Simulation based design environment for multi-agent systems in buildings

Yahiaoui, A., Hensen, J.L.M., Soethout, L. & Paassen, A.H.C. van (2006).
Simulation based design environment for multi-agent systems in buildings.
Proc. 7th International Conference on System Simulation in Buildings, December 11-13, pp. 19. Liège: University of Liege.

Simulation based Design Environment for Multi-Agent Systems in Buildings

Yahiaoui, A., Hensen, J., Soethout, L., & Paassen, A. H. C. v.

ABSTRACT

With increasing experience and understanding of the behavior of users in buildings, it is very often difficult to properly build a control system that operates in the real world. To explore such a potential, this paper addresses a new approach to building automation systems that utilizes hybrids systems in order to model large scale systems typically arising in multi-agents. In fact hybrid systems are crucial for solving complex problems and for designing real-time controllers that can be used to automatically regulate HVAC (Heating, Ventilation and Air-Conditioning) systems and building components. A statechart formalism is also used for modelling of the entire building system behaviour in the structural analysis paradigm, in order to achieve a comfortable indoor climate while fulfilling operating constraints. Particularity, this paper concerns the relevance and reliability of integrating control and building performance simulation environments by run-time coupling, over TCP/IP protocol suite. In addition, this paper involves a case-study with two important steps; first consists of experiments obtained in TU Delft test-cell, and then simulation results are obtained with the use of run-time coupling approach.

1. INTRODUCTION

Modern building automation systems are expected to provide an enhanced functionality of interactive control strategies and of systems management resulting better control, as well as an improved reliability with further reduction in energy costs. Accordingly, building automation and control systems (named BACS or BAS) starts to become well established and more sophisticated by offering a vast diversity of control functions for all the systems that operate within building environmental performance. In BAS, an open distributed control systems composing of various components (sensors and actuators), control systems and an open network interconnecting them, is used to ensure the communication of information with a central computer through routers. For assuring the control analysis and performance of various building equipments and components with the important data exchange on the network, as depicted in figure 1, the simulation of distributed control systems is required. The main goal of BAS is the automation of HVAC (Heating, Ventilation and Air-Conditioning) and lighting systems in order to provide a healthy, comfortable and productive indoor environment while minimizing energy consumption, and to control them in an efficient and rational way (see Yahiaoui et al. 2005). They also perform tasks such as access control, energy management while reducing greenhouse gas emissions, and fault detection and diagnoses. In case when a BAS configuration exploits information between central computer and sub-stations (or terminals) through different network protocols like LonWorks and BACnet, it is necessary to use a router since those protocols utilize different network representations (see e.g. Kastner et al. 2005).

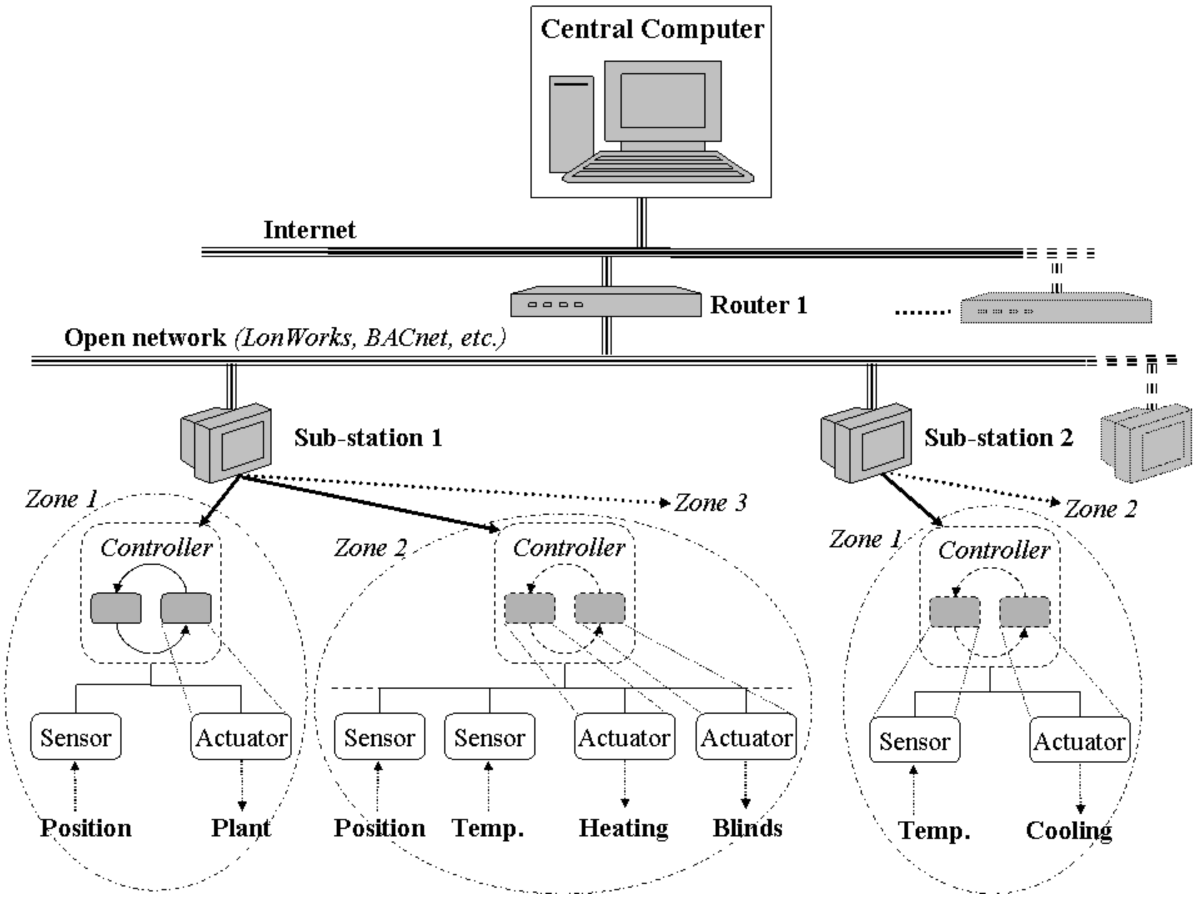

Figure1. Architecture of Building Automation Systems (BAS)

One important complexity in a cooperative simulation between the control system and the plant model of BAS is that most current software tools developed for building performance simulation are of limited use since do not have a flexible way of dealing with control strategies. With this regard, there is no single control method or conventional control application that can solve all the challenges encountered in buildings. An integral approach is, then developed and implemented in order to simulate the comfort and energetic aspects of building design in combination with any control strategies. This approach is therefore based on distributed control modeling and building performance simulation environments by run-time coupling. In this approach, building model and its control system, which are separated and exchange data through Internet socket during simulation can be located on different kinds of hosts. As a result, model-based advanced control

methods such as optimal control, fuzzy logic, and predictive control and so on; which are fast, accurate and robust can now be performed for building performance simulation.
In addition to building performance simulation, control systems of plants and building dynamics require specific methods of specification that can verify the sequence of motions of those plants. Structural methods are useful for a real-time control to easily handle building applications. Although HVAC systems consist of physical (mechanical, hydraulic, electrical, etc.) components and exhibit a mix of discrete and continuous behavior, hybrid systems are essential because they are characterized by interacting continuous-time dynamics (modeled by differential equations) and discrete-event dynamics (modeled by automata). Nevertheless, a hybrid automaton is a Finite State Machine (FSM) formalism extended with discrete states and continuous events. While a perspective used for complex systems consists to divide a system into a set of sub-systems such as sensors, actuators, plant and control laws; the complexity of sub-systems (or components) can in turn be divided into a set of sub-components, and so on. To overcome the parallelism and the dependence of the actions of two or more concurrent sub-systems that can exponentially increase the number of states and transition in hybrid model, StateCharts developed by Harel (1987) can in effect be used as a visual formalism for the specification and modeling of building complex systems. The importance of the modeling with hybrid systems has been demonstrated in various applications such as in aeronautic (Tomlin et al. 1998), in automobile (Lygeros et al. 1998), etc. One way to represent the behavior of hybrid systems is the hybrid automaton. This makes it possible to combine discrete-event or mode switching dynamics with continuous evolution according to differential equations or inclusions. States take values over real and discrete sets. Figure 2 illustrates a complete configuration of hybrid systems.

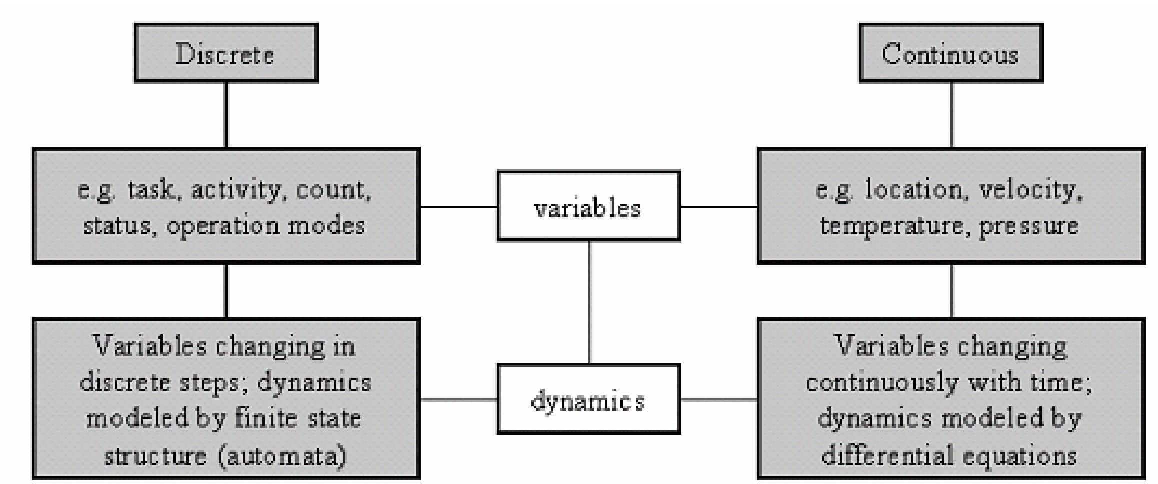

Figure 2. A typical structure of hybrid systems

Hybrid systems typically arise in the computer control to allow a convenient modeling of complex physical systems, e.g., systems containing mechanical, electrical, hydraulic, thermal, and etc. The modeling domain can be the abstraction of continuous systems with phased operation (windows, blinds, … etc.), continuous systems controlled by discrete inputs (switches, valves, … etc.), coordinating processes (Multi-Agent Systems) and so on. A key advantage of this approach offers over existing methods for control and coordination of multi-agent systems is that it does not destabilize the effect of internal dynamics of each agent when agents are abstracted away. For such a reason, hybrids systems are better suited to model large scale systems typically arising in multi-agents in order to have stable internal dynamics. As a result, the concept of *Hybrid StateCharts* integrating discrete logic events and continuous time dynamics in the usual way is developed for the purpose of modeling and designing real-time controllers for integrated building systems.

A simple hybrid StateChart model for a building heating system is decrypted in figure 3. This heating system has two variables: the set-point temperature T_{sp} and the inside air temperature T_{in} and two discrete states *A* and *B*, which they refer to the functioning mode of the heating system. Both states evolve continuously, in which they are described by differential equations. Another variable of negligible temperature $T_{deadband}$ is fundamentally used for the motion that can represent the system internal parameters for real-time specification (e.g. the sensitivity of the controller, the delay of the controlled variable response, and etc.). The automaton starts by default in state *B* where the heater is completely switched down. At transition from the state *B* to the state *A* happens once the value of the internal temperature T_{in} drops below the reference temperature T_{sp} and this must happens just after this transition is verified. When this transition happens, the value of the internal heating rate increases instantaneously within the variable heat flux at between 0 (when the heater is fully off) and 100% (when the heater is entirely on). The transition back to the state *B* happens once the internal temperature T_{in} is above the set-point T_{sp} .

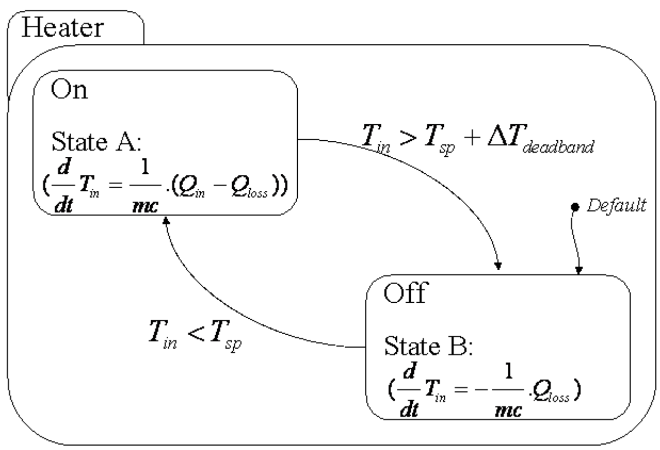

Figure 3. A simple *hybrid StateChart* of the building heating system

Now it should be understandable that this system involves the management of two important steps: the first consists of a logical making decision (e.g. when a transition happens), and the second is a continuous state evolution (e.g. which heat rate needed accordingly). A hybrid control system is therefore used to accomplish proper solutions when complex building systems need to be controlled properly. Furthermore to alleviate those inherent complexities of building control systems such as making decision, learning about occupants, adapting suitable control behaviors and solving problems including needs and constraints. All those building systems can in effect be decomposed and modeled as systems of multiple (intelligent) agents (Wooldridge, 1999). Also Lygeros (et al. 1996) showed that a methodology of designing hybrid controllers for large scale with multi-agent systems based on an optimal control and a game theory. In consequence, this paper presents an approach to defining an agent as a control system (or a zone process controller as shown in figure 1), which is capable of sensing, computing and acting while

fulfilling environmental and operating constraints. The design of those agents can be enhanced so that they act within a flexible autonomous manner to optimize the attainment of their objectives and to affect their surroundings in a desired manner.
The first part of this paper describes a brief description of related work. The next part elaborates the reasoning behind our hypothesis that distributed control and building performance simulation software by run-time coupling will facilitate integrated performance assessment by predicting the overall effect of innovative control strategies for integrated building systems. Then, an analysis to the specification of multi-agent architecture for integrated building systems is described in detail. This is followed by the synthesis related to the design of hybrid control systems for integrated building systems in feedback structure. The last essential part of this paper ends with a case study resulting in finding a balance between theoretical aspects and practical applications.

2. RELATED WORK

In the area of agent based building control systems, a distributed intelligent control application can be enhanced by the introduction of multi-agent systems (MAS) or hybrid intelligent control agent (HICA) formulation. One of principal tasks in this work is based on a cooperative multi-agent architecture using hybrid systems for Intelligent Buildings (IB) of the third generation (see e.g. Sharples et al. 1999). This third-generation Intelligent Buildings is formed when Building Automation Systems (BAS) is based on the capacity of learning about the building environment and its occupants in order to adapt suitable control laws and appropriate set-points accordingly for the current situation. In (Sharples et al. 1999) work, three-generation Intelligent Buildings have been described. The first-generation consists of several control systems or sub-systems in which they operate independently to each other. The second-generation takes place when BAS exploits networks to assure distributed control systems, and the third-generation have in addition to previous (first and second) generations, an extra functionality that arises from the ideal control applications to the intelligent control systems.
Although multi-agent systems (or agent-based control models) are used in the domain of Artificial Intelligence (AI), Information Communications Technology (ICT) and so on; a multi-agent architecture can also be used for designing and modeling distributed autonomous agents that can be exploited in a building environmental performance. In the case of BAS, a cooperative between the controller and the mechanical plant model is required to verify the sequence of functioning activities of the entire system. Hence, it further understood that a simulation is strongly needed in order to evaluate the control behavior in the building environment and to illustrate how distributed multi-agents can react in buildings. In this paper, the main goal aims to derive a useful, general approach to the design of hybrid multi-agent systems for complex systems. This is important for the third-generation Intelligent Buildings (IB) to assure the control performance of distributed building systems in the BAS coordination. In fact by interconnecting building control systems via a network, it becomes possible either to remotely regulate any plant from the management level directly (i.e. from the central computer) or locally control plants since regulators are positioned in an automated level. Another important key concept of development BAS is distributed building control systems that should initially concentrate on design requirements and fundamental principles encountered. A methodology based on hierarchical semantics of multi-agent hybrid systems can now be developed to cover various design stages including modeling, simulation, analysis, implementation, and monitoring of control systems in building design. However, difficulties in implementing appropriate controllers

in buildings motivate us to investigate hybrid StateCharts based on a simple building application as a solution to controlling building HVAC systems, in which time delays, nonlinearities, constraints and so on can be addressed. In addition, integrating hybrid systems together with model based modern (or intelligent) control techniques in the form of multi-agent hybrid systems can solve the complexity of performing autonomous agents in buildings.

2.1 Contributions

One of the main contributions in this paper is to develop a cooperative simulation based design environment for multi-agent systems in buildings. This cooperative simulation use Web services for communications in distributed control and building performance simulation by run-time coupling. The originality of this work consists of generality in terms of extending a mechanism based run-time coupling of ESP-r and Matlab/Siminlink to simulate the environment of using multi-agent systems in building. Although a methodology developed in this paper is based in the encapsulation of data exchanged between ESP-r and Matlab with Web services, both control and building performance applications can run on different operating systems (Windows, Unix and Linux). In this fact, the current distributed simulation mechanism is completely transparent for the users. Consequently, an approach based on the simulation of building automation systems (BAS) can be enabled by similarity to the cooperative simulation of distributed control and building environmental performance applications by run-time coupling. In addition, various protocols used by BAS system are highlighted to be supported within distributed simulation.

3. DISTRIBUTED CONTROL & BUILDING PERFORMANCE SIMULATION

One of several key issues facing us when we want to simulate building and plant models plus control application is that frequently certain system components and/or control features can be modeled in one simulation environment while other components and/or control features are only available in other simulation tools. However, there is domain specific software for building performance simulation (BPS), which is usually relatively basic in terms of control modeling and simulation capabilities (e.g. ESP-r, TRNSYS). On the other hand, there exists a domain for control modeling environments (CME), which is very advanced in control modeling and simulation features (e.g. Matlab/Simulink), but still not totally for building environmental performance. To alleviate such a restricted issue mentioned above, it is necessary to reason behind our hypothesis that marrying two approaches by run-time coupling would potentially enable integrated performance assessment by predicting the overall effect of innovative control strategies for integrated building systems. Additionally, the strategy to run-time couple ESP-r and Matlab developed in this work takes the form of closed-loop control system, as shown in figure 4. This can be performed either in the feedback control loop or in the feedforward control structure. Both control loops can then be used for any simulation of building control systems. In case an open-loop control is performed for building application, it would just necessitate to setting in Matlab side the variables required passing back to ESP-r to zero. More exactly as defined in Mex-file language, those variables are the *rhs* (right hand side) arguments. However, both unidirectional and bidirectional communications can now be carried out within this developed strategy for run-time coupling.

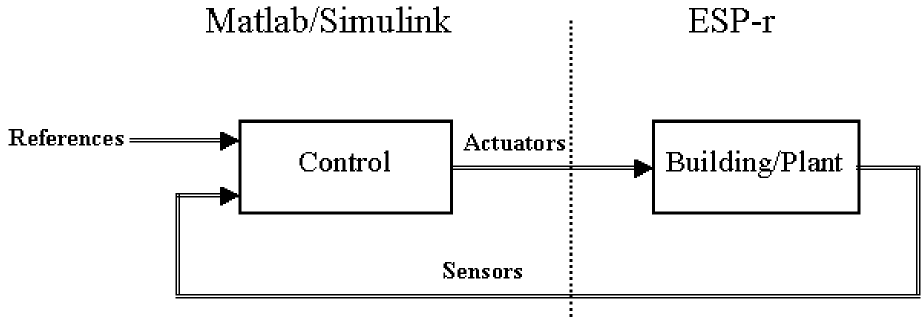

Figure 4. Structure of run-time coupling between control modeling and building performance simulation

In previous work of (Yahiaoui et al., 2003 and Yahiaoui et al., 2005), it has been described that a promising approach to run-time coupling between ESP-r and Matlab/simulink is an IPC (Inter-Process Communication) using Internet sockets. This approach performs a distributed simulation through an open network protocol to exchange data between building model and its controller, as it almost happens in the BAS system. Both building model and its controller which are separated and work together through run-time coupling can be located on a different kind of hosts (Windows, Unix or Linux), in which the performance simulation is much faster than using a single computer. Consequently, the development of this new advent would potentially enable new applications of building control strategies that are not yet possible.
However during the simulation, commands and data are transmitted between ESP-r and Matlab/Simulink. If for instance the building model (i.e. ESP-r) has to send its current measured variable to its controller (i.e. Matlab/ Simulink) with TCP/IP-stream format, a method called encodes and transmits them with a defined control sequence via TCP/IP to a method received. This then receives the control sequence, decodes data from TCP/IP-stream format and sends them to the recipient (Matlab/Simulink). When the controller has to send back the actuated variable to its building model via TCP/IP suite, a new procedure is executed again in the opposite way. Figure 5 illustrates a complete understanding on how the data are exchanged between ESP-r and Matlab/Simulink by run-time coupling.

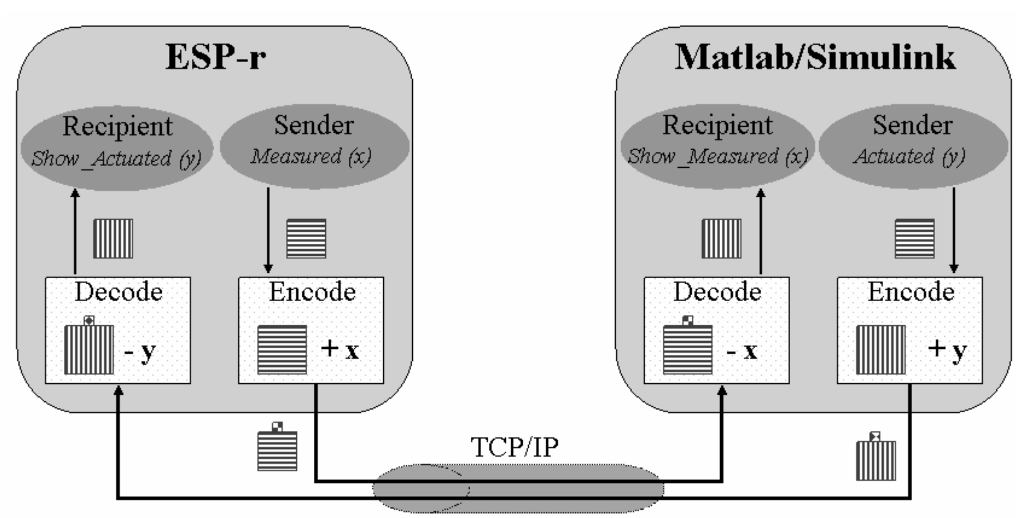

Figure 5. Distributed control and building performance simulation environments

In the current implemented approach of run-time coupling between ESP-r and Matlab, it is ESP-r which starts simulation. Indeed, Matlab is launched at every ESP-r time-step as a separate process. If the connection between ESP-r and Matlab breaks down the data to be exchanged cannot be transferred until the communication between them is reconnected. More detail about distributed control modeling and building performance simulation by run-time coupling can be found in (Yahiaoui et al., 2004 and Yahiaoui et al., 2005).

3.1 Data communication in a real building

In real building operations, it is desirable for control laws to communicate with building components or HVAC systems. The communication between controllers and building systems requires an interface or a gateway, due to their different communication speeds and data formatting. The proper operation of the gateway is dependent on the continued use at the corporate level to ensure the communication between controllers and plants and building components. In consequence, two important standard protocols "BACnet and LonWorks" (see e.g. AutomatedBuildings, 2002) are mostly exploited for building automation. Although BAS (building automation systems) can use TCP/IP as the underlying connection protocol for automated buildings, a data exchange protocol such as XML is dominating the future of interoperability among protocols. With the introduction of run-time coupling of ESP-r and Matlab/Simulink, the approach developed for exchange data between them provides a flexible TCP/IP connectivity to any application of building automation and control systems. By using this Internet protocol and a common Web documents, run-time coupling mechanism can play the role of BAS functions with more or less the same similarities.

However, both protocols cover most or all layers of the OSI (Open System Interconnection) model used for network communications. The similarity of approaching a run-time coupling mechanism to BAS system concerns several ideas developed by considering basically the top-layer of the OSI model (i.e. application layer). The other layers deal with hardware specifications and the low-level details of the network communication. In our work, these details are already covered by the socket implementation. Though the application layer deals with the conceptual part (i.e. with the way of information is modeled), both protocols (BACnet and LonWorks) use different mode of exchange. What follows is a brief overview:

- BACnet (ASHREA 135, 2001) is a specification for a standard protocol published by ASHRAE organization. While the data exchanged in BACnet is encapsulated in objects accessed by services calls, BACnet can ensure the interoperability between devices of different industries. In consequence, the run-time coupling mechanism is implemented in order to create a communication protocol that complies with this specification. The standard defines protocol implementation conformance statements (PICS) that identify different levels of compliance. This meets a field level of building automation services (BEMS or BAS), in which run-time coupling can use various standardized set set of data types to communicate between ESP-r and Matlab as it is over BACnet protocol.
- LonWorks (Loy et al. 2001) was not developed for automated building only, but now it is gaiting a high importance in this field. While supporting communications on a variety of LANs or allowing different components to co-exist in the same network, LON (or LonWorks) communicates via Standard Network Variable Types (SNVT) to facilitate

interoperability by providing a well-defined interface. Basically a proprietary communication protocol is called LonTalk, created by ECHELON Corporation. A chip is required for any device that uses LON. Consequently run-time coupling can also establish standard network variable formats to allow the transfer of data between Matlab and ESP-r by using the LonMark subset of LON capabilities to interoperate with each other.

It is believable that using Web services on BAS architecture would provides a great flexibility to communicate between diverse devices and through various protocols designed with different technologies. In order to similarity approach the utilization of distributed control and building performance simulation by run-time coupling to BAS system, a data exchange format is then developed with Web services in C/C++. The gSOAP (Engelen, 2006) is used in this work to generate efficient Web services and user-defined C/C++ data types. Accordingly, Web services interoperability is achieved with sockets API (Application Programming Interface) to run-time couple of Matlab/Simulink and ESP-r with practical logic applications of automated buildings.

3.2 Data exchange format

The format for data exchange protocols exploited in this work is based on using both XML (eXtensible Markup Language) and SOAP (Simple Object Access Protocol) Web services in order to communicate between ESP-r and Matlab/Simulink with any data interchange format. In reality, SOAP was created to exploit XML documents based protocol in order to communicate between different applications running on different operating systems, with different technologies and programming languages. Although most BAS architecture can use several protocols of different technologies, it is essential for the run-time coupling mechanism to adopt a certain capability of exchanging data between ESP-r and Matlab in heterogeneous distributed environments. Figure 6 illustrates how Web services based SOAP/XML is implemented to run-time couple of ESP-r and Matlab on different operating systems.

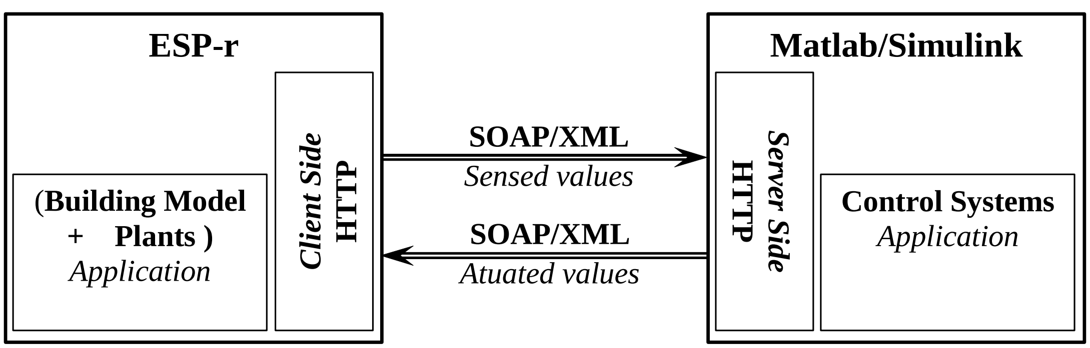

Figure 6. Impletation of Run-time coupling between ESP-r and Matlab/Simulink using Web services based SOAP/XML

To communicate between Matlab and ESP-r in distributed heterogeneous environments, XML data binding is used to manipulate data easily with the XML-Documents via a set of simple objects. But to transfer XML-Documents using HTTP (HyperText Transport Protocol) between ESP-r and Matlab, it requires to also use SOAP in order to ensure packaging, encoding, and exchanging of structured data on run-time coupling. This structured data is then described in the form of XML-Documents. In fact, SOAP is most commonly associated with Web services, serving as the de-facto standard message envelope (O'Reilly Media, 2006). Figure 7 illustrates an example of programming code based SOAP message containing an XML-Document of two elements needed at least for data exchange between ESP-r and Matlab/Simulink environments.

```
<?xml version="1.0"?>
  <soap:Envelope xmlns:soap="http://www ...">
     <soap:Header>
       ... ...
     </soap:Header>
     <soap:Body> ... ...
        <soap:Elt1> ... ... </soap:Elt1>  <soap:Elt2> ... ...</soap:Elt2>
     </soap:Body>
  </soap:Envelope>
```

Figure 7. Sample SOAP message for a travel reservation containing XML data binding

4. ANALYSIS TO MULTI-AGENT SYSTEM SPECIFICATION

Before delving into the specific specification of interactive control systems and building environmental performance, this section describes firstly how the analysis of automated building can be mapped to the design of multi-agent systems. The methodology based multi-agent systems engineering (e.g. DeLoach et al. 2001) is similar to traditional software engineering methodologies. In this paper, the methodology follows the basic steps of the *waterfall* process model since it is the most relevant diagram applied in software development industry. Figure 8 shows a Waterfall model which is structured as a cascade of several phases required for the simulation based design environment for Multi-Agent Systems (MAS) in buildings. This simulation consists of distributed control and building performance simulation by run-time coupling. One of the advantages of using waterfall diagram is that the output of the current phase constitutes the input of the next one, which can be achieved until the previous phase is accepted. The development of this design environment lifecycle begins with the identification of requirements required for run-time coupling between ESP-r and Matlab and ends with the formal verification of this developed cooperative simulation mechanism against those requirements.

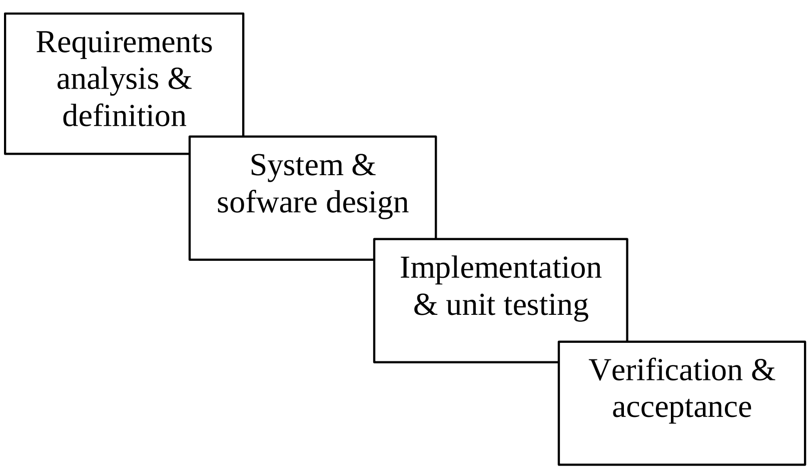

Figure 8. The Waterfall lifecycle model

The waterfall model represents an experience-based refinement of the classical sequential system life-cycle model. This approach modifies the sequential method used by the classical life-cycle model and advances an incremental development strategy. One advantages of this diagram is based on simple detection of a relevant number of design phases desired for the development of multi-agent architecture. The phases of this development model are described in order of importance to be adapted for the current research, but several versions of this diagram can be established for different fields (see e.g. Cheong et al. 2005). A number of phases can also appear differently in another process model such as V diagram, spiral model and so on, and include:

- User requirements – collecting of occupant's needs for the control system (or agent) to be;
- System requirements – specifying the requirements into a set of functionalities and proprieties (i.e. like climatic conditions, changes …) to be provided by the agent to be;
- Design of multi-agent architecture to be;
- Implementation of multi-agent architecture;
- Verification and acceptance of multi-agent systems for operation in building environment.

Although requirements analysis for an agent system is typically translated into a set of goals to be pursued by that agent, the development of multi-agent systems is involved in those phases. Using this approach, it is clear that an agent role in conjunction with a link between tasks (or sub-agents), as shown in figure 9, determines the behavior of each the sub-agent will have. Nevertheless, external events need to be captured from the tasks to coordinate in the design of the agent so that it can behave consistency with the initial concurrent tasks. Internal events should be captures in the design as well. However, based on this approach, figure 9 illustrates how the role models in the analysis phase are transformed to multi-agent systems in the design phase as well as the relationship between multi-agents.

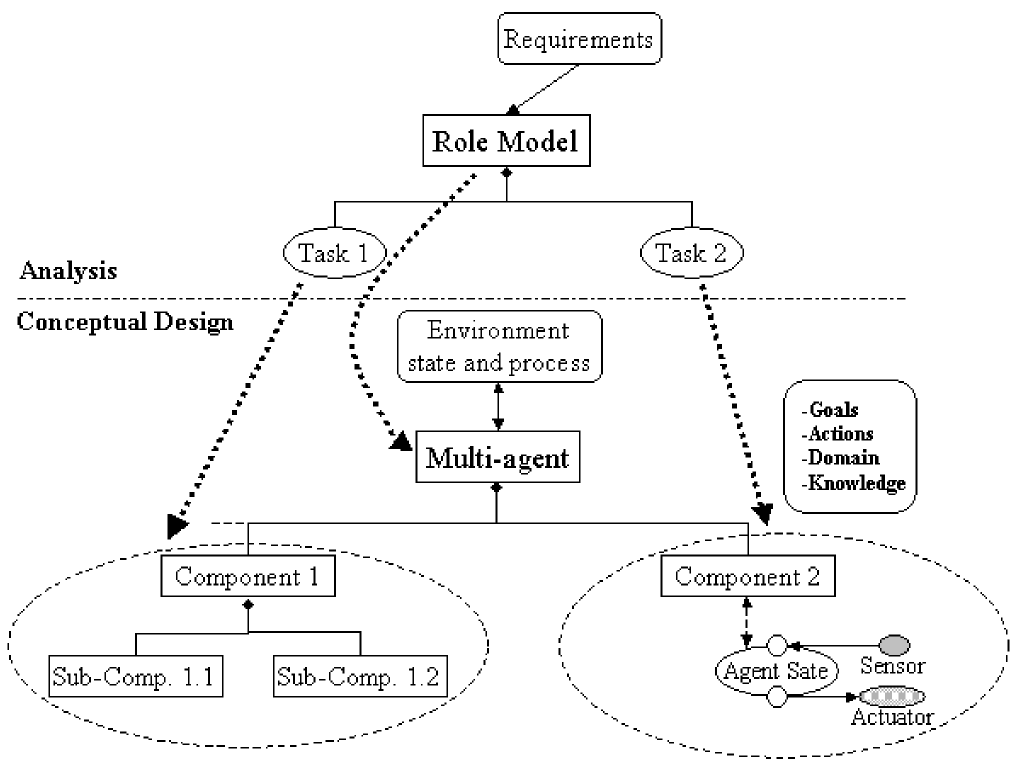

Figure 9 Multi-agents in a building environment

The formal specification of compositional architectures for role models is based on task-based approach to the design of multi-agent systems. As a result of task analysis, hierarchical role models are specified at different levels of abstraction, as the interaction between tasks. Each task can in effect be assigned to one or more agents. Agents themselves perform one or more (sub-) tasks, either in parallel or sequentially. The agents can be designed with a knowledge base to behave by certain intelligence.

5. MATHEMATICAL FORMULATIONS

In this section, an application for a building case-study model is presented as shown in figure 10. This application comprises a test cell unit of dimensions (3.15*3.85*2.6 m^3) constructed in TU Delft with light construction materials that has for the purpose to investigate different causes that influence the indoor environment of passive solar buildings. Those causes can include natural ventilation, radiant or solar heat gain and heat loss coefficient.

5.1 Heating System

A mathematical simple model of the heating system is represented as the rate change of the temperature difference in the heat flow Q_{in} supplied by the heater, and the heat rate Q_{loss} lost through the wall insulation, related by the following equation:

$$mc\frac{d}{dt}(T_{in} - T_{out}) = Q_{in} - Q_{loss} \qquad (1)$$

where m is the building mass (Kg), c is the average specific heat ($J/Kg.K$), Q_{loss} and Q_{in} are heat flow rates (J/s or W), and T_{in} and T_{out} are inside and outside temperatures (^oC).

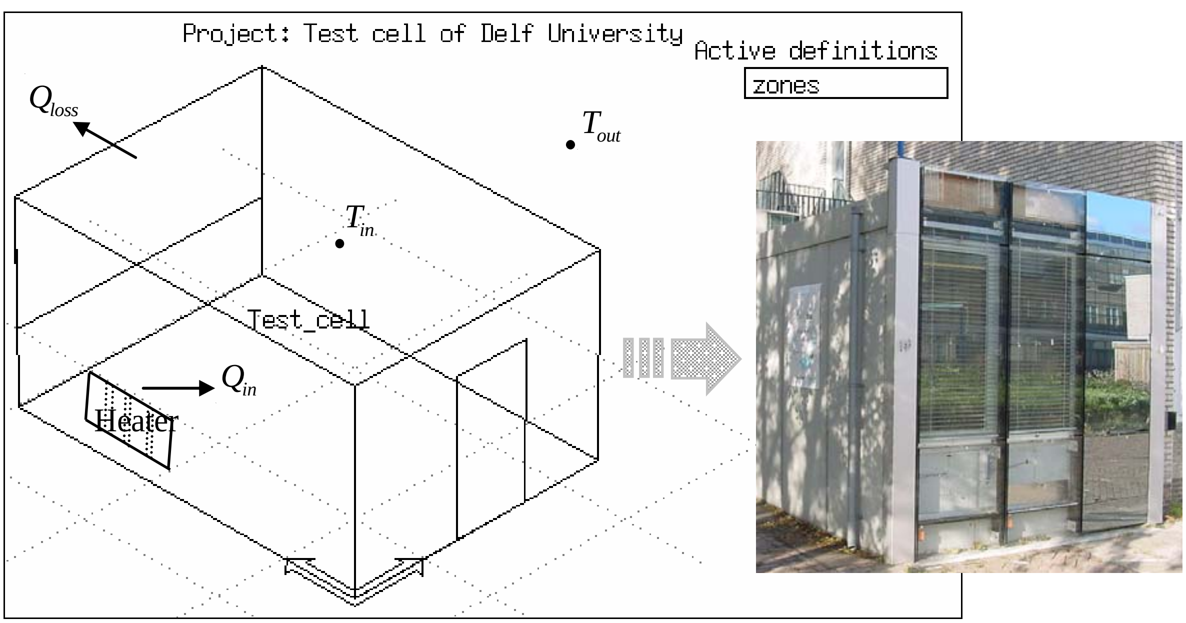

Figure 10. TU Delft test-cell case study

When the outside temperature T_{out} is constant (or very slowly varying), the relation given by equation *(1)* can became:

$$mc\frac{d}{dt}(T_{in}) = {V_h}^2 \big/ R - Q_{loss} \qquad (2)$$

where V_h is the heater voltage, and R is the electric resistance of the heater.

The rate of heat Q_{loss} lost through the wall insulation is proportional to the temperature difference across the insulation, in which it is given by $Q_{loss} = U_0(T_{in} - T_{out})$ *(3)*

where U_0 is a heat loss coefficient (W/K)

Submitting from equation *(3)* into equation *(2)* gives a relation in form of the state-space representation, which is as follow:

$$\frac{d}{dt}T_{in} = -\frac{U_0}{m.c}T_{in} + \frac{1}{m.c}Q_{in} + \frac{U_0}{m.c}T_{out} \qquad (4)$$

where the $\frac{U_0}{m.c}T_{out}$ factor is the effect of the disturbance input.

The value of c for this example consisted of using common proprieties for air temperature in which it is taken from table with respect to the average temperature of the building in wintertime, as mentioned in (ETB, 2005). On the basis of this table, c is something like $1.005\,(k.J\,/\,Kg.K)$. The value of m is also calculated with respect to density ρ, which is in the order of $1.205\,(Kg/m^3)$. The heat loss coefficient U_0 is calculated in relation of U-value defined by each area in relation with all areas of the room.

5.2 Multi-Agent Systems Modeling

In the design phase of multi-agents, the role model is taken in the analysis phase after having defined requirements which are specified in both problem and functional terms. Then an agent paradigm is produced where complex systems are partitioned into simple sub-problems that have limited mutual dependencies. However system decomposition and design of individual agent can sometimes be a problem since it is important to make decisions, hybrid systems can in effect be suitable in decomposition as they support capabilities in decision making. In the case of BAS as shown in figure 1, building zones are represented to explain how distributed control systems work in building. Practically, a building zone contains a controller which forces a building system to behave in a desired way in order to achieve a certain goal. Even though a building zone disposes more than one controller, the objective is the same where system goals need to be accumulated. In order to ensure that all system goals are being satisfied, each role must be played by at least one agent where a traceable link is provided from the goals in the analysis phase to the agents in design phase.

Formally, an agent is described as $H = \langle M, X, I \rangle$, which consists of a set X of variables, a set I of initials states and a set M of modes. The set of variables X are partitioned into input and output variables. The set of initial states I specifies possibles initialisations of the variables of the agent and can further be used to correct the assignments of value to the variables. The set of modes M represent the bahavior of the agent in the system design.

In the context of BAS, each sub-station can be connected to several zones and each zone dispose at least of one controller when a distinct operation is required. In that fact, the controller is an agent and the operation of this controller is defined on agent. When sub-station is connected to more than one zone, agents are composed in parallel. The parallel agents execute concurrently and communicate through shared variables. To enable communication between agents, shared variables can be defined as global so that the agents distribute information without confusion. Finally, the communication between controllers located in different building zones is performed and distributed control systems in BAS environment can be simulated.

5.3 Semantics based Multi-Agent Systems for Real-Time Control and Simulation

A hybrid automaton is a dynamical system that describes the evolution in time of the values of a set of discrete and continuous state variables, as shown on figure 11. A syntax language used, in this paper, for modeling of both a mathematical and a graphical representation is similar to the syntax given by (Alur, et al. 1996). For example, Lygeros (et al. 2000) showed a modeling formalism that moves the game features explicitly into the hybrid dynamical model.
The hybrid automaton for the heating system has discrete and continuous components in its states, its control input, and its disturbance, as the equation *(4)* is described. The control objective is to maintain the temperature of the air in the room around the common set-point T_{sp}, whatever the disturbance happen to be. Then, the hybrid transition relation is given as

$$H = \begin{bmatrix} T_{in} \\ Heater_{status} \end{bmatrix} \in \Re \times \{On, Off\} \qquad (5)$$

A simple model of hybrid automata for building heating system is represented in figure 3. For heating system modes, the states are $Q = \{q_1 = (Off, no-heating),\ \ q_2 = (On, heating)\}$. The component of each state refers to the status of input disturbance variable $t_{out} \in [0, T_{out}]$ and temperature inside the room $\dot{T}_{in} = \{(T_{sp}, T_{in}) \mid (T_{sp}, T_{in}) \in \Re\}$.

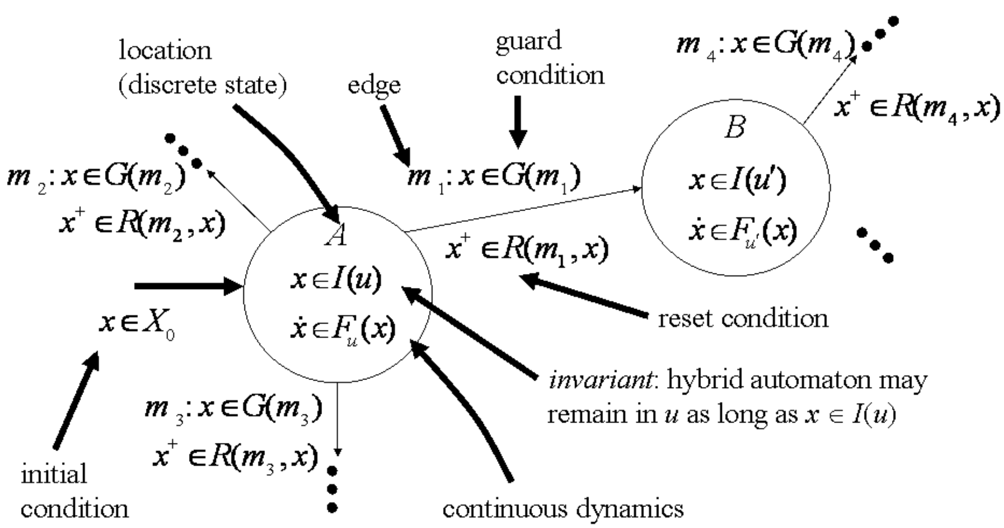

Figure 11. Hybrid automata

Let us assume that $H = \langle M, X, I, A, B, \phi \rangle$ is a linear hybrid system. A state of is pair (m, X_0) such that $\phi_s(X_0)$, that is, the invariant at m is true for X_0. The set of states is denoted by Q. For $q = (m, X_0) \in Q,\ \ q(x)$ is the value x_0 of x at the variation X_0.

H change a state either by a discrete and instantaneous move through an edge, or by a continuous transformation of its variables while time is elapsed. The semantics of H is a labeled transition system $< Q, \rightarrow >$ where the transition relation → between states is defined as follows:

- an edge $< m_0, a, \psi, e, m_1 > \in I$ is enabled at a state (m_0, X_0) if X_0 satisfies the guard ψ. Whenever an edge is enabled at a state if it may be taken, and the resulting state is a pair (m_1, X_1) where X_1 is such that the every variable $x \in X,\ \ x_1 \in v_x$ if $x := v_x$ and $x_1 = x_0$ else.
- time can progress by t at (m, X_0), if there is a valuation X_1 reached from X_0 for a rate $\wedge \in [A_m, B_m]$ and X_1 satisfies the invariant associated to m.

A trajectory σ is an infinite sequence $q_0 \xrightarrow{t_0} q_1 \xrightarrow{t_1} \ldots$ where for all $i \geq 0\ \ q_i \xrightarrow{t_i} q_{i+1}$ is such that the following expressions are introduced:

- $q_i \xrightarrow{t_i} q_{i+1}$, or
- there exists a label a such that $q_i \xrightarrow{a} q_{i+1}$ and $t_i = 0$.

A position π of σ is a pair (i, t) with $0 \leq t \leq t_i$. A total order $\leq$ on the position can be defined by the following rule: $(i,t) \leq (j, t')$ if $i < j$ or $i = j \wedge t < t'$

Let $q_i = (m_i, X_{0i})$ and $q_{i+1} = (m_i, X_{0i+1})$ where $X_{0i+1} = X_{0i} + \wedge_i.t_i$ with $\wedge_i \in [A_{mi}, B_{mi}]$. The state at position (i,t) of σ is $\sigma(i,t) = (m_i, X_{0i} + \wedge_i.t)$. The time elapsed at position (i,t) is $\tau_\sigma(i,t) = t + \sum_{j<i} t_j$, that is the time elapsed since the begenning of the transition.

By expressing the semantics for multi-agents using hybrid systems in buildings, the temperature of the room grows linearly with time until it get up to the reference. For comfort purposes, the temperature can be kept within the small variation around the set-point. A controller can cool the room either by switching on the cooler or by opening the windows if the outside temperature is colder than the one inside the room. Furthermore, each strategy can be used again if a given time has elapsed since the state of the previous event is executed. Figure 3 shows a hybrid statechart controller of the building heating system, in which it can be improved by taking into account more measured variables such as the time elapsed. Initially, the controller takes the state when the heater is off by default for safety raisons. The continuous equations at different states specify the evolutions laws of the variables corresponding to the situations of heating the room.

It should be noted that one of the major difficulty to design multi-agent systems in buildings lies in the incapacity of some methods to express in a generic way the possibilities to exploit the large scale systems. So, to deal with multi-agents in buildings, the hybrid systems can bring a solution to such expectations and to the expression of the agent specifications as well.

6. SYNTHESIS OF HYBRID CONTROLLER

With model-based distributed control and building performance simulation environments, the control algorithms are designed and performed offline building models of which the simulation can be ran on two computers. Hence, a generic diagram depicting the interaction of the controller and its building model is shown in figure 12. Finally it should be noted that the control systems (or controllers) implemented for experiments in Testcel are the same carried out for simulations.

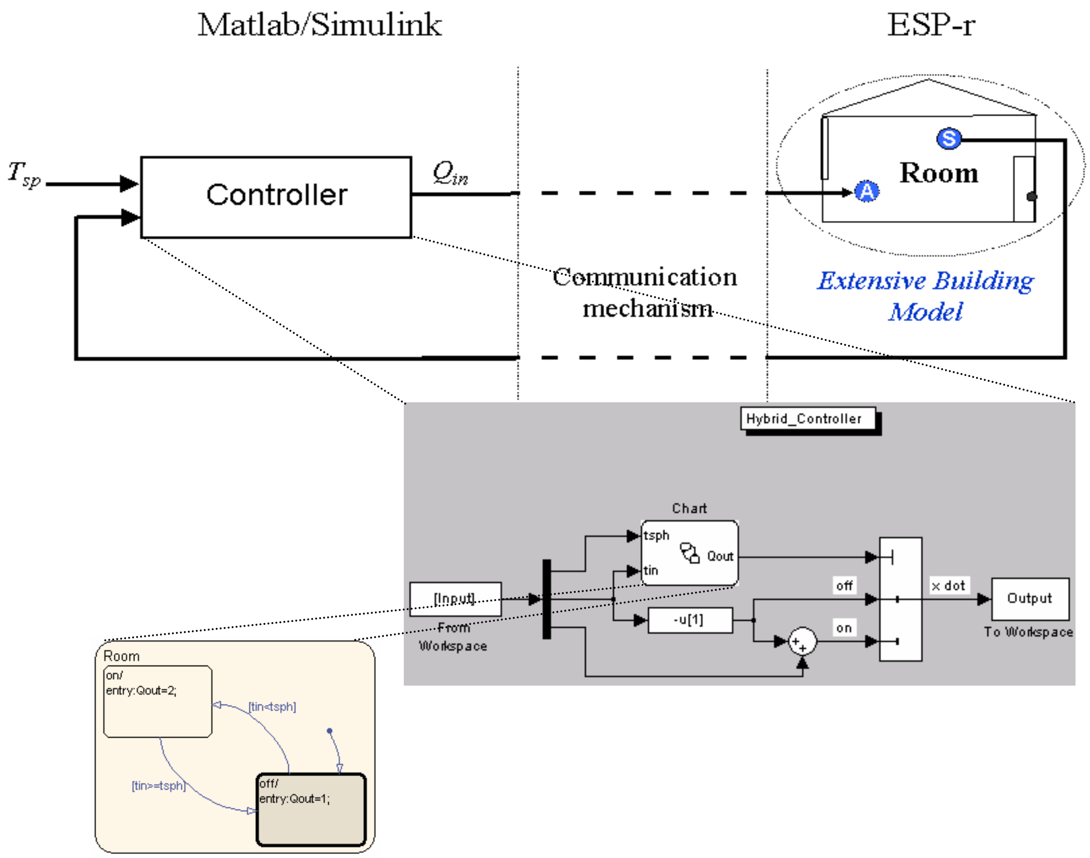

Figure12. A prototype hybrid control system for building plant model

In this application, both Simulink and Stateflow are used as graphical languages. Matlab is used for control and data-flow applications that mix continuous and discrete-time domains. Those tools are based on a particular mathematical formalism, language and necessary to analyze and simulate hybrid system. Figure 11 illustrates also a graphical environment of hybrid controller developed for the building heating plant. In order to integrate multi-agent systems in building performance simulation, the synthesis of model-based hybrid system is presented to generalize a specification of using a multi-agent architecture for integrated building systems. The formalism of Statecharts base hybrid systems is used to represent how an agent can behave in a building. This StateChart lies to the hybrid automaton where states represent all actions of the controller.

7. BUILDING TEST CELL: -CASE STUDY

The current case study is illustrated to investigate an application with two objectives. The first consists of comparing between experimental work and simulation results obtained by the same model-based hybrid controller within the same time step of 1mn/hour. The second qualifies the importance of the run-time coupling approach when it is necessary to apply advanced control systems in building performance simulation.

7.1 Experimental data

A test cell built in TU Delft with light construction materials is used with the aim of designing measurements by controlling the indoor temperature in the room. Although, sensors are installed more or less all over different places in the room to provide timely detection of potential temperature changes, the indoor air temperature is the average of all measures collected by those sensors. A model-based hybrid controller developed for real-time specification is implemented to actuate the electrical heater of 1750 (W) with proper amount of power appropriate for a current situation through a data acquisition located in the room during experiments, shown in figure 13.

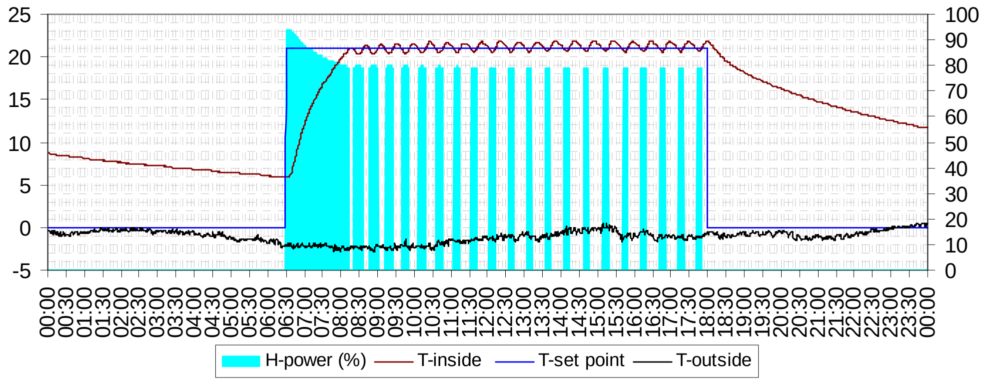

Figure 13 Experimental results

7.2 Simulation Results

A test-cell building model is implemented in ESP-r with new databases created to represent and carry out the same material proprieties that are practically used in the construction of the room, shown in figure 10. The climate measurements are partially integrated as well, but ESP-r considers their values on an hourly basis. The simulation results, shown in figure 14 are obtained within the same model-based hybrid controller implemented on the same (Matlab/Simulink) environments respectively, as shown in figure 13, which is actually realized for experimental work. Though the run-time coupling described approach above is used to exchange data between ESP-r and Matlab, Matlab is synchronously launched at every ESP-r time step as a separate process during occupied period (from 6:30 to 18 o'clock).

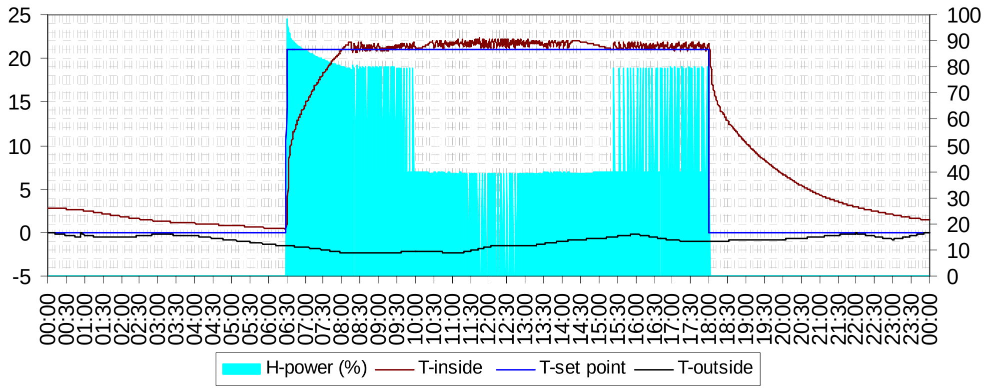

Figure 14 Simulation results

A detailed comparison between experiments and simulation results shows that there are small changes in both responses of the controller used for the indoor temperature in the test cell. Those changes are due to the climate data that highly influence the temperature inside the test cell. In fact this outside temperature is a disturbance that makes changes over time and the controller designed does not takes action to suppress sensitive input noise, which causes chattering at short intervals of few seconds only. However, the controller designed can filter noises if it uses an estimator that operates simultaneously causes with negative values. In addition, hybrid systems can be easily coupled with model-based modern control techniques to eliminate any change that disturbs a system (see e.g. Sazonov, 2003).
Another point to highlight in comparing experiments and simulation results is that the responded signals (or controlled variables) in both figures (13 and 14) are precisely not very close to each other. This is due to ESP-r, which considers the climate on an hourly basis and due probably to theoretical approximations used sometimes to represent closely practical issues. Nevertheless, the controller designed, in this work maintains the measured indoor air temperature with few small and brief variations around the set-point.

8. CONCLUSION

A new approach to building automation systems that utilizes hybrids systems is presented in order to design real-time controllers that can automatically regulate building equipments and components. This type of approach is implemented and experimented within a building model, and tested in simulation within the use of run-time coupling approach developed by means of distributed control modeling and building performance simulation environments. Although a methodology developed in this paper is based in the encapsulation of data exchanged between ESP-r and Matlab with Web services based SOAP/XML, both control and building performance applications can run on different operating systems such as Windows, Unix and Linux.
The use of multi-agent systems for intelligent control in buildings can have a great potential. This potential may well rely on a dynamic autonomy control to be introduced in buildings where it can have a large impact on saving costs. It can also reduce the complexity of distributed control systems used in BAS. For such reasons, it is considered that multi-agents based on hybrid systems can be suitable to design intelligent control models for complex building systems.
Concerning the perspectives of this research, there is a necessity to formalize the run-time coupling mechanism through socket Internets. An approach is envisaged to concurrent design that provides a practical solution for developing knowledge with distributed control and building systems through the use of intelligent agents that can interact with each other, to autonomously regulate HVAC (Heating, Ventilation and Air-Conditioning) systems and building environment. This approach will be based on a framework that utilizes hybrid solutions to learn and to perform aspects related to the comfort of occupants, further saving in energy use, monitoring and supervising functions and decision making.
Future research will involve the development and implementation of hybrid intelligent systems based on the use of fuzzy logic in order to manage reasoning of all interactions among components and to classify suitable gradient-based optimization features under constraints and uncertainties in multi-agent systems.

REFERENCES

Alur, R., Henzinger, T.A. and Sontag, E.D., 1996. Hybrid Systems III: Verification and Control, LNCS 1066, Springer-Verlag.

AutomatedBuildings, 2002. The Automated Buildings website, Inc. < http://www.automatedbuildings.com/news/jan02/art/hk/hk.htm>

ASHREA 135, 2001. *BACnet –A Data Communication Protocol for Building Automation and Control Systems*, ASHREA Inc. Atlanta

DeLoach, S., Wood, M. and Sparkman, C., 2001, Mult-agent Systems Engineering, *Journal of Software Engineering & Knowledge Engineer*, Vol. 11, n. 3, pp. 231-258.

Cheong, C. and Winikoff, M., 2005, Hermes: Designing goal-oriented agent interactions, 6th International Workshop on AOSE, Utrecht, NL.

Engelen, V. R., 2006. gSOAP 2.7.8 User Guide, Florida State University and Genivia Inc. < http://www.cs.fsu.edu/~engelen/soapdoc2.html>

ETB, (2005) The Engineering ToolBox's website, Inc. <http://www.engineeringtoolbox.com>

Harel, D., 1987, Statecharts: A Visual Formalism for Complex Systems, *Journal of Science of Computer Programming*, Vol. 8, pp. 231-274.

Kastner, W., Neugschwandtner, G., Soucek, S. and Newman, H. M., 2005, Communication Systems for Building Automation and Control", *IEEE journal*, Vol. 93, n. 6, pp. 1178-1203

Lygeros, J., Godbole D.N. and Sastry, S., 1996, Multi-agent hybrid system design using game theory and optimal control", 35th IEEE conf. on Decision and Control, pp. 1190-1195, Japan

Lygeros, J., Godbole D.N. and Sastry, S., 1998, Verified hybrid controllers for automated vehicles," *IEEE Transaction Automatic Control Journal*, Vol. 43, pp. 522–539

Lygeros, J., Godbole, D. and Sastry, S., 2000. A game-theoretic approach to controller design for hybrid systems, *IEEE Journal*, Vol. 88, Number 7

Loy, D., Dietrich, D. and Schweinzer, H. J., 2001. Open Control Networks – LonWorks / EIA 709 Technology, *Kluwer Academic Publishers*.

O'Reilly Media, 2006, O'Reilly's site < http://www.xml.com/pub/a/ws/2002/04/16/soap.html>

Sazonov, E. S., Klinkhachorn, P. and Klein, R. L., 2003, Hybrid LQG-Neural Controller for Inverted Pendulum System," 35th SSST Symposium, Morgantown, WV

SmartHome, 2000, SmartHomeFurom's website < http://www.smarthomeforum.com>

Sharples, S. Callaghan, V. and Clarke, G., 1999, A Multi-Agent Architecture for Intelligent Building Sensing and Control, *Inter. Sensor Review Journal l*

Tomlin, C., Pappas, G. J. and Sastry, S., „Conflict resolution for air traffic management: A study in multi-agent hybrid systems," *IEEE Trans. Automat. Cont*., Vol. 43, pp. 509–521, 1998.

Yahiaoui, A., Hensen J.L.M. and Soethout L.L. , 2003, Integration of control and building performance simulation software by run-time coupling, IBPSA Conf. & Exhib., Vol. 3, pp. 1435-1441, Eindhoven, NL

Yahiaoui, A., Hensen, J., and Soethout, L., 2004, Developing CORBA-based distributed control and building performance environments by run-time coupling, 10th ICCCBE, Germany

Yahiaoui, A., Hensen J.L.M., Soethout L.L. and Van Paassen, D. , 2005, Interfacing of control and building performance simulation software with sockets, IBPSA Conference and Exhibition, Canada

Wooldridge, M., Jennings, N. R. and Kinny, D., 1999, A Methodology for Agent-Oriented Analysis and Design, *Journal* of Agents, pp. 69-76.