COMMUNICATING PERFORMANCE ASSESSMENTS OF INTERMEDIATE BUILDING DESIGN STATES

Godfried Augenbroe* and Sten de Wit**

*Georgia Institute of Technology
College of Architecture, Doctoral Program
Atlanta, Ga 30303-0155, USA
Email: fried.augenbroe@arch.gatech.edu
**TU Delft, Faculty of Civil Engineering
PO Box 5048, 2600 GA Delft, The Netherlands
Email: sten@dutct05.tudelft.nl

ABSTRACT

Collaborative building engineering is a team effort in which many elements have to be combined into a unified structure. The aims of the architect, the engineer and all the other players have to merge into a seamless design process. This paper deals with the exchange of information among the design team members, and introduces methods that enhance a particular subset of the exchanged information, i.e. the information related to performance of intermediate designs. A new technique is introduced to control the invocation of performance assessments as the design evolves, based on the availability and level of uncertainty of input data.

INTRODUCTION

Building design is a cooperative team effort, with an input from a multitude of actors, having backgrounds in different disciplines. Recent research efforts center around the objective to provide an integrated approach to the evolutionary design process, enabling the sharing of all relevant information among the team members. Many of these efforts have tackled the problem of making the building shape, it's structure, fabric and physical properties available in such a way that skilled engineers can easily and rapidly deploy their tools for an expert evaluation of the design.

An obvious necessity for effective design support is the ability to evaluate intermediate design states rather than to merely evaluate the finished (detailed) design. Intermediate states are associated with incomplete designs, which pose the harder problem as one is confronted with missing information that prohibit the straightforward use of detailed simulations. A simple, every day example is the assessment of overheating risk when no decision has been made yet about construction types of exterior and interior walls, or on the type of solar shading, or when no knowledge about the presence of other heat sources is available. Still, the design team might raise the question of overheating risk because some decisions about the layout of the rooms and the aesthetic design of the facade have to be made at that stage of the design process.

One way to approach the problem is to provide 'electronic design assistants' which, at the fingertips of the designer, advise him about the possible consequences of design choices. Design assistants usually are based on reasoning with captured expert knowledge, structured according to analyzed previous cases and extensive typologies of building designs and predefined or parametrized design aspects and decisions. Design assistants are ideally suited to predict trends during early decisions, i.e. in the conceptual design phase.

Another way to deal with missing information is to use sound judgment to fill in missing data and perform (detailed) simulations with the tools of the trade.

In many cases both approaches are unsatisfactory. Whereas the first approach suffers from the fact that the expert knowledge base is never complete (particular details of the current design are not taken into account because the knowledge system has no stored knowledge about them), the second approach suffers from an abundance of (added) data and an artificial increase in detail level of the design. The latter may lead to the unjustified deployment of detailed simulations and thus create overconfidence in the results.

An third approach which may be used as alternative or complementary to the two above ones, is introduced below. The basic proposition of it is that any performance assessment is 'as good as' the data

is it is based on. Given the uncertainties in the input data, the result of a simulation and its subsequent evaluation will not just produce a single value but a probability distribution of the evaluation variable, and should be conveyed as such to the ‘client’ of the simulation, usually the design team. This is not any different in the case of detailed simulations of complete designs, when uncertainties stemming from guessed material properties, assumed model parameters and the simulation model as such, can not be avoided. Lack of data caused by incomplete designs just introduces additional uncertainty.
The cornerstones of the approach that is advocated are the following:

- a rich classification of performance concepts, (along with appropriate indicators and measures), associated to their pertaining building objects, on different abstraction levels and for different composition levels of a building
- a technique to control the scheduling of performance assessment tasks during the design evolution
- an uncertainty analysis and parameter screening procedure to add constraints stating when it is appropriate to invoke a performance assessment because it is expected to add to the certainty of the assessment

With these techniques it is possible to determine whether it makes sense to assess a certain performance indicator in the light of certain missing data.

COLLABORATIVE DESIGN

Information technology is a key enabler of team-style design approaches which take full use of rapid exchange of information, thus making it possible for all players to work around the same interactive core design. Based on recent developments in IT, and more specifically in ‘integration technology’, a new generation of integrated design systems is under development.
A recently concluded international research effort, COMBINE (Augenbroe, 1995) was designed to develop systems with the prime objective to increase the level of integration of design evaluation activities. Concentrating on those players that deal with the crucial decisions with respect to energy and comfort performance of buildings, a number of prototype systems were developed. A key deliverable of the COMBINE project is the common building model, designed to unify different representations of the building, i.e. acting as the central medium for sharing information between actors. The model has been successfully used to exchange shape information (e.g. generated by CAD systems) with building performance evaluation tools in the areas of energy, lighting and comfort. However, the model is still quite underdeveloped when it comes to the reverse exchange, i.e. handing the real meaning and rich information content of output results of these tools back to the other members of the design team. How to make the results of these calculation tools available in a comprehensive and meaningful way is a challenge hitherto more or less neglected by the research community. Below, an approach is proposed, based on three basic improvements: *classification* and improved assessments of performances and support for matching performances with requirements. These improvements pertain to the representation semantics and metrics of exchanged performance values. Another aspect is the accurateness and reliability of the performance values, in relation to the availability and accuracy of data with which these values are calculated. The objective is to use uncertainty analyses to determine the reliability of the exchanged performance data and moreover, determine when the availability of new data warrants the evaluation or re-evaluation of certain design performances.

PERFORMANCE CLASSIFICATIONS IN BUILDING MODELS

Electronic exchange of information is based on a definition of the semantics (meaning) of the data we want to exchange and an agreement on the way we physically transfer the data, i.e. syntax. It is obvious that the first aspect poses the biggest challenge. Semantics need to be formalized and hence expressed in (semantic) models in order to be unambiguous and processable by computers. A model contains conceptual structures with which the relevant information is captured in terms of entities (basic concepts) and relationships between entities. A variety of conceptual modeling languages can be used for that purpose. The models that form the basis of this section are developed in a formal modeling language. As full treatment of the models (Dekker, 1995), would be outside the scope of this paper, a more intuitive treatment is attempted, by which the main purpose of the models is clarified. They are a first step towards integrating rich performance concepts into any building model.
In the initial stages of the research the following limitations were introduced:

– only a small subset of requirements and performances are considered
– only requirements and performances associated with building zones are considered, limiting the approach to just that one decomposition level of the building
– only requirements associated with explicit ‘zone-usage’ criteria are considered

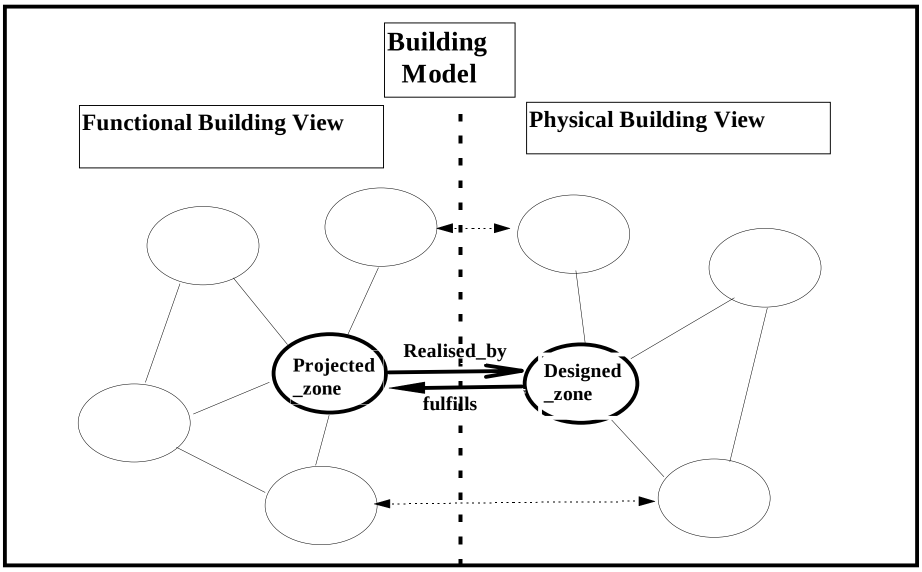

Figure 1 Principle of dual functional and physical views supported by the Building Model

The approach is based on a distinction between '*functional entities*', representing required functions, thus implying required characteristics of (parts of) the design on the one hand, and '*physical entities*' representing chosen design solutions on the other. The basic principle is depicted in Figure 1**.** The expert consultant is called upon to supply performances evaluations (related to energy use, maintainability, fire resistance, etc.) of design solutions.

It should be noted that the satisfaction of required characteristics by design solutions is "what design is all about". Adequate building models that support this association on all different decomposition levels, and through an evolutionary design path are the biggest challenge of future systems (Eastman, 1996). The work presented here is limited to a much narrower scope, i.e. the matching of *Designed_zone performances* with required *Projected_zone characteristics*, and does not deal with the much wider *design evolution* scope. It should be noted however that the need for a full coverage of performance semantics will be imminent in any future system.

Note that all functional entities in Figure 1 on the left have a 'realized by' relationship with the physical counterparts on the right. The relationship can be used to inspect how and to what degree the required characteristics, stored with the *functional entity*, are indeed met by assessed performances, stored with the *physical entity*.

In order to convey the flavor, some highlights of the modeling approach for the functional 'half' are listed below:

- *Projected_zone* can be a subtype of one or more classified *Zone_functions*, identifiable with certain *zone_characteristics* (e.g. *office_zone, meeting_zone, transport_zone, fire_zone,* etc.), each of which carries a reference to documented *rules and regulations*
- each *Projected_zone* is now associated with one or more *Zone_Requirements*; they are derived from the inherited zone characteristics or derived from the *design_brief*
- the type of each *zone requirement* is defined according to a classification of *performance aspects* (*thermal comfort, air_indoor_quality, energy, acoustic, lighting*) and quantified by one or more instances of *performance_ representations.*

Through these constructs, the building model contains the semantics to allow a designer to define a *Projected_zone* (e.g. a corridor), and find relevant regulations (expressed as requirements pertaining to human traffic zones) , derive required characteristics from them (e.g. visibility, way finding, etc.), classify them (lighting, amongst others), relate them to an appropriate *indicator* (e.g. for visual performance) , and link these to one or more *performance_ representations* (e.g. "average illuminance level on floor area by artificial lighting", or "lowest illuminance level on any surface by daylighting with overcast sky") . The latter requires that the model contains a rich set of *performance_representation* instances, with appropriate *measures* (absolute or ordinal value) and *units.* The *performance_ representation* concept is introduced to categorize all types of 'performance expressions' that need to be exchanged in the design process. *Performance_ representation* is a data structure for the results of a performance evaluation experiment. It minimally contains a *measure*, a *value* and a *unit*. While *measure* is a structure designed to contain the relevant conditions that were applied in the performance evaluation 'experiment', *value* relates to the outcome of this experiment. Since some degree of uncertainty will generally be inherent to the outcome, *value* should offer a set of appropriate structures to communicate this. Such a structure would contain, e.g. a mean value and a variance or a number of bins to which probability masses can be assigned. Finally, *unit* is introduced to store the scale to which the content of *value* relates.

The approach to the other half of the model is fully analogous. A *designed_zone* can be subjected to an evaluation, as a result of which it has a *predicted state*, which can be evaluated to classified *performances (thermal comfort, air_indoor_quality, energy, acoustic, lighting,* etc.), each associated with an *indicator* which is quantifiable using one or more *performance_representations* (same as above). This enables the expert to conduct evaluations on a designed zone (e.g. corridor), evaluate the simulated state into certain aspects of performance (e.g. lighting), choose a suitable indicator (illuminance level) and supply a quantified information (e.g. the ones mentioned above). It is important to note that this process involves interpretation and engineering judgment by the expert consultant in terms of what and how he conveys relevant performance information to other members of the design team. The constructs in the building model enable him to express the information in a pre-conceived, but comprehensive way.

ASSESSMENT OF INTERMEDIATE DESIGN STATES

Design is an evolutionary process. Many different parts of the building are being designed separately, whereas many alternatives are in different stages of refinement and detailing. Some partial design solutions are retained, whereas others are discarded or kept dormant to emerge at a later time in the process. Rather than study the process itself, our approach aims to be generic to whatever type of design rationale and design decision management is used. We study the role of the consultant and in particular, how his assessment roles vis à vis his communication with the design team can be enhanced.

It is assumed that at any moment in the process there may be issued an evaluation request, directed at a consultant. If he is consulted at some intermediate design stage, the first decision that he has to make is whether or not the information he has at his disposal enables to make that kind of assessment with a reasonable level of certainty. As actor in an evolutionary design path, he might have been consulted earlier with the same assessment request, so he will want to check whether there is any new information, relevant to the request, that warrants a new assessment. Multiple representations of performances along with a probabilistic analysis of the available information are a key ingredient to allow him to make these judgments. A task model constraining the sequence of assessment tasks, based on the availability of design data as input, is the other basic ingredient. It will be introduced in the next section.

Matching performances with requirements

'Design for performance' is adopted as the leading paradigm for the approach that is advocated here (Kalay, 1996), (Mahdavi, 1996). Our focus is on the support of information exchanges that enable the paradigm, in particular with respect to the quantifiable 'technical' performance aspects that are determined to a large extent in the area of building technology where, traditionally, the 'handshake' between architectural designer and building engineer takes place.

The ultimate aim of integrating richer performance concepts in building models is to enable the design team to inspect at any stage in the design the performance of the design as a whole. Designers and experts will want to check which requirements are met and which are not, inspect the rating of the values of certain performance indicators, or choose a part of the design and inspect the list of all requirements and performance aspects that pertain to that building part. All this should be adaptable to the context of the design task at hand, or the role and expert level of the design actor, who is accessing the information. The latter is crucial in order to present only that information which is relevant to, and can be understood by the observer. It is not hard to see why these goals are difficult to reach. It will require rich models of requirements and performances, fully integrated in a complex building model supporting multiple abstractions and decomposition levels, and with semantically rich (context dependent) associations between them.

In (Dekker, 1996), a prototype of a design system has been developed in order to show the potential use. The prototype is built as a consultant's back end tool to the building model, discussed in the previous section. After the consultant has chosen a particular aspect that he is specialized in (the aspects being fixed by the classifications introduced before), he selects a (designed) zone and the system provides him with all relevant requirements, i.e. projected functions which the zone is to fulfill, and those performance aspects that match the expertise ('expert profile') of the consultant. Each item in the list of required performances is associated with specific evaluation tools which, at the fingertips of the consultant, can be deployed (automatically extracting data from the building model through dedicated interfaces). Based on the results of the calculation

runs, the expert will supply information in the matching performance representations and add them to the building design instance in the database. The prime benefit of this approach is that both the requirements as well as the evaluated performances are now accessible by any down-stream actor in the design process in a way that enables a straightforward comparison, checking for completeness, and inspection of performance information in the format and detail level that the observer has selected.
As explained above, probability information should be an essential part of the measure of a performance indicator. This additional dimension of the information enables to control the information content (in the true sense of the word) during incremental design and evaluation.

A MODEL OF DESIGN TASK CONTROL

This section explains the approach to a task scheduling model, i.e. a formal, computer interpretable representation of the tasks, the actors that perform them and the constraints that control their execution and sequencing. The model has been adopted from the COMBINE project It is based on the metaphor of a Project Window, whereas a Petri-net modeling technique, serves to derive the formal Project Window reference model. Both will be introduced briefly below.

Project Window

The Project Window (PW) metaphor is introduced as a ‘window’ on any suitable, limited portion of the overall design process. Such a portion is usually limited to a small time period of the project (typically a project phase with a well defined start and end state) and only a limited numbers of actors (typically limited to a specific sub-system in the building or dedicated to the exchange that takes place among actors in a particular setting, i.e. to reach a predefined decision point). The PW approach has been proven to be an adequate instrument to implement integrated building design system prototypes directed at a particular industrial need. A number of PW models were generated with input from industry and IBDS prototypes were developed for them.
A PW model describes actors, their roles and tasks, the in and output data for each task and pre and post conditions that regulate the execution of the tasks. It should be noted that a PW model is no attempt at a generic process description. Rather, it is the result of a workfloor approach to the project at hand. PW modeling tools should allow easy configuration at the start of the project and ‘on the fly’ reconfiguration. The execution of the PW model is done in interaction with the Project Manager, who is always in control over what task to execute next.

The Combi-Net technique

PW models are expressed using a flavored Petri-net technique. Petri-nets are well suited to model control over the sequence of tasks. Several types of controls apply:

- conditions determining when a specific actor can perform a particular operation. (temporal logic control)
- conditions to ensure that the model remains in a consistent state while the design progresses. (data integrity control)

A PW model consists of two diagrams The first one captures all the “Actors”, their “Design_Roles”, and the “Design_Tools” they can execute. The diagram describes all the communicating entities in the system and their inputs and outputs.

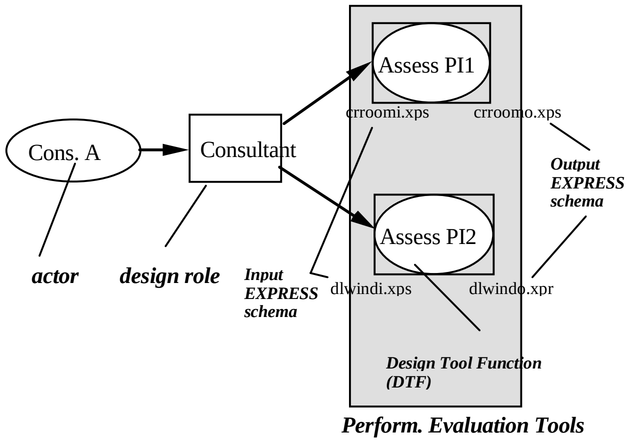

Figure 2. Combine PW Diagram

Figure 2 shows a example for one *actor* (Consultant A), having the *design_roles* to assess two Performance Indicators, PI1 and PI2 , both of which are identified within the system as *Design Tool Functions*, associated with expert performance evaluation tools. The diagram also shows the explicit reference of each function to a predefined subset of the building model. The latter aspect is not dealt with here.
The second diagram type, known as the "Combine_Petri" diagram, describes the order in which these DTF's are allowed to execute in the system. These diagrams use the basic structures of Petri_net diagrams, i.e. *Places* and *Transitions*, where *Places* represent (on the lowest granularity) design tool functions and *Transitions* represent all potentially allowed sequences from one Place to the other, subject to constraints.

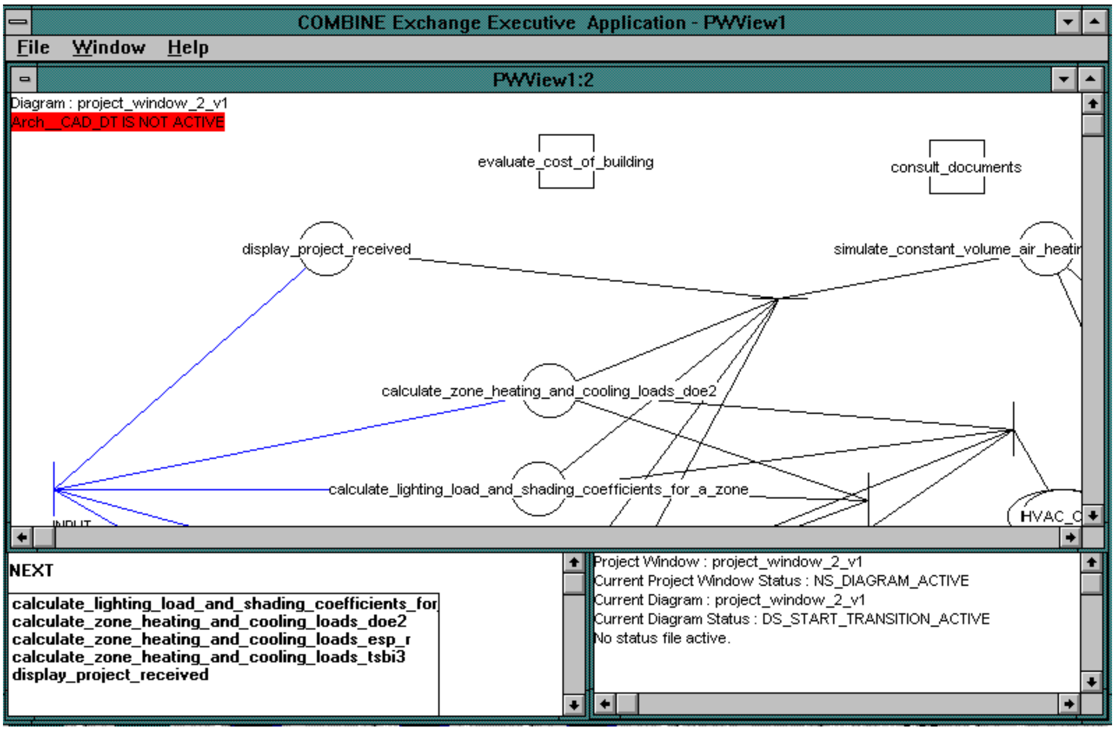

Figure 3: Sample ExEx screen**.**

Figure 3 shows a snapshot of a part of a Petri_net, in the runtime support tool that we will briefly introduce below.

Supervision and Exchange Executive

In order to translate the Combi-Net control model into an agent actively controlling the building design system, an application called the Exchange Executive (ExEx) was developed within the COMBINE project (Figure 3) The project control based on the PW model was essentially developed to control scenarios in which DTFs were allowed to access the central database. The ExEx is capable of reading in the PW model, create a binary image of the model in RAM and execute it. The ExEx tool determines data dependencies between all individual DTF's and evaluates the state of the system based on the evaluation of the pre and post conditions. After the execution of a DTF the ExEx will show a list of all possible next DTF's that the project manager ca choose from. It will also show the list of already executed DTFs which might have to be re-executed because part of their input data has changed.

This paper adds an additional category of control options to this approach, i.e. those relating the available information to the usefulness of performing a certain evaluation. In that case, the 'project manager' would usually be the local manager of the consultant's office.

TASK MODEL OF INTERMEDIATE ASSESSMENTS WITH UNCERTAINTY

To apply the approach of the preceding section to model and execute the control over if and when to perform a performance evaluation, the following premises should hold:

- design tasks and their decompositions into granular design tool functions (DTF) have been defined according to the rules explained above
- for each DTF there is prior knowledge of the part of the building it needs as input and the data it generates, e.g. in the case of a performance assessment tools the input data and the output data (performance concepts related to the entities in the input) have been modeled prior and have been identified as a subset of the complete building model
- each DTF has a one to one relationship with a performance concept and a tool or method to evaluate that particular performance
- an explicit two-way mapping has been defined to correlate the input parameters to the input/output models of the DTF

This corresponds to the standard approach, which always assumes that the input data dependency of the DTF execution is purely a matter of whether that input data is available or not. This prohibits the defaulting of missing data and it also lacks the context of the performance request. In many cases it is permissible, given the early stage of the design and the low required accuracy of the evaluation to 'default' missing data (locally in the DTF application) and still be able to present meaningful information to the design team. In many cases however, the lack of information will make an evaluation useless and even misleading in a given design context.

Extensions based on uncertainty analysis of DTF input data

For appropriate DTFs in the PW, an uncertainty analysis is performed in order to formulate the constraints that determine whether the execution of the DTF (remember that a DTF has a one to one relationship with a particular performance concept and a specific indicator for it) makes sense, given the availability of the input data. This results in additional control over the execution of DTFs:

- the Petri-Net itself specifies the general order in which a DTF may be invoked (the standard control). It also specifies at what stage in the process more elaborate tools may be invoked generating new or re-generating previous performance assessments
- when invoked, a DTF uses the design context (the position and history of the token in the Petri-Net) and prior uncertainty analysis to determine whether it is permissible to default missing data, using some 'internal' default values

In the case of design loops, when the DTF is repeatedly invoked in an incremental evaluation sequence, prior uncertainty analysis is used to determine whether the change in input data warrants a new evaluation (the change in the input parameters might be such that no significant decrease in the uncertainty of the result can be expected).

The next section will show how the information for the uncertainty related control of the execution of performance evaluations can be generated

UNCERTAINTY ANALYSIS OF MISSING DATA

In present day practice, the DTF's are executed deterministically, i.e. inputs and parameters of the underlying model are treated as known values (Figure 4) However, the values of these parameters and inputs will often be uncertain. Examples of sources of uncertainty are:

1) buildings are not constructed exactly to specifications.
2) the building design specifications are insufficiently detailed
3) the model does not fully capture the complex (physical) processes it claims to describe.

The actors in the design process need quantitative information about these uncertainties in order to control the reliability of their design decisions. One way to achieve this, is to make uncertainty analysis (Figure 5) an integral part of the performance evaluations (de Wit, 1997) and enrich the information structures in order to allow the uncertainty assessments to be shared in the design process.

If we want to integrate a full uncertainty analysis into the DTF, a joint probability distribution over the input parameters would be required as input for the DTF, and each execution of the DTF would imply a cyclic refinement of this pdf. Since rigorous changes

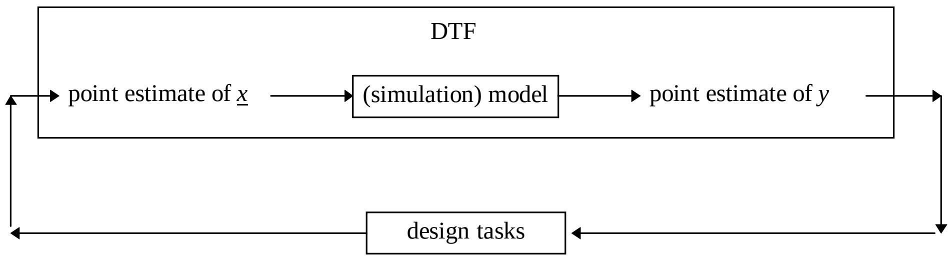

Figure 4 Traditional uncertainty analysis, no update of information:

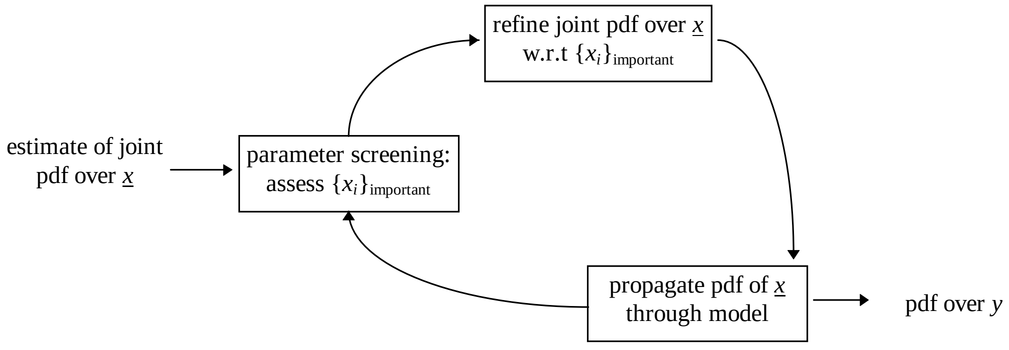

Figure 5 The uncertainty analysis cycle is repeated until the resulting pdf (probability density function) over y no longer changes significantly.

of the parameter vector may be expected in each design cycle, these refinements would often prove irrelevant. Moreover, the estimates for the uncertainty in the parameter vector will be of such a crude nature, that instead of a full probability distribution to quantify the parameter uncertainty, their covariance matrix suffices. This means that the input required by a DTF will consist of the parameter data: (μ_x , C_x) where μ_x is the parameter mean vector and C_x is the parameter covariance matrix. Consequently, the output of the DTF will consist of the data (μ_y , C_y).

Here, we make the additional assumption that the model behaves sufficiently smooth to use a simplified model for the uncertainty assessment, i.e. a linearization $\hat{y}$ of the model y around the parameter mean vector $\underline{\mu}_x^0$.

$$\hat{y}(\underline{x}) = y_0 + \sum_i \frac{\partial y}{\partial x_i}\left(x_i - \mu_{x,i}^0\right) \qquad (1)$$

Wit this assumption, the execution of the DTF can be very efficient:

The first time the DTF is executed, the linear model must be built, i.e. the value of y_0 and the differential sensitivities are assessed.

Each call of in the design cycle, including the first one, requires as input data the parameter mean vector μ_x and the parameter covariance matrix C_x at that stage. On the basis of the covariance matrix, the DTF calculates the variance $V_{\hat{y}}$ of the linearized model $\hat{y}$:

$$V_{\hat{y}} = E\left\{\left(\hat{y} - y_0\right)^2\right\} = \sum_i \sum_j \frac{\partial y}{\partial x_i}\frac{\partial y}{\partial x_j} C_{x,ij} \qquad (2)$$

as an estimate for the actual model variance V_y. To check whether y_0 is still a valid estimate for the mean model output y, given the new parameter values, the deviation Δ of the mean value of y from y_0 is estimated by:

$$\hat{\Delta} = E\left\{\hat{y}\right\} - y_0 = \sum_i \frac{\partial y}{\partial x_i}\left(\mu_{x,i} - \mu_{x,i}^0\right) \qquad (3)$$

If $\hat{\Delta} << V_{\hat{y}}$ we will assume that the change in the mean value of y with respect to y_0 due to the new parameter values is insignificant. Now, y_0 is no longer exactly equal to the mean of y but an acceptable approximation.

If, however, $\hat{\Delta} \geq V_{\hat{y}}$, y_0 can no longer serve as an approximation for μ_y . In that case, a new linearized model has to be built around the current parameter mean vector and the variance estimate has to be recalculated. At this stage, the DTF can decide on the basis of the variance estimate for y whether the calculation output should be released. In other words, if the variance has an unacceptable magnitude, the DTF can decide that storage of the results in the central database is unacceptable. If release of the output is deemed to be acceptable, the DTF outputs (y_0 , $V_{\hat{y}}$).

CONCLUSIONS

A novel approach has been introduced to deal with incremental performance analysis strategies in collaborative building design teams. Based on rigorous definitions of performance assessments and their metrics with included probability information, the approach enables to control which assessments become ‘available’ as the design information content grows. It also indicates if and when a prior assessment is invalidated by new information.

Future work is targeted at establishing the reverse links, i.e. to determine which performance indicators can be evaluated within a predefined level of uncertainty for a given incomplete design.

Much effort has to be put into the mapping of data between evaluation applications, keeping track of all correspondences of changed design information and the local parameters in the applications. The Combine approach is too crude in most cases. Other approaches (Eastman, 1996) will be used in future extensions.

The technique is presently being tested on overheating risk assessments in different stages as the design evolves.

REFERENCES

Augenbroe, Godfried. COMBINE Final report. CEC-DGXII report (1995)

Dekker, Sil. “A Desktop COMBINE interface for the building physics consultant” (in Dutch). MSc Thesis Report, TU Delft, 1995.

Eastman, C.M. “Managing Integrity in Design Information Flows”, Computer Aided Design, May 1996, 28:6/7, pp. 551-565.

Kalay, Yehuda E., “Performance-based Design”, Application of the Performance Concept in Building, Tel Aviv, Dec. 1996, pp. 4.1-4.13

Mahdavi, Ardeshir, “Computational Support for Performance Based Reasoning in Building Design”, Application of the Performance Concept in Building, Tel Aviv, Dec. 1996, pp. 4.1-4.13

Wit, Sten de, “Influence of modeling uncertainties on the Simulation of Building Thermal Comfort Performance, These Proceedings.