Simulation Modelling Practice and Theory 17 (2009) 1143–1159

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

Modelling thermal processes in buildings using an object-oriented approach and Modelica

Anton Sodja, Borut Zupančič *

University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000 Ljubljana, Slovenia

A R T I C L E   I N F O

Article history:
Received 1 August 2008
Received in revised form 26 March 2009
Accepted 2 April 2009
Available online 21 April 2009

Keywords:
Object-oriented modelling
Multi-domain modelling
Modelica
Intelligent building model
Thermal and radiation flows

A B S T R A C T

Most of today's modelling and simulation concepts originate from the times and methods of analog computers. Usually, it is assumed that the model must be expressed in an explicit state-space form. Consequently, the topology of the system gets lost and any future extension and reuse of the model is tedious and error-prone. In other words, it is the modeller's task to consider the computational order of the operations during a simulation.

In this paper we discuss the re-implementation of a passive-solar- building simulator in an object-oriented environment; it was originally built in the non-object-oriented simulation environment of *Matlab–Simulink*. The former simulator was designed to resemble a real physical test chamber with regard to the thermal and solar radiation flows. However, due to the lack of object orientation in *Matlab–Simulink* it was very difficult to apply any configuration modifications and extensions.

We start with a brief description of the mathematical modelling which includes thermal dynamics and solar radiation. Then the implementation in *Modelica* is presented. So, a much superior environment in comparison with *Matlab-Simulink* was obtained, giving us the possibility of high-level modular and object-oriented modelling. The model is also extremely efficient in multidisciplinary projects in which control-engineering specialists (our group) cooperate with specialists from civil engineering, because civil engineers can more easily understand graphical and textual models in *Modelica* than schemes in *Simulink*.

We expect that such a model will fulfil and significantly improve several model properties in comparison to the *Matlab–Simulink* implementation, i.e., a better understanding of the influences of thermal and radiation flows on comfortable living conditions, a model-based control-system design, which will enable the harmonization of active and passive energy resources, important energy savings, and a very suitable environment for education in modelling, simulation and control.

1. Introduction

A model that reflects, as much as possible, the behavior of a real system is indispensable if we want to obtain deeper knowledge about the system. In addition, a good model is also very important for successful control design. Usually, complex systems result in complex models; however, it is extremely important to clearly define the modelling aims and to balance all the procedures with regard to a reasonable complexity and accuracy.

The literature from this area is very comprehensive. In particular, there are many studies dealing with traditional modelling. In addition to some papers from our group, which will also be referred to later [14,17], there are some interesting

* Corresponding author. Tel.: +386 14768306.
E-mail addresses: anton.sodja@fe.uni-lj.si (A. Sodja), borut.zupancic@fe.uni-lj.si (B. Zupančič).

1569-190X/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.simpat.2009.04.003

recent publications that also deal with bond-graph approaches [16] and object-oriented approaches, mainly with the implementation of particular *Modelica* libraries. In [4] the thermal model of a building is described and implemented in *Modelica* and the so-called AT plus library was developed. The convection modelling is rather simplified. As in many other papers the final diagrams are rather ambiguous, especially the parts with solar radiation, which are implemented with many icons. There is also no discussion in this paper about the transfer of direct and diffuse solar radiation through the window and the distribution of direct radiation inside the room. In [12] the authors introduce a commercial simulation environment 'IDA ICE' as one of the first attempts to introduce new approaches with symbolic equations modelling. In addition, there is a comprehensive overview of simulators for buildings. In [6] a group from the Herman–Rietschel institute deals with an interesting ceiling solution, which includes the capillary pipes. In this case the process is modelled with finite elements. However, some parts are perhaps too simplified, with the emphasis being more on the implementation of the model rather than its development. Using a simulation it is shown that with new techniques, based on storing heat energy from rooms during the daytime and releasing it during the night-time, a low-energy-demand chilling can be realised.

Although the majority of the presented papers are focused on the thermal effects in buildings, which are modelled with a theoretical approach, there are also approaches with experimental or combined modelling. In [15,13] genetic algorithms for optimising the model parameters are used. There are also many interesting papers dealing with control [10]. They are particularly interesting for us as our modelling is also mostly intended for control purposes. In [7] the Dymola–Simulink interface, which gives the efficient possibility to use the Matlab environment (and Toolboxes) for control-system optimization, is presented. Such an approach is relatively universal as the Simulink model is used only for the evaluation of the criterion function. However, the Modelica concept enables more efficient optimization solutions, which are, unfortunately, still far from maturity usage (see [1] with a Modelica extension Optimica and appropriate extensions in Modelica compiler).

We also noticed a lack of papers dealing with the harmonization of thermal and illuminance effects [9].

In this paper we are dealing with the modelling of thermal and radiation processes in buildings. The basic aim of the model is to obtain a better understanding of the influences of thermal and radiation flows on comfortable living conditions and to use it for control-system design. This control system should act to harmonize the active and passive energy resources and so important energy savings can also be expected. Furthermore, a very suitable environment for education in modelling, simulation and control was expected.

Our previous activities were based on theoretical modelling of the thermal processes and illumination inside a "real test chamber", which was represented by a small house. An appropriate and well-validated simulator in the *Matlab–Simulink* environment [14] was developed. With the aid of the simulator the efficiency of the energy consumption was improved and comfortable living conditions with regard to the temperature and the illumination were ensured [8,9].

However, *Matlab–Simulink* is a rather conventional modelling tool; a lack of object orientation is its most significant drawback. As a result, the model has to be implemented, more or less, in a state-space form. This is a mathematical description that is very far from basic mass- and energy-balance equations, which are normally understandable by engineers. With such transformations the topology of the basic system is lost, which results in a less-intuitive and transparent model.

Furthermore, traditional modelling tools like *Matlab–Simulink* require the models to be causal, so we have to determine which variables are the inputs (causes) and which are the outputs (consequences) of the model components. This is quite a harsh restriction, often meaning that slight changes to the modelled system or an adaptation of an existing model to a similar system is a very time-consuming and error-prone task, unless the original model was designed with these specific extensions in mind. Our model, which was developed for a test chamber, was so restricted to a rectangular-shaped object, with one room and at most one window in a single wall. Merging two room submodels into a double-room building was impossible without a complete rearrangement of the equations. An additional drawback of *Matlab–Simulink* is its inability to easily incorporate the documentation of a model directly into that model. So the documentation is separated from the model implementation and, usually, the consequence is that the documentation is not up to date with respect to all the recent changes.

In order to obtain the significant support of a tool, also in the modelling phase, so as to achieve better flexibility of the simulator and to make it more generally applicable we decided to re-implement the former simulator in *Matlab–Simulink* in a very sophisticated object-oriented environment, *Dymola* [3], which uses the *Modelica* modelling language standard [11,5]. *Dymola* with *Modelica* is a high-level but general-purpose OO modelling tool, well proven in many complex applications. It is also extremely efficient and helpful in the modelling phase. However, such developed models can easily be used in the *Matlab–Simulink* environment, so one can benefit from all the experimental possibilities and toolboxes, which finally results in a really efficient modelling, simulation and experimentation environment.

Using the object-oriented modelling approach one can build complex model libraries on basic principles, i.e., physical laws and energy- and mass-balance equations, and such model classes can easily be reused in different configurations. So our design was focused on the development of a new library, rather than focusing on a single object (i.e., the test chamber). This approach provides easier future adaptations and extensions in complex intelligent-building modelling.

A very important advantage in the *Dymola–Modelica* environment, is the documentation feature, allowing annotations of formatted descriptions to be added to the model. So, the documentation and the model itself are just two different layers of the same model class, which always gives us the possibility to update both layers simultaneously.

As the variables are also objects, they become real physical quantities, with many other attributes: quantity name, unit, display unit, minimal and maximal values, etc. The environment also uses descriptions given in code at variable and parameter declarations. These descriptions are very helpful in the graphical-editing mode.