Building Simulation

Seventh International IBPSA Conference
Rio de Janeiro, Brazil
August 13-15, 2001

BUILDING SIMULATION TRENDS GOING INTO THE NEW MILLENIUM

Godfried Augenbroe
Georgia Institute of Technology, College of Architecture, Doctoral Program
Atlanta, U.S.A.

ABSTRACT

Recent decades have witnessed the proliferation of simulation software for a broadening range of building performance assessments. The next challenge is now to mature building simulation into a recognized and indispensable discipline (with a unique expertise and toolset) of all professions, involved in the design, engineering, operation and management of buildings. The two key targets to achieve this are an increased level of quality assurance and better integration of simulation expertise and tools in the overall building process. The latter target must address all stages of a project, i.e. from early conceptual design to commissioning, through to operation and, finally deconstruction of the facility. Meanwhile, the 'appearance' and use of simulation is changing rapidly, mostly as a result of the Internet revolution. This is exemplified by new forms of ubiquitous, remote, collaborative and pervasive simulation. These and other trends and manifestations are discussed in the light of the challenges that the building simulation discipline faces at the start of the new millennium.

KEYWORDS

Building simulation, design integration, web hosted services, e-Simulation

1. INTRODUCTION

The role of simulation tools in the design and engineering of buildings has been firmly established over the last two decades. Simulation is credited with speeding up the design process, increasing efficiency, and enabling the comparison of a broader range of design variants, leading to more optimal designs. Simulation provides a better understanding of the consequences of design decisions, which increases the effectiveness of the engineering design process. With the groundwork done in the 60s and 70s, a rapid proliferation of advanced simulation packages for many aspects of building performance has taken place over the last twenty years. The challenge of the next decade is to better integrate simulation in the design process as a whole, increase quality control and exploit the explosion of opportunities that the Internet offers.

The ubiquitous and 'instant' accessibility of domain experts and their specialized analysis tools through the Internet has de-emphasized the need to import 'designer friendly' tools into the nucleus of the design team. Instead of migrating tools to the center of the team, the opposite migration may now become the dominant trend, i.e. delegating a growing number of analysis tasks to (remote) domain experts. The latter trend recognizes that the irreplaceable knowledge of domain experts and their advanced tool sets is very hard to match by 'in-house' use of their designer friendly variants. With this recognition, sustaining complete, coherent and expressive communications between remote simulation experts and other design team members has surfaced as the real challenge. This is strongly related to the changing team context of simulation, which is discussed first in sections 2 and 3. Although tool interoperability is an important underpinning technology, it is by no means a complete solution for efficient communication between the design team and domain experts. This observation is discussed in section 4, and is elucidated by a recent research initiative that deals with these issues in a novel fashion. Section 5 revisits the 'old' wish list of simulation tools, only to find that in spite of tremendous progress, some important wishes have remained unfulfilled.

It is important to realize that design analysis does not exist in isolation. The whole *analysis process,* from initial design analysis request to model preparation, simulation deployment and interpretation needs to be managed in the context of a pending design or maintenance decision. As the primary objective of a design analysis is to better inform and rationalize design decisions, the causality between design request and analysis has to be made explicit. This entails that associations between certain design considerations and a building simulation must be maintained and enforced in the process and across all members of the design engineering team. A new category of web-enabled groupware is emerging for that purpose. This development may have a big impact on the simulation profession once the opportunities to embed simulation facilities in this type of groupware

are fully recognized. There are early indications that the web revolution and explosion of e-commerce opportunities have ignited a spark in the AEC industry. Many 'AEC.com' startups have begun to offer web hosted collaboration services to project teams (Doherty, 2001). These services offer information exchange and group management support through dedicated subscriber web sites. The opportunities of new forms of Internet enabled simulation such as distributed ('outsourced') simulation and commercial web hosted simulation services, also known as 'e-Simulation' are some of the new forms that are discussed in section 6.

2. THE CHANGING TEAM CONTEXT OF SIMULATION

Building simulation tools have become an integral part of the ensemble of computer applications for the design, engineering and (to some extent) operation of buildings. The primary objective of their use is to conduct a performance analysis that informs for instance a pending design decision, a dimension parameter choice or a budget allocation for maintenance. How effectively this is done, is as much dependent on the quality of the tools and technical skills of the consultant as it is on other factors. Among these other factors, the management and enforcement of the causality between certain design considerations and a requested analysis are crucial. If the interaction between design tasks and engineering analysis is incidental and unstructured the potential contribution of building simulation to achieve better buildings will not be reached. Better tuning of the coupling between design intentions and simulation deployment is needed therefore. A new category of *simulation environments* will emerge for precisely that purpose. The tools embedded in these environments focus on data integration and simulation interoperability but above all on rapid and timely invocation of the most adequate simulation function (rather than simulation tool) in a given design context. The two major areas of improvement for the building simulation profession can be identified as (1) *tool-related,* targeting advancements of tool functionality and (2) *process-related*, targeting functional integration of simulation tools in the design process. Close-up observations about the developments of these two fields will be made in later sections.

Designer friendly versus design-integrated tools

The development of 'simplified' simulation tools for architectural designers has received a lot of attention from the research community in the past, but seems to be fading lately. Many researchers still hold the belief that simulation tasks should be progressively moving towards the non-specialist, in this case the architectural designer. We argue against this and find that attempts to provide *designer-friendly* tools have been overcome by recent events, such as the World Wide Web and the continued increase in computing power. The ubiquitous and 'instant' accessibility of project team partners and their advanced tools creates a stimulus to involve as many experts as desired in a design decision. These experts are expected to use the best tools of the trade and infuse their irreplaceable expertise in the communication of analysis results with other design team members. There seems to be no apparent reason to try to des-intermediate the domain expert. Indeed, in an era where the Internet stimulates delegation and specialization of remotely offered services, such des-intermediation would be counter-productive.

Figure 1The Sony Center

In every project there are distinct stages that call for different types of assessments to assist design evolution. Early stage assessments are mostly based on expertise and experiential knowledge of consultants. The current generation of simulation tools plays no significant role in this stage. If computer tools are to have an impact in this stage they will probably have to be less based on simulation and more on expert knowledge representations. AI based approaches have attempted this in the past but have not made a lasting impact, which is not surprising if one realizes that the need for early expert intervention is greatest if the problem is complex and novel. But this is exactly

where knowledge based tools are traditionally weak. A close look at one of the most publicized projects of recent years, the Sony center (Fig. 1), may elucidate this. It is intuitively clear that this project poses extreme challenges. It has very large internal spaces, three different levels of microclimate, a great variety of functions and occupants, different climate control regimes and more. A project of this complexity cannot be realized without involving experts at the very early stage of the design process. No architectural firm would in this case take the risk of relying on designer friendly analysis tools, because it will take a high degree of expertise to judiciously apply simplified analyses to this case. Although this design case is an extreme example, it raises the more common issue of accountability for the choice of a particular analysis tool in a design problem. Modern building project partnering strategies deal with accountability as an integral part of team building and management. Accountability of performance analyses should be treated from the same team perspective. Designer friendly analysis tools have typically ignored this issue by assuming that the non-expert designer will take responsibility for use of the tool. The problem with this is that to the designer, the tool is essentially a 'black box', which does not make any of its applicability limitations explicit. The above assumption seems therefore not justifiable.

Figure 2 reflects how designer friendly tools are generated by 'reduction' or 'boiling down' of expert domain knowledge and expert tool functionality. The premise is that this leads to the type of tool that can be absorbed by the non-expert users in the inner design team. Tool developers typically use expert judgment in the reduction process (Fig. 2).

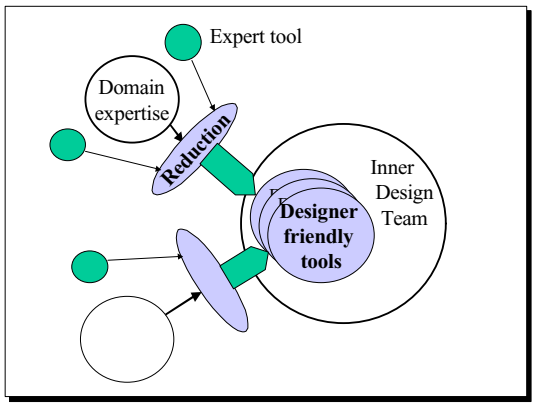

Figure 2 Reduction of domain knowledge and migration of designer friendly tools

The once popular research area seems to have been replaced by the opposite strategy, which is to delegate ('outsource') design analysis to domain experts and their increasingly complex expert tools. The latter effort concentrates on an efficient communication layer that supports the delegation of tasks and interpretation of results. Figure 3 shows four distinct versions of this approach that are discussed below. Whereas designer friendly tools emphasize the import of 'packaged' domain expertise into the design team, *design integrated* tools emphasize the export of formalized analysis requests along with an explicit design context. Equally important is the import of analysis results in the form that supports rational decision-making. The basic distinction between designer-friendly tools (Fig. 2) and design-integrated tools (Fig. 3) is the reduction and encapsulation of domain knowledge in the first case versus enrichment and externalization of design context in the second. This has repercussions for the design team. Instead of a tool user, the inner design team needs to become a central design manager, maintain a central design repository and act as a coordinating agent for domain experts. Variants A, B, C and D of Fig. 3 show different versions of how this advanced form of design analysis integration may be realized. The four variants differ in integration concepts and integration architecture.

Variant A represents the 'classical' integration case attempted in projects like COMBINE (Augenbroe, 1995). In this variant, design context and analysis results are mapped between the inner design team and remote experts and their tools, preferably embedded in an interoperability layer to allow easy data exchange. The latter is realized through a shared building simulation model (also referred to as Aspect Model). The interface is data oriented, with little or no support for process management and task flow logic. As the latter is a basic weakness of this variant, the COMBINE team introduced a so-called Exchange Executive module to control data exchange by embedded workflow logic. This approach may now be recognized as an early and rudimentary form of Variant B, which maintains comprehensive data and workflow management and enacts task coordination during the design analysis interaction. In addition to variant A, the interface has explicit knowledge about the design analysis scenarios that are delegated to a consultant. A recent initiative in this area is introduced in more detail in section 4 below.

Variants C and D take a different approach to the team integration challenge. Variant C represents a practical partnering approach, whereas variant D represents rigorous software based approaches. In variant C a team of simulation experts is invited into the inner design team, and a high bandwidth discussion with designers is maintained throughout all stages. In (McElroy and Clarke, 1999) it is shown that this indeed provides the guarantees for expert simulation to be used effectively. An important

condition for this variant to succeed is that it needs upfront commitment from the design team. In Variant D the emphasis is on functional and behavioral interoperability across different performance characteristics. (Mahdavi et. al., 1999) describe how this can be implemented rigorously on the object level of the next generation of integrated design environments. In (Pelletret and Keilholz, 1999) an interface to a modular simulation back-end is described with similar objectives. Both approaches have in common that they rest on the assumption that these environments will ultimately be sufficiently transparent to be accessible to members of the design team without a significant reduction of domain expertise. This assumption takes us back to the origin of designer-friendly tools of Figure 2. The four variants are expected to further develop and possibly merge over the next ten years.

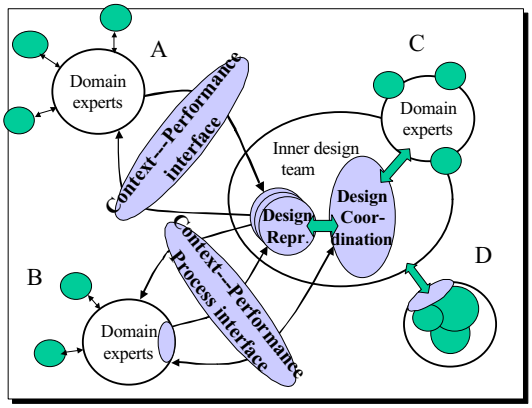

Figure 3Four variants of delegation of expert analysis to domain experts and their tools

A spin-off benefit from employing expert simulation, not always fully recognized, is the improved discipline it places on *decision-making*. As the simulation process itself is systematic, it enforces a certain level of rigor and rationality in the design team decision process. As we are progressively moving towards dispersed teams of architectural designers and analysis experts, full integration of all disciplinary tools in a collaborative design framework is the ultimate goal. This form of *'new integration'* distinguishes itself by fostering a remote engineering culture enabled by group messaging, distributed workflow management, distributed computing, supply side component modeling and delegated simulation. Engineering design in general faces additional external challenges in the area of sustainability, resolution of conflicts across design team members, and above all performing better risk and uncertainty analyses of their performance predictions through all life cycles of the building, as described in (de Wit, 2001) and (de Wit and Augenbroe, 2001).

Future additions to the tool sets of the building simulation profession must be able to respond to all of these challenges.

3. The COMING OF AGE OF BUILDING SIMULATION

Design analysis supported by simulation involves the 'creation' of a behavioral model of a building in a given stage of its development, e.g. reflecting its 'as-designed' or 'as built' or 'as operated' specification. The actual simulation involves executing this model on a digital computer, and analyzing its observable states, which are made up of the post-processed outputs of the simulation runs.

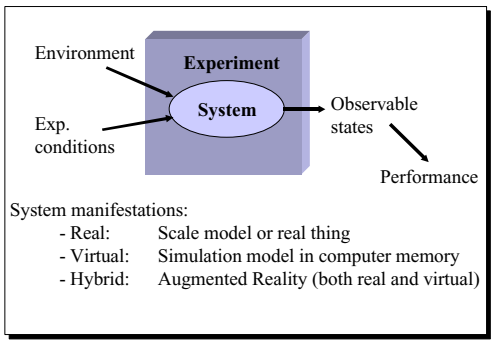

Figure 4 Simulation as a (virtual) experiment

Models are developed by reducing real world physical entities and phenomena to an idealized form on some desired level of abstraction. From this abstraction, a mathematical model is constructed by applying physical conservation laws. A classic overview of modeling tasks in the building physics domain can be fund in (Clarke 1985). Comparing simulation to the design and execution of a virtual experiment (Fig. 4) is not just an academic exercise. The distinction between computer simulation and other means to predict the behavior of a design can become rather artificial indeed. Interesting hybrid variants of simulation have emerged through combinations of real and virtual simulation. One of the more recent 'experiments' that blur the distinction between real and virtual simulation is the deployment of augmented reality (AR) (Malkawi and Chaudhary, 1999). Continued developments in this area will eventually allow a user (*observer* is the better term in this case), while being immersed in the real environment, to interact with a running simulation of himself and his environment. The results of the simulation can be overlaid in real time on the sensory input of the immersed observer, e.g. via computer generation of an image on a transparent visor. But we don't have to resort to exotic AR environments to appreciate the opportunities that this technology offers. We are part of hybrid simulation experiments in everyday life around us. In fact, changing the setting of a thermostat is a 'real' experiment that can

easily be turned into a hybrid experiment if we are given access to a behavioral representation of the building and its control logic. Through an appropriate interface we could interrogate a simulation model about the consequences of the system intervention we are about to make. This is in fact an example of 'invisible' simulation, another variant of which will be encountered in section 6.

Modeling and simulation of complex systems requires the development of a hierarchy of models, or *multimodel*, which represent the real system at differing levels of abstraction (Fishwick and Zeigler 1992). Selection of a particular model is based on a number of (possibly conflicting) criteria, including the level of detail needed, the objective of the simulation, available knowledge and resources etc. The earliest attempts to apply computer applications to the simulation of building behavior ('calculation' is the proper word for these early tries) date from the late 60s. Maturation and expansion of software applications occurred through the 70s. The earliest 'building simulation' codes dealt with heat flow simulation using semi-numerical approaches such as the heat transfer function approach (now virtually extinct). The next generation of tools started applying approximation techniques to the partial differential equations directly, using finite difference and finite element methods (Augenbroe, 1986) that had gained popularity in other engineering domains. The resulting system is a set of differential algebraic equations (DAE), derived through space averaged treatment of the laws of thermodynamics (Fig. 5).

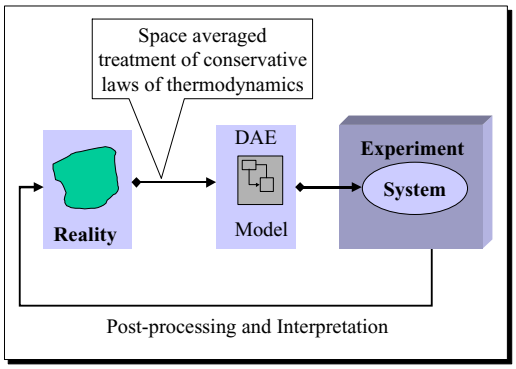

Figure 5 Standard approach to simulation

Since these early days, the finite element method and special hybrid variants such as finite volume methods have gained a lot of ground and a dedicated body of knowledge has come into existence for these numerical approximation techniques. Due to inertia effects, the computational kernels of most of the leading computer codes for energy simulation have not profited much from these advancements.

In the late 70s, and continued through the 80s, substantial programming and experimental testing efforts were invested to expand the building simulation codes into versatile, validated and user-friendly tools. Consolidation set in soon as only a handful tools was able to guarantee an adequate level of maintenance, updating and addition of desired features to a growing user base. As major software vendors continue to show little interest in the building simulation area, the developer community has started to combine forces in order to stop duplication of efforts. The launch of EnergyPlus (Crawley et. al. 1999) is another indication of this.

As to computational elegance, it cannot escape closer inspection that computational kernels of the energy simulation tools (the largest and oldest category of building simulation tools) date back more than 15 years. Rather primitive computing principles have remained untouched as the bulk of the development resources have gone into functional extensions, user interfaces and new components. But thanks to the fact that Moore's law* has held over the last 25 years, current building energy simulation codes run efficiently on Windows based PC's.

Although there is still a dominance of energy assessment tools in the arsenal of the building simulation professional, a host of software applications has been developed for other building performance domains in recent years. Numerically sophisticated computer applications for lighting, CFD and acoustics have emerged in the late 80s and 90s. As these codes are more demanding on computer power, their breakthrough came with the increasing processor speed of low range machines and the advent of dedicated numerical techniques. At this juncture in time, the tool landscape for the consulting building performance engineer is diverse. One of the best overviews can be found on the DOE web site: http://www.eren.doe.gov/buildings/tools_directory/. The list reveals that the emphasis has shifted from an early focus on energy consumption to many other building performance characteristics. The hundreds of man-years that have been invested in building performance analysis tools have paid dividend. One can now choose from a range of tools in each pertinent performance domain. The knowledge embedded in these tools is based on solid investigation of the physical behavior of building components leading to component modules that have in most cases been validated appropriately. A skilled guild of tool users has emerged through proper training and education, whereas the design profession appears to have acquired enough confidence in the accuracy of the tools to call on their expert use whenever needed. In spite of the

* In 1965, Gordon Moore promised that silicon device densities would double every 18 months.

growing landscape and sophistication of tools, many challenges still remain to be met though before the building performance consultancy reaches the level of maturity that a vital role in design decisions demands.

'Old' tool challenges

Many of the challenges that the development community is facing today concern the usual 'old' wish list of desired tool characteristics, concerning learning curve, GUI, documentation, output presentation, animation, interactivity, modularity, extensibility, error diagnostics, usability for 'intermittent' users, and others. Apart from these the user community at large has begun to identify a number of additional challenges. They relate to the value that the tool offers to the design process as a whole. This value is determined mostly by application characteristics. Among them, the following are worth mentioning: (1) the tool's capability to inspect and explicitly 'validate' the application assumptions in each particular case, (2) the tool's capability to perform sensitivity, uncertainty and risk analyses, (3) methods to assert pre-conditions (on the input data) for correct tool application, (4) support of incremental simulation cycles, (5) standard post-processing of output data to generate performance indicators quantified in their attached metrics. Some of these challenges will resurface later in this paper where a 'new' wish list is compiled.

One 'development issue' mentioned above deserves special attention. It concerns the modularity and extensibility of (large) computer codes. In the late 80s many came to realize that the lack of modularity in current 'monolithic' programs would make them increasingly hard to maintain and expand in the future. Object oriented programming (OOP) languages such as C++ were regarded as the solution and 'all it would take' was to regenerate existing codes in an OOP language. Projects attempting to regenerate existing programs and add new functionality in the object oriented programming paradigm were started. EKS (Tang and Clarke 1993), SPARK (Sowell and Haves 1999) and IDA (Sahlin 1996a) are the best known efforts of that period that started new software applications that were intended to be modular and re-usable. Ten ears later, only IDA (Björsell et. al. 1999) has evolved to an industry strength application, in part due to its pragmatic approach to the object-oriented paradigm. An important outcome of these attempts are the lessons that have been learned from them, e.g. that (1) reverse engineering of existing codes is hard and time consuming (hardly a surprise), (2) it is very difficult to realize the promises of OOP in real life on an object as complex as a building, (3) the class hierarchy in an OOP application embodies a particular semantic view of a building. This view necessarily reflects many assumptions with respect to the building's composition structure and behavior classification. This particular developers view may not be suitable or acceptable to other developers, thus making the original objective of re-usability a speculative issue. Another important lesson that was learned is that building an OO simulation kernel consumes exorbitant efforts and should not be attempted as part of a domain specific effort. Instead, generic simulation platforms, underway in efforts such as MODELICA (Elmqvist et. al. 1999), (http://www.modelica.org/) should be adopted. An important step in the development is the creation of a building performance class hierarchy that is identical to, or can easily be mapped to widely accepted 'external' building models. The emerging model proposed by the International Alliance for Interoperability (IAI) (http://www.iai.org.uk) seems the best candidate at this time to adopt for this purpose. This is only practical however, if a 'semantic nearness' between the OO class hierarchy and the IAI model can be achieved. Whether the similarity in the models would also guarantee the seamless transition between design information and building performance analysis tools (a belief held by many IAI advocates, see for instance (Bazjanac and Crawley 1999)) is a matter that needs careful study. The next section will argue that such seamless transition can and should in general not be automated as design analysis translation requires intervention of human judgment and expert modeling skills, strongly influenced by design context and analysis purpose. In an attempt to put the observations of this section in a broad historic perspective, Figure 6 identifies building simulation trends between 1970 and 2020.

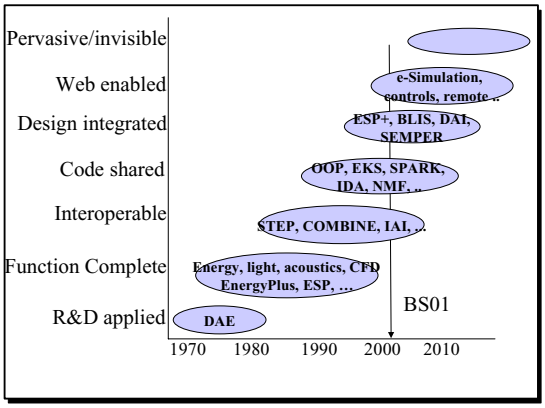

Figure 6 Trends in technical building performance simulation tools

The foundation for building simulation as a distinct class of software applications came with the advent of first principles based formulation of transport

phenomena in buildings, leading to DAE formulations that were amenable to standard computational methods. The next step was towards broader coverage of other aspects of technical building behavior. This movement towards function complete tools led to large software applications that are being used daily by a growing user base, albeit that this user base is still composed of a relatively small expert guild. The next two major movements started in parallel in the 90s and had similar goals in mind on different levels of granularity. Interoperability targets data sharing among (legacy) applications whereas code sharing targets re-use and inter application exchange of program modules. Whereas the first tries to remove inefficiencies in data exchange, the latter is aiming for functionally transparent kits of parts to support modeling and the development of large simulation models.

Design integration adds an additional set of process coordination issues to its predecessor movements. Ongoing trials in this category approach different slices of a very complex picture. It is as yet unclear what approach may eventually gain acceptance as the best framework for integration.

The two most recent trends in Fig. 6 have in common that they are Internet driven. The Web enables a new breed of simulation services that is offered at an increasing pace, mostly in conjunction with other project team services. Ultimately, simulation will be available 'anywhere and anytime', and may in many cases go unnoticed, such as in the case of intelligent control devices that, based on known user preferences, take action while running a simulation in the background. At the moment this simulation would need a dedicated simulation model specifically hand-made for this purpose, but eventually it will be driven by a generic, complete 'as built' representation of the artifact, metaphorically referred to as 'electronic building signature' (Dupagne, 1991). Future simulations may have a 'hybrid' nature, as they deal with both physical objects as well as occupants that may be regarded as 'simulation agents' while interacting with building systems. Occupant behavior, for now usually limited to a set of responses to environmental changes, may be codified in a personalized knowledge map of the building together with a set of individual comfort preferences. Hybrid simulation may be combined with embedded simulation if the knowledge maps and preference sets are encoded digitally on a personal smart card that is read automatically when an occupant enters a building zone. Applications of these types of invisible and pervasive simulation are still in the R&D trial stages.

Fig. 6 shows that a 2001 snapshot cuts through four different layers for which no consolidated solutions exist. This makes any prediction on the time scale for solutions in the upper layers even more speculative. In the next section a closer look at the crucially important integration challenge is taken.

4. DESIGN INTEGRATION

The availability of function complete simulation tools is by itself no guarantee that building simulation plays an important role in design evolution. For that, we need to be able to guarantee that an expert simulation is called upon at the right time, and for the right design decision. Giving this guarantee is a matter of quality assurance (QA) and adequate design process coordination. Proper coordination requires a dynamic view of all design activities, verification of their interrelatedness and anticipation of expected downstream impacts of alternative decisions. Such a dynamic view can be made explicit in a *process model* that (among others) captures the role of the building performance expert analysis in the design decision process. In the AEC industry, the road towards a *generic building product model* is a long and windy one (Eastman, 1999). A *generic process model* does not exist, for many obvious reasons. For one, every building project creates its own 'one-off' volatile partnership and only on the highest abstraction level some established process patterns may exist that can be applied to every project. On finer granularity levels, it is commonly left to each project partnership to define and enforce the most adequate form of coordination among designers and domain experts. An important part of QA is to enforce the right type of coordination at critical decision moments in design evolution. Few unbiased empirical studies exist about the impact of building performance tools on particular design choices. Some surveys have shown that early decisions about environmental technologies are taken without adequate evidence (e.g. backed up by simulation) of their performance in a particular case. A recent 'post mortem analyses' on a set of design projects revealed an ominous absence of the building performance analysis expert in the early stages of the design process (de Wilde et. al. 2001). The study shows that, once the decision for a certain energy saving technology is made on the grounds of overall design considerations or particular owner requirements and cost considerations, the consultant's expertise is invoked later for dimensioning and fine-tuning. By that time the consultant is restricted to a narrow 'design space' which limits the impact of the performance analysis and follow-up recommendations. In the light of these observations, it appears a gross overstatement to attribute energy efficiency improvements of recent additions to our building stock directly to the existence of simulation tools.

In order to become an involved team player, the profession needs to recognize that two parallel research tracks need to be pursued with equal vigor: (1) tools that respond better to design requests, and (2) tools that are embedded in team ware that manages and enforces the role of analysis tools in a

design project. One way to achieve the latter is to make the role of analysis explicit in *design project models*. A project model is intended to capture all process information for a particular building project, i.e. all data, task and decision flows. It contains information about how the project is managed and makes explicit how a domain consultant interacts with other members of the design team. It captures what, when and how specific design analysis requests are handed to a consultant, and it keeps track of what downstream decisions may be impacted by the invoked expertise. Design iterations are modeled explicitly, together with the information that is exchanged in each step of the iteration cycle. Process views of a project can be developed for different purposes, each requiring a specific format, depth, and granularity. If the purpose of the model is better integration, an important distinction can be made between data and process integration. Data driven tool integration has been the dominant thrust of the majority of 'integrated design system' research launched in the early 90s. Their purpose was to develop a common information model for sharing design information among architectural design tools (mostly CAD) and performance analysis tools. An early pioneer project in the energy/HVAC area was the aforementioned COMBINE project (Augenbroe 1995). Similar projects have been deployed in other domains and, although these projects have delivered significant results, a real breakthrough in open integration platforms has not happened for a number of reasons (Augenbroe et. al., 1998). The first reason is the lack of a 'scoping' mechanism, i.e. a systematic way of identifying clusters of applications 'worth integrating'. The COMBINE team addressed this by introducing so-called 'Project Windows' (PW). Useful PW's may be identified by analyzing real projects and should be supported by a business case that defines who benefits from better integration within the PW. Other, technical reasons that have prevented a breakthrough become apparent by inspecting the state of affairs in current 'integration technology' and standardization:

(1) integration technology developed in STEP-related efforts (http://www.nist.gov/sc4/www/stepdocs.htm) has still not delivered on its early promise of *easy* integration. Building a robust STEP interface remains a laborious effort that requires a combination of advanced programming skills and deep domain knowledge. Furthermore, a STEP interface should cover all possible cases as there is no mechanism for ad hoc repair of exceptions. With the 80-20rule in mind this leads to over-engineered solutions requiring exorbitant resources.

(2) lacking standardization in the AEC domain; the STEP initiative in the AEC domain has failed whereas the ongoing IAI effort is still far removed from providing a 'complete' and proven standard.

Until this date, the following fundamentals seem to have been ignored: (1) many tools are semantically too far apart to assume that there exists a *computable mapping* between their information models. This fact renders it impossible to generate one model from the other by a procedure (current integration technologies provide mapping languages to write automatic procedures or translators for those cases where a computable mapping exists), (2) *data exchange* is always dependent on a *process event*, i.e. data exchange without a process context is meaningless, as it at least requires a triggering and acknowledgment mechanism. As a consequence, we argue that tool integration has to start with scenario development and mappings should be dealt with as they occur at specific events in those scenarios. A generic data integration solution (generic in the sense that it assumes a 'one fits all' mapping) may be possible when tools are in *semantically close* domains but will prove elusive in most other cases. It is interesting to note that the IAI development of their Industry Foundation Classes (IFC) reflects awareness of these fundamental issues (http://aiaweb.lbl.gov) but as yet offers no workable solutions for it. The IFC model harbors many domain models 'concurrently' but supports no mechanism to translate between them, and although there is the notion of a (generic) process model with design stages and tasks, exchange events between existing applications are not defined explicitly. The recently started BLIS (http://cic.vtt.fi/projects/blis/) activity, which builds on the IFC model, addresses these issues in concrete application settings. BLIS is one of a series of ongoing efforts born out of the observations made above. Another new initiative is introduced below.

A new initiative in design analysis integration

A new wave of collaborative engineering research deals specifically with process integration and workflow control. In contrast with a data centric view, this approach takes a task centric view reflecting its priority to control decision flow. This control may for instance enforce that the right consultants are involved at the right time in the right decision, whilst providing them access to all relevant information. For the building industry in general and the role of building performance analysis in particular, there is not yet much evidence of a similar wave of research. In (Augenbroe and Eastman, 1999) a new research initiative is proposed exploring a new avenue for design analysis integration. The underlying objective is to enable more effective use of analysis tools in design engineering teams. Based on a 'first principles' review of solutions built with current data driven integration technologies, the initiative starts from the premise that these solutions fall short of the typical needs of design analysis integration. The conjecture is that input generation for an analysis tool cannot be done with a standard mapping

between design entities and simulation entities. The reason is that every (non trivial) case requires case specific interpretation, modeling and expert 'translation', in many instances based on 'idealization' of available design information. With current mapping technology, only a part of this can be captured in a procedural mapping between source and target schemata. The challenge is therefore to develop user driven, constructive interfaces. These interfaces should furthermore be 'lightweight' to avoid the over-engineering risk that was mentioned before and that has plagued so many STEP interface projects. The new integration initiative supports the view that the next generation of tools should be able to respond to explicit analysis scenarios, i.e. be able to handle a particular design request and produce output that is cognizant of the context of the request. This can be accomplished by defining project models that contain workflows with explicit team interactions and decision moments. Typically, a building project is too complex and involves too many parties to be captured in one complete and comprehensive project model. Rather, many smaller process models, each dealing with a small 'project window' could be developed. With the help of a proper modeling tool, the design team defines what experts will be consulted for what input and what decision moments will drive the engineering design process. The challenge is to represent a particular design decision as a set of tasks and events and associate the appropriate analysis requests with them. The development of the models is a joint upfront team responsibility. This has the immediate benefit that all domain experts are involved in the early 'project modeling' stage and thus have a chance to control their own 'destiny' in the later execution of the project.

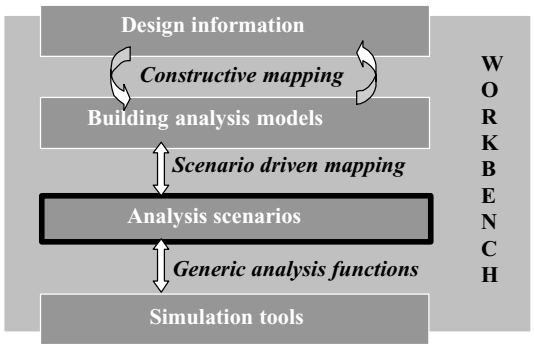

Figure 7Architecture of the Design Analysis Integration (DAI) Initiative

These observations have led to the formulation of a design analysis integration (DAI) initiative with a four-layered workbench architecture, shown in Figure 7.
The central entry point in the workbench is the *Scenario* layer. It offers a *workflow modeler* to specify the task flow logic and the 'interaction events' that connect design and analysis, for a given analysis request. A simulation scenario schedules and coordinates the variety of simulation tasks such as data collection, model preparation, actual analysis tasks, QA, sensitivity study, post-processing, etc..
Relationships and flow logic between tasks are part of the scenario. Tasks may span the spectrum of all typical consultant tasks including designer interrogation tasks. Interrogations may be modeled as distinct events in which case the designer can use the workbench to respond to these 'requests for information' and, if necessary engage in 'design analysis dialogues'. Only in the simplest cases, a scenario will be an ordered list of sequential tasks. In the general case, a scenario is composed of complex sequences of tasks, with branches and iterations. Eventually, typical scenario templates will emerge for frequently recurring analysis jobs. Templates may de-facto represent a consultant firm's workflow of a particular type of 'simulation job'. Templates also make local interoperability requirements and a firm's internal QA procedures explicit.
The *Scenario* layer communicates with the *Tool* layer through generic tool function calls, which are not bound to any specific software. The Tool Layer will resolve a function call and map it to a particular software application run.
The *Scenario* layer communicates with the *Building Analysis Model* layer. Data instances are extracted from an analysis model to be used in scenario tasks whereas the results of an analysis are committed back to the model. 'Raw' outputs are first post-processed into quantified performance indicators before they are committed. The analysis models are populated from available design data with the help of an *interface development kit* with which one defines and executes constructive mappings. The interfaces that execute these user driven mappings provide access to heterogeneous design information, ranging from structured CAD files and other representations (e.g. based on IFC) to unstructured data from product libraries, catalogs etc. The interfaces provide intelligent access to this information aiding the user to populate the building analysis models. A demonstration prototype of this approach is expected to be available in 2002.

5. A TOOL FUNCTION WISHLIST

The proceedings of IBPSA conferences contain many publications that show the increasing palette of functions offered by current building simulation applications (BS97, 1997), (BS99, 1999). Progress has been significant in areas such as performance prediction, optimization of system parameters, controls, sensitivity studies, nonlinear HVAC components, etc. The recent progress in the treatment of coupled problems is also significant, as reported in (Clarke, 1999), (Clarke and Hensen, 2000) and

(Mahdavi, 2001). Yet, there is a set of functions that have been receiving relatively little attention, maybe because they are very hard to realize. A wish list of some of these hard options is given below.

Rapid evaluation of alternative designs by tools that facilitate quick, accurate, and complete analysis of candidate designs. This capability requires easy pre and post processing capabilities and translation of results in selected performance indicators that can easily be communicated with other members of the design team. For rapid evaluation of alternatives, tools need mechanisms for multidisciplinary analyses and performance based comparison procedures that support rational design decisions.

Design as a (rational) decision making process enabled by tools that support decision-making under risk and uncertainty. Tools should be based on a theory for rigorous evaluation and comparison of design alternatives under uncertainty. Such a theory should be based on an ontology of unambiguously defined performance requirements and their assessments through quantifiable indicators. The underlying theory should be based on modern axioms of utility theory and apply them to derive overall measures of building utility.

Incremental design strategies supported by tools that recognize repeated evaluations with slight variations. These tools should respond to an explicitly defined design parameter space and offer a mechanism for trend analysis within that space, also providing 'memory' between repeated evaluations, so that each step in a design refinement cycle requires only marginal effort on the part of the tool user.

Explicit well-posedness guarantees offered by tools that explicitly check embedded 'application validity' rules and are thus able to detect when the application is being used outside its validity range.

Robust solvers for nonlinear, mixed and hybrid simulations, going beyond the classical solving of a set of differential algebraic equations (DAE). The generation of current DAE solvers have limitations in the presence of building controls as they add a discrete time system to the overall set, leading to a mixed problem, and often to synchronization problems. Many DAE solvers fail to find the right solution in the presence of nonlinear state equations (requiring iterative solution techniques) and time critical controls (Sahlin 1996a). Hybrid problems occur when intelligent agents enter in the simulation as interacting rule-based components, as is the case when occupants interact with the control system on the basis of a set of behavioral rules. To cover all these cases, a robust type of multi paradigm solvers is needed.

Two more general, industry wide perspectives should complete this wish list: certification (the end user perspective), and code sharing (the developers perspective). Tool certification is an important aspect of QA, often perceived as enforcing the use of qualified tools and procedures. A recent PhD study (de Wit, 2001) compares certification to a more general approach based on uncertainty analysis. It is argued that at best a calibration (in relation to its peers) of the combination of firm/consultant and available tools and expertise makes sense.

Code sharing is perceived as the ultimate target of efficient collaborative programming, and object oriented environments are considered as the paradigm that will make it happen. As introduced before, the benefits of object-oriented frameworks, e.g. modularity, reusability, and extensibility, are well understood. Frameworks enhance modularity by encapsulating volatile implementation details behind stable interfaces, thus localizing the impact of design and implementation changes (Schmidt 1997). These interfaces facilitate the structuring of complex systems into manageable software pieces and object-based components that can be developed and combined dynamically to build simulation applications or composite components. Coupled with graphical environments, they permit visual manipulation for rapid assembly or modification of simulation models with minimal effort. This holds the promise of a potentially large co-developer community as these platforms offer the capabilities to exchange whole systems or parts with other developers. Wherever co-development is practiced, it is predominantly on code level but the WWW evolution holds strong promises for functional sharing (i.e. borrowing the functions rather than the code). A prime manifestation of this is distributed simulation, which is dealt with in the next section. The biggest barrier for the opportunities of shared development is the level of resources that need to be spent on the re-development of existing tools and the reverse engineering efforts that come with it. Unfortunately it is often regarded a safer route to expand legacy tools, but in other domains it has been shown that this approach requires more effort and produces less desirable results than a completely new design (Curlett and Felder 1995).

Realistically, it must be acknowledged that not many items on the wish list stated in this section will be realized within the near future.

6. BUILDING SIMULATION AND THE WWW REVOLUTION

Many engineering disciplines are discovering the World Wide Web as a prime enabler of remote collaborative partnerships and the opportunities for 'e-simulation' are imminent. The Web facilitates new forms of data sharing and distributed engineering through web hosted services. Grand scale use of web hosted simulation services in the building industry is not expected in the near future, but much will depend on how 'Internet ready' the

current simulation applications will be by the time that Application Service Providers (ASP) discover the growing market of e-simulation. Especially Web hosting providers of project team spaces (as explained earlier) will soon be ready to expand their services in other areas.

To understand the role of the web in various manifestations of distributed simulations such as 'delegated' and collaborative simulations, it is useful to classify the various forms of concurrency in distributed simulation into four types (Fox 1996): data parallelism, functional parallelism, object parallelism, and 'component parallelism'. The latter form is suitable for distributed simulation, if building components can be defined that encapsulate behavioral physics (i.e. thermal, acoustic, etc.) of building subsystems. This would stimulate shared development and distributed simulation if the following demands are met: (1) availability of a high level interoperability architecture allowing independent development of new components, (2) availability of a simulation methodology and component classification that is suited for distributed simulation. High-level simulation architectures have been introduced for tactical military operations where loosely coupled components such as autonomous agents in a battlefield interact at discrete events (see http://www.dmso.mil). A building is an inherently tightly coupled system, which from a distributed simulation viewpoint leads to high bandwidth data parallelism. This would not fit the distributed component supplier model. There is a need to classify the generic aspects of building systems behavior in order to define a common engineering representation consisting of building components, subcomponents and subassemblies. A major step in this direction is provided in the current Neutral Model Format specification (Sahlin 1996b). Object class morphology provides the necessary structure to accommodate a common engineering model, and to define the essential interfaces for component coupling.

A more immediate and low-level application of e-simulation is web hosted simulation through an ASP business model. This variant offers web access to a 'stand alone' simulation running on a remote machine. Access to the simulation application is typically enabled through a standard web browser. Although this form existed prior to the WWW revolution in traditional client sever modes, the web and Java have now created a more 'natural' environment for this form of application hosting. The ASP model is rapidly expanding from IT services into engineering domains and it is only a matter of time before dedicated e-simulation services will be offered to the building industry. Technically there is a distinction between pure server based simulation and client server federation. In the latter case either a Java 'applet' is downloaded and executed on the client machine or a Java 'servlet' is created which is executed on the server while maintaining a dedicated communication with the client. From a functional viewpoint both can be treated as identical. Some of the business cases where web hosted simulation has expected benefits over the traditional desktop simulation are:

1. Installation and maintenance on the desktops of the client organization can be avoided, whereas the latest update is always available to all users. Moreover, the developers do not need to support multiple operating platforms
2. The developer has complete control over the program and authorizes its use on case by case basis if so desired. This control may also extend to 'pay per use' revenue models for commercial offerings. The opportunities to store and re-use audit-trails of user runs on the server are other benefits.
3. The model can be made 'invisible', which is useful when the user does not interact with the simulation directly but only indirectly, for instance through a web enabled decision support environment. An example of this use, reported in (Messadi and Augenbroe 2001) deals with an Intranet enabled control system for smart façade technologies. Figure 8 shows the principle of the approach, consisting of an 'Internet ready' building system that plugs into the Internet making it instantly accessible from any node on the Internet. In this instance, the model performs autonomous simulations in response to proposed user interventions in the system controls.

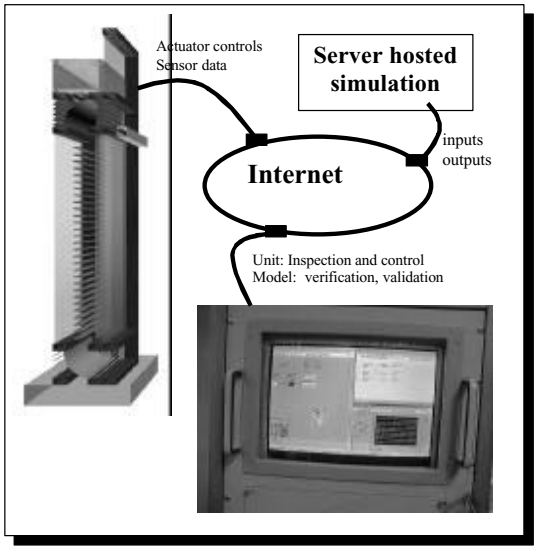

Figure 8Architecture of smart facade demonstration unit

4. The simulation may be part of an e-business service, such as the Web hosted electronic catalogue of a manufacturer of building components Each product in the catalogue could be accompanied by a simulation component that allows users to inspect the product's response to user specified conditions. Web hosting makes a lot of sense in this case, as manufacturers are reluctant to release the internal physical model with empirical parameters of a new product to the public. Taking this one step further, the simulation component may be part of the selling proposition and will be available to a buyer for downloading. This will for instance allow the component to be integrated into a whole building simulation. Alternatively the component could remain on the server and be made to participate in a distributed simulation. Obviously, there are important model fidelity issues like applicability intervals and validation that need to be resolved before this simulation service could gain broad acceptance.

7. FINAL REMARKS

We predict that future trends in building simulation tools are driven by the needs for better informed design decisions and better quality control over performance assessments. Although there remain weaknesses and gaps in tool functionality, the more immediate challenge is to better integrate simulation in all phases of the building process.

Object-oriented frameworks respond to the needs for co-development by leveraging proven software design. The approach should be based on a reusable component-based architecture that can be extended and customized to meet future application requirements. Such a high level architecture is still elusive in the building industry although important building blocks are already available.

The Internet is the natural environment for distributed simulation but the building performance research community faces the uphill task of developing a common engineering representation that is capable of providing the high level architecture for component sharing. With this architecture in place, the Web could act as a catalyst for top down development of interoperable components, and simulation could become an integral component in team ware for remote collaboration in design and engineering teams. With this in mind, building performance experts should proactively engage in the deployment of a new breed of team ware for project management, as this will ultimately guarantee the profession to control its own destiny in a project through proper input and role specification at project definition.

With the advent of these resources, it may ultimately be appropriate to enforce a formal agreement between design team and building simulation expert, concerning the model assumptions that underlie a delivered design analysis. Model specifications that are suitable for such formal agreement do not exist in current practice. Research in this area should deal with certification and expert calibration, based on approaches that use uncertainty and risk analysis.

REFERENCES

Augenbroe, G.L.M., 1986: Research-oriented tools for temperature calculations in buildings*, Proc. 2nd Int. Conf. on System Simulation in Buildings, Liege.*

Augenbroe, Godfried. 1995: *COMBINE 2 Final Report*, CEC Publication, Brussels, 1995 (also available on http://dutcu15.tudelft.nl/~combine/ and http://erg.ucd.ie/)

Augenbroe, G.; W. Rombouts and M Verhoef. 1998: Product Data Technology in A/E/C Systems, Past, Present and Future, *ECPPM conference, Watford, UK, October 1998, pp. 47-54.*

Augenbroe, Godfried and Charles Eastman, 1999: Needed progress in bulding design product models. Internal white paper, Gorgia Instituite of Technology.

BS97, 1997: *Proceedings of Building Simulation '97*, Fifth IBPSA conference, Prague. Editors: J. Spittler and JL.M. Hensen.

BS99, 1999: *Proceedings of Building Simulation '99,* Sixth IBPSA conference, Kyoto. Editors: N. Nakahara and JL.M. Hensen.

Bazjanac, Vladimir and Drury Crawley. 1999: Industry Foundation Classes and interoperable commercial software in support of design of energy-efficient buildings. *Proceedings of Building Simulation 99, Sixth International IBPSA conference, Kyoto, September 1999, Paper B-18.*

Björsell, Niclas; Axel Bring; Lars Eriksson; Pavel Grozman; Magnus Lindgren; Per Sahlin; Alexander Shapovalov and Mika Vuolle. 1999: IDA indoor climate and energy. *Proceedings of Building Simulation 99, Sixth International IBPSA conference, Kyoto, September 1999, Paper PB-10.*

Clarke J A. 1985: *Energy simulation in building design*, *Adam Hilge*r, Bristol and Boston.

Clarke J A. 1999: Prospects for truly integrated building performance simulation. *Proceedings of Building Simulation 99, Sixth International IBPSA conference, Kyoto, September 1999, Paper P-06.*

Clarke, J.A., and J.L.M. Hensen, 2000: Integrated simulation for building design: an example state-of-the-art system. *Proc. Int. Conf. Construction*

Information Technology 2000 (CIT2000), CIB-W78/ IABSE/ EG-SEA-AI, Vol. 1, pp. 465-475, Icelandic Building Research Institute, Reykjavik.

Crawley Drury B.; Curtis O. Pedersen; Richard J. Liesen; Daniel E. Fisher; Richard K. Strand; Russell D. Taylor; Linda K. Lawrie; Frederick C. Winkelmann; W. F. Buhl; A. Ender Erdem and Y. Joe Huang. 1999: ENERGYPLUS, A new-generation building energy simulation program. *Proceedings of Building Simulation 99, Sixth International IBPSA conference, Kyoto, September 1999, Paper A-09.*

Curlett, B. P. and Felder, J. L., 1995: *Object-oriented Approach for Gas Turbine Engine Simulation*, NASA TM-106970.

Doherty, Paul, 2001: *Cyberplaces: The Internet Guide for Architects, Engineers, Contractors and Facility Managers* (2nd edition), RS Means, 2001

Dupagne, Albert (ed.), 1991: ESPRIT-CIB Exploratory Action "Computer Integrated Building", CEC-report DG XIII.

Eastman, Charles M. 1999. *Building Product Models: Computer Environments Supporting Design and Construction*. NewYork, CRC Press.

Elmqvist, Hilding; Sven Erik Mattson and Martin Otter. 1999: Modelica - A Language for Physical System Modeling, Visualization and Interaction. *IEEE Symposium on Computer-Aided Control System Design, CACSD'99, Hawaii.*

Fishwick P. A. and Zeigler, B. P., 1992: A Multimodel Methodology for Qualitative Model Engineering," *ACM Transactions on Modeling and Computer Simulation*, Vol. 12, pp. 52-81.

Fox, G. C. 1996: *An application perspective on high-performance computing and communications*. Technical Report SCCS-757, Syracuse University, NPAC, Syracuse, NY, April 1996.

Mahdavi, Ardeshir, Mustafa Amre Ilala, Paul Matthew, Robert Ries, Georg Suter, Rohini Brahme, 1999: The architecture of S2. *Proceedings of Building Simulation 99, Sixth International IBPSA conference, Kyoto, September 1999, Paper A-38.*

Mahdavi, A. 2001: Distributed multi-disciplinary building performance computing. Proceedings of the 8th Europia International Conference Delft, The Netherlands pp. 159 – 170 . (Ed.: Beheshti, R.)

Malkawi, A., Choudhary, R. 1999: *Visualizing the Sensed Environment in the Real World.* Journal of the Human-Environment Systems, Vo3: No.1:61-69, 1999.

McElroy, L. and J.A. Clarke, 1999: Embedding simulation within the energy sector business. Proceedings of Building Simulation 99, Sixth International IBPSA conference, Kyoto, September 1999, pp 262-268.

Messadi, Tahar and Godfried Augenbroe. 2001: Configurable smart façade unit for onsite calibration and control optimization. ASHRAE transactions (under review)

Pelletret, R., W. Keilholz, 1999: Coupling CAD Tools and Building Simulation Evaluators. *Proceedings of Building Simulation '99, Sixth International IBPSA Conference, Kyoto, Japan, September 13-15, 1999, p.1197-1202*

Sahlin, P. 1996a. *Modelling and Simulation Methods for Modular Continuous Systems in Buildings*, Doctoral Dissertation KTH, Stockholm, Sweden. 1996 (also available at: http://www.brisdata.se/ida/literature.htm)

Sahlin, P., 1996b: *NMF Handbook, An Introduction to the Neutral Model Format, NMF Version 3.02*, ASHRAE RP-839, report from Building Sciences, KTH, Stockholm.

Sowell, Edward F. and Philip Haves. 1999: Numerical performance of the SPARK graph-theoretic simulation program. *Proceedings of Building Simulation 99, Sixth International IBPSA conference, Kyoto, September 1999, Paper A-05.*

Schmidt, D. C., 1997: *Applying Design Patterns and Frameworks to Develop Object-Oriented Communications Software*. Handbook of Programming Languages, Volume I, P. Salus, ed., MacMillian Computer Publishing.

Tang, D. and J. A. Clarke. 1993. Application of the Object Oriented Programming Paradigm to Building Plant System Modelling. *Proceedings of Building Simulation '93, Third International IBPSA Conference, Adelaide..*

P. de Wilde, M. van der Voorden, J. Brouwer, G. Augenbroe and H. Kaan. 2001: The need for computational support in energy-efficient design projects in The Netherlands. *This Proceedings.*

Wit, M.S. de, 2001: *Uncertainty in predictions of thermal comfort in buildings*. Doctoral Dissertation, TU Delft, June 2001.

Wit, Sten de, and Godfried Augenbroe, 2001: Uncertainty analysis of building design evaluations. *This Proceedings.*