Discrete event heat transfer simulation of a room.

Victor Manuel Soto Frances[a,*], Emilio Jose Sarabia Escriva[a], Jose Manuel Pinazo Ojer[a]

[a] *Departamento de Termodinamica Aplicada. ETSII. Universitat Politecnica de Valencia. C/ Camino de vera s/n. 46022 Valencia (Spain)*

Abstract

The paper presents a new proposal of a discrete event simulation (DEVS) model for heat transfer in an enclosure. To our knowledge this is the first proposal in this direction, therefore the model is not comprehensive (has no windows, no ground coupling,etcetera). It is a first step of DEVS into the heat transfer in buildings. The model presented is just one possible implementation for a single thermal zone, although it has been designed to be used in multi-zone models. Common methods like CTF (conduction transfer functions) or RTF (response transfer functions) cannot be used. Instead it employs the successive transition state formulation for the 1D conduction heat transfer through multi-layered walls. An example test room has been calculated and the results compared with software that uses well known methods (CTF,RTF). Finally the DEVS model has been tested with a random convective internal-load signal.

Keywords: Successive state transition method, Building energy simulation, Discrete event simulation, Devs

Nomenclature

N_i Number of boundary conduction elements of the i-zone, equal as well to the number of its boundary surfaces

N Number of volumes or thermal zones

[*] Corresponding author

Email addresses: vsoto@ter.upv.es (Victor Manuel Soto Frances), emases@upvnet.upv.es (Emilio Jose Sarabia Escriva), jmpinazo@ter.upv.es (Jose Manuel Pinazo Ojer)

ΔU_{lg} Change of internal energy of H_2O from liquid to vapor $[Jkg^{-1}]$

ϵ_k Emissivity of surface k

$\sigma_{Stefan-Boltzmann}$ Stefan-Boltzmann constant $5.67 \cdot 10^{-8} [Wm^{-2}K^{-4}\ ]$

F_{kj} View factor between surface k and j

$q_{conv,j}$ Convective heat flux into the zone air from surface j $[Wm^{-2}]$

$Q_{conv,side}$ Convective heat power at either side of a conduction element $[W]$

$q_{conv,src}$ Convective heat flux into the zone air from sources $[Wm^{-2}]$

$q_{rad,src,side}$ Radiation heat input to *side* of any wavelength, due to known sources $[Wm^{-2}]$

$q_{rad-lw,j}$ Longwave radiation heat exchange which depends on superficial temperatures, at surface j $[Wm^{-2}]$

$q_{rad-lw,side,m}$ Longwave radiation heat exchange at side *side* of conduction element m $[Wm^{-2}]$

q_{rad-lw} Vector of longwave radiation heat exchanges within a zone $[Wm^{-2}]$

T Vector of superficial temperatures of the boundary surfaces of the zone

T_0 Superficial temperature at one side of a conduction element $[^\circ C]$

T_1 Superficial temperature at the other side of a conduction element $[^\circ C]$

T_{zone} Temperature of the zone air $[^\circ C]$

C_{da} Specific heat capacity of dry air $[Jkg^{-1}K^{-1}]$

C_{H_2O} Specific heat capacity of water vapor $[Jkg^{-1}K^{-1}]$

E_i Internal energy of the air in the zone i $[J]$

m_{da} Mass of dry air $[kg]$

t Time $[s]$

T_{da} Dry bulb air temperature $[^\circ C]$

W	Absolute humidity $[kg_{H_2O}kg_{da}^{-1}]$

Discrete Event Simulation

$\delta_{ext}()$	External function
$\delta_{int}()$	Internal transition function
$\lambda()$	Output function
σ	Scheduled time for the next call to a certain component $[s]$
dQ_{abs}	The minimum absolute quantum of the QSS integrator
dQ_{rel}	Relative quantum of the QSS integrator
$ta()$	Time advance function
e	Elapsed time from the last call of a certain component $[s]$
S	Set of internal states of a model
s	Internal state of a model
X	Set of input events
x	Input event
Y	Set of output events
y	Output event

Subscripts

01	Refers to the set formed by the magnitude at both sides
$cond$	Conduction
$conv$	Convection
i	Index for zone or volume i
j	Index for surface j
m	Index for conduction element m

side Takes values 0 or 1 to represent each side of a 1D conduction element

src Sources

1. Introduction

The origin of this paper was to study the possibility of using a discrete event simulation (DEVS) model for the calculation of the annual thermal energy needs; demand, consumption, and other related magnitudes of an enclosure. Although there are many well known and powerful software tools as; TRNSYS [1], EnergyPlus [2], Esp-r [3] to name a few, which already perform such computations, we want to start exploring DEVS. The DEVS formalism has many advantages which can be exploited within this field. In the engineering practice it is desirable to compute quickly the heat transfer in buildings to size the equipment (heat exchangers and so on). But at the same time, the heat transfer model employed previously, should be able to cope with more complex calculations, including the thermal dynamical response of the equipment in order to increase the overall efficiency. Unfortunately, the heat transfer problem becomes more difficult because the characteristic response time of the walls and the equipment is very different. In contrast with the aforementioned solutions, DEVS deals with it naturally. Moreover since DEVS calculation is driven by events the model could be used afterwards in optimal real-time control of the heat transfer processes.
Since DEVS method is not very widespread, readers not familiarized with the DEVS formalism are advised to read [4] or other books on the subject.

1.1. Motivation

In general the software tools face the simulation by making an integration which focuses on the time-axis as the leading dimension. The simulation manager keeps track of the components by looking at them from the time axis, just making a picture of everything at a time. EnergyPlus [5] and Esp-r [6] use an adaptive time step. Roughly, the first uses a fixed zone time step and chops it into smaller time steps if the quickest temperature evolution of a zone crosses a tolerance value ($0.3°C/$per zone time step) then all the evolutions of all components proceed with this shorter time step inside the current zone time step.The second splits the overall matrix of the simulation into sub-matrix. They are grouped by their membership (or domains) into ; the building fabric, air flow network, the HVAC system, and so on. Each domain is evaluated at its own frequency.An analogy, for the EnergyPlus case, would be like using a single stroboscope and adjusting its frequency to see everything at the speed of the quickest component, while ESP-r uses several stroboscopes tuned to keep track of each domain.

This way of thinking of integration comes from classical integration methods of differential equations which in turn were designed in times where no high computing devices were available. DEVS formulation was devised to solve the time axis view constraint. Initially it dealt with discrete state computer systems but recently it was extended to continuous systems, like the differential algebraic systems DAEs [7].

When trying to simulate real buildings and HVAC systems, one set of problems comes from the fact that controls like thermostats or forcing functions like lighting, occupancy and equipment (i.e. the internal gains) act suddenly in a ON/OFF manner at any time. This and other types of highly non-linear behaviour causes difficulties to the smooth and "stroboscopic" classical methods in many fields. Even the structure of the differential equations can change due to the action of controls. The other set of problems comes from the fact that the time response of different parts of the model may be quite different. An evident example is the change of the conduction heat flux through walls in common constructions, compared with the response time of the heat power output from the HVAC equipment, as was pointed out before. The first may take hours while the second may take minutes. Moreover the spread of the response times of different parts of the system may lead to stiffness problems. Adaptive methods try to adapt the time step to the most rapidly changing subsystem or even try to discover when a sharp change (ON/OFF type) event is going to occur. Esp-r simulation methods resemble the most, the DEVS proposal of this article. The subdivision is done based on some features shared by the elements of that part: time response, spatial resolution level, the type of problem to be solved (algebraic linear/non-linear equations, differential equations). Upon this subdivision a coordinator algorithm tries to solve the coupling among the parts or modules (which they call onion strategy, while for the decoupled components they use the term "ping-pong") [8]. Anyway it drives the simulation by tracking the time axis and uses several strategies to adapt the time step; *".. boundary condition look ahead (monitors user specified control variable(s) and reduces time-step value if rate of change greater than user specified value), time-step reduction by iteration (reduces time-step value until difference in control variable for current time-step and previous time-step* *is within user specified limit), user specified time-step value, iteration without time-step reduction, simulation rewind (rewind simulation clock to user specified start period if user specified control variable is outside user specified limit)..*"[9]. In article [10], Clarke makes a resume of the iterative solution of nested domains. In the case of EnergyPlus as the Engineering Manual points out [5], the simulation is driven by the integration of a set of ODEs (ordinary differential equations).(To see a discussion about the coordination of the coupling among these domains see [11, 12]).

2. General philosophy and implementation.

A DEVS model is formed by coupling atomic models, which send messages to each other through ports. Any DEVS model is defined by the following set: its internal state set S, the time advance function $ta()$ (used to schedule the time elapsed until its next call, named σ), the external function $\delta_{ext}()$ which deals with the arriving external input events (note that the time elapsed e from the last call is always, $e < \sigma$) and the *internal transition* which in turn is composed of $\lambda()$ and $\delta_{int}()$ functions (the first issues an output event and the second computes the evolution of the internal state)(see [4]). In what follows each atomic element is defined through its own set. The guiding principles used in our design, have been:

- A *static model* behaves like a function. It has no internal dynamics. The $ta(s)$ function returns 0 after a δ_{ext}. So whenever there is an input event, it will follow an *internal transition* and therefore an output at the same time (no time advance). After δ_{int}, $ta(s)$ returns ∞ and the component remains waiting.

- Atomic does not necessarily means simple neither a single spatial point. Processes inside an atomic component will share the same time call.

- Trying to solve an algebraic system of equations (ASE) by an iterative calling of atomic models is not efficient. Therefore whenever possible an ASE must be built and solved within an atomic component.

- The zone energy content is modelled as an ODE. It will be integrated using the QSS (Quantized State System) model (see [13] and [14]). The aim is that every zone evolves in time according to their own needs, therefore all the building zones will not be lumped into an atomic model thus avoiding the EnergyPlus or ESP-r schemes.

- From the point of view of DEVS, *concurrency* is a singularity. This occurs naturally in very quick phenomena like heat radiation exchange which interlinks atomic elements. This singularity must be treated efficiently without time advance.

On this basis we lay down now the atomic models which we propose to simulate a room (zone) (without windows and not ground coupled, so the floor is in contact with the outside air). These are (see Fig. (2));

1. Twofold local surface balance (twofold surface or conduction element).
2. Infrared radiant heat exchange in an enclosure-zone (RAD-lw).
3. Global surface balance (acts as an iterator).
4. Weather (source).
5. Reiterative sum (sums all convective contributions to a zone)(rsum).
6. Integrator.In our case we use the first order accurate QSS.
7. Air node (static function which converts the zone energy content into dry temperature).

2.1. Twofold local surface balance.(Twofold-surface-local-balance)

Here we show the formulation of the atomic DEVS model of a conduction heat transfer element (multi-layered 1D). There can be as many elements of this type as conductive boundaries. It responds passively to excitations at both sides. That means that although not computed, it has an implicit response as time goes by. Only when an external event arrives, the internal state of the element is computed based on; the time elapsed, its characteristic dynamics and the external input by solving the thermal power balance at both surfaces simultaneously (this includes any type of the three heat transfer mechanisms; radiation, conduction and convection, see [5]). This leads to a pair of linear equations were the unknowns are the temperatures at both sides. The function which solves the system is called $solveT()$. It may include variable convection coefficients. In this paper, the values used in the example, are equal to those of the simple model of EnergyPlus (inside:3.076$[Wm^{-2}K^{-1}]$ for walls, 0.948$[Wm^{-2}K^{-1}]$ blocked natural convection, 4.04$[Wm^{-2}K^{-1}]$ non-blocked for ceiling and floor, outside:10.79$[Wm^{-2}K^{-1}]$)(see [5]). The CTF (conduction transfer functions) cannot be employed here, since they use a fixed time step. Instead the *successive transition state method* (see [15]) has been chosen and in Appendix A we present a formulation that fits DEVS. A drawback is that this method needs to evaluate quite costly exponential functions. The Appendix A assumes that the evolution of the superficial temperatures between transitions is linear. In another paper we will show how to extend this hypothesis.

The state S of the wall is characterized by the number of poles n chosen (see Appendix A and section 4) . The internal states are {number of poles (K) × surface temperature} at each side (0 or 1) × state × time advance, thus:

$$S = S_{01} \times T_{01} \times state \times \sigma$$

where:

$$S_{01} = \mathbb{R}^K \times \mathbb{R}^K$$

$$T_{01} = \{T_0, T_1\} \in \mathbb{R}^2$$

$$\{state \in \mathbb{Z} | \text{ re-compute=0, converged=1}\}$$

$$\sigma \in \{0, \infty\}$$

In the surface balance equations, any type of excitation is considered at both sides whether it exist or not. If it does not exists then an event will never arrive at that port. There are nine input ports in total, two sets like this $\{T_{zone}, q_{rad-lw}, q_{rad-src}, weather\}_{side}$ where $side \in \{0, 1\}$ plus a $message$ port. To each port an ordinal integer is assigned starting at zero, using an even/odd numeration to identify each side (see Fig.(2)). T_{zone} is the dry temperature of the zone at that side (if it exits). Otherwise is left null and other known boundary condition is taken. In the same manner, q_{rad-lw} is the longwave infrared radiation exchange with other surfaces of the zone and depends on their temperatures, while $q_{rad-src}$ is a known radiation input from a source (people,lights, equipment) of any wavelength, which does not depend on any temperature explicitly. These, in turn, could be modelled independently as atomic components or as a unique component of internal gains with several components (convective, radiant, latent, etc.) and several connections. Finally the $weather$ is a set of climate conditions (see section 2.4).The $\delta_{ext}(x)$ [1] is given by:

[1] Remark: when an event different from the weather arrives, the weather variables are interpolated linearly since its trajectories are piecewise linear (see section 2.4)

$$\delta_{ext}(s,e,x) = \begin{cases} \text{transition(s,e)} & \text{, if } e > 0 \\ T_{01,new} \leftarrow \text{solveT()} & \text{, if x.port} \in \{T_{zone}, q_{rad-src}, weather\} \\ \text{state} \leftarrow \text{re-compute} & \\ \sigma \leftarrow 0 & \\ \sigma \leftarrow \infty & \text{, if x.port } \in q_{rad-lw,side} \\ \text{state} \leftarrow \text{re-compute} & \text{, if x.port=message \& x.value=re-compute} \\ T_{01,new} \leftarrow \text{solveT()} & \\ \sigma \leftarrow 0 & \\ \text{state} \leftarrow \text{converged} & \text{, if x.port=message \& x.value=converged} \\ s_{01} \leftarrow \text{evolvestates(s,e)} & \\ T_{01} \leftarrow T_{01,new} & \\ \sigma \leftarrow 0 & \end{cases} \tag{1}$$

(For a discussion about *transition()* and *evolvestates()* functions, see Appendix A). The time advance function $ta(s)$ just returns the σ value when called. An external transition can be summarized as follows. If an input event arrives after an elapsed time e the transition and the $T_{01,new}$ are computed, the state is set to $re-compute$, and gets ready for an immediate internal transition. If the event message is "re-compute" then computes the $T_{01,new}$ and immediate internal transition is scheduled. In case of concurrency of models, they are called by their ordinal appearance in the input file. A special input is the q_{rad-lw} signal. The design forces concurrency of all conduction elements within a zone in order to solve the thermal radiation exchange. If an infrared event arrives then the time advance is set to infinite and therefore it is kept waiting until a $re-compute$ or $converged$ message arrives. These q_{rad-lw} and $message$ events are sent by the iterator (section 2.3). Finally, if a $convergence$ message arrives in, then all the elements of a zone will be scheduled for immediate internal transition and the state is set to $converged.$

The value of an output event depends on the output port (see the conduction element in (Fig.(2))). There are two ports, numbered as ; 0 and 1, used when the internal $state$ is $re-compute$ or $converged$ respectively. In case of output-port=0, $Y_0 = \mathbb{R}^4$, $\{y \in Y_0 | y(0) = T_0, y(1) = T_1, y(2) = q_{rad-lw,0}, y(3) = q_{rad-lw,1}\}$ for iteration purposes, while in case output-port=1, $Y_1 = \mathbb{R}^2$, $\{y \in Y_1 | y(0) = Q_{conv,0} = Area \cdot h_{conv,0} \cdot (T_0 - T_{zone,0}), y(1) = Q_{conv,1}\}$

, converged values are passed to the air zone node element. Finally:

$$\lambda(s) = \begin{cases} (y \in Y_1, port = 1) & \text{if state = converged} \\ (y \in Y_0, port = 0) & \text{if state = re-compute} \end{cases}$$

$$\delta_{int}(s) = \{ \ \sigma \leftarrow \infty$$

Thus as a passive element, it will keep waiting for an external input event to arrive.

2.2. Infrared radiant heat exchange in an enclosure.(RAD-lw)

This element calculates the infrared radiant heat flux exchange for a given set of temperatures. It solves a linear system of equations for the q^{new}_{rad-lw} (or q^{new}_{rad} for short, when needed) of all the surfaces of a zone for a given set of their superficial temperatures. We assume, as usually done, non-participating media and grey surfaces. Thus no linearisation of the radiation heat exchange is done. The equation for the k-surface of a thermal i-zone, which forms the linear system for its N_i surfaces, can be written as a function of emissivities (ϵ_k), view factors (F_{kj}) and Kronecker's delta (δ_{kj}) as:

$$\sum_{j=1}^{N_i} \left(\frac{\delta_{kj}}{\epsilon_k} - F_{kj} \frac{1-\epsilon_j}{\epsilon_j} \right) q^{new}_{rad-lw,j} = \sum_{j=1}^{N_i} (\delta_{kj} - F_{kj})\, \sigma_{Stefan-Boltzmann} T_j^4 \qquad (2)$$

The proposed DEVS model needs to be of the static type. The RAD-lw states are:

$$S = T \times q^{old}_{rad} \times q^{new}_{rad} \times \sigma$$

where N_i = number of surfaces in volume i, $\sigma \in \{0, \infty\}$, and:

$$T \in \mathbb{R}^{N_i} \text{vector of superficial temperatures}$$

$$q^{old}_{rad-lw} \in \mathbb{R}^{N_i} \text{vector of old infrared radiant heat fluxes}$$

$$q^{new}_{rad-lw} \in \mathbb{R}^{N_i} \text{vector of new infrared radiant heat fluxes associated with } T$$

This element is concurrent with the previous conduction elements (see 2.1). There should be as many as different zones. In the priority queue all RAD-lw go after all the Twofold-surface-local-balance. Therefore after the first input event coming from a (2.1) element, the RAD-lw $\delta_{ext}(s, e, x) = \{\sigma \leftarrow 0$, thus forcing this element to wait at the priority queue while it is gathering all the new temperatures from the surfaces to be used here to compute q^{new}_{rad}. The time advance function just returns σ. The output function has just one port 0:

$$\lambda(s) = \begin{cases} \text{compute } q_{rad}^{new} = f(T) \\ y = (T, q_{rad}^{old}, q_{rad}^{new}, port = 0) \end{cases}$$

The output goes to the global-balance (2.3) element. Finally, $\delta_{int}(s) = \{\sigma \leftarrow \infty$.

2.3. Iterator.(Surface-global-balance)

This element is placed after the RAD-lw elements (2.2) and there should be only one element of this type. So its priority under concurrency conditions comes after them. Its mission is to check that the thermal superficial power balance of the zone or zones (in case of multi-zone building) is fulfilled up to some tolerance. This time by the zone surface we understand the boundary surface of the zone, i.e., *the union of all the boundary surfaces of such zone.* By calling the local surface (2.1) elements with a certain $q_{rad-lw,side}$ fixed at each side, we are forcing just a local balance;

$$q_{rad,src,side} + q_{cond,side} + q_{conv,side} = q_{rad-lw,side} \tag{3}$$

which provides the T vector for a zone (*note: the sub-script "side" is to emphasize that it could be 0 or 1 depending on the conduction element orientation*). Then by calling RAD-lw (2.2) with these new temperatures gives us a new vector q_{rad-lw}^{new} for the zone. It happens that $q_{rad-lw}^{new} - q_{rad-lw}^{old} = \Delta q_{rad-lw} \neq 0$. If the set of equations (3) were balanced at the i-zone level for its vector T, summing over its N_i surfaces (or boundary conduction elements) would give us:

$$\sum_{m=1}^{N_i} q_{rad,src,side,m} + q_{cond,side,m} + q_{conv,side,m} = \sum_{m=1}^{N_i} q_{rad-lw,side,m} = 0 \tag{4}$$

, but using in the right hand side, the right values q_{rad}^{new}, those which are in agreement with the actual T used to compute the left hand side, one gets:

$$\begin{aligned} \sum_{m=1}^{N_i} q_{rad,src,side,m} + q_{cond,side,m} + q_{conv,side,m} = \sum_{m=1}^{N_i} q_{rad-lw,side,m}^{new} = \\ = \cancelto{0}{\sum_{m=1}^{N_i} q_{rad-lw,side,m}^{old}} + \sum_{m=1}^{N_i} \Delta q_{rad-lw,side,m} \end{aligned} \tag{5}$$

since the infrared radiation values are always balanced. Therefore the last term of eq. (5) measures the unbalance at the zone level (in fact we computed the residual as the $\| \bullet \|_\infty$ norm, $\| \Delta q_{rad-lw} \|_\infty < tol_{zone}$). Therefore some kind of iteration over T and thus over q_{rad-lw} is needed. In our case we just use the q^{new}_{rad-lw} as the new estimate, which is sent back to the corresponding Twofold-surface-local-balance or conduction elements (see Fig.(2)). The states are:

$$S = Flags \times state \times \sigma$$

$\sigma \in \{0, \infty\}$, where in case there were N zones (or volumes) :

$$Flags \in \mathbb{B}^N \quad , \mathbb{B} \in \{0,1\} \quad ,\text{set when not converged volume}$$

$$state \in \mathbb{Z} \quad , \quad \{\text{converged,re-compute}\} :\rightarrow \mathbb{Z}$$

$$\delta_{ext}(s,e,x) = \begin{cases} flags(x.port) = 1 & , \text{ if } residual(x.value) \geq tol_{zone} \\ flags(x.port) = 0 & , \text{ else} \\ \sigma \leftarrow 0 & \end{cases}$$

The advance function just returns σ. $ta(s) = \{\sigma$. The output port and value, depend on the i-zone and on the global convergence:

$$Y_1 = q^{new}_{rad-lw} \times port \quad , port \in \{1, 2, \cdots, N\}$$

$$Y_0 = message \times port \quad , port = 0$$

The output function is:

$$\lambda(s) = \begin{cases} y = (message \leftarrow converged, 0) \in Y_0 & , \text{ if all Flags==0} \\ state \leftarrow converged & \\ y = (q^{new}_{rad-lw}, i) \in Y_1 & , \text{ if Flags(i)==1} \\ state \leftarrow \text{re-compute} & \\ Flags(i) \leftarrow 0 & \\ y = (message \leftarrow re - compute, 0) \in Y_0 & , \text{ if state==re-compute \& all Flags==0} \end{cases}$$

The internal transition function is just

$$\delta_{int}(s) = \begin{cases} \sigma \leftarrow 0 & , \text{ if(still Flags on or message to be sent)} \\ \sigma \leftarrow \infty & , \text{ else} \end{cases}$$

In short. Only those m-conduction elements (2.1) belonging to a not converged zone will receive an input event about the new $q_{rad-lw,side,m}$ and they are kept waiting until a $re - compute$ message arrives. If all zones have converged the corresponding *message* event is issued to all the (2.1) which in turn will send out an event to the reiterative sum (2.6), thus breaking the iteration. Notice that by selecting properly tol_{zone} not necessarily all zones should be called, only those unbalanced. This will be studied in another paper.

2.4. Weather source.(weather).

We give here the basic aspects of this model. It is a static DEVS function. It only has output events, thus acting as a driving source. The main points are:

- The weather data is read from a binary file. It contains standard hourly data; dry temperature, wet bulb temperature, total solar radiation on horizontal plane, beam or direct radiation, cloudiness, etc.
- When initializing this component, it pre-calculates excitations which depend on the orientation (like the solar radiation on a tilted surface). It does as many pre-calculations as different orientations are encountered. Although there are no windows this component could be initialized to estimate the hourly internal heat gains from the sun for each conduction element.
- Although many scheduling possibilities exists, every hour sends an output event. Another possibility would be to schedule an output each time a certain signal crosses some threshold and assigning several output ports for each one. However we preferred to keep the model simple.
- The output trajectories are not piecewise constant but piecewise-linear making up a continuous function.

2.5. Air volume.(Air node).

Assuming a spatial nodal model for the zone, the differential equations should be written for; energy balance, water vapour mass balance and air mass balance. The three dynamical variables are correspondingly; internal energy of the air, mass of water and mass of air. However in this DEVS model proposal, it has been assumed that the volume is closed, so there are no changes of water or air masses.

So the differential equation is just:

$$\frac{dE_i(t)}{dt} = \sum_{j=1}^{N_i} q_{conv,j}(t) + q_{conv,src}(t) \tag{6}$$

Although equation (6) keeps track of an extensive variable, the intensive thermodynamic variables are needed like (dry temperature, wet bulb temperature and pressure). In our case we only need the dry temperature $T_{da,i} = T_{zone,i}$ and can be obtained from the internal energy of the i-volume as equation (7).

$$T_{da,i}(t) = \frac{E_i(t) - W_i \cdot \Delta U_{lg}}{m_{da,i}(C_{da} + W_i \cdot C_{H_2O})} \tag{7}$$

,where E_i is the internal energy of the air of the zone, ΔU_{lg} is the change of internal energy of H_2O from liquid to vapor, C_{da}, C_{H_2O} are the specific heat of dry air and water vapor respectively and W_i is the absolute humidity. This DEVS element is also a static function (see Fig.(2)).

2.6. Reiterative sum. (rsum)

We are not going into the details. We just give the idea behind this component. It is a static function. Its aim is to evaluate the right hand side of eq.(6). This component appears after the Twofold-surface-local-balance or conduction elements, since when convergence is achieved the q_{conv} from those elements must be added. The output event is the total heat power going into the air and it is sent to the QSS integrator (see next 2.7 and Fig.(2)).

2.7. Integrator of quantized state system QSS

This is a new way of integrating differential equations using DEVS formalism. We just point out its main features. QSS means *Quantized State Systems.* The idea is to change the continuous variable by its quantized counterpart. The size of the quantization must be chosen by the user (there is a minimum and a relative quantum). The first order QSS1 (QSS in short), gets a piecewise constant trajectory. In our case the rate of convection heat (power) going into the zone air. The integration of this trajectory is an internal piecewise linear (continuous) evolution of the energy content of the zone air. This approximation is called first order quantization. The energy content of the zone (continuous variable) is chopped into quantum values inside QSS element. Whenever the integrated internal variable (which in first order approximation, is linear) reaches the next quantized value, an output event with the new quantized value is sent out, and the QSS state changed to this new value. To avoid cycling QSS posseses hystheresis. Summing up, QSS reveives a piecewise constant trajectory (the value of the derivative in eq. (6)) and outputs also a piecewise constant trajectory (the quantized integrated value of the energy content of the zone). Notice that the QSS-integrator needs to be able to schedule its next internal transition by computing the necessary elapsed time required to reach the next quantized value. One may think that the evolution of the energy content of the zone (i.e., the dry temperature), would look more like an exponential profile as EneryPlus does (see options of integrators in [5]). However the integrator, as we will show, will follow this exponential-like evolution using this piecewise linear approach inside QSS by issuing more frequent output events when the T_{zone} is changing quickly. There are other higher order implementations and variants ; QSS2, QSS3, BQSS, CQSS, LIQSS, LIQSS2, etcetera. Briefly, QSS2 and QSS3 are similar to QSS. QSS2 gets and outputs a piecewise linear (not continuous) trajectory of events. The input contains the velocity (derivative) and the acceleration (second derivative), and the output, its integrated value and slope. Inside the QSS2 the integrated continuous variable has a piecewise parabolic (continuous) trajectory. It is easy to imagine that the more information is available at the input event (like in QSS2) the less frequent will be the issue of output events since the approach is better, thus increasing the speed of the computation. The QSS3 follows the same philosophy but with piecewise parabolic input/output trajectory of events. In another paper it will be explained in detail, how the DEVS model presented in this paper, might be extended to cope with QSS2 and QSS3 integrators and their comparison. To get more information about QSS integrators please see [14] and [13].

3. Calculation: example.

In this section we show the results of a single room: not coupled to the ground, without windows and internal gains, for testing the implementation. The room shown in fig. (1), has a volume of $45m^3$. The weather file employed has been the *DRYCOLD.EPW* of the *BESTEST* standard [16]. The exact view factors are shown in table 1 and the constructive solutions for the walls, floor and roof are shown in table 2. The shortwave length emissivity at both sides is 0.85 and the infrared (longwave length) has been kept at 0.8 inside and 0.9 outside for all the enclosures.

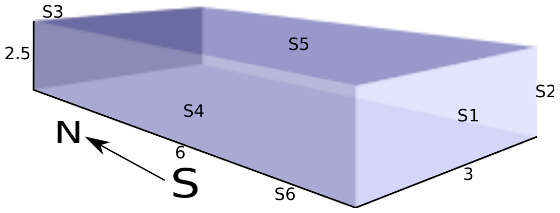

Figure 1: Test room,dimensions (meters), orientation and surface numbering.

Number	-6	-5	-4	-3	-2	-1
-6	0.00000	0.34276	0.22224	0.106380	0.22224	0.106380
-5	0.34276	0.00000	0.22224	0.106380	0.22224	0.106380
-4	0.26668	0.26668	0.00000	0.107750	0.25114	0.107750
-3	0.25532	0.25532	0.21550	0.000000	0.21550	0.058366
-2	0.26668	0.26668	0.25114	0.107750	0.00000	0.107750
-1	0.25532	0.25532	0.21550	0.058366	0.21550	0.000000

Table 1: View factors.(Because of the implementation the numbers are changed their sign and ordered starting at the smallest).

3.1. Ordering of the models and scheme.(Select function)

As mentioned PowerDevs [17] uses the ordinal position in the input file to solve the concurrency problem. In our implementation the components appear in the next sequence: (1) Air-node [2], (6) Twofold-surface-local-balance, (1) Weather, (1) RAD-lw, (1) RSUM, (1) QSS-integrator, (1) Output-report. All these are atomic elements. There is one more component, the root element that is a coupled element which contains all the previous elements. As mentioned it acts as the simulator manager or engine.
Roughly and to sum up, Fig. (2) shows that each conduction element may receive input events due to the weather, known sources of radiation, infrared radiation and the zone temperature at either side (0) or (1) and a *message* from the global balance iterator. If *message* is *converged* then output-port 1 contains an event with the convective power at each side. This power may

[2] Note: the meaning is; (amount of elements)-type of element

Name	k$[W/m/K]$	ρ $[kg/m^3]$	c_p $[J/kg/K]$	Thickness$[m]$
Wall				
Solid brick	0.870	1800.0	1380.0	0.120
Concrete mortar	1.400	2000.0	1050.0	0.015
Polystyrene foam	0.033	25.000	837.00	0.040
Void brick	0.490	1200.0	920.00	0.040
Gypsum	0.300	800.00	920.00	0.015
Roof				
Pavement	1.100	2000.0	1380.0	0.050
Waterproof film	0.190	1100.0	1680.0	0.010
Light concrete	0.350	1000.0	1050.0	0.100
Concrete filler block	1.540	1254.0	1050.0	0.260
Gypsum	3.000	800.00	920.00	0.020

Table 2: Constructive solutions. The order is from outside to inside. (The floor is the same as the roof but the order of the layers is reversed).

define the evolution of a zone air temperature (if it exists at that side), in which case is integrated by; rsum, QSS and Air node elements. Otherwise (not converged case) the output-port 0 of the conduction element contains its superficial temperatures and corresponding $q^{old}_{rad-lw,side}$ at both sides. This event goes to input port 0 or 1 of RAD-lw, according to the conduction element orientation towards the zone. There should be one RAD-lw per zone. RAD-lw calculates the q^{new}_{rad-lw} vector, based on the conduction superficial temperatures, and sends it to the global balance iterator at the corresponding input port assigned to the zone. Finally, the iterator checks the convergence or not of every zone and accordingly sends back a *message* and/or the q^{new}_{rad-lw} to the corresponding conduction elements. In the latter case a new iteration starts.

3.2. Parameters

The list of the main internal parameters of the simulation is presented here. See the section 4, for a discussion of their values.

- The relative quantum (as a percentage of the signal) (dQ_{rel}) and the minimum absolute quantum (dQ_{abs}).

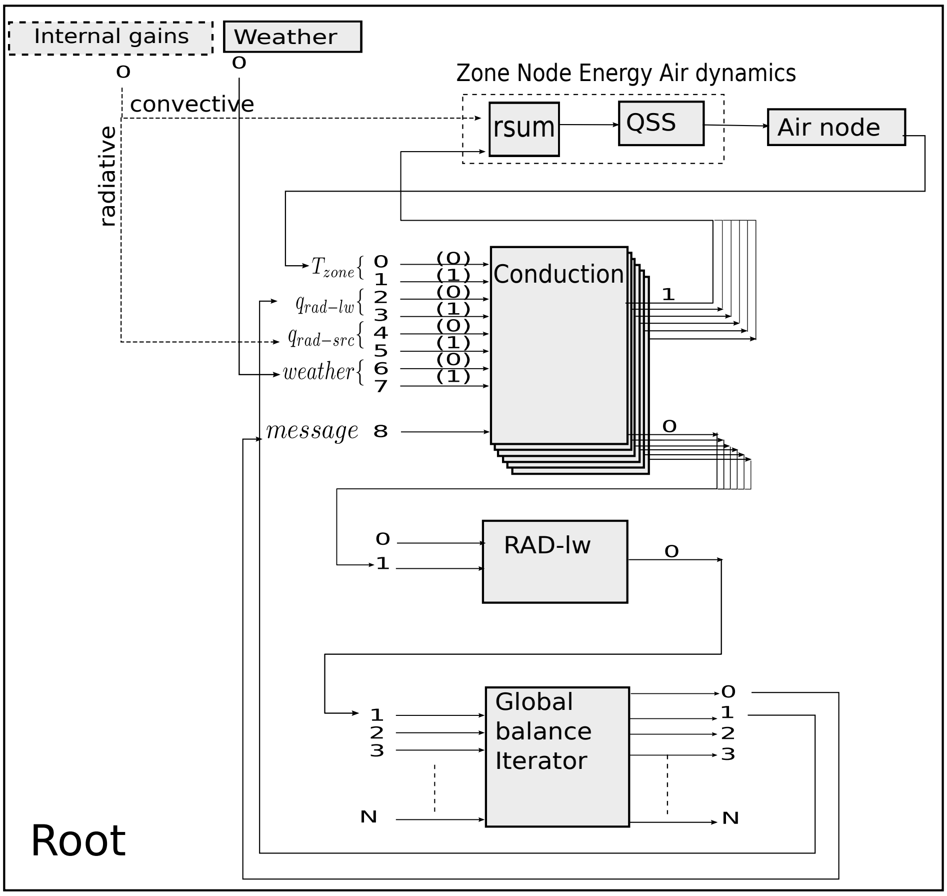

Figure 2: Scheme and relationships of the atomic and root elements for the proposed DEVS model, applied to a single thermal zone.

- The highest absolute value of the pole used α_{max}, that is $\alpha_p \leq \alpha_{max}, \quad p \in \{1, 2, \cdots, P\}$. The amount of poles P depends on the construction.
- The truncation value of the poles α_{trunc}, that is $\alpha_k \leq \alpha_{trunc}, \quad k \in \{1, 2, \cdots, K < P\}$. The poles and residuals $k \in \{K+1, \cdots, P\}$ are used to compute Q^m (see Appendix A).
- The tolerance of the superficial thermal power balance at the zone level tol_{zone}.

4. Results and discussion

As a validation or comparison test, we have checked the results of the example (see section 3) obtained by our DEVS implementation to those obtained by EnergyPlus v7.1. We have tried to use the same assumptions in our implementation, as EnergyPlus. Nevertheless the main differences are; we use an isotropic model for the radiation from sky and although we have used the same sky temperature model used by EnergyPlus, we still found some differences. Moreover, EnergyPlus employed a zone time step of 15 minutes, while the elapsed time is variable and much shorter in DEVS (around a few minutes or seconds) and we think that this causes some differences in the results (see Fig.(3)). Finally EnergyPlus uses a warm up cycle at the start of the simulation which we did not use. The step response dynamics computed by the CTF-values issued by EnergyPlus and that achieved by our computed poles/residuals were checked. Figure (3) shows that the overall agreement is good. Notice that the trajectories are piecewise constant altough a linear interpolation has been used in both cases (EnergyPlus and DEVS). Our DEVS implementation gives a slightly lower T_{zone} and daily peaks of convective heat gains than EnergyPlus but both follow the same pattern.

In the implementation although care has been taken in making computations quick and efficient, the goal has not been the speed. This point will be treated in more detail in future works along with the effect of the parameters on the speed and precision. The test were done on a linux operating system and the CPU was IntelCore i5 650.

First, We notice that in case the zone temperature is imposed (i.e. it is fixed), the only driving source is the hourly weather component. In this case the simulation proceeds very fast (takes seconds), since no QSS integration is performed and only very few states remain after an hour, in the successive state transition. Therefore we focus on the floating temperature case, which needs QSS integration to keep track of the zone state.

We computed the conduction elements with $\alpha_{max} = 500$, resulting 2392 poles for the wall and 3801 for the floor and roof. In order to get a good response from the conduction elements we arrived at $\alpha_{trunc} > 100$ by trial and error. We chose $\alpha_{trunc} = 400$. That means that starting from a state, if the elapsed time is short enough, as many states could be excited per each

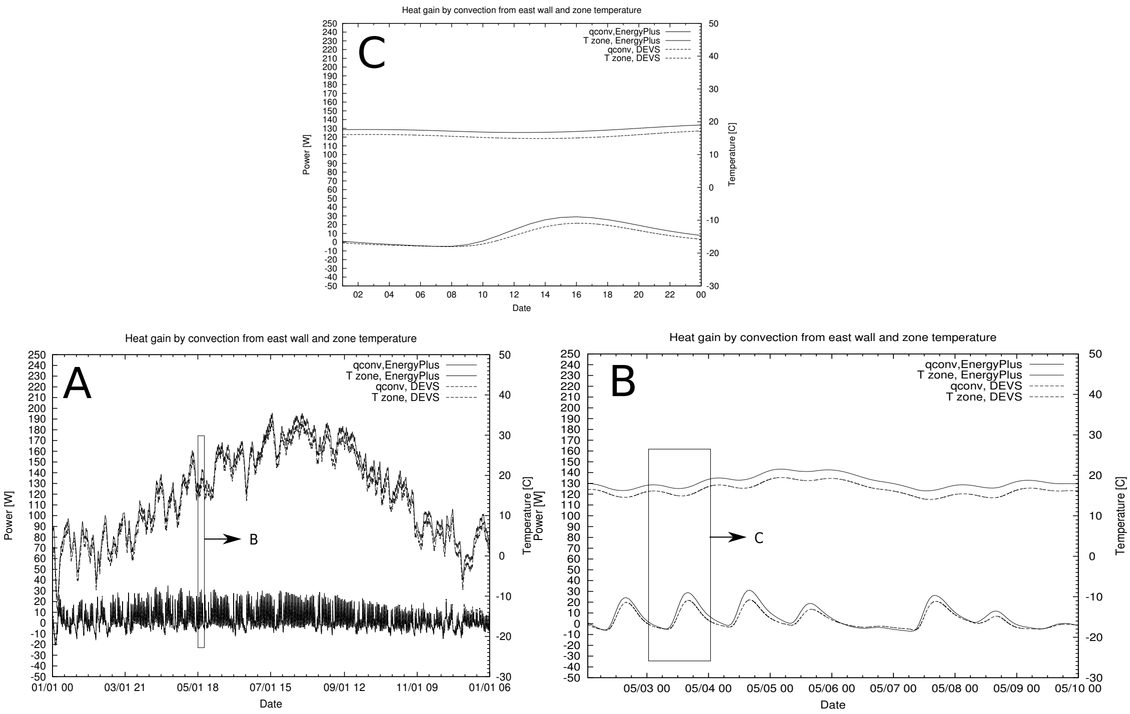

Figure 3: The time format is date:"month/day hour:minute". A) year B) week C) day (only hours shown). The bottom curves in each graph refer to heat gain and the top curves to the zone temperature.

side as poles exist beetwen 0 and α_{trunc}. In order to save computation effort, it is needed to know beforehand and quickly how much of them are actually excited. The expression $\varphi_k = e^{-\alpha_k t_e} \leq 10^{-10}$ was used (10^{-10} arbitrarily chosen). So for a given elapsed time t_e we used the inequality:

$$\alpha_{k-state-max} > \frac{10ln(10)}{t_e} \tag{8}$$

.i.e, $\forall k \geq k_{-state-max}, \varphi_k = 0$. A difficult point arises when an input event arrives in with a very short elapsed time t_e. The $\frac{Q^m}{t_e}$ (see Appendix A), that actuates as a small corrective term, dominates. So the dynamics is modified. Starting from the general solution (A.1) and in the limiting case $t_e \to 0$, we assumed $0.001|\sum_{k=1}^{K} A_k^m| > |Q^m|/t_{e-min}^m$. Taking into account the expression for Q^m we get:

$$t^m_{e-min} = 1000 \frac{|\sum_{k=K+1}^{P} A^m_k/\alpha_k|}{|\sum_{k=1}^{K} A^m_k|} \tag{9}$$

so finally $t_{e-min} = max\{t^m_{e-min}, m \in \{X, \pm Y, Z\}\}$. In our example $t_{e-min} =$ 2.78 seconds for the walls and $t_{e-min} = 9.72$ seconds for the roof and floor. Therefore while computing the transition in eq. (1) if the $e = t_e < t_{e-min}$ then $t_e = t_{e-min}$ is forced. This could be troublesome if a signal of very high frequency arrived at these elements but is not the case. If eventually happens then it appears as a very small spike in the q_{conv} signal (in the order of 0.01[w], not visible in figures (4) and (3)). Therefore in practice this approach seems to work.

Finally for QSS-integrator the parameters $dQ_{rel} = 10^{-6}$ and $dQ_{abs} = 50[J]$ were selected by trial and error. Normally literature suggests $dQ_{rel} = 10^{-2}$, but in our case the signal chosen is the energy content in the zone (which is a big extensive magnitude) (nor the temperature, an intensive one). The smaller the dQ_{rel}, the greater the calculation time. Unfortunately since the integration is first order, this tolerance must be small in order to keep track properly of the zone temperature. This leads QSS to issue a frequent output event thus making also necessary to use a higher α_{trunc}. In another paper it will be shown how the model can be improved using QSS2 and QSS3 (second and third order integrators). The increment in the order makes, the DEVS model, less sensible to the value of the parameters, much quicker in calculating since it requires to keep track of a lesser amount of states.

Finally the tol_{zone} employed in the Global-surface-balance element was $0.01[W]$.

Looking at the scheme of Fig. (2) the source of events from the weather, instead of being read from a file, could arrive to the coupled root model from sensors outside the PC (through its serial port, for instance). In the same way, the internal gains events could be an external input to the root. In order to show the potential of the DEVS formalism, we have modified a random signal generator from PowerDevs [17] trying to emulate a pure (and a priori unknown) random pure convective internal heat gain of $100[W]$ which is switched ON and OFF alternatively. The randomness comes from the duration of each ON/OFF state. The time elapsed between each transition of state is modelled as a statistical variable with normal distribution with mean 5 hours and standard deviation of half an hour. Thus, sometimes takes more than 5 hours and sometimes less. A remarkable point is that in a real situation the model would not know when a change would happen.

Figure 4 shows the results at several time scales; annual, an arbitrarily chosen week and a day of that week. Within this day, graph D shows a detail of the exponential-like shape of the evolution of the T_{zone} and the convective heat gain from the east wall. Notice that it is a switch-ON event, so the heat flows out of the room and T_{zone} rises. As Fig. 4 shows, it takes around 15-20 minutes for the room to reach an almost constant temperature. The capability to capture such different response time scales is of practical interest to calculate losses of energy efficiency due to ON/OFF or partial load of thermal equipment.

The DEVS implementation self-adapted to the events without previous knowledge of them to occur. The QSS element issued more frequent events (due to the change in the zone energy content) during these shorter periods of time.

5. Conclusions

As section 4 shows, DEVS formalism might be a promising technology for energy simulation in buildings or even, since the simulation is driven by events, the model could be used in controlling actual building facilities.

The aim of this paper was just to pay attention to the application of the DEVS simulation methodology. Although it is used, currently, in the computers science field, the idea was to extend it to the simulation of energy in buildings. An important reason for that is that it has a great potential to improve speed, integrate HVAC dynamics and controls and to keep modularity. From its design, it allows the self-adaptation to the different dynamics of the components.

This self-adaptation capability to different dynamics and to very non-linear models or even cope efficiently with random signals is very desirable to extend its use in the engineering practice. Moreover, the model can be used in a standard/quick simulation which looks just for the energy demands or peak loads, but at the same time, can also be used in more costly and detailed full-dynamic situations, where the dynamics of the HVAC systems are included. DEVS changes the point of view from the time axis to the states of the components (time is just one more).

To our knowledge this is the first proposal or attempt to develop such a

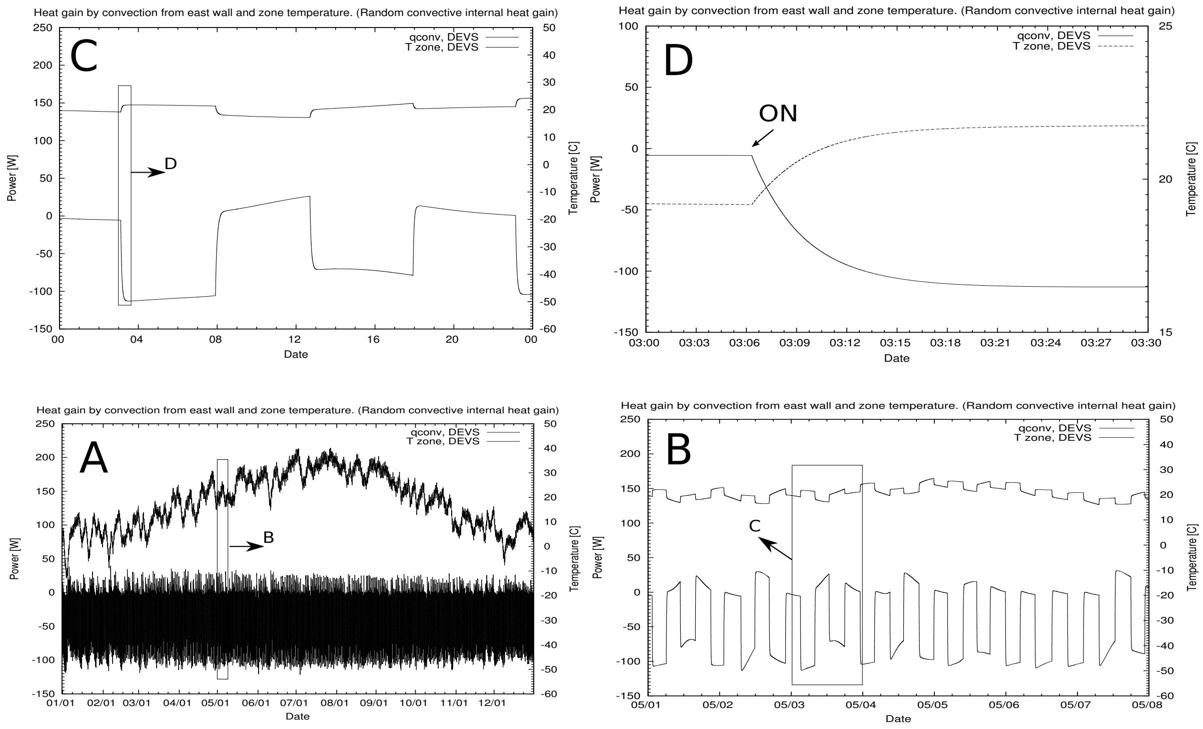

Figure 4: The time format is date:"month/day hour:minute". A) year B) week C) day (only hours shown) D) switch ON event detail (hours:minutes). The bottom curves in each graph refer to heat gain and the top curves to the zone temperature.

model and for sure it will need improvements and fixings. The implementation from the DEVS point of view is not so straight forward, although once accustomed, it is not so difficult to write down a model.
In the future our intention list includes:

- Topics which affect the kernel of the method: to increase the order of the quantized integrators to QSS2 and QSS3; to increase speed and reduce the number of events; to reduce the number of states needed to represent the dynamics of multi-layered conduction elements within the practical frequency range.

- Topics which extend the applicability of the DEVS model: to include other models (windows, ground coupling, and so on); to check the behaviour of this proposal, in the multi-zone case with excitations at the periphery and at the core of the building; to implement controls; to implement dynamics of equipment.

Acknowledgements

We would like to acknowledge the help in using the PowerDevs software [17] to Federico Bergero.

Appendix A. Successive transition state

The successive transition method can be tracked back to Urano et al (see [18], [19]) or Testuo Hayashi et all (see [20]). A short resume of the method is the following. The response in heat flux on a surface of a multi-layered wall to a unitary (step) temperature excitation is expressed as:

$$\phi^m(t) = A_0^m + \sum_{k=1}^{\infty} A_k^m e^{-\alpha_k t} \tag{A.1}$$

Where $m \in \{X, \pm Y, Z\}$. The well-known matrix [21] expression for the heat fluxes (W/m^2) in Laplace domain, in our case is [3]:

$$\begin{bmatrix} \mathcal{L}q_0''(s) \\ \mathcal{L}q_1''(s) \end{bmatrix} = \begin{bmatrix} \mathcal{G}_X(s) & \mathcal{G}_{-Y}(s) \\ \mathcal{G}_Y(s) & \mathcal{G}_Z(s) \end{bmatrix} \begin{bmatrix} \mathcal{L}T_0(s) \\ \mathcal{L}T_1(s) \end{bmatrix} \tag{A.2}$$

The $A_0^m, A_1^m, \cdots$ in equation (A.1), are the residuals at the corresponding poles -α_k. These poles have been calculated using the Hittle method (see [22]) and characterize completely the dynamic thermal response of the element. The greater the k the biggest the α_k becomes. There are four A sets, well in fact three since $A^{-Y} = -A^Y$. In all of them A_0 is the thermal conductance of the element. The infinite series cannot be computed so the series must be truncated at $k = K$.

$$\phi^m(t) = A_0^m + \sum_{k=1}^{K} A_k^m e^{-\alpha_k t} + Q^m \delta(t) \tag{A.3}$$

[3] (0 ≡ origin of spatial axis and 1 ≡ the other end side).

In eq.(A.3) δ is the Dirac's function and $Q^m = \sum_{k=K+1}^{\infty} A_k^m/\alpha_k$. Finally the transfer function $\mathcal{G}_m(s)$ is obtained multiplying the Laplace transformation of $\phi^m(t)$ by s.

$$\mathcal{G}_m(s) = A_0^m + \sum_{k=1}^{K_0} \frac{A_k^m s}{s+\alpha_k} + Q^m s \tag{A.4}$$

The hold or shaping function between sampled values of the forcing temperatures is not a triangle but a trapezoidal shape. This shape allows to adapt itself to a variable time step. Multiplying eq.(A.4) by the transfer function of the hold function, using the Z-transform and after some involved manipulations, we arrive at the different equations about the heat fluxes at each side of the multi-layered element:

$$\begin{aligned} q_0(t_n) &= a_{00}T_0(t_n) + a_{01}T_1(t_n) + D_0 \\ q_1(t_n) &= a_{10}T_0(t_n) + a_{11}T_1(t_n) + D_1 \end{aligned} \tag{A.5}$$

The eq.(A.5) is just a linear system of equations relating the new temperatures and the density of the conduction heat flux $[W/m^2]$ at both sides and at t_n. The coefficients a_{xx} and D_x depend on the time elapsed t_e from the last evolution of the states and their previous value $w_{k,n-1}^{side}, T_{side,n-1}, \quad side \in \{0,1\}, \quad k \in \{1,\cdots,K\}$. The succesive state transition computation has been re-organised as follows to adapt it to the DEVS model implementation:

$$\varphi_k = e^{-\alpha_k t_e} \tag{A.6a}$$

$$p_k = (1-\varphi_k)/\alpha_k t_e \tag{A.6b}$$

$$a_{00} = A_0^X + \frac{Q^X}{t_e} + pA^X \tag{A.7a}$$

$$a_{01} = -A_0^Y - \frac{Q^Y}{t_e} - pA^Y \tag{A.7b}$$

$$a_{10} = -a_{01} \tag{A.7c}$$

$$a_{11} = A_0^Z + \frac{Q^Z}{t_e} + pA^Z \tag{A.7d}$$

$$b_{00} = -(pA^X + \frac{Q^X}{t_e}) \tag{A.7e}$$

$$b_{01} = (pA^Y + \frac{Q^Y}{t_e}) \tag{A.7f}$$

$$b_{10} = -b_{01} \tag{A.7g}$$

$$b_{11} = -(pA^Z + \frac{Q^Z}{t_e}) \tag{A.7h}$$

$$D_0 = b_{00}T_{0,n-1} + b_{01}T_{1,n-1} + W_\varphi^0 \tag{A.7i}$$

$$D_1 = b_{10}T_{0,n-1} + b_{11}T_{1,n-1} + W_\varphi^1 \tag{A.7j}$$

with $pA^m = \sum_{k=1}^K p_k A_k^m$ and $W_\varphi^x = \sum_{k=1}^K \varphi_k w_{k,n-1}^x, \quad side \equiv x \in \{0,1\}$. Equations (A.6) and (A.7) are used in the *transition() function* call (see eq.(1)). Recall that not all the terms in eq.(A.6) are computed. It depends on how much time has elapsed t_e and the value of the poles, since φ_k goes to zero as time goes by.

Equations (A.8) are used in the evolvestates() call (see eq.(1)), and serve to update the states at both sides of the conduction element.

$$w_{k,n}^0 = \varphi_k w_{k,n-1}^0 + p_k\{(A_k^X T_{0,n} - A_k^Y T_{1,n}) - (A_k^X T_{0,n-1} - A_k^Y T_{1,n-1})\} \tag{A.8a}$$

$$w_{k,n}^1 = \varphi_k w_{k,n-1}^1 + p_k\{(A_k^Y T_{0,n} + A_k^Z T_{1,n}) - (A_k^Y T_{0,n-1} + A_k^Z T_{1,n-1})\} \tag{A.8b}$$

References

[1] [link].
URL http://www.trnsys.com/ http://sel.me.wisc.edu/trnsys/

[2] [link].
URL http://apps1.eere.energy.gov/buildings/energyplus/

[3] [link].
URL http://www.esru.strath.ac.uk/Programs/ESP-r.htm

[4] E. Kofman, Simulacion y control de sistemas continuos por eventos discretos., Ph.D. thesis, Facultad de Ciencias Exactas,Ingenieria y la Agrimensura. (2003).

[5] U. S. D. of Energy, EnergyPlus Engineering Reference (2012).
URL http://apps1.eere.energy.gov/buildings/energyplus/

[6] J. A. Clarke, N. J. Kelly, D. Tang, A review of esp-r's flexible solution approach and its application to prospective technical domain developments, Advances in Building Energy Research 1 (1) (2007) 227–247. arXiv:http://www.tandfonline.com/doi/pdf/10.1080/17512549.2007.9687277, doi:10.1080/17512549.2007.9687277.
URL http://www.tandfonline.com/doi/abs/10.1080/17512549.2007.9687277

[7] E. Kofman, Quantization-based simulation of differential algebraic equation systems, SIMULATION 79 (7) (2003) 363–376. doi:10.1177/0037549703038881.
URL http://sim.sagepub.com/cgi/content/abstract/79/7/363

[8] J. Hensen, A comparison of coupled and decoupled solutions for temperature and air flow in a building (se-99-13-04), ASHRAE Transactions part 2. 105.

[9] J. Hensen, Modelling coupled heat and air flow: Ping-pong vs onions, Tech. rep., University of Strathclyde Energy Systems Research Unit, 75 Montrose Street, GLASGOW G1 1XJ, Scotland (1995).

[10] N. J. K. J. A. Clarke, D. Tang, A review of esp-rs flexible solution approach and its application to prospective technical domain developments, Advances in Building Energy Research 1, issue 1,(special). doi:10.1080/17512549.2007.9687277.

[11] L. M. Djunaedy E., Hensen J., Towards external coupling of building energy and airflow modeling programs, ASHRAE Transactions (part 2) (SYMP-00057-2002) 109 (2003) pp. 771 – 787.

[12] M. Bartak, I. Beausoleil-Morrison, J. Clarke, J. Denev, F. Drkal, M. Lain, I. Macdonald, A. Melikov, Z. Popiolek, P. Stankov, Integrating cfd and building simulation, Building and Environment 37 (89) (2002) 865 – 871. doi:10.1016/S0360-1323(02)00045-8.
URL `http://www.sciencedirect.com/science/article/pii/S0360132302000458`

[13] G. Migoni, Simulacion por cuantificacion de sistemas stiff, Ph.D. thesis, Facultad de Ciencias Exactas, Ingenieria y Agrimensura. Universidad Nacional de Rosario (2010).

[14] S. J. Ernesto Kofman, Quantized-state systems: A devs approach for continuous system simulation, TRANSACTIONS of The Society for Modeling and Simulation International 18 (1) (2001) 2–8.
URL `www.fceia.unr.edu.ar/ kofman/files/qss.pdf`

[15] W. Toshiyuki, O. Akihito, U. Yoshimi, H. Tetsuo, R. Yuji, Practical methods for calculating unsteady-state heat conduction of multi-layer walls based on pulse transfer function, Journal of architecture, planning and environmental engineering. Transactions of AIJ (391) (1988-09-30) 8–19.
URL `http://ci.nii.ac.jp/naid/110004074588/en/`

[16] ASHRAE, Standard method of test for the evaluation of building energy analysis computer programs (bestest) (2001).

[17] F. Bergero, E. Kofman, Powerdevs. a tool for hybrid system modeling and real time simulation, Simulation: Transactions of the Society for Modeling and Simulation International 87 (1–2) (2011) 113–132.

[18] U. Yoshimi, W. Toshiyuki, An analysis of multi-layer wall heat transfer by state transition matrix : part 1 an approximate transfer functions model and its accuracy, Transactions of the Architectural Institute of Japan. (305) (1981-07-30) 97–111.
URL `http://ci.nii.ac.jp/naid/110003881844/en/`

[19] U. Yoshimi, W. Toshiyuki, An analysis of multi-layer wall heat transfer by state transition matrix : part 2 a successive calculation method and

its accuracy, Transactions of the Architectural Institute of Japan. (311) (1982-01-30) 57–66.
URL http://ci.nii.ac.jp/naid/110003881951/en/

[20] T. Hayashi, Y. Urano, T. Watanabe, Y. Ryu, Passive system simulation program pssp and its applications., in: IBPSA (Ed.), Building Simulation, International Building Performance Simulation Association, 1985, pp. 346–.
URL http://www.ibpsa.org

[21] Y. Chen, S. Wang, Frequency-domain regression method for estimating ctf models of building multilayer constructions, Applied Mathematical Modelling 25 (7) (2001) 579 – 592. doi:10.1016/S0307-904X(00)00067-6.
URL http://www.sciencedirect.com/science/article/pii/S0307904X00000676

[22] D. C. Hittle, R. Bishop, An improved root-finding procedure for use in calculating transient heat flow through multilayered slabs, International Journal of Heat and Mass Transfer 26 (11) (1983) 1685 – 1693. doi:10.1016/S0017-9310(83)80089-1.
URL http://www.sciencedirect.com/science/article/pii/S0017931083800891