

MODEL AND TOOL REQUIREMENTS FOR CO-SIMULATION OF BUILDING PERFORMANCE

Marija Trcka, Jan Hensen
Technische Universiteit Eindhoven
Den Dolech 2, 5600MB Eindhoven
The Netherlands
e-mail: m.trcka@bwk.tue.nl

ABSTRACT

The use of building performance simulation (BPS) can substantially help in improving building design towards higher occupant comfort and lower fuel consumption, while reducing emission of greenhouse gasses. Unfortunately, current BPS tools do not allow inter-tool communication and thus limit a modeler to the component models available in the simulation software which happens to be used.

A pragmatic way forward would be to enable co-simulation by externally coupled (legacy) tools. This means that each coupled software would represent only that part of the overall building and system configuration that it is able to model. The overall system is represented by the coupled models, which exchange simulation data during run-time. In this way, shortcomings of each tool can be overcome, and advantages of individual tools can be exploited.

The work underlying this paper addresses co-simulation of building energy and heating, ventilation and air-conditioning (HVAC) models. So far, the research focus has been on thermodynamic issues such as which variables should be exchanged and at what frequency, rather than on mathematic or computer science aspects. This paper specifies and discusses the requirements for BPS software in order to enable co-simulation of building and HVAC system configurations.

KEY WORDS

Building performance simulation, co-simulation, distributed modeling and simulation, run-time coupling, reuse, external coupling

1. Introduction

The traditional (manual) methods for designing heating, ventilation and air-conditioning (HVAC) systems are being surpassed by simulation tools because:

- buildings become more and more complex in terms of shape, lay-out, functionality and services;
- requirements for flexibility and adaptability increase;
- modern building standards and codes are performance based rather than prescriptive; i.e. addressing questions such as “how many hours per year will the temperature rise above a certain limit?” and “what will be the annual energy consumption per square meter floor?”

Advances in hardware and software resulted in a flood of building simulation tools. However, each tool is applicable only to a subset of the overall problem, and is limited both in scope and resolution. The majority of the tools are legacy codes often originating from the seventies. On the whole, they are domain specific, not reusable, large, complex monoliths that are difficult to maintain, but still useful.

Previously [1, 2], it has been argued that in the area of system simulation, there is still enormous amount of work to be done. System modeling and simulation capabilities develop very slowly and take up an enormous amount of resources (time wise and financial). An efficient way forward would be to share developments and to reuse existing component models.

An overview of research concerning the programs interoperability is given in [3]. It could be accomplished on either the product model level [4, 5] or the level of physical process models [6, 7]. Both data and process model reuse follow the traditional approach, where all components models are brought together in a monolithic stand-alone simulation program. The integration takes place before run (or execution) time, as shown in the upper part of Figure 1.

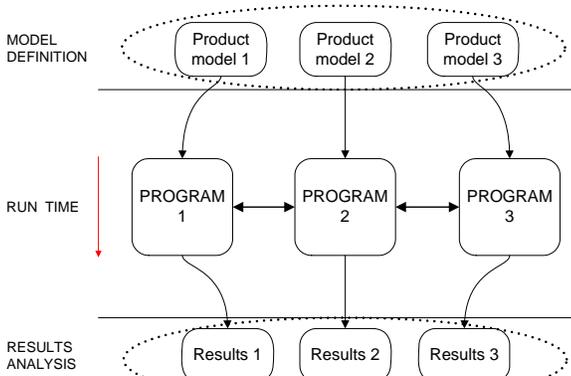


Figure 1 External run time coupling

Enabling the run-time communication between the legacy simulation programs would allow a modeler to model across various environments, while exploiting advances of each. So far, only initial steps towards a general external coupling approach in the domain of building systems were made. This paper refers to these early implementations and discusses requirements and applicability of the domain legacy tools for their distributed utilization.

2. External coupling

In literature, the approach where two simulators are coupled in run-time, has been researched under several terms, such as: external coupling, co-simulation and distributed simulation. *Co-simulation* represents a particular case of simulation scenario where two solvers interact. It concerns both (1) two different solvers running together and (2) two different integration logic within the same program. *Distributed simulation* refers to the technology concerned with integrating various simulators over the network.

Since neither of the terms by their definition identifies exactly the approach undertaken in this paper, we introduce the term external coupling. External run-time coupling addresses the case where at least two programs are executed simultaneously (potentially on separate machines) and where information (i.e. simulation results) is exchanged between them during their execution time.

The external coupling definition is more specific than the definition for co-simulation as it does not concern coupling of two integrators within the same program, and broader than the definition of distributed simulation, as the coupled programs do not necessarily have to be distributed over the network, i.e. coupled programs could resign and be executed on the same machine, since the models are distributed in aspects, and not necessarily by location.

However, the differences are not significant in regards to the general questions (i.e. consistency and stability of the overall simulation, etc), and many of the issues that are

researched and discussed in all three approaches are common. All three terms are used in this paper to address run-time communication between legacy tools.

Compared to the traditional approach, external coupling generates several advantages [8, 9, 10]:

- reusability of already existing (legacy) COST software,
- combination of heterogeneous technologies,
- collaborative model design and development process,
- information hiding,
- scalability and fault tolerance,
- geographically distributed components, and
- potentially reducing model execution time & more available memory.

The external coupling approach breaks boundaries between different simulations and by that introduces the potential to “pool resources”, i.e. to use the best simulation model available without being limited to those available “locally”.

Additionally, building performance simulation can benefit from the external coupling approach as:

- at the moment there is no a single tool that can be used to solve all simulation analysis problems encountered by designers,
- each tool can benefit from future simulation models developments of engaging (renewable) technologies, such as micro heat and power generators, fuel cell etc.,
- different scale of modeling and simulation would easily be applicable. Rather than performing a simulation on a zone/building scale, one can combine different buildings and systems models and simulate various scenarios on the scale of a town, or a region (Figure 2).

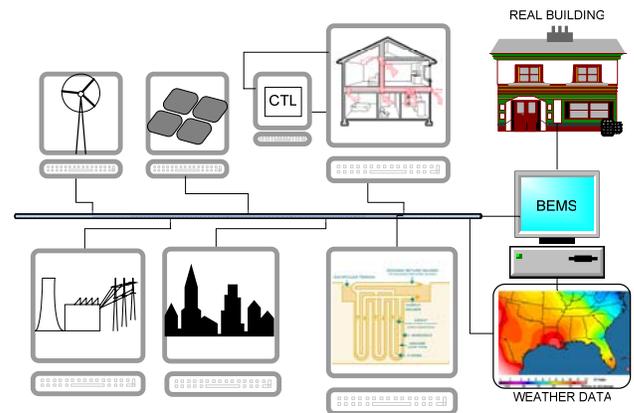


Figure 2 Distributed BPS simulation

Some software-specific work has been done by others regarding the external run-time coupling in the field of building performance simulation in general, i.e. ESP-r and

Radiance [11], ESP-r and Fluent [12] and EnergyPlus and MIT-CFD [13].

In addition to the general run-time coupling developments, there are several developments regarding co-simulation in the domain of HVAC systems and control. Here we address some of them.

TRNSYS developers introduced a new type 155, defined as MATLAB connection. Matlab is launched at every TRNSYS time step as a separate process. The type 155 communicates with the Matlab engine through a Component Object Model (COM) interface. Any Matlab command (including Simulink simulations) can be run within a TRNSYS simulation [14]. The similar approach is implemented in TRNSYS coupling with EES. TRNSYS is able to execute EES at each time step to solve a given set of equations.

A link between EnergyPlus and TRNSYS was used before EnergyPlus has obtained its own photovoltaic component model [15]. EnergyPlus module communicated information found in the EnergyPlus input file concerning photovoltaic arrays to TRNSYS. TRNSYS was then automatically launched during an EnergyPlus simulation to determine the performance of the PV array before returning control back to EnergyPlus. EnergyPlus then waited for TRNSYS to complete, then recuperated the output files that TRNSYS generates during its run and incorporated them into its native output-reporting format. The use of Windows API calls was used. However, the link is not a real application of external coupling as there is no communication between programs on the time step basis.

The assortment of efforts within the BPS domain presented show that there is a need for interoperable simulation environments. However, the research by others has not yet offered a general standardized framework for building interoperable simulation environments. The authors' work preceding this paper has pioneered mechanisms for external coupling in HVAC simulation domain.

2.1 Implementations

The approach undertaken by the early prototypes [16] is to develop components within each BPS environment that will be used to interface other environments. So far, two distinct mechanisms evolved: mechanism with discontinuous and continuous external program run.

In the former, the external program is invoked each specified time step from the base program. While the external program does its calculations, the base program waits. While the base program continuously runs, the external program starts and stops during the overall simulation time. To keep the dynamic evolution of the results, the necessary information, if any, is externalized.

In the latter, the coupled programs are invoked separately and are running simultaneously. Distributed components need to exchange data at the run-time and to synchronize their local (simulation) clocks. Therefore, besides data

transfer management, a time management is required in order to successfully accomplish the simulation tasks. We distinguish internal and external time management approaches. The internal time management indicates that the synchronization checking procedure is coded within the "interface" component itself. On the other side the synchronization can be compassed within the inter process communication (IPC) mechanisms, applying blocking mode, for example.

The minimum set of variables is exchanged in both directions. The exchanged variables should be real physical quantities, because as such they are readily available in any software program and allow run-time model coupling with a real building or components. The distributed models connections are considered analogous to the physical components connections, i.e. via pipes, ducts and potentially wires. The set of data that uniquely determine the information passed through the connections should be exchanged. This comprises state variables of the working fluid, defined by applying Gibbs phase rule, a transport variable (quantification of the flow), and if required a control variable. If the internal time management approach is employed the information about the current simulation time should be exchanged as well to ensure programs synchronous execution.

2.2 Coupling strategy

There are two different coupling strategies [17]:

- ping-pong coupling, and
- onion coupling.

In the former, distributed models run in sequence, where each model uses the known (from the previous time step calculation) output values of the coupled model. The latter coupling strategy requires that models iterate within each time step until the difference between the calculated and estimated values falls within a specified predefined tolerance.

Table 1: Coupling strategy in different implementation mechanisms

	<i>implementation mechanisms</i>					
	<i>continuous</i>	<i>discontinuous</i>				
<i>coupling strategy</i>	<i>ping-pong</i>	<table border="1"> <tr> <td>steady state comp.</td> <td>dynamic comp.</td> </tr> <tr> <td><i>any</i></td> <td><i>ping-pong</i></td> </tr> </table>	steady state comp.	dynamic comp.	<i>any</i>	<i>ping-pong</i>
steady state comp.	dynamic comp.					
<i>any</i>	<i>ping-pong</i>					

The early prototypes [16] employ coupling strategies as summarized in Table 1. For coupling to a steady state component model in the discontinuous manner, iterations do not pose additional issues. The program's rewind due to iteration requirements is not necessary as the steady state components models output is only a function of input data (boundary condition) and it is not influenced by its initial state (state history) and the evolution of the simulation time.

However, the situation is different for the continuous mechanism in any case, as well as for the discontinuous mechanism applied for coupling to a transient component model. In any case, a “*passive*” *simulator*, i.e. a coupled program that does not control the iteration process, will have to have a mechanism to rewind its state in order to ensure consistency of simulation data and synchronization of simulation time between the federates. This can significantly increase the effort for the code adjustments. Therefore, the ping-pong approach is applied to continuous mechanism as well as to discontinuous mechanism if the model of the coupled component is transient (results depend on the simulation time evolution). It has been shown that for sufficiently small coupling time step, both coupling strategies give the same results. There is only one information exchange between the coupled programs in each direction per simulation time step.

3. Building performance simulation tools

For the last thirty years many building performance simulation programs have been developed. Based on the heat and mass balance equations, given hourly weather profile, building geometry description and its attribution, and description of a mechanical system and its control strategies, the simulation programs provide time step based calculations, based on which building performance indicators can be determined.

In this paper we focus on mechanical system modeling capabilities of the BPS programs. Three major programs (ESP-r, TRNSYS and EnergyPlus) are chosen and their system modeling capabilities are compared. The programs implement the component-based approach to system simulation, i.e. a modeler can choose from components available in each simulation program and combine/link them together to form various system configurations. However, the programs’ calculation procedures differ and those differences influence the level of program applicability for external coupling.

3.1 ESP-r mechanical system modeling capabilities

ESP-r uses simultaneous modular integration approach. System parts are represented by a discrete nodal scheme. Interactions between nodes are represented by deriving a set of time averaged heat and mass balance equations. The system matrix is then solved for each time step obtaining the simultaneous solution for the overall system. However, the mass and energy matrix equations are solved sequentially, applying iterations if the difference between assumed and final values of marked variables exceeds specified tolerance.

HVAC components models, found in ESP-r take into account the components heat capacitance.

System control is explicitly modeled, defining:

- sensor location and sensed variable,

- actuator location and actuated variable, and
- control law with its settings.

Control variable(s) for each component is predetermined by the model algorithm itself. It does not always have a real world complement. Different sensor/actuator/control law combinations are possible. There are various (although limited number of) controllers in the software database.

3.2 EnergyPlus mechanical system modeling capabilities

The program uses an input-output based component solving approach with iterations for simultaneous solutions (independent modular integration approach). EnergyPlus system representation is based on fluid loops. All system components are attached to them. The loops define the movement of mass and energy within the system. The difference is made between: air loop, plant loop and condenser loop. The loops are indirectly connected (through coil model for example). Each loop has two logical simulation blocks: a supply and a demand side. The demand side places a load to the supply side. These sides are simulated independently, while the convergence between their interaction points is checked and if necessary the iteration procedure is employed. The overall iteration is required to ensure that the results among the loops are balanced.

Most of the components are modeled in steady state fashion.

The control modeling is less explicit then in ESP-r. There is not always a plain sensor-control law-actuator specification as such. The representation of control is somewhat artificial, since it uses the knowledge of the zone load, available only in the simulation model, but not in reality [18]. The control is modeled using a two-level hierarchy: controllers and set point managers. The former can not span a loop manager boundary, meaning that the sensed node and the controlled device must be in the same loop. The latter is able to cross the loop manager boundary, but the application approach does not exactly mimic reality. A set point manager can sense any node variables and use them to determine the set point of any other node variables. This requires that any system controllable variable (air mass flow rate, supply air temperature etc.) is formulated as a function of the zone temperature, i.e. the real controlled variable and the zone load. For example, a set point manager senses a zone temperature, and each time step resets the supply air temperature set point (variable temperature control), that is further used for simulation of a water/air heat exchanger. Alternatively if variable flow control is applied, the required flow rates are set by the zone requirements and passed upstream. As long as the system capability allows the requirements, i.e. mass flow and heating/cooling load will be satisfied.

The components in EnergyPlus are driven by either the (known) load to be satisfied or the set point values at their outlet to be reached, which can be established by the set point managers and varied in simulation time.

The calculation is then “reversed”, i.e. given the required output (load or set point value) the component’s input is calculated. Whatever the driving force is, the components will try to satisfy the requirements as long as they are below or equal to their assigned capabilities (“ideal” control). Apart from some local hard wired control of few components, there are no control laws (P, PI, PID etc.) per se.

The zone temperature is recalculated taking into account the actual system output.

The user specified time step is limited to ten minutes. To ensure the stability of the results, an adaptive time step that is calculated during run-time, is used for system simulation and zone temperature updates.

3.2 TRNSYS mechanical system modeling capabilities

TRNSYS, again an input-output based component solving approach with iterations for simultaneous solutions (independent modular integration). The components are simulated sequentially while the balance of the results is obtained by iteration procedure.

The control in TRNSYS is defined in an explicit manner, as in ESP-r.

4. Comparison of the simulation tools with regards to their applicability to external coupling

4.1 Control simulation features

The possibilities to simulate different control strategies are very limited in all the programs that this paper addresses. Nevertheless, there are differences among them, as mentioned above.

In reality the closed loop control system (control with a feedback), includes the sensor that samples a real world (measurable) variable. It sends the information to the controller that based on the set point value and measured value, and according to the controller-specific control function (algorithm), calculates the control signal that feeds the real world actuator.

However, in the simulation environment a modeler can make use of variables that can not be sensed or actuated in reality, as well as apply a control logic that would not be applicable in reality. For example, a modeler can directly actuate heat flux in his/her model that in reality would be only indirectly actuated by either changing a valve/dumper position or inlet temperature of the working fluid. Furthermore, in simulation a concept of “ideal” control becomes possible due to the accessibility to many variables not known in real world, such as the zone load. The “ideal” control means that the actuated variable will

be adjusted to satisfy the set point requirements (determined from the known load) for the controlled variable, without specifying the explicit control function and by numerically inverting the (forward) simulation components models (from the given output calculate the input).

However, the external coupling creates restrictions upon the control modeling, as not all the variables in a distributed model are readily available in each part of the model. If the control model is distributed, i.e. actuator and sensor represented in distinct simulation environments the control variable, i.e. sensed value or actuated value should be exchanged between the programs. The concept of the load-driven control strategy modeled across multiple environments is implicitly excluded since the information about the required load is not available.

The distributed parts of the simulated system can have different arrangements, depending on the simulation objectives and simulation programs suitability to accomplish it. In regards to the control modeling, two distributed configurations are possible: 1) the controller and actuator are simulated in different environments, and 2) the controller and actuator are simulated in the same environment. The former distributed control model configuration requires that the control logic (controller) is modeled and simulated apart from the actuator. However, the latter enables the use of a component’s hardwired or even “ideal” (if control is local) control model. Therefore the use of hard-wired component’s control excludes its applicability in some distributed system modeling configurations.

The sensed and actuated variables do not have to have real world counterparts, as long as the exchanged variables can be interpreted by both of the coupled programs. The same applies for the control logic itself providing the explicit value of the sensed variable.

4.2 Variable time step

In most of the BPS programs, the user selects the simulation time step prior to the simulation run. Based on the selected time steps of the coupled programs, the coupling frequency can be determined (the coupling time step can not be smaller than the largest chosen simulation time step of the coupled programs). Hence, if any of the programs reduces its simulation time step in the run time (e.g. EnergyPlus), the coupled variables will be kept constant during the reduced time steps calculations. This may lead to the stability problems, due to the shorter time scales of the system and the zone temperatures responses compared to the coupling frequency, which is limited to the maximal predefined simulation time step in the coupled programs, e.g. ten minutes. However, it should be mentioned that only in case of coupling to a steady state component model in discontinuous style, the coupling frequency is not bounded by the specified simulation time steps as long as the synchronization of externally fed disturbances (weather data) is taken care of.

Issues related to the limitation of the user specified simulation time step length (limitation of coupling frequency) are associated with the distributed control modeling. The behavior of a real controller depends on the controlled system, controller settings and controller sampling time. The real controller samples the sensing condition on a very short time scale. However, a mathematical/numerical model of such controller “samples” the sensing condition on the discrete time intervals that are limited with the simulation/coupling time step. Assigning inadequately long simulation/coupling time step can lead to oscillatory/unstable feedback between sensed and actuated variables.

To illustrate the above we use a simple example as shown on the Figure 3. The system employs proportional heating output control to maintain the zone temperature within the desired range, set between 20 and 22°C (throttling range set to 2°C).

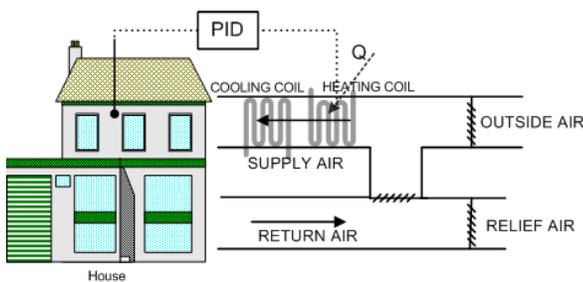


Figure 3 Schematic outline of the simple simulated system example

For a demonstration purpose, the system is modeled in two distinct programs. The zone is modeled in EnergyPlus, while the system and control part is modeled in TRNSYS. Consequently the control model (sensor, actuator and control logic) is distributed.

The simulation output of the PID (proportional mode only) controller depends on the controlled system, the specified throttling range as well as the applied simulation/coupling time step. Two simulations are performed:

- simulation 1: the coupling frequency follows the variations of the time step according to EnergyPlus calculation procedure. This was possible to achieve since this particular TRNSYS model is a steady state model and its output is not influenced by the time step duration and components initial state. In any other case the adaptable coupling frequency is not achievable.
- simulation 2: the coupling frequency set to the common minimum allowed user predefined simulation time step (10 min)

The results from the two one-day simulations are shown on Figure 4.

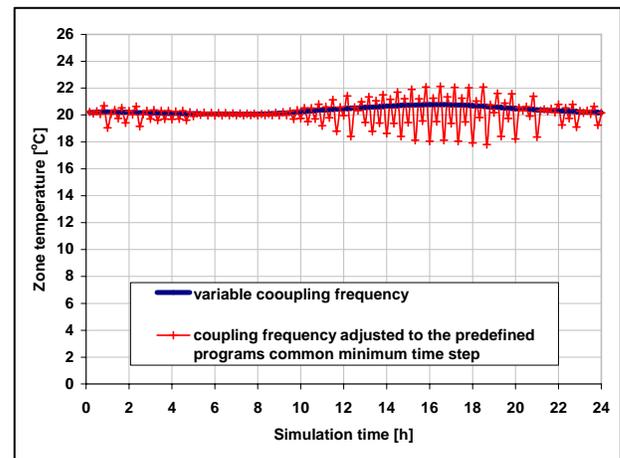


Figure 4 Simulation results for variable and fixed coupling frequency

The coupling time step of ten minutes is inadequately long for the simulated system and its control settings. If the programs are coupled on such small frequency the resulting zone temperature is instable and oscillates. The calculated controller output is kept constant during the reduced time step calculations within the coupling time step. Since the coupling time step is too long, the resulting heating output under-/overshoots the requirements and results in oscillating zone conditions, unless the zone requirements are close to the maximum heating output (Figure 3, zone temperature between 6-8h).

However, if the coupling time step is adjusted according to the simulated system and its control strategy (and the controller settings) the resulting zone temperature shows stable slowly oscillating behavior.

Due to the reasons mentioned above, the restriction of the user predefined simulation time step in some simulation environments limit their applicability for the external coupling. This is particularly the case for the distributed system control simulation.

6. Conclusions

Co-simulation by external coupling of sub-system models can alleviate the limitations of current BPS software. The advantages of individual (legacy) tools can be combined, resulting in a more flexible use of modeling and simulation in general.

The paper summarizes implementation of co-simulation in early prototypes, where the implemented coupling strategies are discussed in more detail. In order to compare legacy programs' applicability for external coupling, the paper provides an overview of the three major BPS tools. They are compared on the basis of two aspects: (1) control simulation features and (2) specification of simulation time step. The comparison shows that not all legacy tools are equally easy to be externally coupled to others.

The main BPS tools' requirements discussed in this paper can be summarized as follows.

- Control modeling and simulation in a distributed manner (sensor and actuator modeled in distinct tools) requires an explicit definition of sensor/sensed variable, actuator/actuated variable and control law. The use of *load-driven* control is excluded, since the information of the required load is not exchanged between programs during run-time;
- A modeler should be able to specify the simulation time step, sufficiently small to assure stability of the results due to different time scales of the simulated system parts, and according distributed control model settings. The early prototypes implement fixed size of coupling time step (the decision is based on the fact that most of the legacy BPS tools use fixed, pre-determined simulation time step size). Therefore, the use of programs with variable time step size in co-simulation is limited with regards to the minimum allowed simulation time step specified by the user. This value makes a lower bound for the coupling time step, as the coupling time step cannot be smaller than the maximum simulation time step assigned in coupled programs.

References

[1] J.L.M. Hensen *On the thermal interaction of building structure and heating and ventilating system*, PhD thesis (Technische Universiteit Eindhoven 1991).

[2] J.L.M. Hensen, J.A. Clarke, Building systems and indoor environment: simulation for design decision support. *Architecture (International Conference on Design and Decision Support Systems in Architecture & Urban Planning)*, Technische Universiteit Eindhoven. 2000, 177-189.

[3] J.L.M. Hensen, E. Djunaedy, M. Radosevic, and A. Yahiaoui. Building Performance Simulation for Better Design: Some Issues and Possible Solutions. *Proc. 21st International Conference PLEA. Passive and low energy architecture*, Technische Universiteit Eindhoven, 2004, 1185-1190.

[4] S.R. Lockley, W. Rombouts, W. Plokker, The COMBINE Data Exchange System. *First ECPPM Conference*, Dresden, 1994.

[5] V. Bazjanac and D.B. Crawley, Industry foundation classes and interoperable commercial software in support of design of energy-efficient buildings. *Proc. 6th International Building Simulation Conference '99*, Kyoto, Japan. 1999, 661-667.

[6] A. Bring, P. Sahlin, M. Vuolle, *Models for building indoor climate and energy simulation. A report of task 22 Building Energy Analysis Tools*. (International Energy Agency Solar Heating & Cooling Programme, Stockholm, 1999).

[7] M.M. Tiller, *Introduction to physical modeling with Modelica* (Kluwer Academic Publisher, Norwell, Massachusetts 2001).

[8] C.A. Boer, *Distributed simulation in industry*, PhD thesis (Erasmus University Rotterdam 2005).

[9] A. Ganse, Distributed processing, http://staff.washington.edu/aganse/presentations/EIS_Distributed_Systems.ppt, University of Washington, Seattle, (accessed: December 2005).

[10] R.M. Fujimoto, Parallel and distributed simulation - introduction and motivation [http://www.cc.gatech.edu/classes/AY2003/cs4230_fall/Lectures/01-intro%20\(8-20-02\).ppt](http://www.cc.gatech.edu/classes/AY2003/cs4230_fall/Lectures/01-intro%20(8-20-02).ppt), Georgia Institute of Technology, College of Computing, accessed: December 2005.

[11] M. Janak, Coupling building energy and lighting simulation. *Proc. 5th International Building Simulation Conference '97*, Prague, Czech Republic, 1997, 307-312.

[12] E. Djunaedy, J.L.M. Hensen, and M.G.L.C. Loomans, Towards external coupling of building energy and air flow modeling programs. *ASHRAE Transactions*, 2003, 771-787.

[13] Z. Zhai, *Developing an Integrated Building Design Tool by Coupling Building Energy Simulation and Computational Fluid Dynamics Program*, PhD thesis. (Massachusetts Institute of Technology, 2004).

[14] CSTB, Type 155 - a new TRNSYS type for coupling TRNSYS and Matlab <http://software.cstb.fr/articles/18.ppt>, accessed: December 2005.

[15] TESS, EnergyPlus - TRNSYS PV, <http://www.tess-inc.com/tess/eplplus.htm>, accessed: December 2005.

[16] M. Trcka-Radosevic, J.L.M. Hensen, and A.J.Th.M. Wijsman, Integrated building and system simulation using run-time coupled distributed models. *ASHRAE Transactions in press*, Chicago, Illinois. 2006.

[17] J.L.M. Hensen, A comparison of coupled and decoupled solutions for temperature and air flow in a building. *ASHRAE transactions*, 105(2), 1999, 962-969.

[18] EnergyPlus, *EnergyPlus Engineering Reference* (University of Illinois and Lawrence Berkeley National Laboratory, 2005.)