amount of power needed to heat a generic room. The temperature is given in degrees Celsius, and the time is given in hours. We assume the room is rectangular with four neighboring rooms sharing one wall. The FMU takes the following parameters:

- **targetTemp** - The goal temperature that should be maintained indoors in °C(Celsius degree).
- **roomVolume** - The volume of the room in m³ (cubic meter).
- **insulationThickness** - The thickness of the insulation. For our calculations, we assume it is fiberglass.
- **surfaceArea** - The surface area of the room (including the ceiling and floor) in m² (square meter).
- **windowThickness** - The thickness of the windows in the room: they are glass and square-shaped.
- **outdoorTemp** - The average air temperature outdoors (°C).
- **wall1, wall2, wall3, wall4** - The temperature of the neighboring rooms (°C).

The FMU outputs the following for each simulation step:

- **PowerDissipated** - The amount of power required to heat the room (Watt).
- **IndoorTemp** - The current indoor temperature (°C).
- **OutdoorTemp** - The current outdoor temperature (°C).

2.3.1 Room Heating Model Components

The Room Heating Model is made out of five major components: the outdoor temperature, the Thermostat subsystem, the Heater subsystem, the Power Calculation, and the House block.

The outdoor temperature is calculated using a sine wave. The outdoorTemp parameter specifies the average air temperature outdoors, and we assume that the daily temperature fluctuation is 10 degrees Celsius, so the current outdoor temperature is calculated as $10 \sin(t) + \text{outdoorTemp}$, where t is the current time step.

The Thermostat is a subsystem that contains a Relay block. The thermostat allows fluctuations of 5 degrees Celsius above

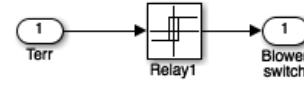


Figure 9: The Thermostat component

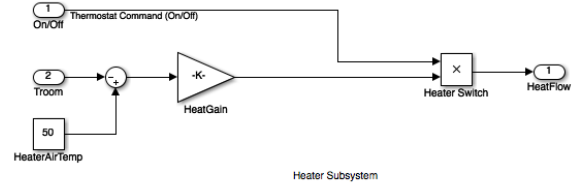


Figure 10: The Heater component

or below the desired room temperature. If air temperature drops below 5 degrees Celsius lower than the targetTemp, the thermostat turns on the heater.

The Heater subsystem calculates the amount of heat flow from the heater into the room ($\frac{dQ}{dt}$). It assumes that temperature of the hot air from the heater (T_{heater}) is 50 degrees Celsius, and the air has a constant flow rate of (M) 3600 kg/hr, and the heat capacity of air at constant pressure (c) is 273 K or 1005 J/kg-K. These constants were given from the base model [1]. The heat flow in the room is expressed using the previously defined constants and the current room air temperature, T_{room} in following differential equation:

$$\frac{dQ}{dt} = (T_{heater} - T_{room}) \times M \times c$$

The Power Calculation is a gain block which integrates the heat flow over time, and divides it by the current time step.

The House subsystem calculates room temperature variations. It takes into consideration the heat flow from the heater ($\frac{dQ_{heater}}{dt}$) and heat losses due to the environment ($\frac{dQ_{losses}}{dt}$). It first calculates the current temperature of the room (T_{room}) as the average temperature of its neighbors. It calculates the heat losses and the temperature time derivative using the mass of the air inside the house, M_{air} , and the thermal resistance of the house, R . The differential equations for the heat flow and heat loss are expressed below:

$$\frac{dQ_{losses}}{dt} = \frac{T_{room} - T_{out}}{R}$$

$$\frac{dT_{room}}{dt} = \frac{1}{M_{air} \times c} \times \left(\frac{dQ_{heater}}{dt} - \frac{dQ_{losses}}{dt} \right)$$

2.3.2 Heating Simulation

The heating model simulates a 48 hour time window. We aim to create some sort of hierarchy of house heating models and compare the results we have at a macro and micro level. Each of the heating models in the coupled simulation exchange information about their indoor temperature at each time step.

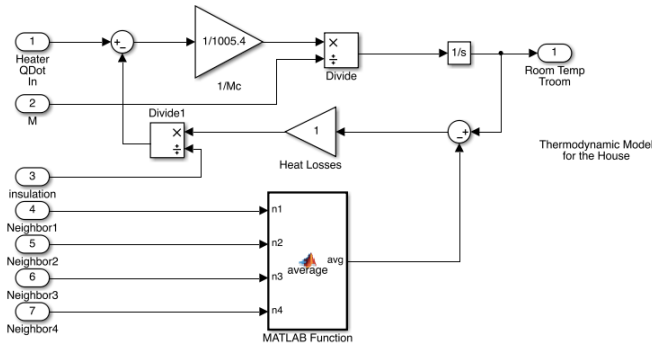


Figure 11: The House subsystem

3. SIMULATION CHALLENGES / CAVEATS

There are many simulation challenges associated with using the FMI standard (and its implementations) and interfacing with all-in-one simulators such as NS-3. While some have been overcome along the project, other have forced us to postpone some of our goals. Primarily, the first challenge is creating the FMU itself. The entirety of the heating model is written in Matlab, however, many FMI Toolbox exporters for Matlab/Simulink were incompatible with many of the Matlab/Simulink versions we tested. In addition, once the .fmu files are created, they must be run on the same operating system on which they were created. The development of a coherent process chain is necessary. The use of tools based around Modelica (JModelica/OpenModelica) instead of Matlab/Simulink is now in our action plan.

In addition, there were many difficulties with NS-3 as well. Mainly, we encountered difficulties running NS-3 network simulation across different operating systems. We also wanted to add “home automation related” protocols to the NS-3 network, in particular, incorporating Power Line Communication (PLC) links was originally on our radar. We tried using existing PLC frameworks, but unsuccessfully regarding our time constraints. This will be needed when going to the full Smart Grid simulation, but the specific protocols models will be provided when necessary.

4. SIMULATION RESULTS

We ran four initial simulation scenarios. We ran two simulations for each scenario: one entirely in Simulink using four room’s linked together, and one using the FMU room models connected with AA4MM-FMI. Both simulations for each scenario generated identical outputs at each time step, which was our validation step.

The “base case 4 room model” connects four basic house heating room models together. Each of the house four heating models are identical. The house heating room model for this simulation has the following input specifications:

- **targetTemp** - 21 degrees Celsius.
- **roomVolume** - 10,000 m³
- **insulationThickness** - .02 meters
- **surfaceArea** - 600 square meters.
- **windowThickness** - .01 meters.
- **outdoorTemp** - 15 degrees Celsius.

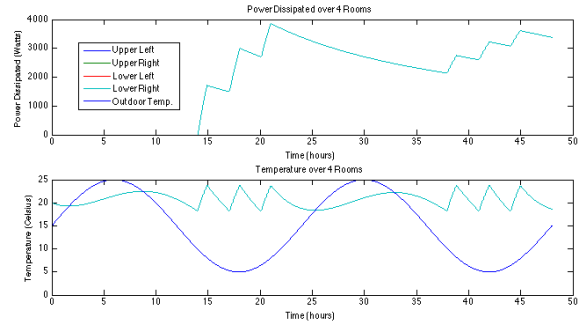


Figure 12: Base case 4 room model

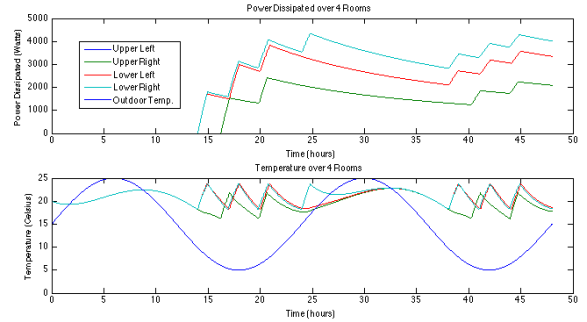


Figure 13: Cold 4 room model

- **wall1, wall2, wall3, wall4** - Two neighboring walls, and two outdoor temperatures.

We see that the power consumption and the temperature fluctuations for each room are identical in this situation, which is to be expected.

The “cold room model” connects three basic house heating models together, with one “cold” room in the upper right corner. The “cold room” has the exact same specifications as the basic house heating room model, except that its targetTemp is 19 degrees Celsius. Though 19 degrees Celsius is not necessarily considered to be “cold”, changing one target temperature in one room of the model does affect the power dissipated in the “cold” room and each of its neighboring rooms. This same “cold” room construct could be used to model garages, wine cellars, and various controlled environmental rooms.

The big room corresponding to the base case and the big room corresponding to the “cold” room generated similar results to their 4 room simulation counterparts.

5. FUTURE WORK

Our proposed smart space co-simulation is a small piece to a larger simulation environment. We hope to produce a smart grid neighborhood using the same constructs and ideas. As stated earlier, we also aim to fully incorporate the NS-3 network simulations into our smart space co-simulation and extend this to Smart Grids scaled environment within the MS4SG (Multi simulation for Smart Grids) project. Our partner EDF, an electric power utility company, will then

validate our models and tools with real data.

The integration of the FMI standard with our multisimulation framework (AA4MM) is successful but needs to be validated regarding scaling issues. We plan to add more "bricks" (rooms, networked objects) and more complex models (networks) and evaluate the performance of this solution compared to pure "mono-tool" or pure "mono-standard" ones.

Regarding the heating simulation, we also hope to incorporate and develop the capability to produce smart space rooms with more complicated geometric information. At this time, we limit our rooms to have exactly four neighboring rooms. However, we also want to examine multi-story smart spaces, and spaces with more complicated housing geometries (like rooms with various shapes and sizes). We also want to change the room models to support changing the heater position in the model. Currently, each room has its own heater without a power constraint. However, it would be interesting to look at smart spaces where rooms share heaters. It would be particularly interesting to examine power dissipated by the system when rooms are not guaranteed to have their own heater.

We also acknowledge that this model is fairly simplified. First of all, the current indoor temperature at each time step is calculated as the average of the neighboring temperatures. In addition, we assume that the constants provided by the Matlab house heating model [1] are accurate without a sensitivity analysis. Further, many of the calculations for air flow and resistance are simplified. More complex and detailed calculations would be needed for a more precise model. Finally, we also use a very simple initialization process for our simulation. Each FMU is given their initial values, and they are updated in a round-robin order at each time step. However, in larger and more complex system, we would have to designate a more elaborate initialization process. We also acknowledge that the paper is missing a performance analysis. In general, the simulations run faster with AA4MM-FMI, but we will quantify and analyze this result with larger models in our future work.

6. CONCLUSIONS

In this paper, we have defined and simulated a smart space heating co-simulation environment. This is a preliminary work seen as a small-scale vision of what would be required to have a full-scale smart evaluation toolkit. Firstly, we have developed and tested the AA4MM-FMI framework as a "master" in FMI with a decentralized view and intrinsic synchronization. Secondly, within our smart space simulation, we have taken a component-based strategy. We split our smart space environment into individual FMU rooms to create a hierarchy of smart space models and simplify the distribution of multiple components simulation. These proofs of concept need to be evaluated against larger scale problems, using our distributed version of AA4MM. Moreover, our ongoing work shows the capability for the integration of other models and simulators like HLA federates, or direct AA4MM that are currently being integrated in MS4SG.

We are also developing the idea of a macro simulation in the context of smart space heating scenarios. Using a large single room model is nearly equivalent to our 4 room con-

nected model: the "aggregated" result (energy/power) will be similar. At a high level, the 4 room connected model will be able to retain heat better than the large single room due to the increased number of walls and insulation. However, the biggest difference is that the individual rooms will have different behaviors and comfort temperature can be modulated. Depending on the level of details, the scale of our simulations, and their overall computation cost, multiple combinations of "single rooms" or their more detailed "multi-room" models can be built. It can pinpoint specific low level behavior while maintaining the total number of "FMU" and the computation and data exchange levels within reasonable limits. We now aim to create a macro-scale smart space neighborhood, and eventually a smart grid using the same constructs defined in this paper at a higher level of abstraction. The small FMU rooms in the models will be combined into housing units, instead of individual rooms. Using this higher level of abstraction, we will be able to define more intricate and realistic smart grid macro-level simulations from our initial micro level smart space housing model approach.

7. REFERENCES

- [1] House Heating System. The MathWorks, Inc., 2007.
- [2] Alessandro Ricci and Mirko Viroli and Andrea Omicini. Give agents their artifacts: the A&A approach for engineering working environments in MAS. In *AAMAS'07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–3, New York, NY, USA, 2007. ACM.
- [3] T. Blochwitz, M. Otter, J. Akesson, M. Arnold, C. Clauß, H. Elmqvist, M. Friedrich, A. Junghanns, J. Mauss, D. Neumerkel, H. Olsson, and A. Viel. The Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In *9th International Modelica Conference*, 2012.
- [4] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Clauß, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J.-V. Peetz, and S. Wolf. The Functional Mockup Interface for Tool independent Exchange of Simulation Models. In *8th International Modelica Conference*, 2011.
- [5] B. Camus, C. Bourjot, and V. Chevrier. Multi-level modeling as a society of interacting models. In *Spring Simulation Multi-Conference, Agent-Directed Simulation (ADS) Symposium*, 2013.
- [6] J. Siebert and L. Ciarletta and V. Chevrier. Agents and artefacts for multiple models co-evolution. Building complex system simulation as a set of interacting models. In *9th Int. Conf. on Autonomous Agents and Multiagent Systems - AAMAS 2010*, Toronto, Canada, May 2010.
- [7] Julien Siebert and Joris Rehm and Vincent Chevrier and Laurent Ciarletta and Dominique Mery. AA4MM coordination model: event-B specification, RR-7081. Technical report, INRIA, 2009.
- [8] T. Leclerc, J. Siebert, V. Chevrier, L. Ciarletta, and O. Festor. Multi-modeling and co-simulation-based mobile ubiquitous protocols and services development and assessment. In *7th International ICST Conference on Mobile and Ubiquitous Systems -*

Mobiquitous 2010, Australia Sydney, pages 273–284. Springer, 2010.

- [9] T. Leclerc, J. Siebert, V. Chevrier, L. Ciarletta, and O. Festor. Multi-modeling and co-simulation-based mobile ubiquitous protocols and services development and assessment. In *7th International ICST Conference on Mobile and Ubiquitous Systems - Mobiquitous 2010*, Sydney, Australia, Dec. 2010.
- [10] J. Siebert, L. Ciarletta, and V. Chevrier. Agents and artefacts for multiple models co-evolution. Building complex system simulation as a set of interacting models. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, volume 1, pages 509–516, Toronto, Canada, May 2010. ACM.
- [11] E. Widl, A. Elsheikh, P. Palensky, M. U. Awais, and S. Matthias. The High Level Architecture RTI as a master to the Functional Mock-up Interface Components. *2013 International Conference on Computing, Networking and Communications (ICNC)*, pages 315–320, 2013.