18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
http://mssanz.org.au/modsim09

Object-Oriented Modelling and Control in Buildings

Zupančič, B. [1], **A. Sodja** [1]

[1] *University of Ljubljana, Faculty of Electrical Engineering,*
Laboratory for Modelling, Simulation and Control
Email: borut.zupancic@fe.uni-lj.si

Abstract: This paper deals with some of the important aspects of the harmonisation of thermal and radiation flows in buildings using an object-oriented multi-domain modelling approach. The traditional modelling approach is based on block-oriented schemes in which causal relations play an important role. However, this causality is artificially generated in order to fulfil the appropriate conditions for a simulation on conventional sequential computers. Fortunately, new concepts based on object-oriented approaches, use physically oriented connections and algebraic manipulations to enable the so-called acausal modelling. Here, all these basic concepts are described and the advantages of the new concepts in comparison with traditional approaches are briefly discussed.

We have developed a mathematical model that gives the dependence of the external temperature, the solar radiation, the artificial heating and the roller-blind positioning on the internal room temperature. The approach was a combination of theoretical and experimental modelling. The heat dynamics is caused by the heat conduction, the convection and the solar radiation. In the latter case the transfer of the sun's rays through a glass window was studied, and in many cases simplifications were introduced. Some of the model's constants were determined experimentally, comparing the experimental and simulation results. However, only some basic equations are presented (one can find them in the appropriate references) as the emphasis here is on the object-oriented simulator paradigm.

The simulator was originally developed in the Matlab-Simulink environment. However, the main drawback of such a conventional modelling approach soon became apparent – the simulator could only be used for the developed configuration. Almost every structural change demanded a new development from scratch. We were not able to establish a library of reusable sub-models for the walls, windows, etc. For this reason we decided to implement the model in Modelica. Modelica is a multi-domain object-oriented modelling language that supports both high-level modelling using pre-prepared complex model components and detailed modelling with equations. Modelica is currently the most promising result in the sense of the standardization efforts for the modelling of continuous and hybrid dynamical systems. The basic idea of the implementation in Modelica is to decompose the described system into components that are as simple as possible and then to start from the bottom up, connecting the basic components (classes) into more complicated classes, until the top-level model is achieved. The components provided by the *Modelica Standard Library* were sufficient to start with the Modelica implementation of the described mathematical model in a graphical way – by connecting the appropriate components (icons) from the *Thermal* library (the components for 1-dimensional heat-transfer, e.g., *ThermalConductor*, *BodyRadiation, Convection*, etc). Some classes were modified and some new classes were designed. The basic classes, such as walls, windows, furniture, floor, ceiling and some others, were developed and then connected into a complex top-level building model, which was validated with the aid of real experiments in a specially developed experimental room.

The final part of the paper shows how the simulation model can be used for the control system design, which results in the creation of pleasant living conditions. The multi-domain model is very efficient, as the control, thermal and illumination components can be efficiently combined. Several control algorithms were designed, compared and validated with regard to the power consumption and the integral square error criteria. Besides taking into account the temperature conditions we also measured the illumination, which gave us the possibility to study how to effectively harmonize the living conditions in buildings, i.e., the temperature and illumination are always treated separately, in spite of the fact that they are closely dependent.

Keywords: *Object-oriented modelling, Simulation, Acausal modelling, Thermal flows, Intelligent building*

1. INTRODUCTION

Traditional, general-purpose simulation tools have a lack of object-oriented properties, which makes it impossible to develop truly reusable components. For this reason some special-purpose tools were developed (for mechanical, electrical, chemical systems, etc.). However, such systems do not support multi-domain modelling (systems from different areas) – which is frequently needed, particularly in the case of automotive, aerospace and robotics applications. For modelling such systems, new concepts are needed.

When analysing the advantages and disadvantages of traditional and more advanced modelling and simulation tools the basic distinction appears from the term “causality”. This term can explain the evolution, which in the past was declared as the evolution from block-oriented tools into object-oriented tools.

2. OBJECT- AND MULTI-DOMAIN-ORIENTED APPROACHES

When using general-purpose simulation software for predominantly continuous systems (e.g., Simulink) we assume that a system can be decomposed into block-diagram structures with causal interactions. Often, a significant effort in terms of analysis and analytical transformations is needed to obtain the problem in this form. This procedure requires a lot of engineering skills and manpower and, in addition, it is error-prone. However, in order to allow the reuse of component models, the equations should be stated in a neutral form, without any consideration of the computational order. This is the so-called acausal modelling approach. In nature, for example, real systems are acausal. We never know whether a flow causes a pressure difference in a tube, or vice versa. Causality is artificially made because the physical laws have to be transformed into a convenient computational description. It is much easier, more convenient and more natural then to use acausal modelling tools, such as Dymola with Modelica (Cellier, 1991, Fritzson, 2004, Dymola, 2008, Modelica, 2005). First, we write balance and other equations in their natural form as a system of differential-algebraic equations. Then, computer algebra is utilized to achieve an efficient simulation code.

There are some important processing tasks and features in acausal-oriented modelling tools that will be briefly described in the following.

Object orientation In each object-oriented programming the principles *encapsulation*, *information hiding* and *inheritance* are very important. Inheritance is a way of forming new classes (instances of which are called objects) using classes that have already been defined. These new classes take over (or inherit) the attributes and behaviour of the pre-existing classes, which are referred to as the base (ancestor) classes. So an existing code can be used with little or no modification.

However, we do not intend to discuss the well-known general concepts in OO programming. From a modeller point of view, object orientation means that we build a model similar to a real system, using a pump, a pipe, a valve, etc., and then put them together.

Component coupling The connections between sub-models are based on variables, which define the proper relations and influences between movements, flows, temperatures, etc. There are two types of variables in the connectors of subsystems: *across (potential) variables,* which become equal at the connection points (potentials, temperatures, pressures) and *through (flow) variables,* the sum of which equals zero (currents, momentum, forces). A connector is a structure in which all the mentioned variables are collected, and by joining the connectors the sub-models are connected. During the processing the tool generates the appropriate equations from the *connect* statements and the *connector* definitions, and this is a fundamental concept in making acausal modelling work.

Hybrid features This means that continuous-time, discrete-time and discontinuous modelling is supported. Such an approach demands many original solutions in symbolic processing and in the run time during the simulation. Events can be periodically or relation (state) triggered. The variable structure models that can be studied in such environments are very useful, especially in control-systems studies.

Symbolic processing Multi-domain OO modelling tools have a modelling pre-processor with some basic functions. Each variable (unknown) must be calculated from a single equation. The equations should be sorted in such a way that all the variables on the right-hand side are calculated before the equation is executed (in each step size). The size and complexity are also reduced at this stage (elimination of trivial equations v1=v2, symbolic elimination of algebraic loops, reduction of the problem size (number of unknowns) with tearing, index reduction, etc.).

Modelica – OO modelling standard The ideas about OO multi-domain modelling environments originate in the 1990s, and even earlier, in particular with many activities at Lund University within the group of Prof. Åström. Later, a member from this group – Hilding Elamqvist – founded a company named Dynasim, and another important milestone was the foundation of the Modelica association in 2000, which took care of the Modelica language standardization. Modelica (Modelica, 2005, Fritzson, 2004)) is a multi-domain object-oriented modelling language that supports both high-level modelling using pre-prepared complex model components and detailed modelling by equations. The graphical diagram layer and the textual layer can be efficiently combined. Modelica is currently the most promising result in terms of the standardization efforts for the modelling of hybrid dynamical systems.

The described concepts will be demonstrated with the modelling and control of the thermal and radiation flows in a building. This is because bioclimatic conditions are extremely important for pleasant and healthy living conditions. As such, they represent a process with inexhaustible possibilities for studying the modelling, simulation and control concepts (Sodja and Zupančič, 2008, Sodja and Zupančič, 2009, Škrjanc et al., 2001, Lee and Brown, 2008, Zupančič and Sodja, 2008).

3. MODELLING AND CONTROL OF THE THERMAL AND RADIATION FLOWS

There are very many aspects when talking about building automation. Our activities were focused on one important aspect: how to ensure good living conditions, but also energy savings, with the appropriate harmonization of thermal and daylight flows. Our approach was based not only on real experiments, but also on mathematical modelling. Our modelling experiences will be used to evaluate the described traditional and more advanced object-oriented multi-domain concepts.

3.1. Mathematical modelling

The main modelling goal was to efficiently use the model for control systems design. The theoretical modelling of the heat dynamics of a room was based on energy-balance equations that take into account thermal conduction, thermal convection and solar radiation (Škrjanc et al. 2001, Lee and Brown, 2008). The model was basically developed for a cube-shaped room with several layer walls. Each wall can contain a double-glazed window with a gas between the two panes. The model for thermal conduction and solar radiation was derived with the aid of Figure 1.

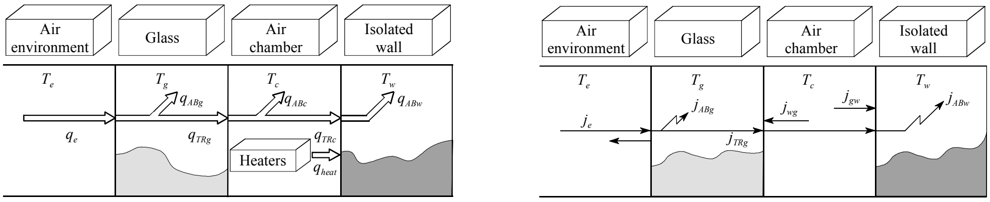

Figure 1. Thermal and radiant flows

The incoming flow q_e is partly transmitted (q_{TRg}) and partly absorbed by the glass (q_{ABg}) and by the air inside the room (q_{ABc}). In the isolated wall only absorption takes place (q_{ABw}). q_{heat} represents the artificial heating (or cooling). The following equations were used in the mathematical modelling:

$$q_e = \alpha (T_e - T_g) \qquad q_{ABg} = q_e - q_{TRg} \qquad q_{TRg} = \alpha (T_g - T_c) \qquad q_{ABc} = q_{TRg} + q_{heat} - q_{TRc}$$

$$q_{TRc} = \alpha (T_c - T_w) \qquad q_{heat} = n_{heat} P_{heat} / S_{heat} \qquad q_{ABw} = q_{TRc} \tag{1}$$

A part of the global solar radiation (j_e) is reflected from the surface of the glass and a part is absorbed by the glass (j_{ABg}). The part that comes into the room (j_{TRg}) is absorbed in the isolated wall (j_{ABw}) because of the assumption that the wall is black in the sense of infrared radiation. The effect of the infrared radiation is modelled with flows between the wall and the glass (j_{wg}) and between the glass and the wall (j_{gw}). The following equations were used:

$$j_{TRg} = TR\, j_e \qquad j_{ABg} = AB\ TR\, j_e \qquad TR = \frac{1-R}{1+R} TR_0 \frac{n^2+1}{2n} \qquad AB = 1 - TR_0 \frac{n^2+1}{2n}$$

$$j_{ABw} = j_{TRg} \qquad j_{wg} = \sigma (T_w^4 - T_g) \qquad j_{gw} = \sigma (T_g^4 - T_w^4) \tag{2}$$

Our final goal was to obtain an appropriate mathematical model that describes the temperature changes in the different media. Using basic thermodynamic laws the final equations were derived using Eqs. 1 and 2

$$\text{Glass} \qquad \frac{dT_g}{dt} = \frac{S}{m_g c_g} \left(q_e - q_{TRg} + j_{ABg} + j_{wg} \right) \qquad \text{Chamber} \qquad \frac{dT_c}{dt} = \frac{S}{m_c c_c} \left(q_{TRg} - q_{TRc} + q_{heat} \right)$$

$$\text{Isolated wall} \quad \frac{dT_w}{dt} = \frac{S}{m_w c_w}\left(q_{TRc} - j_{TRg} + j_{gw}\right) \qquad (3)$$

where m is the mass of a medium, c is the specific heat and S is the appropriate area. These final equations represent only the influence of the basic physical laws and some initial modelling considerations. So this part represents only the main modelling idea. Many other phenomena were included, many suppositions were used and many simplifications were considered (see details in Škrjanc et al. 2001, Sodja and Zupančič 2009). However, the details are beyond the scope of this paper, as the modelling itself was not the main aim here.

The basic inputs (influence variables) of the simulation model are the outside conditions as well as the changeable properties of the envelope: the outdoor air temperature, the temperature of the terrain, the global solar radiation, the properties of the opaque elements (thermal capacity and resistance), the properties of the transparent elements (the geometry of openings, the optical characteristics of the glass and the resistance of the fill between the panes are variable), interior properties (the absorption, the emission coefficients of the walls and the thermal capacity of furnishings are changeable), some other characteristics (the changeable orientation, the additional heating and cooling – the power of the heater, cooler and ventilator). The outputs of the simulation model are the different thermal and radiation flows, the indoor temperature, the walls, the windows and the surface temperatures.

3.2. Implementation of the simulator in Dymola-Modelica

The simulator was originally developed in the Matlab-Simulink environment (Škrjanc et al. 2001). However, the main drawback of such a conventional approach became very evident – the simulator components could only be used for the developed configuration. Almost every structural change demanded a new development from scratch. We were not able to establish a library of reusable components for the walls, windows etc.

So we decided to implement the model in Modelica (Fritzson, 2004, Modelica, 2005) within the Dymola environment (Dymola, 2008). The basic idea of the implementation in Modelica is to decompose the described system into components that are as simple as possible and then to start from the bottom up, connecting basic components (classes) into more complicated classes, until the top-level model is achieved.

The components provided by the *Modelica Standard Library* were sufficient to start with the implementation of the Modelica model in a graphical way – by connecting the appropriate components (icons). The *HeatTransfer* library contains the components for 1-dimensional heat-transfer modelling with lumped parameters, e.g., *ThermalConductor*, *BodyRadiation, Convection*, etc. All these components include a single connector (port) that contains two variable declarations: the temperature *T* and the heat-flow rate *Q_flow*.

Implementation of the wall

A wall normally consists of several layers. The resulting scheme for a one-layer implementation, as seen in *Dymola*'s GUI, is depicted in Figure 2. The block called the *LayerCapacity* is a model of a heat capacitor, while the blocks *InnerSide* and *OuterSide* are models of the thermal conduction through the layer, and are connected on one side with the *LayerCapacity* and on the other side with the stand-alone connectors *inside* and *outside*. The described structure is defined as a *Layer* model class. We observed three connecting points, with three different temperatures: in the middle the average temperature of the layer, and two boundary-layer temperatures on either side. The model of the wall is obtained by simply connecting several layer sub-models in series. The structure of the wall is further connected to the other connectors according to the wall's boundary conditions.

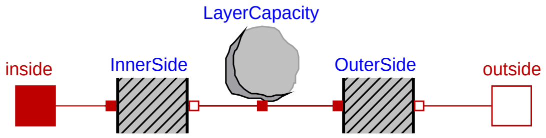

Figure 2. Scheme of a wall layer in Modelica

Implementation of the window

A similar procedure was used to implement the model class of a window. The scheme obtained in *Dymola*'s GUI is shown in Figure 3. The heat capacities of the outer and inner panes are modelled with two *HeatCapacitor* model classes, *OuterPane* and *InnerPane*, the connectors of which also contain the panes' average temperatures. Both panes interact with each other via thermal radiation and thermal conduction through the air in the gap

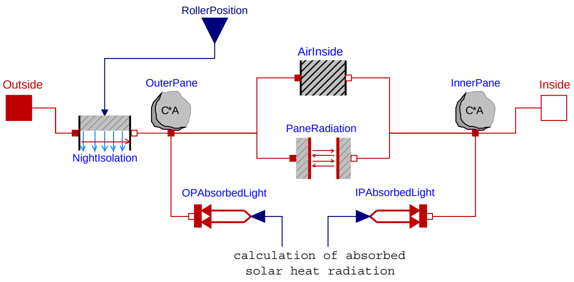

Figure 3. Scheme of the model describing the thermal processes in a window in Modelica

between the panes. Therefore, *OuterPane* and *InnerPane* are connected with the model classes *AirInside* and *PaneRadiation*. There are also model classes named *OPAbsorbedLight* and *IPAbsorbedLight* in Figure 3. These are conversion blocks and transform the absorbed solar-radiation flows into connections of the panes' heat-capacity blocks. They are needed to convert the absorbed radiation flows, calculated as a real variable, into the *HeatPort* connector type. A more detailed description of the solar radiation flows and the appropriate *Modelica* implementation can be found in (Sodja and Zupančič, 2008, Sodja and Zupančič, 2009). All the other blocks, which model other thermal flows coming from the window's surroundings, are connected to the stand-alone connectors *Outside* and *Inside*, respectively. It is clear that the connector *Outside* is not connected directly to the heat-capacity model class *OuterPane* of the pane in Figure 3, but through the *NightIsolation* model class that models the influence of a (partially) shaded window, which influences the thermal conductance to the air outside (prepared for control).

Implementation of the room

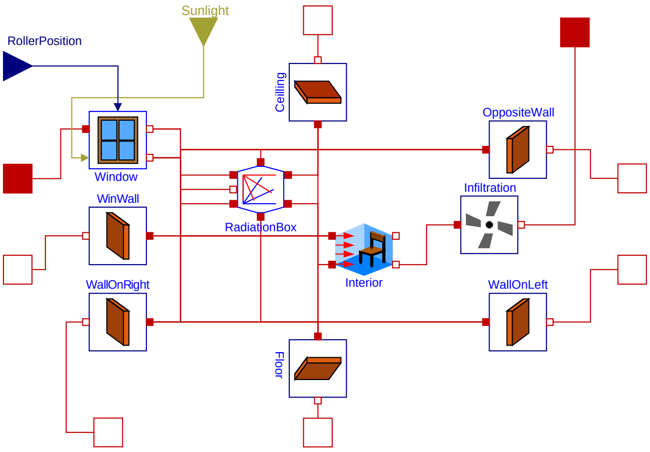

Figure 4. Modelica model of the room

Finally, a model of a room can be built from the prepared model classes. The overall scheme consists of classes that model the room's envelope and those from the interior model class. The appropriate model scheme is shown in Figure 4. The class *Interior* in the middle is surrounded with the classes of the room envelope. The inner surfaces of the envelope (represented by the connector facing towards *Interior)* are connected to the *RadiationBox* class, which models the thermal radiation exchange between the surfaces (and is beyond the scope of this paper – see Sodja and Zupančič, 2009), *Interior* class (model of the air mass and the furniture inside) and to the lower-right connector of the *Window* class, which is an array of solar-radiation heat flows received by each surface. The external surfaces of the envelope are connected to connectors that are visible from the outside of the model of the room, and were used in the top-level model. The blocks that model the convection between the outdoor air and the walls of the building (ceiling, north, south, east and west walls) are therefore connected to those connectors. The *Floor* connector is connected to a constant ground temperature. The intensity of the solar radiation is routed to a class named *Sun* in the top-level model, where the direction vector of the solar rays is also calculated from a specified start date and simulation time and packed together with the solar-radiation component intensity into one connector. *(Sunlight* in Figure 4*).*

3.3. Model validation

A small cube-shaped ‘test chamber’ with the appropriate sensors for temperature, solar radiation and illumination measurements was built on the roof platform of the Faculty of Civil and Geodetic Engineering, University of Ljubljana, with the main goal to validate the developed models and control systems. One set of measurements was used for the appropriate final-parameters tuning of the mathematical model of the test chamber. Another set of measurements was used for the simulator validation. Simulations were performed with the measured outdoor temperature and the global solar radiation as the input variables, taken from experiments, as well as with the variable signal for the roller-blind moving regime.

3.4. Control system

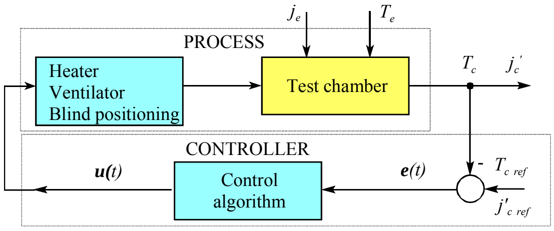

Figure 5. The skeleton of the chamber control system

The simulator was used for different control systems’ implementations. The solutions designed with the help of the simulator were then tested on the test chamber. Besides the conventional approaches with control actions on the heater/cooler and the ventilator, the emphasis was put on the envelope’s dynamical adaptations. Therefore, the position of the roller blind was controlled in order to achieve the appropriate harmonisation of the thermal and daylight flows that influence the indoor temperature T_c and the illumination j'_c. The external temperature T_e and the global solar radiation j_e were treated as external disturbances. The skeleton of the control strategy is depicted in Figure 5.

Temperature control with conventional control algorithms

The design of the temperature control using conventional approaches was the first activity. Different control strategies with different aims were studied with a simulation and with real experiments on the test chamber. A special expert control system, an on-off controller, a PID controller, a fuzzy controller and a deadbeat controller were designed and tested. They were compared with the energy consumption $W = \int_0^{t_{\max}} P_{heat}(t)dt$ and the integral square error $ISE = \int_0^{t_{\max}} e^2(t)dt$. The conditions for the experiments were the following: a step change in the reference signal T_{ref} from 20°C to 22°C , a step change in the solar radiation flow j_e of 40W/m^2, a step change in the external temperature T_e of 2°C with a simultaneous change in the solar radiation flow j_e of 10W/m^2.

The criteria values for all three step changes for the five designed algorithms are shown in Figure 6.

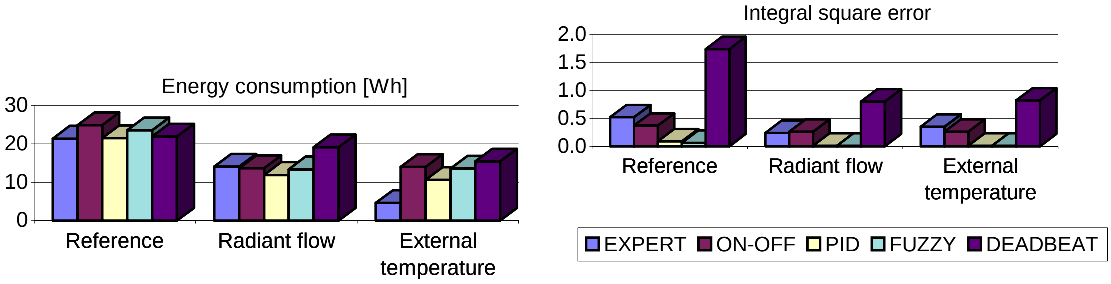

Figure 6. The comparison of the defined criteria

Energy consumption The algorithms are more or less equivalent. The expert algorithm seems to be superior in the case of the external temperature change.

Integral square error The FUZZY and PID are superior in this case. The deadbeat algorithm is perfect when using a linear mathematical model with the continuous control signal. However, when using a non-linear process model and a discrete controller the control system's performance was very bad, also with regard to some other criteria.

Harmonisation of temperature and illumination

Up to now we have developed only the thermal part of the model, i.e., the influence of the external temperature and solar radiation on the temperatures and the different thermal and radiant flows. However, the daylight in the buildings is also very important for pleasant living and working conditions. (Kristl et al., 2008, Lah Trobec et al., 2006). Daylight can be used to reduce the consumption of lighting and heating energy.

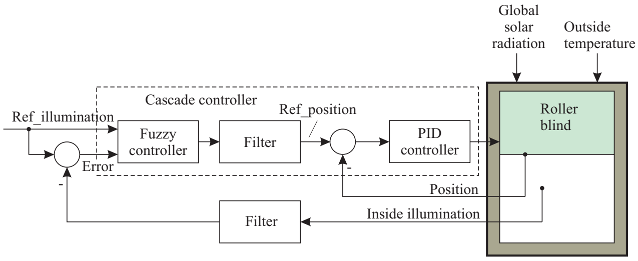

Figure 7. Illumination control system

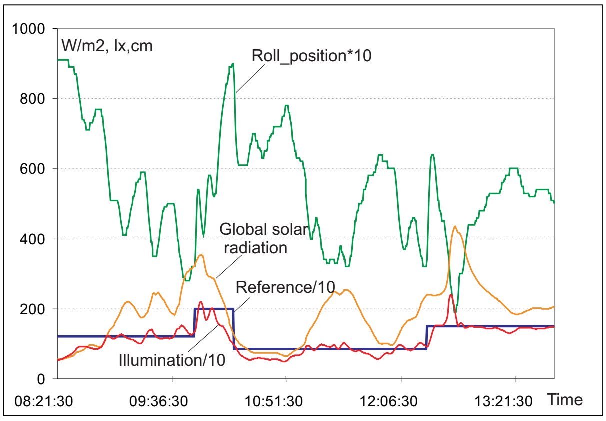

Figure 8. Cascade control of inside illumination

So the harmonisation of the temperature and the illumination was only tested with real experiments on the test chamber. An example of one solution is depicted in Figure 7. The controlled external roller blind on the window (Figure 8) is supposed to harmonize the illumination response. Two signals are measured: the indoor illumination, which is the main controlled variable, and the position of the roller blind, which is the auxiliary variable of the cascade control system. The main controller is a fuzzy controller and the auxiliary one is a conventional PID controller. The main controller gives the reference value for the roller blind's position and the auxiliary controller feeds the motor of the roller blind.

The results of such an experiment are shown in Figure 8. The system was influenced by the set-point illumination step changes and by the global solar radiation changes as a disturbance. The difference between the controlled inside illumination and the desired profile is small, in the range ±100 [lx]. The influence of the roller blind's position on the inside temperature was also studied. We also succeeded in harmonizing the inside temperature and the illumination by simultaneously influencing two control loops – the thermal and the illumination to the roller blind. Of course, passive energy resources can only be used efficiently in some narrow bands with regard to the appropriate reference values. Usually, active resources – heating, cooling, electrical lightening – must be used to drive the system close to the reference signals.

4. DISCUSSION AND CONCLUSIONS

In this paper the simulator, which was primarily built for the design of the control of the heat and radiation flows in buildings, was used primarily for a comparison of traditional block-oriented causal approaches and more advanced object-oriented multi-domain approaches. The structure of the block-diagram model in Simulink does not reflect the topology of the physical system. The effort required to produce the Simulink simulation model cannot be compared with that of the Modelica model. One needs much more time and much more modelling knowledge for the Simulink-based approach. In the Modelica model (Figures 2,3,4), the connections between the physical system and the mathematical model are very transparent. All the components are fully reusable in other configurations. The combination of textual and diagram programming is efficient. This means that Modelica can be used for very complex models and is also superior for model documentation. We can conclude that Matlab-Simulink is more appropriate for the design and implementation of control schemes as it has more facilities, especially in conjunction with some toolboxes – e.g., Control System Toolbox, Optimization Toolbox. Modelica is superior when modelling physical systems, when the concept of algebraic manipulation and the specially defined connectors bring many advantages. This approach is also very useful in education. We propose to begin modelling courses with the OO acausal approach, especially when one can deal with implemented libraries that do not demand a deep theoretical modelling background. This can motivate students much more than the low-level Simulink approach.

The illumination model is currently under development. When it is integrated with the thermal model, the combined model will be suitable for the development and testing of multivariable control systems.

REFERENCES

Cellier F.E. (1991), *Continuous system modeling,* Springer Verlag.

Dymola, Multi-engineering modelling and simulation (2008), Users manual, Ver. 7.0, Dessault System, Dynasim AB, Sweden, Lund.

Fritzson, P. (2004), *Principles of object oriented modelling and simulation with Modelica 2.1*, IEEE Press, John Wiley&Sons Inc., Publication, USA.

Kristl, Ž., Košir, M., Lah Trobec, M., Krainer, A. (2008), Fuzzy control system for thermal and visual comfort in building, *Renew. energy*. [Print ed.], Vol.4, No. 33, pp. 694-702.

Lah Trobec, M., Zupančič, B., Peternelj, J., Krainer, A. (2006), Daylight illuminance control with fuzzy logic, *Solar Energy*, Vol. 80, pp. 307-321.

Lee, K., Braun, J.E. (2008), Model-based demand-limiting control of building thermal mass, *Building and Environment*, Vol. 43, pp. 1633-1646.

Modelica - A unified object-oriented language for physical system modelling (2005), Language specification, Ver. 2.2, Modelica Association, http://www.modelica.org/documents /ModelicaSpec22.pdf

Sodja, A., Zupančič, B. (2008), Some aspects of thermal and radiation flows modelling in buildings using Modelica, *Proceedings of 10th International Conference on Computer Modelling and Simulation UKSIM/EUROSIM 2008*, Cambridge, UK, pp. 637-642.

Sodja, A, Zupančič, B. (2009), Modelling thermal processes in buildings using an object-oriented approach and Modelica. Simulatation Modelling Practice and Theory, doi:10.1016/j.simpat.2009.04.003 (article in press)

Škrjanc, I., Zupančič, B., Furlan, B., Krainer, A. (2001), Theoretical and experimental fuzzy modelling of building thermal dynamic response, *Building and Environment*, Vol. 36, No. 9, pp. 1023-1038.

Zupančič, B., Sodja, A. (2008), Object oriented modelling of variable envelope properties in buildings. WSEAS transactions on systems and control, vol. 3, no. 12, pp. 1046-1056.