The Methods of 2020 for Building Envelope and HVAC Systems Simulation - Will the Present Tools Survive?

Per Sahlin, PhD, member ASHRAE

EQUA Simulation Technology Group AB[1]

Summary

Computer simulation of buildings, HVAC and control systems is likely to grow in importance in the years to come. The fundamentals of the computer methods in use today were primarily developed during the seventies and early eighties. They were optimized to enable full-year whole-building simulations on available hardware. Consequently, they are highly specialized and very difficult to modify in order to meet the needs posed by new users and technologies. On the other hand, they have been proven to do the job and are well-known to developers and end-users. In this paper, we compare the traditional methods to a new generation of computer modeling tools: equation based methods. These have the capacity to model virtually any system that can be described with equations and are thus able to grow in arbitrary new directions as needs evolve. The main obstacles have up until recently been availability of development tools, difficulty of use, lower execution speed and frequent crashes. The paper discusses these and other issues involved.

Introduction

Determining the performance and economy of a technical system requires evaluation under realistic operating conditions. Evaluations conducted using computer-based simulations are often more useful than "real-life" experiments, especially during the early, critical stages of the design process. Consequently, computer-based analysis is quickly replacing physical measurements for many problem types in R&D departments throughout the world. In the building industry, computer-based simulation has gained wider acceptance only in the last few years. It seems likely that simulation will continue to grow in use and importance in the building design, operation and maintenance process in the next twenty years. A primary driving mechanism in this process is the emerging generation of computer-

---

1 EQUA Simulation Technology Group AB
Västerlånggatan 27
111 29 Stockholm, SWEDEN
Phone: +46-8-20 81 00
E-mail: per.sahlin@equa.se
Web: www.equa.se

aware young engineers. In this situation, it is natural to think about what the simulation tools of the future might look like. Will the predominant tools of today be able to evolve organically and meet future needs?

The question is not new. Already in 1985, a group of leading building simulation specialists gathered in Berkeley, California, to discuss future directions [Clarke 1985]. There was a consensus that most of the tools, that had been developed until then, were too rigid in their structure to be able to accommodate the improvements and flexibility that would be called for in the future. Each added feature to the existing tools required a larger implementation effort than the previous one. Basic methodological improvements, such as a complete change in solution strategy, were close to impossible to carry out since most of the program structure would be affected. In response to the conclusions reached, several significant projects were initiated to develop modular architectures for building simulation tools that would give the desired flexibility, maintainability, and that would be possible to upgrade in all respects.

Fifteen years have passed. The tools that were deemed inadequate at the Berkeley meeting are still dominant, although today often wrapped in modern GUIs. However, the Berkeley analysis still applies. Further confirmation of this was obtained at two international workshops that were organized jointly by the US departments of energy and defence in the summers of 1995 and 1996, again to discuss the future of the field. A summary from the workshops [Crawley et al. 1997] confirms that a majority of researchers and users still have high expectations on the performance of future methods and that the hopes are low that the present generation of tools will be able to meet the standards. Another major new building simulation development project, EnergyPlus [Crawley et al. 1999], was started after these workshops.

The difficulties with the practical implementation of good "new generation" tools have obviously been underestimated. Several projects have failed to deliver as expected. In the author's opinion, the primary cause of this is an unwillingness by building simulation developers to learn from other engineering fields. A tendency exists on the part of program developers to overestimate the uniqueness of building simulation problems, and to solve them with building-specific methods. Many other fields, such as chemical process engineering, mechatronics and power systems, have a sufficiently similar problems and often better resources.

The basic characteristics of the equation based simulation methods - our focus in this paper - have emerged in independent projects from a several domains. Only recently, in conjunction with the development of the new modeling language, Modelica, has an inventory of modeling needs from different domains been carried out (www.modelica.org). The author of this paper has a twelve-year history as developer of equation based methods and their application, primarily in building simulation, and is therefore far from an impartial observer. On the other hand, this experience has provided a large number of opportunities to examine the issues involved. This paper is an attempt to summarize this knowledge and draw some conclusions.

There are many types of challenges ahead. Key issues for any simulation development project include: tool interoperability (especially regarding CAD and product modeling tools), Internet, object-oriented programming, and graphical user interface (GUI) technology. Ability to utilize new opportunities in these areas will – along with all non-technical aspects of a successful enterprise – determine success or failure. In this attempt to speculate into the future we will, however, regard this group of factors as being equal for all and instead focus on the fundamental modeling and solution methods used.

In the subsequent two sections, the stage will be set, from a business as well as from a technical perspective. Then we will briefly present the two groups of methods that will be compared, here defined as the traditional and the equation based approaches. After that, the yardsticks by which, in the author's opinion, the respective methods should be measured are presented. The issues are briefly discussed and some conclusions for the future are drawn.

Present and Likely Future Penetration of Building Simulation Tools

Before we continue with the technical discussion, it seems appropriate to attempt to estimate the likely future importance of computer building simulation in monetary terms. After all, if simulation will continue to play a marginal role in the design process of a building, the methods used are of equally minor consequence. One can distinguish four rather separate types of economical impact of building simulation tool usage:

1. ***Improved quality and life cycle cost of produced buildings and service systems.*** This is naturally by far the most important factor. It is however also very hard to estimate correctly. Some serious attempts have been made, for example in the evaluation of the British EDAS programme [ETSU 1998] – a state subsidized scheme to foster improved building design practice. It was estimated that the U.K. government investment of £ 5.66 M gave rise to savings in energy costs worth £ 25 M per annum.
2. ***As a marketing vehicle for equipment and systems manufacturers.*** Several examples indicate that being able to illustrate the function of ones equipment in a whole-system context can be

an effective marketing method. Such tools form the bulk of commercial simulation tool usage today, the perhaps most prominent examples being the HAP[2] and TRACE programs by Carrier Corporation and Trane, respectively.

3. ***Modeling services*** for design, energy compliance etc**.**
4. ***License Sales of Computer Programs.*** This is – and is likely continue to be – the least important of the economic factors. However, estimating present and future market size for software is established practice and most required numbers are readily available from, e.g., the DOE Building Energy Software Tools Directory. Here, it is assumed that license sales varies roughly in proportion to the overall impact of all four factors and we will use it here as an indicator.

We will use Sweden as our probe and extrapolate results to a world-wide basis. Sweden has a long history of building simulation tool usage in the commercial sector and a fairly high level of general computer literacy. Our primary measure of tool penetration will be number of single user licenses per million of inhabitants (lpm) in the primary target market. For example, the Swedish manufacturer's tool TeknoSim (www.teknoterm.se) is distributed in some 300 copies in Sweden. Since free copying is allowed within a design office, this is likely to correspond to some 1000 single user licenses (a low estimate). Divided by the size of the Swedish population (~ 10 million), this gives us an approximate lpm-value of 100. Table 1 provides a summary of some similar estimates for a few well-known existing simulation tools. Note that the accuracy is limited. The author has in some cases estimated the percentage of a given total number of licenses that corresponds to a specific "home" market. All of the tools perform time dependent simulation of thermal behavior but otherwise they differ quite substantially in many ways.

---

[2] All building simulators that do not have explicit references can be found on the DOE Building Energy Software Tools Directory (www.eren.doe.gov/buildings/tools_directory).

Table 1. lpm for a selection of building simulation tools

tool	*lpm*	*"home" market*	*category*	*focus*
TeknoSim	100	Sweden	manufacturer, low cost	Single-zone indoor climate equipment sizing
IDA ICE	22	Sweden	commercial, high cost	Multi-zone indoor climate and energy
TSBI3	30	Denmark	semi-commercial, high cost	Multi-zone indoor climate and energy
TAS	5	U.K.	commercial, high cost	Multi-zone indoor climate and energy
ESP-r	1	U.K.	academic, free	Multi-zone indoor climate and energy
HAP	17	U.S.A	manufacturer, medium cost	Multi-zone energy and equipment sizing
DOE-2	5	U.S.A.	semi-commercial, low cost	Multi-zone energy and equipment sizing
Energy10	4	U.S.A.	semi-commercial, low cost	Few-zone energy, mainly architectural use
TRNSYS	4	U.S.A.	semi-commercial, high cost	Multi-zone indoor climate and energy

In a country like Sweden, the approximate user potential is 2000 HVAC consultants, 1500 architects and perhaps 1000 others (contractors, manufacturers, energy advisers, academics), in total some 4500 potential licenses. If a single tool manages to penetrate into every user's computer, this tool would consequently reach an lpm value of 450, a number we can assume to be a theoretical maximum. This max can be assumed to remain rather constant also in 2020, since the number of potential users is likely to change only marginally until then.

In Sweden, TeknoSim has consequently reached about a quarter of its full potential since its introduction in 1994. Let us make the assumption that by 2007, there will be a leading tool on each market with a similar penetration in all of the "technologically advanced" world of say a billion inhabitants and furthermore that by the year 2020 some tool on each market will have reached three quarters penetration.

If a total yearly revenue of 1000 Euro (or equivalently $) from each user is assumed, the total world market for the leading tools will amount to 100 MEuro in 2007 (100 [lpm] * 1000 [tot. pop] * 1000 [revenue/license]) and to 300-400 MEuro in 2020. This market is significant enough to motivate a few substantial companies that focus on building simulation. Today, perhaps a total of a few hundred commercial professional software developers are active in this field world-wide. Due to market fragmentation, they are to a very large extent isolated from each other. If the given market estimates are in the right range, this number will increase by almost an order of magnitude by the year 2007 and international companies may start to appear.

Technological Motivation and Mathematical Problem Complexity

A building is a complex system with a multitude of simultaneously interacting physical processes. Today, most significant systems within the building are being modeled in some context.

Manufacturers develop sophisticated simulation models of their own equipment for in-house product development. A visit to any of a range of academic conferences within building physics, indoor air quality (IAQ), HVAC, refrigeration, building simulation etc. will also reveal an enormous effort to model reality in mathematical terms at different levels of sophistication and integration. There is ample evidence of a plethora of physical systems within a building that meaningfully lend themselves to modeling. This is an interesting observation in itself, but it is hard to predict the consequences of all this modeling activity in more concrete terms. If one instead tries to gain overview over all computer tools that attempt to package a simulation model in a useful form for some end-user, a similarly overwhelming complexity emerges (a sample of some two hundred energy related tools can be studied at www.eren.doe.gov/buildings/tools_directory).

The focus here will be on thermal, whole-building, deterministic simulation models, since these form the most significant niche of building simulators. The primary purpose of these tools is to predict indoor thermal climate, energy consumption and life cycle cost. Some of the phenomena that must be modeled to some extent in order to form a meaningful simulation model in this class are:

- Outdoor climate conditions, including temperature and incident solar radiation
- Dynamic heat flux through the building envelope and internal structure
- Heat balance of each room (or ventilated zone)
- Air, water and heat flow through the primary and secondary HVAC system

Many tools in this category also model some of the following closely coupled quantities:

- Moisture transport within the building
- Cost of supplied energy
- Thermal sensation (comfort) of building occupants
- Daylight
- Naturally occurring air-flows between and within zones
- Indoor Air Quality
- Performance of ground-coupled systems

Three typical purposes (and corresponding time-scales) of thermal simulations are:

1. Prediction of extreme conditions (design day)
2. Prediction of the energy consumption (per annum)

3. Prediction of controller action (time-scales down to seconds)

As can be expected, a model that is capable of studying all of the mentioned processes and time-scales becomes quite complex. Detailed models may solve about a thousand of simultaneous equations per zone in the building (unless the air-flow pattern within each room is modeled, in which case this number can be several orders of magnitude larger). Highly simplified models may on the other hand use only half a dozen equations per zone.

Presently, the most complex models are mainly used in academia, since they require substantial time and expertise to use. ESP-r is an example of such a tool. A key observation, however, is that there is also a growing demand for state-of-the-art modeling in the commercial sector. The DOE-DOD study previously mentioned indicates this [Crawley et al. 1995]. The specification of the detailed model in IDA ICE was developed by commercial actors [Björsell et al. 1999]. The EnergyPlus project [Crawley et al. 1999] in the U.S. approaches more detailed modeling. A new version of the widely used Hot 2000 software is likely to be based on the ESP-r model [Haltrecht 1999].

The need for simplified models is primarily driven by two mechanisms: (1) shorter execution time and (2) less need for input data. Acceptable execution times are governed by the user's tolerance to wait for results. In normal applications (i.e. not when a very large number of repeated simulations are required) the acceptable waiting time has some correlation to the time it takes to describe input data and analyze results. It is the total cycle time of a simulation experiment that is of primary interest. It is usually not worthwhile to decrease the simulation time from ten to five seconds in a case where it takes several minutes to describe input and understand results. The time it takes to collect and enter input data is usually a more serious bottleneck than execution time. On the other hand, for most complex models, it is possible to define wizards which provide "intelligent" defaults and thereby reduce the time of entering data. A bound on meaningful model simplification is therefore given by the time it takes to enter input data into and analyze results from the most radically simplified but still useful wizard. With modern solution methods and hardware, numerous equations can be solved quickly and this eliminates the need for the simplest models. Today, in the author's opinion, it is difficult to motivate models of less than, say, a hundred equations per zone. In the future discussion, we will therefore focus on models of considerable complexity.

Two Different Approaches to Solving Complex Models

Let us define two families of approaches to building, managing and solving complex thermal simulation models for buildings: the traditional and the equation based methods. The traditional methods are the combined result of countless numbers of building engineering man-years in the development of successively more complex simulators. Equation based methods, on the other hand, have been developed to deal with simulation problems from several different domains. Only recently have they been applied to full-complexity building simulation models. Before we go on with the further analysis, we will characterize the two method families a bit further.

The Traditional Approach

A different name for this family might be Interacting Tailored Solvers. A thermal building simulation model can be broken down into a number of separable sub-problems that each have some specific characteristics that can be taken advantage of. Traditionally, utilizing *all* such special characteristics has been necessary to enable whole-year and whole-building simulations. In this approach, each sub-problem is solved separately in sequence with a tailored method, then some means of synchronizing the different results is applied to arrive at a satisfactorily accurate solution of the combined problem. Many different sub-problem partitions are possible and the following account has no ambition of being exhaustive:

- If (close to) constant room air temperature and moisture level is assumed, the solution of the HVAC system can be treated separately from the zone model. This de-coupling is done in, e.g., BLAST and DOE-2.
- Envelope heat transfer. The transfer function and response factor methods [ASHRAE 1997] are well-known ways of efficiently solving this sub-problem, used in, e.g., BLAST and DOE2. ESP-r uses a different tailored approach to this problem.
- Multi-zone air flow. The non-linear algebraic equations for this problem have a very special structure that lends itself to excellent tailored methods. ESP-r and TRNSYS use two of these.

In recent development projects, more general, but still specialized, classes of solvers are used for sub-problems of certain types:

- Linear state-space models. Modal reduction methods can provide tremendous efficiency on linear sub-problems that must be solved repeatedly. m2m [Lefebvre 1999] is an example of a tool where these are heavily relied on.
- Any sub-problem with a positive-definite system matrix. The conjugate gradient method is applicable to any problem of this nature. Many such problems occur in building simulation. The multi-zone air exchange problem is an example. The BUS++ environment is based on a powerful conjugate gradient solver.

The principal architecture of a program in the traditional approach is depicted in Figure 1. Large variations exist. The most important characteristic is that the Common Framework controls the execution and synchronization of the sub-problem solvers in some problem-dependent manner.

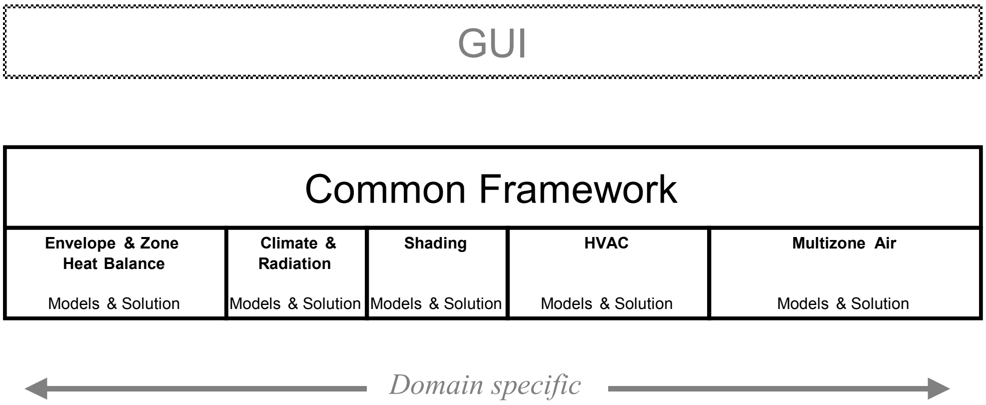

Figure 1. Typical architecture of a simulator in the traditional approach

Naturally, the definition of the traditional approach becomes rather vague since many different divisions into separable sub-problems are possible. The cleverness of the sub-problem partitioning and the efficiency of the synchronization method used will vary significantly between different projects. In spite of this spread, we hope to be able to extract some useful conclusions on the basis of this categorization. ESP-r and EnergyPlus can be regarded as two leading projects in this tradition.

Equation based methods

If all the relationships that govern a thermal building simulation problem are written down as equations (partial differential equations are first discretized in space,) one arrives at a so called hybrid DAE problem. “DAE” stands for differential-algebraic equations, i.e. a mixture of ordinary differential and algebraic equations. “Hybrid” denotes the fact that the system may contain discontinuities and hysteresis. The resulting system of equations will contain many different time-scales and have several non-linear equations. Corresponding system matrices will have no special properties, such as being symmetric, diagonally dominant or positive definite. In the equation based method, this complex system of equations is formulated and solved using general-purpose methods, i.e. methods that have no strong relationship to any engineering field or more specific problem category. Hybrid DAE:s occur naturally in many domains such as chemical process engineering, analog electronics, mechatronics, multi-body mechanics, power generation and distribution, electrical power and pipe flow networks to name a few. Some commercial software systems that deal with this class of problems are: ALLAN [Jeandel et al. 1997], Dymola [Elmqvist et al. 1996], ESACAP [Stangerup 1991], gPROMS [Oh and Pantelides 1996], IDA [Bris Data 1999] and Saber. In addition to these, there is a large number of academic systems: ASCEND [Piela et al. 1991], OmSim [Andersson 1994], MOSES (www.elet.polimi.it/section/automeng/control/oo), SMILE (www.first.gmd.de/smile/smile0.html) [Jochum 1993] and SPARK [Buhl et al. 1993] to name a few.

The DAE problems are formulated using special *modeling languages*. Several such languages have been developed, the first being Dymola [Elmqvist 1978] which was defined in 1979. NMF (the Neutral Model Format) was proposed as a possible standard for building simulation models in 1989 (http://home.swipnet.se/nmf) [Sahlin and Sowell 1998]. The first formal standard is VHDL-AMS (www.vhdl.org/analog) [IEEE, 1997], an extension of the VHDL language for logical circuitry simulation. Modelica (www.modelica.org) was proposed in 1997 as the first completely domain-independent standard.

We will not digress here on the issue of DAE modeling language design - it is a research area of its own - but merely point out a few main characteristics about this way of expressing models. The two main advantages of using such a language are:

1. The modeler can forget entirely about the way equations are solved, he/she is merely required to *formulate* the model, not to *solve* it. This enables domain experts to concentrate on their specialty and not have double duty as numerical analysts.
2. The resulting model can be automatically transformed in many ways. Most importantly, it can be translated into the format required by more than a single simulation environment. Most often, generated code is in the form of directly executable C or Fortran. But it is also possible to translate into a different DAE modeling language and continue to work with the code as source in the new language. This means that available model material is not tied to a single language as long as the required translators are available.

Translation from a DAE language to algorithmic, executable code can thus be done. Indeed, this is the main feature of the DAE based systems. However, automatic translation of existing C or Fortran into a DAE language is not possible. Nevertheless, it is often possible to re-use algorithmic code in the DAE context by calling existing routines from the DAE model. Figure 2 shows existing and planned translators around NMF, Dymola and Modelica.

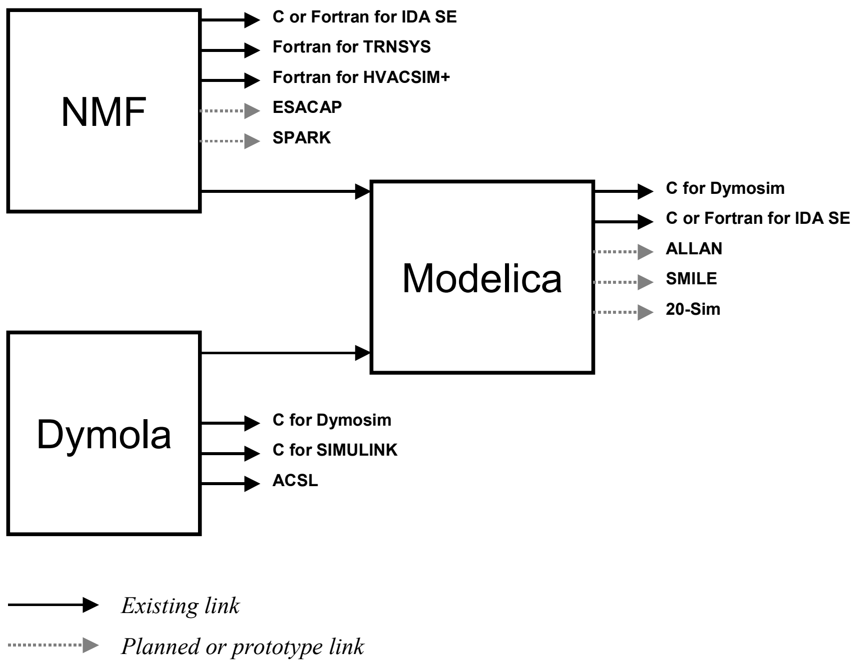

Figure 2. Translators around NMF, Dymola and Modelica.

The architecture of an equation based building simulator is divided into domain-dependent and domain specific modules as depicted in Figure 3.

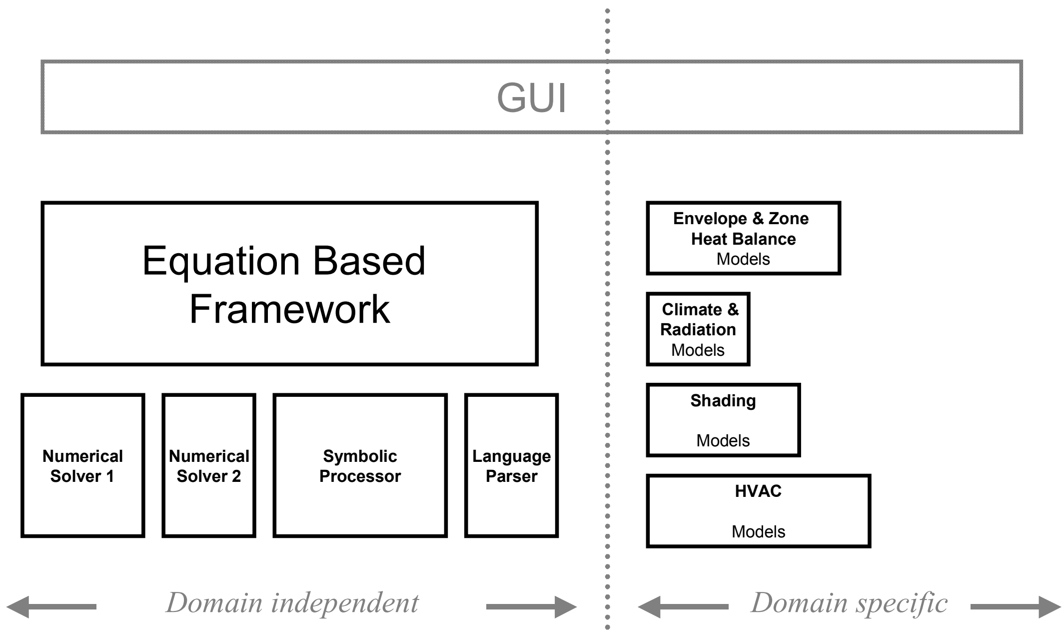

Figure 3. Basic architecture of an equation-based building simulator

To the best of the authors knowledge, there are only two examples of full-complexity equation-based thermal building models. The first, CLIM 2000, has been developed at Electricité de France and is only internally used [Bonneau 1993]. The other, IDA Indoor Climate and Energy (ICE), has been developed by the author and his colleagues. The mathematical models of ICE have been developed in the context of IEA Task 22 [Bring et al. 1999] and are available as NMF source code at http://home.swipnet.se/nmf.

Pros and Cons

Most of the discussion about traditional vs. equation based building simulation revolves around the credibility of the latter. Soundly skeptical engineers are reluctant to embark into new uncharted territory. In the remainder of the paper an attempt will be made to identify and briefly discuss the main issues involved.

The End-User Perspective

Robustness

It is absolutely imperative that an end-user tool is completely robust. Crashes for models with reasonable input data must not occur. It is also unacceptable to require any selection of solver tuning parameters or model initial guesses from the user in order to simulate a model.

The successful experience with IDA ICE shows that this is indeed possible [Björsell 1999]. With proper solution methods – especially for non-linear algebraic equations – and well-stated mathematical models, the robustness of equation based models may be as good as that of most traditional implementations.

Execution Speed

This is the issue that attracts the most attention and skepticism from developers in the traditional approach. Is it really possible to attain sufficiently high execution speed on large problems? The main problem here is to create fair testing conditions for the two approaches. Comparable phenomena must be modeled with the same fidelity. Both solutions must have the same accuracy.

Again, the only results available to the author stem from the development of IDA. The first results were not so encouraging, when IDA was applied to pure multizone air-exchange problems in 1993. The tailored solvers proved to be significantly faster [Sahlin and Bring 1993] and have more favorable scaling properties on large problems. Since then, the IDA methods have been significantly improved and, although the comparative tests have not been repeated to verify this, it is likely that the tailored methods still outperform the general by significant factors while the poor scaling properties of the latter have been largely remedied.

More relevant is of course comparisons with respect to standard thermal simulation problems. Again, the IDA experience in 1995 was that a single zone high-fidelity model executed 2-4 times faster in a traditional implementation [Sahlin 1996]. This value may vary significantly in both directions depending on quality of implementation. Regarding scalability, recent IDA development has provided methods where execution time increase close to linearly with problem size for normally connected thermal building models. This is very near the ideal asymptotic behavior (completely linear) and no further improvement can be expected.

An example of execution speed from IDA ICE may be of interest. A ten-zone model of totally 2297 equations with a normal amount of discontinuities takes twelve minutes for a yearly simulation on a 450 MHz PC. The time resolution of the results is then on the order of a few minutes. Independent bench-marking between traditional and equation based implementations is clearly necessary to gain more insight into the speed issue.

Transparency

An important qualitative property of a simulation tool is the ease of understanding the structure and detail of the underlying mathematical model. In traditional, monolithic tools, the mathematical model is generally presented separately in the documentation. Quite often, essential bits of information have been left out, i.e., it is not possible to recreate a program with the same behavior based on the given documentation.

The equation based methods are clearly in advantage here. It is possible for every user to “zoom in” and investigate the model in its smallest detail, i.e. the individual equations. Normally, one can also plot any variable in the model during a simulation, i.e. not just the ones that have been selected for plotting by the developers.

In commercial applications, it is often desirable to hide certain aspects of a simulation model that contain critical company knowledge. Some equation based approaches are built to enable this; others require the full model in symbolic form.

The Developers Perspective

Availability of Suitable Equation Based Tools

In order for the equation based methods to be of interest at all, supported appropriate tools must be available. For several years, this condition has been fulfilled, but barely. There is still a shortage of commercially and technically stable systems at affordable prices.

For building simulation applications, it is also necessary to have access to separable runtime systems. A runtime system is the part of the development system that must accompany a shipment of an end-user application. Unfortunately, most existing equation based environments do not distinguish between developer's and runtime versions at all. This makes it economically unfeasible to ship end-user applications based on these, since each shipped license also must carry the full cost of a development environment license.

Dependence on External Tools

Any software developer strives to use only established, standard development tools. The equation based environments have not yet reached full maturity and may therefore be unattractive to many from this perspective. This is an issue that will be remedied automatically with time, as successful applications of equation based tools become abundant.

The equation based tools actually, at the same time, provide significantly better investment protection than home-grown solutions. Since the equation models (the capital) are stored in a processable format, they can at any time be converted into a new form, for which tools with the desirable properties exist.

Ease of first implementation

The time it takes to build and debug the first version of a model is obviously a key parameter. The less code one has to write and verify the better. This is a strong point with the equation based approach, since only the model equations and no intertwined solution algorithms have to be developed by the application experts. Furthermore, the "common framework" is given by the development environment used.

The first implementation of a thermal building simulator in IDA (in 1995) was actually a re-implementation of a traditional model and this provided some opportunities for comparison also from this perspective. Table 2 lists estimated development times for the two approaches from [Sahlin 1996]:

Table 2. Main phases in the development of a special purpose tool

phase	special purpose time in weeks	IDA time in weeks	comment
design program architecture	4	2	NMF model architecture in the IDA case.
design and implement a suitable numerical method	8	0	Method is already available in the IDA case
implement math models as formal code	4	4	More information must be input in the NMF case, but the structure is given. In the special purpose case, the model code interacts with the solution procedure.
write I/O interfaces	12	0	I/O already available in IDA case
testing and tuning of full implementation	10	4	Less new code to be tested in IDA case. IDA tuning includes selection of suitable methods.
documentation of implemented models	4	4	Written account of model equations.
development of user's manual	4	1	IDA Solver file structure is already documented.
total time in man-weeks	**46**	**15**	

Re-Use of Existing Code

The tedious task of converting a possibly large amount of already existing algorithmic code to equation form may seem stifling. Many people have the misconception that it must be done all at once. Granted, the more of the code that is accessible to symbolic processing and optimization the better. However, modern DAE modeling languages such as NMF and Modelica allow calling existing external subroutines and functions. This way significant portions of an existing system may be re-used as is.

The two approaches are fairly comparable in this respect, perhaps with a slight advantage for the traditional approach.

Coping with Change – User Needs and Building Technology

Users and modeling needs evolve. New building technologies are also continuously introduced. It is necessary for a successful approach to follow this moving target closely.

At least two types of modeling flexibility are needed. First, one must be able to easily modify the internal structure of a component model without upsetting the solution strategy. Similarly, one must be able to model new global phenomena, such as adding CO_2 equations in all air-flow paths of the model, without invalidating everything else.

Strict separation of model from solution method is necessary to obtain good properties in this respect. Mixing them together is the very definition of the traditional approach.

Coping with Change – Platform and New Methods

The scenery of computing is constantly and rapidly evolving. In the eighties, mainframe implementations were ported to workstations. In the early nineties, everything should be wrapped in a Windows GUI. Today, Internet has a profound impact on the methods used, also for so-called number crunching activities, since the Internet once again opens the possibility to have central, powerful servers. These servers will soon be parallel and this will necessitate yet another major conversion of methods used. Keeping up with and taking full advantage of this development requires significant resources. Again, the larger base of applicability for the equation based methods provide a relative advantage, since the needed development cost can be shared between several domains.

The very structure of traditional implementations is also a problem here. Since models and solution algorithms are inextricably bound together, it is usually very difficult, also for the original developers, to improve and change the structure of the numerical methods used.

Maintaining Customized Versions

Few natural boundaries exist for the functionality of a building simulation application. The most tangible evidence of this is the enormous number of existing tools. When the dimensions of user sophistication, modeled processes, simulation purpose, nationality, organization, and computer platform are all multiplied together, one obtains this combinatorial explosion. Successfully dealing with this manifold is perhaps the most challenging task of all for a building simulation methodology.

Stable and validated versions of models must be maintained, yet it must also be possible to modify these models for almost every simulation project. Skilled users must be able to modify the model in fundamental ways themselves, yet designers with no simulation experience must be able to make meaningful runs without too many questions asked.

Excellent methods for systematizing and organizing models is perhaps the most important feature of the equation based approach. In this area the traditional methods are completely at loss.

Benefiting from Independent Development

Many successful engineering systems capitalize on creative users. AutoCAD, Matlab, and LabView are just a few examples of well-known systems that allow independent application development. The concept of tool boxes or separable model libraries makes it easier to apply the system to ones own problem without having to start from scratch each time.

For simulation programs, again, separating the models from the solution methods is the enabling factor. The main motivation for languages such as NMF and Modelica is to be able to build large, distributed but yet compatible model libraries.

Conclusions

Aside from the issue of attainable execution speed, developers from all camps largely agree that the equation based approach has superior properties in key respects. Even the developers of a completely new system such as EnergyPlus present their effort as a temporary solution, while waiting for suitable "new generation" tools to be ready (http://gundog.lbl.gov/dirsoft/eplusmerge.html). It seems to be more a question of "when" rather than "if" the equation based methods will break through. One may ask then why still an absolute majority of all work presented is in the old tradition and why the number of papers dealing with development or application of equation based methods actually seems to be dropping?

We have a classical situation of a technically inferior group of methods with a tremendous momentum against a poorly understood, rather shaky but promising new technology. Ideally, it would be just a question of time until the new technology grows strong enough to attract not only a small group of bold "early adopters" but larger numbers of developers. However, if everyone continues with the mainstream, the growth process will be so slow that it might actually die altogether.

Fortunately, exactly the same battle is fought on other arenas but with considerably more vigor. Equation based techniques are rapidly gaining a foothold in several areas including simulation of chemical process plants, multibody systems, mechatronics and power systems. This process is likely to eventually have some impact also on building simulation.

Execution speed is a critical issue for the success of equation based methods in most fields and consequently significant development efforts are devoted to improvements in this respect. The various possibilities on sequential hardware have not nearly been exhausted yet. No building simulation benchmarking exercises have yet been carried out between the DAE tools. With the development of

Modelica it is now for the first time practically possible to move DAE models between environments and thereby carry out comparative testing of both models and environments. This should sharpen the competition to the advantage of the stronger tools and thereby speed up the evolutionary process further.

In this paper, we have focused exclusively on the fundamental modeling and solution methods. However, there are synergetic effects to be benefited from also in terms of GUI development. Several aspects of a simulation system are common to a range of application domains, e.g., treatment and presentation of time series, graphical editors, data base objects with numerical parameters, integration with optimization tools etc. This is another way in which development can be made more efficient by cost-sharing over multiple domains.

In conclusion, it seems likely that the field of building simulation will have grown to some economical importance by the year 2020. The equation based methods have superior properties in almost all key technical aspects. Some argue that the traditional methods will continue to be faster and that this will motivate their continued prosperity. There is a lack of sufficient data to draw any firm conclusions regarding execution speed, but the author believes that once available, this data will prove that the equation based methods are competitive already also in this respect. Equation based methods are likely to gain momentum primarily by becoming successful in other fields. This will then eventually lead to enhanced interest also in the building simulation community.

References

Andersson M.: *Object-Oriented Modeling and Simulation of Hybrid System*s. PhD thesis ISRN LUTFD2/TFRT--1043--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, December 1994

ASHRAE 1997, Handbook of Fundamentals, ASHRAE, Atlanta, 1997

Björsell, N., A. Bring, L. Eriksson, P. Grozman, M. Lindgren, P. Sahlin, A. Shapovalov, and M. Vuolle, *IDA Indoor Climate and Energy*, Proc. of the IBPSA Building Simulation '99 conference, Kyoto, Japan, Sept. 1999

Bonneau, D., F.X. Rongere, D. Covalet, B. Gautier, 1993, *CLIM 2000 - Modular Software for Energy Simulation in Buildings*, Conference proc. Building Simulation '93, IBPSA, Adelaide, Australia, Aug. 1993

Bris Data AB: *IDA Simulation Environment – User's Manual*. Bris Data AB, Västerlånggatan 27, Stockholm, Sweden, 1999

Buhl, W.F., E. Erdem, F.C. Winkelmann, E.F. Sowell, 1993*, Recent Improvements in SPARK: Strong Component Decomposition, Multivalued Objects, and Graphical Interface*, Conference proc. Building Simulation '93, IBPSA, Adelaide, Australia, Aug. 1993

Clarke, J.A., E.F. Sowell, the Simulation Research Group, 1985, *A Proposal to Develop a Kernel System for the Next Generation of Building Energy Simulation Software*, Lawrence Berkeley Laboratory, Berkeley, CA, Nov. 4, 1985

Crawley, D.B., L.K. Lawrie, F.C. Winkelmann, W.F. Buhl, A.E. Erdem, C.O. Pedersen, R.J. Liesen and D.E. Fisher, *What Next for Building Energy Simulation – A Glimpse of the Future*, Conference proc. Building Simulation '97, IBPSA, Prague, Czech Republic, Sept. 1997

Crawley, D. B., C.O. Pedersen, R.J. Liesen, D.E. Fisher, R.K. Strand, R.D. Taylor, L.K. Lawrie, F.C. Winkelmann, W. F. Buhl, A.E. Erdem, and Y.J. Huang, *EnergyPlus, a New-Generation Building Energy Simulation Program,* Proc. of the IBPSA Building Simulation '99 conference, Kyoto, Japan, Sept. 1999

Elmqvist H.: *A Structured Model Language for Large Continuous System*s. PhD thesis, ISRN LUTFD2/TFRT--1015--SE, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, May 1978

Elmqvist H., D. Brück, and **M. Otte**r: *Dymola --- User's Manua*l. Dynasim AB, Research Park Ideon, Lund, Sweden, 1996.

D. Haltrecht,R. Zmeureanu and I. Beausoleil-Morrison, Defining the Methodology for the Next-Generation Hot 2000 Simulator, Proc. of the IBPSA Building Simulation '99 conference, Kyoto, Japan, Sept. 1999

IEEE: "Standard VHDL Analog and Mixed-Signal Extensions". Technical Report IEEE 1076.1, IEEE, March 1997

ETSU (1998) 'Final Report on the Operation of the Energy Design Advice Scheme, 1992-7', Prepared for the Energy Technology Support Unit by Eclipse Research Consultants The Eden Centre, 47 City Road, Cambridge CB1 1DP

Jeandel A., F. Boudaud., and E. Larivière: "ALLAN.Simuation release 3.1 description"M.DéGIMA.GSA1887. GAZ DE FRANCE, DR, Saint Denis La plaine, FRANCE, 1997

Jochum, P., M. Kloas, 1993, *The Dynamic Simulation Environment Smile*, Institute of Energy Engineering, TU-Berlin, 1993

Lefebvre 1999, *M2m&ROOFSOL: A Tool for Evaluating the Cooling Performance of Passive Roof Components using the Model Synthesis Method***,** Proc. of the IBPSA Building Simulation '99 conference, Kyoto, Japan, Sept. 1999

Oh M., and C.C. Pantelides: "A modelling and simulation language for combined lumped anddistributed parameter systems". *Computers and Chemical Engineerin*g, **2**0, pp. 611--633, 1996

Piela P.C., T.G. Epperly, K.M. Westerberg, and A.W. Westerberg: "ASCEND: An object-oriented computer environment for modeling and analysis: the modeling language". *Computers and Chemical Engineerin*g, **15:**1, pp. 53--72, 1991

Sahlin, P., *Modelling and Simulation Methods for Modular Continuous Systems in Buildings*, Doctoral Dissertation, Building Services Engineering, Bulletin No 39, KTH, Stockholm, May 1996

Sahlin, P., A. Bring, *Applying IDA to Airflow Problems in Buildings*, ITM report 1993:3, the Swedish Institute of Applied Mathematics, Gothenburg, Sweden, Sept. 1993

Bring, A., P. Sahlin and M. Vuolle, *Models for Building Indoor Climate and Energy Simulation*, Report of IEA SHC Task 22 Building Energy Analysis Tools Subtask B Model Documentation, KTH, Stockholm, Sweden, 1999 (available at http://home.swipnet.se/nmf/iea_bice/T22Brep.pdf)

Sahlin, P., E.F.Sowell, A Neutral Format for Building Simulation Models, Proceedings of the IBPSA Building Simulation '89 conference, Vancouver, Canada, 1989

Stangerup, P., 1991, *ESACAP: A Simulation Program and Description Language for Interdisciplinary Problems,* proc, 1991 European Simulation Multiconference, Copenhagen, Denmark, June, 1991