THE DESIGN ANALYSIS INTEGRATION (DAI) INITIATIVE

Godfried Augenbroe[1], Pieter de Wilde[2], Hyeun Jun Moon[1], Ali Malkawi[3],
Rohini Brahme[4] and Ruchi Choudhary[5]
1. Georgia Institute of Technology, College of Architecture, Atlanta, USA
2. TNO Building and Construction Research, Delft, The Netherlands
3. University of Pennsylvania, GSFA, Philadelphia, USA
4. School of Architecture, Carnegie Mellon University, Pittsburgh USA
5. University of Michigan, Doctoral Program in Architecture, Ann Harbor, USA

ABSTRACT

The Design Analysis Integration (DAI)-Initiative aims to steer towards new solutions for design analysis integration. These solutions should be able to overcome the limitations of current interoperability approaches that assume the existence of generic and static interfaces in a 'perfect world' in which all information is structured and all mappings between design and analysis representations are computable. This paper reports on the first phase of the development, a first-generation prototype in 'workbench' style for managing a process driven design analysis dialogue. The workbench is meant to enable a more robust use of existing building models such as IFC for the mapping to simulation tools. The paper presents the underlying theories, prototype development, and findings from the DAI-Initiative and concludes with a discussion of future work, targeting extension and benchmarking of the current prototype.

KEYWORDS

Building design, performance analysis, building model, interoperability

1. INTRODUCTION

The Design Analysis Interface-Initiative (Augenbroe and de Wilde, 2003; DAI-Initiative, 2003) was launched to develop credible solutions for the integration of building performance analysis tools in the building design process. These solutions are driven by the need to enable a more effective and efficient use of existing and emerging building performance analysis tools by collaborating building engineering teams. The longer term objectives are better functional embedding of performance analysis tools in the design process, increased quality control for building analysis efforts, and exploitation of the opportunities provided by the Internet. The latter refers to the possibilities for collaboration in loosely coupled teams where the execution of specific building performance analysis tasks is delegated to (remote) domain experts). It is obvious that in such teams process coordination is the critical factor with interoperability as a support act rather than the main objective.

The DAI-Initiative starts from the premise that available solutions for integration based on building product modeling and standardization efforts alone will not be able to meet the longer term objective for a number of reasons (Augenbroe and Eastman, 1998):

- current product models and standards are focused on data exchange; they do not take process context into account and therefore are unable to deal properly with data exchange control issues that are related to process logic
- current developments in building product models focus on single uniform ('neutral') building models. Yet neutral models have some distinct disadvantages which will be discussed briefly. Firstly, interfaces between neutral models (containing all available data about a building) and specific tools (dealing with one performance aspect only) have to filter out only the relevant information, making these interfaces overly complex ('over-engineered'). Secondly, the mapping of data in typical design domains to technical and performance evaluation domains (e.g. to lighting or acoustics) might not be possible (current interoperability research has failed to address the fundamental issue of computability of mappings). Thirdly, the use of neutral models might have implications for the order of execution of the steps in a building design process, imposing a rigid order for the use of tools and models. In fact the success stories about interoperability seem to be invariably based on a set of rigid assumptions about workflow.
- current product models and standards assume that all information about a building design is well structured and stored in 'structured idealizations' of reality. Yet, as a fact of life, a vast proportion of information will live in unstructured media such as text documents, informal memos, personal notes etc.

- current product models assume that data mapping can be automated; this ignores that there will always be a need for additional expert driven idealizations, based on schematization skills and engineering judgment

There are many relevant past and ongoing efforts in the design analysis interoperability field. Two efforts that have significance for the origins of the DAI effort are worth mentioning in particular. These two efforts are the COMBINE effort in the early 90s (Augenbroe, 1995) and the ongoing IAI-IFC effort (Bazjanac and Crawley, 1999; International Alliance for Interoperability, 2002). COMBINE represents an early effort to embed interfaces in a scenario management layer based on a Petri-Net approach (Augenbroe et al, 1998). Although this approach provides necessary control over design analysis integration, it has proven to be inadequate to meet all the challenges mentioned in the previous section. The IAI effort has concentrated mainly on the construction of one 'complete' building model. The IFC development has not attempted an approach to provide a process context to the efficient deployment of the interfaces between the central IFC model and the analysis applications. Only recently, these fundamental shortfalls have been recognized in projects that attempt to deploy the IFC in actual project settings (BLIS website. But a clear framework and architecture for a design analysis communication platform has as yet not resulted from these efforts.

Other design analysis integration efforts have taken a different approach. Two major efforts are reported in (Papamichael, et. al 1999; Papamichael, 1999), where the Building Design Advisor (BDA) is described, and in (Clarke and Mac Randal, 1991) introducing the Intelligent Front End (IFE). Although these are good examples of how data exchange technology and analysis logic can be deployed they suffer from the drawbacks mentioned in the introduction, building on explicitly (BDA) or implicitly steps and interaction modes between the core design team and domain experts. The resulting tools have in common that they result in templates of "design analysis" dialogues that are too far removed from the idiosyncratic, spontaneous and self organizing behavior that is so common for building teams. There are a series of other design analysis integration efforts that are not rooted in product data technologies but take an alternative route. In (McElroy and Clarke, 1999; McElroy et al, 2001) it is described how a team of simulation experts is integrated into the inner design team. It is shown that this indeed provides the guarantees for expert simulation to be used effectively. This approach can be viewed as being located at the one extreme end of the full spectrum of human-driven on the one end to automated integration at the other end. Yet another approach is presented by Mahdavi et al (1999) and (Pelletret and Keilholz, 1999) who try to develop an analysis environment that ultimately will be sufficiently transparent to be accessible to all members of the design team without a significant unwanted reduction of the role of domain expertise or limitations in the analysis functionality offered.

Within this context of these efforts the DAI-Initiative assumes that it is important to capitalize on all the efforts that have been invested in the development of building product models over the last decennium, notably the IAI-IFC effort, while enabling maximum input of human domain expertise, while making no limiting assumptions about the design process or the logic of the design analysis interaction flow. For these reasons the efforts cited in the previous paragraph were not deemed adequate starting points for the DAI effort.

2. OBJECTIVE, ASSUMPTIONS AND CONSTRAINTS

The DAI development targets a layered support of the interaction between the building design process and a wide array of building performance analysis tools. This is realized by the development of a 'workbench' with four layers. The workbench positions building design information and software applications (and more generically 'analysis tools') on opposite layers; in order to move from information to tool or from tool to information one has to pass through the intermediate layers. Those intermediate layers provide context to a specific interaction by capturing the relevant process step (the associated task) and model aspects, each on a separate layer, as shown in Figure 1.

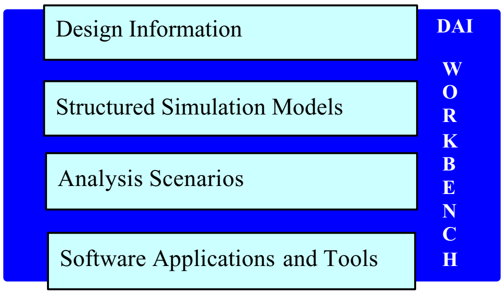

Figure 1 Four Layered DAI workbench

It is interesting to note that interoperability is usually understood to connect the top and bottom layer through an adequate data interface. The DAI approach could effectively been regarded as the 'fat' version of the traditional view on interoperability. The top layer contains all building design information in partly structured, partly un-structured format. The building model layer contains semantic product models of varying granularity that can be used for specific analysis and engineering domains or specific performance aspects. The scenario layer captures the process logic (workflow), allowing to plan a process as well as to actually go through ('enact') that process. These functions are accomplished by current main stream workflow design and workflow enactment applications. The bottom layer contains software applications (mainly building performance analysis tools) that can be accessed from the scenario layer to perform a specific analysis. The software applications are accessed in a neutral way, i.e. a particular analysis function is called from the scenario layer rather than specific software. This concept is fundamental to the workbench. It is based on the introduction of a set of pre-defined 'analysis functions'. Analysis functions act as the smallest functional components in the definition of analysis scenarios. Each analysis function is defined for a specific performance aspect, a specific building (sub)system and a specific measure of performance. An analysis function acts as a scoping mechanism for the information exchange between design information layer, model layer and application layer.

The following assumptions are fundamental to the development of the workbench:

- The workbench is process-centric, and should allow for explicit definition, management and execution of analysis scenarios (represented as a workflow process). This provides a number of additional useful functions, as this allows to store audit trails of any building analysis, reuse of previous scenarios in new projects and provides a learning instrument for novices. As analysis scenarios can be repeated it is easy to support incremental design analysis cycles.
- Expert knowledge and expertise are essential elements of performance assessment. Judgment of the applicability of performance assessment methods and evaluation of the validity of results obtained with (computational) tools are essential human skills that the workbench recognizes.

3. APPROACH

The development of the prototype took the following three steps:

Requirement specification

In order to get a first overview of requirements for the DAI-prototype, a mock-up was made to demonstrate the major functions of the intended prototype, which can briefly be stated as: transparent access to analysis tool functions, configuration of building product models leading to right-engineered data interfaces, workflow design and workflow enactment in the analysis office, capturing of audit trails, and explicit quality assurance procedures. It was used for discussion of the underlying ideas with the tool user community (expert tool users and consultants) through a number of workshops. As the DAI-Prototype needs to be process-centric, repetitive activities in the day-to-day practice of energy analysis work that can benefit from process support (typical analysis scenarios) were identified. A representation method was selected to represent analysis scenarios, whereas three typical analysis functions were selected for further development. Based on the experiences with the mock-up, the feedback from the workshops and the findings regarding analysis scenarios the requirements for the DAI-Prototype were analyzed and written down as a formal specification of requirements.

Development of the prototype

The next step was the actual development of the DAI-Prototype. The prototype was built on an existing commercial platform for process management, adding specific components to interface with the other layers. Note that only some of these components allow for actual live connections; others are only mock-up connections in the current state of the prototype.

In order to demonstrate the concept of analysis functions and their central role in the design analysis dialogue, three typical analysis functions were formally specified and developed. For the actual analysis, they rely on two software applications: EnergyPlus (US Department of Energy, 2002) for building energy simulation, and IDEA-L (Geebelen, 2001) for lighting simulation. Both tools were equipped with interfaces to respond automatically to the three analysis functions.

Demonstration and assessment

The resulting DAI-prototype was tested during a final workshop with the user community; this provided feedback on the viability of the workbench in real practice, and helped to define follow-up efforts for further development of the prototype.

4. REQUIREMENT SPECIFICATIONS

The specification of requirements for the prototype was guided by the development and trial execution of a scenario for a simple design analysis process. Based on these efforts the following implementation requirements were stated:

Implementation requirements:

- In order to allow the DAI-Prototype to provide support for design professionals working in a team context, a web-based application is preferred. This will allow the use of the internet for data exchange and communication. For demonstration purposes all components must also work on a stand-alone computer.

- The DAI-Prototype needs to incorporate a module for easy design, adjustment, enactment and documentation of workflows. This module should be based on a teamwork-paradigm with clearly separated workflow design and workflow enactment stages.
- In order to demonstrate the universality of the system, the prototype needs to contain at least two different fully functional building performance assessment software tools on the tool layer.

Process modeling requirements:

- The process modeling method must be able to capture temporal logic as well as other dependencies between tasks; it should be suitable to support loosely coupled teams. It must support a task centric approach to model the workflows, and it should be possible to associate tasks with automatic invocation of software applications.
- It should provide a graphical workflow representation, with models that have a form that can easily be communicated with end users.
- Existing, easy to use commercial software should be used. The tool should have an API which allows the configuration of dedicated workflow management functions to be programmed and embedded in the DAI workbench.

Requirements for links between the scenario and adjacent layers:

- The design of the analysis scenario should allow flexible and rapid design of logical workflows, driven by the skills of experienced consultants. The workflow management (WFM) engine in the scenario layer should deliver the necessary functions to store proven scenarios as local (firm owned) "best practices" for re-use on other projects.
- The task granularity is determined by the definition of a set of universal analysis functions that have validity across different scenarios. It is only on the level and scope of an analysis function that the integration with the two adjacent layers will take place. The analysis function concept delivers the central scoping mechanism that allows both the data interface with the upper layer as well as the tool interface with the lower layer to be managed, thus decreasing overhead and avoiding over-engineering of the interfaces.

Data topology in the DAI workbench:

- The building simulation model layer should preferable contain a series of 'minimal' product models for different performance aspects. These models should indeed be minimal in the sense that they cover a specific functional domain of the building. These so-called 'Aspect Models' contain the neutral description of the essential building elements for a specific performance aspect (Eastman, 1999; Augenbroe, 1995). In some cases an Aspect Model may turn out to be very large, in which case further down-scoping may be necessary to keep the model manageable. A technique that could be used for this is the definition of so-called project windows, as used in (Augenbroe, 1995).
- An interface is needed that translates the data contained in a populated Aspect Models into the representation that has been associated with each "analysis function model". An analysis function model consists of a schema (the building representation needed to describe the object in the analysis) and an 'operational' specification of the analysis that is to be performed.
- For the expression of all data models and exchange of instances, XML technology (Refsnes Data, 2002) was identified as the preferred option.

Requirements for Analysis Functions:

- Analysis functions need to capture the smallest analysis tasks that routinely occur in analysis scenarios. The three analysis functions embedded in the current prototype are based on a random selection
- Each analysis function (AF) is identified by a well defined "virtual" experiment on the object, which is defined to reveal building behavior that is relevant to the performance aspect that is to be analyzed. The analysis function must be defined in a tool independent way, and formally specified by way of an AF-schema. The AF-schema defines the data model of the building system that the AF operates on as well as the experiment and the aggregation of behavioral output data.

5. PROTOTYPE

The DAI-prototype development started with the development of the mock-up (in the form of a storyboard), which was then gradually developed into the prototype by adding and incorporating workflow management, analysis functions, tools and their data interfaces.

Storybook example

The ‘storybook’ example describes a simple exemplary analysis process that supports the selection of a glazing system during the design of an elementary office space. It provides the elements that demonstrate the building analysis process, starting with the initial design analysis request from a designer/architect, followed by the planning of the analysis efforts, the actual execution of the analysis, potential mid-stream modification of the analysis plan, and ends with the feedback provided to the designer/architect. The storybook explains how workflow design can be used to plan the analysis effort, and how workflow enactment can be used to execute it; it also shows how audit trails can be captured and how reuse of previous analysis scenarios fits in the overall picture. A page of the storybook is depicted in figure 2.

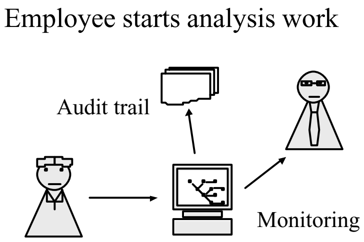

Figure 2: Page of the storyboard

Workflow design and workflow enactment

The first scenario modeling was undertaken with a commercial workflow process modeling tool.

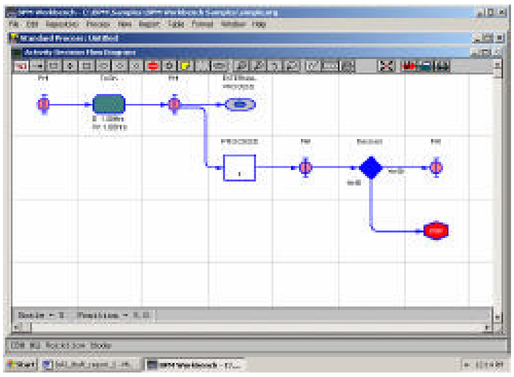

Figure 3: Workflow modeling window

The software offers a graphical front end to a variety of ‘workflow enactment engines’, i.e. computer programs that automate the dispatching of tasks to assigned task performers, transfer of documents, coordination of dependencies between information, tasks, tools and actors in an organization. It allows graphical representation of tasks in process flow diagrams that can be constructed using drag-and-drop capabilities. It also allows easy decomposition of complex processes (Holosofx, 2002). A screenshot of the process modeling window is shown in figure 3.

Processes can be exported to so-called workflow management engines. In this case the IBM MQ Workflow (IBM, 2002) engine has been used. This is a robust, task-based system (MQ = message queue). One of the expected advantages of using a generic workflow engine is the easy integration of the workbench in environments where workflow management is already used to manage other businesses processes. The integration of the simulation process with internal business processes such as invoicing, reporting and resource allocation within the same (or across) collaborating firms is an exciting future prospect for the DAI workbench. It would indeed add significantly to the management and QA of jobs within the engineering enterprise.

Many other useful ‘in-house’ applications could be easily integrated. For instance, keeping track of simulation results across the life cycle of a building the results from efforts like Metracker (LBNL, 2003) could be integrated into the DAI Workbench.

Analysis functions

Analysis functions are the key to the connection of the scenario layer with the building simulation model and the software application in the tool layer. They allow the expert to specify exactly what needs to be analyzed and what results (captured as quantified performance indicators) need to be conveyed. The analysis itself is fully embodied in the choice of an analysis function. It should be noted that not all analysis function calls need to be performed by a software application. For instance, some analysis function may define the daylighting performance of a window as the result of real experiment, e.g. by putting a scale model in a daylight chamber. Other analysis functions may be defined such that the measure is qualitative and subjective, based on expert engineering judgment. Allowing all these different type of analysis functions to be part of the same environment and controlled by a transparent workflow model adds to the control over the process, especially if one realizes that many different experts may be called upon in the same project, each using their own analysis expertise.

In the DAI-Prototype three analysis functions were fully developed. The choice of analysis functions was inspired by the storybook example. Relevant analysis functions in this storyboard were (1) 'maintain thermal comfort in office space' and (2) 'make office space energy efficient', Thermal comfort can be measured through PMV values as defined by Fanger (1970), whereas normalized heating and cooling loads are adequate measures of energy performance. Another (non-thermal) analysis function was added to show the multi performance domain approach: (3) 'maximize daylighting autonomy of an office space' (defined as the percentage of office hours that the office does not need artificial lighting).

The development of an analysis functions requires a number of steps. First of all, a description of the required performance analysis is created using common language descriptions. The next step is to develop an unambiguous, machine-readable version of the analysis function dealing with the object (a building sub(system)), the experiment and the method of output aggregation. This provides a formal representation of the building system data and behavior (in the context of the experiment defined as "observable output as resulting from a given input") associated with an analysis function. The formal expression of the entities is done with the help of a conceptual modeling language. In this case EXPRESS was used (Eastman, 1999). In the third and final step the EXPRESS models are translated into XML, as this proved advantageous for the implementation of the data exchange software The AF descriptions, EXPRESS models and XML versions of three analysis functions are available on the DAI website (DAI-Initiative, 2003). The following brief discussion of the main elements in an analysis function description is given in order to convey the underlying ideas:

An analysis function defines an experiment needed to generate behavior (building states over time) that can be observed; from these observed states different measures for different aspects of the functional performance of the building (or building sub-system) can be derived. Each experiment is defined by the following elements:

- The experimental set-up being observed (the "test box")
- The experimental conditions to which the set-up is exposed (the "load" that is applied to the test box)
- The observation schedule that is used for observation of the generated states (the "measurement protocol" or "time series")
- The aggregation procedure; the observed states (the output of the experiments) are intrinsic to each experiment. Depending on the analysis function there is an option to specify how the observed states are to be aggregated into a Performance Indicator.

For each individual analysis function, the entities and attributes that are described by the experiment are based on a design analysis view. For instance, the analysis function for the assessment of thermal comfort is based on the decision to evaluate thermal comfort using PMV-values. Because of this, the analysis function needs to describe those entities that are needed to calculate PMV-values: there needs to be an internal air zone that has an average air temperature, and there need to be surfaces that have temperatures that can be used to calculate a mean radiant temperature. Also, occupants need to be defined that have a metabolic rate and clothing value. However, if the decision had been made to base the thermal comfort analysis function on a different measure, for instance the use of degree hours for the air temperature, then there would not have been a need to include any occupant and occupant properties, and the treatment of surfaces might have been different. Note that different analysis functions for thermal comfort, like a PMV-based and a degree hour-based function, can co-exist in the DAI-Workbench, and can be defined independently of their software realization.

Data exchange interfaces to software applications

In order to do the analysis as identified for the three analysis functions (energy efficiency, thermal comfort and daylight autonomy), existing simulation tools have been embedded in the tool layer of the prototype. For the thermal building aspects the dynamic building simulation tool EnergyPlus (US Department of Energy, 2002) has been selected. EnergyPlus is only a simulation engine; at the current time it does not come with extensive user interfaces or shells. This allows the DAI-Prototype to interact directly with the engine. For the quantification of the daylight autonomy, use has been made of the lighting tool IDEA-L (Geebelen, 2001). Like EnergyPlus, IDEA-L is basically a computational engine, allowing easy configuration for specific analysis functions, and thus optimally suitable for use in the DAI-Prototype. For both software applications the information contained in the AF data model is parsed to an input file that is handed over to the simulation tool and handled by the AF specific (small) interface of each application. These interfaces are generic, based on XML schemas that describes the structure of the AF information on the one hand and the structure of the native AF model on the application side. These interfaces are relatively easy to develop, as they parse information from a dedicated, small AF schema, rather that a complex model for the whole building. The development and validation of these 'lightweight' interfaces is therefore a more manageable task.

Prototype demo

All of the above elements, that is the workflow design tool, the workflow enactment engine, three AF models and the simulation tools EnergyPlus and IDEA-L with their specific interfaces to the three AF's have been brought together in the DAI-Prototype. The current prototype encompasses the following live functions:

- An analysis scenario is defined in the prototype in the form of a workflow. During the definition of the workflow, previously made processes can be reused. A standard process has been made that includes all tasks that can be supported in the prototype. The elements of this standard process can be rearranged and reconnected to make new processes, without having to redefine all tasks and links between those tasks and the software applications embedded in the prototype.
- Once the workflow has been defined it can be enacted. This will result in tasks being dispatched to the allocated workbench user in the appropriate sequence, whereas the prototype will provide automatic access to applications that are associated with each of those tasks.
- The core analysis functionality in the example scenario is operational for the three analysis functions mentioned. The current implementation provides access to the analysis function model, e.g. for population, control over the operation of the interfaces, and additional viewing and reporting functionality (driven by XML interfaces). Other tasks that do not relate to the core issues are not implemented.
- Apart from the actual enactment, the workflow engine allows to monitor progress of the analysis effort, and keeps an audit trail that can be inspected during and after execution of the analysis process.

The overall workbench-structure incorporating these elements is depicted in figure 4.

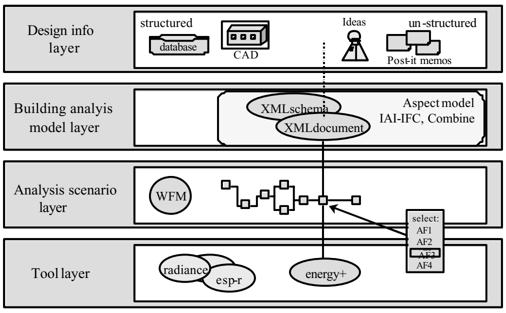

Figure 4: workbench with main elements

5. CONCLUSIONS AND REMARKS

A new approach to design analysis integration has led to a four layered first-generation workbench for building analysis experts. The workbench enables expert teams to respond to an analysis requests with the design of configurable analysis scenarios which can consequently be enacted.

The scenarios are built from a set of re-usable analysis functions, which constitute the basis of the modular design analysis dialogue. Each analysis function defines an experiment that is linked to the study of a specific performance aspect. It has been shown that on the premise of this modularity, many of the obstacles for efficient data integration can be overcome. Another important consequence is that the loose coupling of analysis functions and specific software tools leads to open and flexible integration of legacy and new applications, fostering the innovation of simulation tools.

The prototype shows promise for a viable design analysis integration that can overcome the drawbacks of current data driven interoperability efforts:

- no large, all encompassing building model needs to be assumed for the workbench to function;
- structured information models reside on a separate layer in the workbench, allowing many smaller models to co-exist, and be partly redundant; these models may be domain and "process window" specific and may contain varying levels of "idealized" representations;
- the mapping between design information and the set of co-existing structured representations is supported by constructive interfaces that are integrated in the workbench. Each interface operates in pull mode, initiated by the occurrence of analysis functions in scenarios. This approach leads to minimal interfaces;
- the constructive interfaces are "procedural" replacements of the integration of all domain models. This strategy is driven by the rationale that progressive integration of models will ultimately lead to unsustainable large models and over-engineered interfaces;
  all functionality is AF centric leading to fully automated data transactions between the neutral Aspect Models and the native application models.

The scenario layer of the workbench allows the design and enactment of workflow processes with state of the art technologies. Through this, the DAI approach effective integrates groupware with a novel approach to data interoperability.

Future developments of the DAI-Prototype should first of all deal with the expansion of the set of applicable analysis functions together with the population of the tool layer with additional software tools, each equipped with multiple small analysis function based interfaces. The next step should then deal with the development of the constructive interfaces that populate analysis function models from design information. These interfaces will be small and manageable and thus be easily adaptable to the changing and growing neutral product models (such as IFC) from which they are mapped.

ACKNOWLEDGEMENTS

The reported work was funded by the US Department of Energy.

REFERENCES

Augenbroe, G., 1995. *COMBINE 2, Final Report* [online]. Commission of the European Communities, Brussels, Belgium. Available from http://dcom.arch.gatech.edu/bt/Combine/my_www/document.htm [Accessed 27 April 2002]

Augenbroe, G., Eastman, C., 1998. Needed progress in building design product models. White paper, Georgia Institute of Technology, Atlanta, USA

Augenbroe G. and de Wilde P., 2003. *Design Analysis Interface (DAI). Final Report.* Atlanta: Georgia Institute of Technology.

Bazjanac, V., Crawley, D., 1999. Industry foundation classes and interoperable commercial software in support of design of energy-efficient buildings. *In*: Nakahara, Yoshida, Udagawa and Hensen, ed. *Building Simulation '99, 6th International IBPSA Conference*, Kyoto, Japan., September 13-15 1999, 661-667

BLIS, website, http://www.blis-project.org/

Clarke, J.A. and D. Mac Randal (1991). An Intelligent Front-End for computer-aided building design. *Artificial Intelligence in Engineering*, 1991, Vol. 6, 1, 36-45.

DAI-Initiative, 2003. Homepage [online]. Georgia Institute of Technology, Atlanta, USA. Available from: http://dcom.arch.gatech.edu/dai/

Eastman, C.M., 1999. Building product models: computer environments supporting design and construction. Boca Raton, Florida USA: CRC Press

Fanger, P.O., 1970. *Thermal Comfort.* Copenhagen: Danish Technical Press

Geebelen, B., Neuckermans, H., 2001. Natural-lighting design in architecture: filling in the blanks. *In:* Lamberts, Negrao and Hensen, ed. *Building Simulation '01, 7th International IBPSA Conference*, Rio de Janeiro, Brazil, august 13-15 2001, 1207-1213

Holosofx. (2002). *Holosofx web site* [online]. Avialable from: http://www.holosofx.com [Accessed 18 July 2002]

IBM. (2002). *IBM MQSeries Workflow Product presentation* [online]. New York, USA, 2002. Available from:http://www-3.ibm.com/software/ts/mqseries/workflow/ [Accessed 30 July 2002]

Larence Berkely National Laboratory, *2003. Design Intent Tool Homepage.* Available from: http://eetd.lbl.gov/BTP/CBS/BPA/dit.html [Accessed 11 April 2003]

Mahdavi, A., Ilal, M.E., Mathew, P., Ries, R., Suter, G., Brahme, R., 1999. The Architecture of S2. *In*: Nakahara, Yoshida, Udagawa and Hensen, ed. *Building Simulation '99, 6th International IBPSA Conference*, Kyoto, Japan., September 13-15 1999, 1219-1226

McElroy, L.B., Clarke, J.A., 1999. Embedding simulation within energy sector businesses. *In*: Nakahara, Yoshida, Udagawa and Hensen, ed. *Building Simulation '99, 6th International IBPSA Conference*, Kyoto, Japan., September 13-15 1999, 263-268

McElroy, L.B., Clarke, J.A., Hand, J.W., Macdonald, I., 2001. Delivering simulation to the profession: the next stage? *In:* Lamberts, Negrao and Hensen, ed. *Building Simulation '01, 7th International IBPSA Conference*, Rio de Janeiro, Brazil, august 13-15 2001, 831-840

Papamichael K., H. Chauvet, J. La Porta and R. Dandridge (1999). *Product modeling for computer-aided decision-making. Automation in Construction*, Volume 8, January 1999.

Papamichael K. (1999). *Application of information technologies in building design decisions.* Building Research & Information*,* Volume 27, Number 1.

Pelletret, R., Keilholz, W., 1997. Coupling CAD tools and building simulation evaluators. *In*: Nakahara, Yoshida, Udagawa and Hensen, ed. *Building Simulation '99, 6th International IBPSA Conference*, Kyoto, Japan., September 13-15 1999, 1197-1202

Refsnes Data, 2002. *XML Tutorial Homepage* [online]. Available from: http://www.w3schools.com/xml/default.asp [Accessed 30 July 2002]

US Department of Energy, 2002. *EnergyPlus* [online]. Washington, USA. Available from: http://www.eren.doe.gov/buildings/energy_tools/energyplus/ [Accessed 31 March 2003]