International Journal of Thermal Sciences 75 (2014) 105–115

Contents lists available at ScienceDirect

International Journal of Thermal Sciences

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

Discrete event heat transfer simulation of a room

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

Departamento de Termodinamica Aplicada, ETSII, Universitat Politecnica de Valencia, C/Camino de vera s/n, 46022 Valencia, Spain

ARTICLE INFO

Article history:
Received 15 November 2012
Received in revised form
16 July 2013
Accepted 29 July 2013
Available online 2 September 2013

Keywords:
Successive state transition method
Building energy simulation
Discrete event simulation
DEVS

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.

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

* Corresponding author.
E-mail addresses: vsoto@ter.upv.es, vsotofrances@gmail.com (V.M. Soto Frances), emases@upvnet.upv.es (E.J. Sarabia Escriva), jmpinazo@ter.upv.es (J.M. Pinazo Ojer).

1290-0729/$ – see front matter © 2013 Elsevier Masson SAS. All rights reserved.
http://dx.doi.org/10.1016/j.ijthermalsci.2013.07.024

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
ΔU_{lg}	change of internal energy of H_2O from liquid to vapour [J kg^{-1}]
ε_k	emissivity of surface *k*
$\sigma_{Stefan-Boltzmann}$	Stefan–Boltzmann constant 5.67 × 10^{-8} [W m^{-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* [W m^{-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 [W m^{-2}]
$q_{rad,src,side}$	radiation heat input to *side* of any wavelength, due to known sources [W m^{-2}]
$q_{rad-lw,j}$	longwave radiation heat exchange which depends on superficial temperatures, at surface *j* [W m^{-2}]
$q_{rad-lw,side,m}$	longwave radiation heat exchange at side of conduction element *m* [W m^{-2}]
q_{rad-lw}	vector of longwave radiation heat exchanges within a zone [W m^{-2}]
T	vector of superficial temperatures of the boundary surfaces of the zone
T_0	superficial temperature at one side of a conduction element [°C]
T_1	superficial temperature at the other side of a conduction element [°C]
T_{zone}	temperature of the zone air [°C]
C_{da}	specific heat capacity of dry air [J kg^{-1} K^{-1}]
C_{H_2O}	specific heat capacity of water vapour [J kg^{-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 [°C]
W	absolute humidity [$kg_{H_2O} kg_{da}^{-1}$]

Discrete event simulation

δ_{ext} ( )	external function
δ_{int} ( )	internal transition function
λ ( )	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

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 an 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 Refs. [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 δ_{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 λ ( ) and δ_{int} ( ) functions (the first issues an output