One aspect where there is general agreement is that the decision making process must not be automated to the extent that the designer becomes merely a provider of data, with the computer system providing the answer. Instead there has to be some form of co-operation between the computer system and the designer with the final decision being taken by the designer. Generally such systems go under the generic title of decision support.

The system that is the main focus of this paper is a Decision Support System (DSS) for the conceptual design of small to medium span road bridges (50m maximum span). The general basis of the system is heuristic substitution (Miles et al, 1995). The system itself has no in-built search engine but is sufficiently fast and easy to use to enable the designer to look quickly at a significant number of options. One of the main features of the system is the use of metaphors both in determining the form of the software itself and also in the user interface. As is shown below, the software allows a substantially enhanced level of accuracy and detail to be introduced into the conceptual design stage of the process. Given that this is where the major cost decisions are taken, such an approach can only be of benefit. The costing model uses the work of Poh & Horner (1995) on cost significance to provide a simple but yet accurate method of cost estimation.

2. METAPHORS – AN OVERVIEW

All IT users are familiar with the usage of metaphors in the user interface with, for example, menus from the restaurant metaphor, radio buttons from the car radio metaphor and of course the desktop metaphor with its folders. Interestingly to many, familiarity with such interfaces means that the purpose of the original metaphor is lost and all that remains is the terminology.

Metaphors are increasingly used in software because it is then possible for the user to map his/her existing knowledge onto that of the system. There is a number of possible ways in which metaphors can help a user in their usage of a system, viz:-

- *Identification*: What is this?
- *Transition*: Where have I come from; where am I going?
- *Orientation*: Where am I?
- *Choice:* What can I do now
- *Demonstration:* What can I do with this?
- *Explanation:* How do I do this
- *Feedback:* What is happening?
- *History*: What have I done?
- *Interpretation*: Why did this happen?
- *Guidance:* What should I do now?

Metaphors when used in computer systems are generally used to enable people rapidly and easily to build a mental model of how the system works. Building user interfaces using everyday concepts helps to make computers seem more a part of our everyday world and thus enhances their acceptability, although as Stubblefield (1998) points out there has to be a reasonable match between the metaphor and the application domain. Also using common concepts in user interfaces tends to speed up the learning process, thus reducing the need for training. As Flemming et al (1997) show, this is not always beneficial but for most well designed software, interface metaphors can give substantial gains in productivity (Kim, 1999).

Other aspects of software besides the user interface can also usefully employ metaphors. In many object oriented systems, metaphors can be used to devise the structure of the software so that it directly reflects that of the application domain (Meyer, 1988). For the software which forms the main subject of this paper, a similar approach has been used but the following description concentrates on the user interface.

It is inevitable that the employment of software in design offices will increase, both in terms of more intensive usage of current types of software and with the addition of new forms of software support. As the number of software systems increases it is likewise inevitable that much of it will only be used occasionally, especially for conceptual design where because of the nature of the task, the software will be problem specific. In such circumstances, software which requires users to undergo training will impose a significant cost overhead on the design process. This will not be an economic proposition for many design practices and therefore there will be a growing demand for software which can be used in an intuitive manner with little or no prior training. In such circumstances the use, in the user interface, of metaphors which help the user to map his/her current world view onto that of the software is inevitably going to be of increasing significance. So far the choice of metaphors for user interfaces has tended to be made in a somewhat opportunistic manner (e.g. restaurants, car radios). For design specific interfaces, it makes sense to use metaphors which directly reflect the design process itself. The work described below has been guided by this philosophy.

3. COSTEST

The system which forms the subject of this paper is known as COSTEST. As explained above it is a system to assist with decision making during the conceptual design of small to medium span bridges. The software is written in Visual C++. A schematic of the system architecture is given in figure 1 which shows the essential features. The aims of the system are :-

- To devise a system which can be used in an intuitive manner thus avoiding the need for training;
- To allow more options to be considered in much greater detail than is possible with current conceptual design techniques;
- To substantially increase the level of accuracy and detail in conceptual design. This is particularly useful when the design is part of a competitive bid;
- To allow the designer to control the decision making;
- To achieve the above in the sort of time that is currently allocated to conceptual design;
- To allow the user access to the details of all calculations and methods used in the system so they can evaluate and hence have confidence in its accuracy.

This last feature is one which in evaluations with practising designers has been stressed many times. They are ultimately responsible for the design and hence they need to be able to undertake some form of check, albeit not necessarily an exhaustive one.

As figure 1 shows COSTEST has four major components, these being :-

1. The Site/Bridge arrangement which deals with the data relating to site topography, the obstacle which is to be bridged and the basic dimensions of the bridge in terms of number of spans, span length, etc.

2. The Design Offices; these contain a number of sub-functions as shown. The purpose of these is discussed in more detail below.

3. The detailed calculations, this is where the relevant design code calculations are contained. COSTEST contains both Eurocode and British Standards information, the designer being able to choose between the two options

4. The client cost report; this component assembles the findings from the various processes within COSTEST and produces a client report.

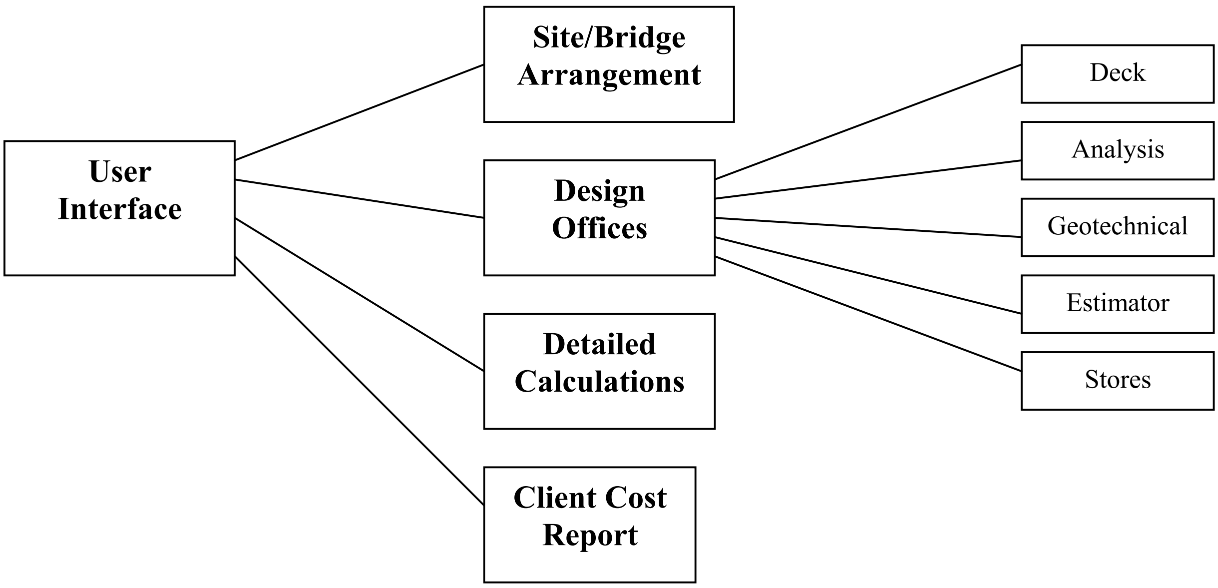

Figure 1. System Architecture

4. THE USER INTERFACE

It is in the user interface that the most noticeable impact of the metaphors within COSTEST occurs. The arrangement of the main interface screen is shown in figure 2. This has been carefully designed to try and bridge the gap between user goals and intentions and the system state. The underlying concepts behind this are based on the theory of action.

This attempts to deal with the conflict between the user's goals, which are psychological and the system's mechanisms and states which are expressed in physical terms. For a good user interface the discrepancy between the psychological and physical variables should be minimised. From a purely psychological point of view there is very little published work that one can use as a guide to achieve this minimisation, possibly the best available being the contributions of Miller et al (1960), Powers (1973) and Card et al (1983). The gulfs between the psychological state of the user and the system's physical state are illustrated in figure 3. The gulf of execution is bridged by making the commands and interface mechanisms of the system as close as possible to the intended user's thoughts and goals. The evaluation gulf is minimised by making the interface display a realistic model of the problem and its related features. For the system, the interface style is governed by its creator. For the user, the interface can become less of a barrier with training and experience but with a well designed interface, the initial barrier should be minimal.

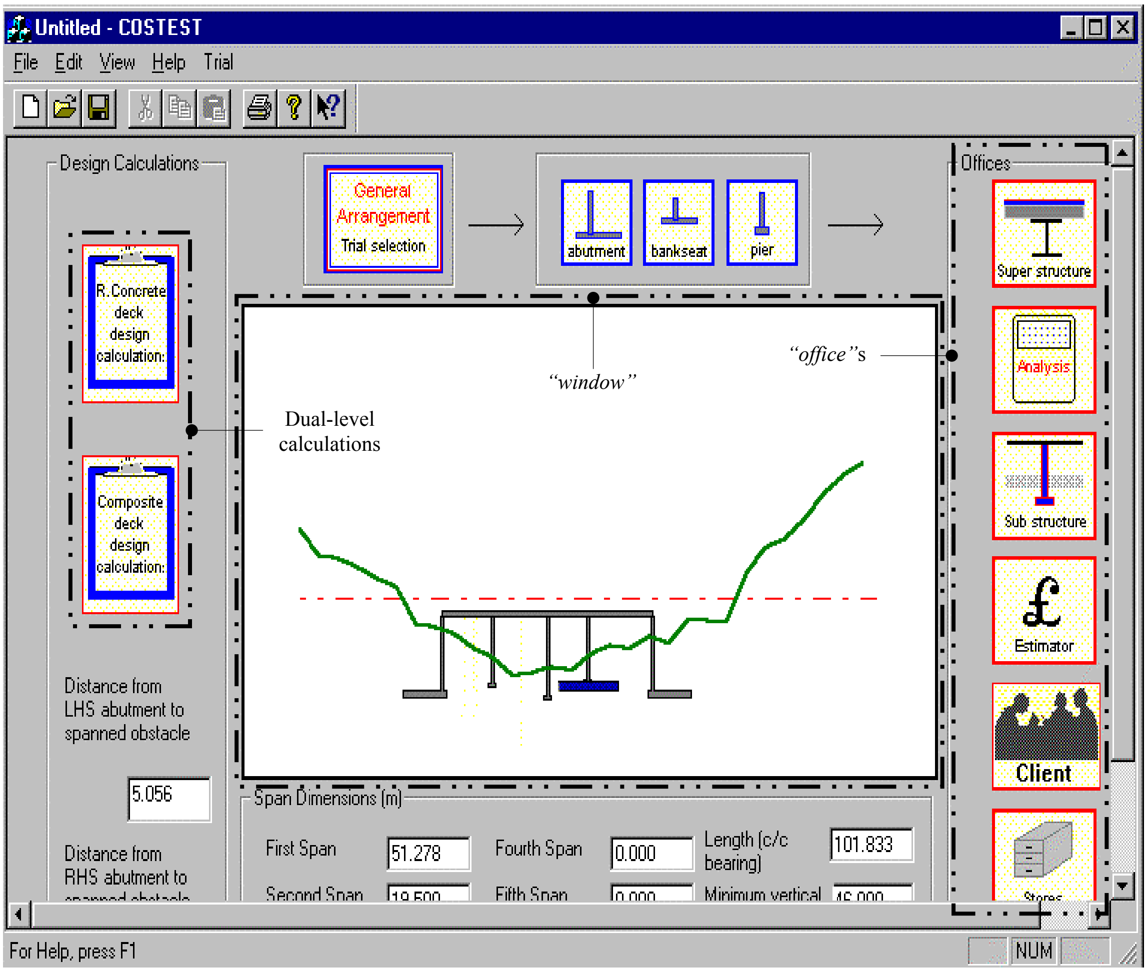

Figure 2 . The COSTEST main user screen. The "office corridor" runs along the right hand side of the form view. Within the corridor the individual "office"s are to be found. The left-hand side is showing the dual-level calculations that are accessed by clicking the buttons containing clipboard icons.

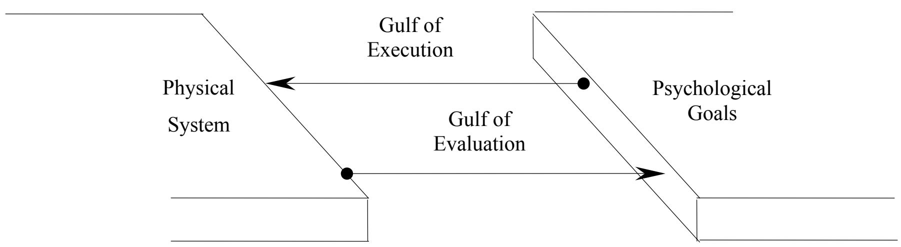

Figure 3: The gulfs of Execution and Evaluation as described by Norman (1986)

There are various ways of minimising the difference between the user and the system states. Two of the most common and powerful are direct manipulation interfaces and the use of metaphors. Direct manipulation interfaces allow the user and the system to interact graphically, thus avoiding abstract methods such as numerical data input. Also direct manipulation interfaces are in themselves extensions of the metaphor concept, providing a conceptual model of the problem and thus reducing the cognitive demands on the user. As is shown in figure 2, in COSTEST, the bridge and the obstacle to be spanned are shown visually in cross section in a form which closely relates to reality. All parts of the bridge can be directly manipulated to stretch and shrink its dimensions so that the user is able easily to fit the bridge to his/her mental picture of what is required. Direct manipulation interfaces allow the user to apply their prior knowledge of the subject to their usage of the system.

For direct manipulation interfaces, the discrepancy between the psychological goals and the physical state can also be expressed in terms of the following two distinct aspects :-

- Distance
- Engagement

Distance is a measure of the discrepancy between the user's thoughts and the physical requirements of the system. Engagement refers to the qualitative impression that one is directly manipulating the objects of interest. There are two metaphors for the nature of human-computer interaction, these being the conversation metaphor and the model world metaphor. The former relates to systems in which the user and the system have a conversation about an assumed but not explicitly represented world. This style of interface can impose a significant cognitive load on the user. For model world interfaces, the interface is itself the world in a metaphorical manner. The user can interact with the world and there is no intermediary between user and the world and thus the user is directly engaged. This obviously reduces the cognitive effort required by the user.

4.1 Initial Data Input

One of the main uses of metaphors is to help users employ the facilities of the software to navigate through a decision space. Kim (1999) shows, using the example of a cyber shopping mall, that spatial metaphoric aids lead to easier navigation and a similar approach has been used in COSTEST.

The main interface of COSTEST is shown in figure 2. The user starts their interaction with the system by accessing the General Arrangement. This brings up a screen which enables them to insert the relevant topographical data for the site and the obstacle which is to be bridged (shown as a black rectangle in figure 2). Following this the user then chooses the form of end support (i.e. abutment, bankseat) and inserts any piers which may be required. The system automatically inserts a default vertical alignment for whatever is to be carried over the obstacle (shown as a chain dashed line on figure 2) which can be moved to the appropriate location by the user. To save on programming effort the current version only provides a horizontal alignment. If the system is developed further this will obviously have to be changed.

As the user builds up the design, the section of the bridge in the main window is appended appropriately and the dimensions at the base of the screen are updated. Should the user wish to change any of the dimensions then the mouse can be used to alter structural shapes or drag components to a new location.

4.2 The office corridor

Once the above section is complete, the arrows towards the top of the user interface guide the user to the office corridor. The metaphor of an office corridor has been used to provide the user with a concept of linked activities and the metaphor has been extended so that the office is a design office, thus directly mapping onto the user's cognitive model of the design process. Each icon within the corridor then forms a part of a whole process and it is intended that the user should visit each office, although the "stores" (the last icon) as its name implies only needs to be visited if something extra is needed.

The first office on the corridor is the S*uperstructure*. Currently COSTEST contains the facilities to allow the user to design 2 types of deck, reinforced concrete and steel/concrete composite. Obviously for a complete design tool more options would be required. On entering the office, the user chooses which type of deck form is required. The system provides a default deck profile which the user can alter to the correct size by dragging in the required direction. Likewise, for example the number of steel beams can be altered and the spacing changed as required. Additional facilities allow the user to apply surfacing and pavements (sidewalks). The dimensions of these can again be altered by dragging. The finished profile will then look something like figure 4.

The Universal Beams for the steel/concrete composite are selected from the stores. If the user is unhappy with the selection, the stores can be searched for a more suitable beam.

The dimensions from the *Superstructure* office are passed to the A*nalysis* and the E*stimator* offices. The *Analysis* office takes the deck dimensions and calculates the self weight. Material densities are taken from the data in the stores. Should the user wish to change these then the stores have to be entered and the relevant combo boxes amended. Live loading cases are also calculated and then the system activates an FE package which analyses the deck, the results being fed back to COSTEST for further checking later in the process.

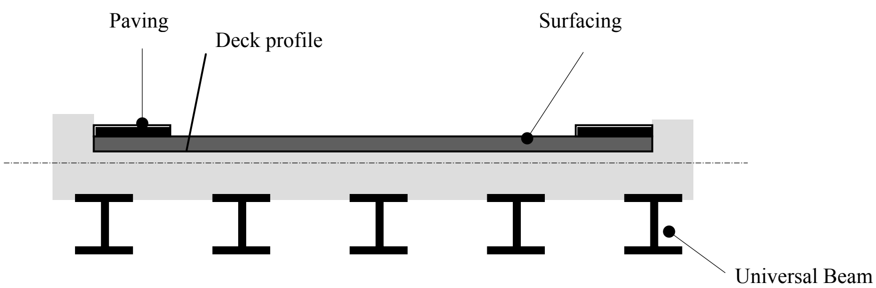

Figure 4: A typical composite deck. This diagram represents the direct manipulation facility of the superstructure "office". The deck's components are defined and dragged in all directions to suit the designers' requirements

The next office on the corridor is the substructure. This uses heuristics to provide approximate sizes for the substructure and foundation elements. In future it is hoped to link the foundation and substructure design system of Cadogan (1998) to COSTEST to remove the need for the heuristics. This is particularly desirable as for a typical bridge, 55% of the costs are due to the substructure and foundations (Foley, 2000). As currently configured COSTEST can only deal with spread footings and as with all other aspects of the system, the user can resize the components graphically using a direct manipulation style of interface.

The *Estimator* office provides user access to the cost modelling facilities of COSTEST. These are based on the concept of cost significance (Poh & Horner, 1995) and provide a fast and accurate method of estimating costs while avoiding the effort of taking off full quantities. The basis of the method is the concept that for any given structural form, 20% of the components within the structure account for 80% of the costs. So if one analyses given structural forms to determine the cost significant items, then to cost similar structures, it is only necessary to work out the quantities of the cost significant items, apply costs to these and then divide by 80%. Foley (2000) examined the validity of this approach for a number of bridge types and found that it works well with an accuracy of +/- 5% which is substantially better than current conceptual design costing models. COSTEST allows the user to look at a number of solutions to a given problem and these are collected in the *Estimator* office.

When the user tries to access the *Client* office, he/she is first directed to the *Design Calculations* on the left hand side of the screen. This has to be done for each design option which it is wished to pass from the *Estimator* to the *Client* office but as the *Client* office is next on the corridor, its purpose will be described here. This office collects a series of design solutions which are chosen by the designer. The choice is based on the concept of *bias*; that is the user can choose a number of options which exhibit particular features. This is typical of real design where for instance a designer may wish to employ a particular form of deck construction. COSTEST allows 3 types of solution, no bias, superstructure bias and substructure bias. No bias is the cheapest of all the feasible options considered, superstructure bias is a preference towards a particular type of deck, etc (the preference could for example be on the basis of aesthetics or the need to build over an existing road). COSTEST provides a client cost report for each of the types of bias which have been indicated as being desirable by the user (Foley, 2000)

The final office on the corridor is the *Stores*, which as has already been indicated contains all the default values used in the system for material properties, structural sections, etc.

4.3 Dual Level Calculations

The dual level calculations contain all the relevant formulae from the British Standards and Eurocodes for bridge deck design (lack of time prevented the inclusion of a section for substructure and foundations). They are called dual level because two levels of detail are available to the user. The top level is in summary form only with no information regarding how the answers were obtained. However, an often stated requirement that arises in the evaluation of design systems is that designers need to understand how a given answer was reached so at the lower level, COSTEST provides a full description of the calculations and associated procedures.

5. EVALUATION

An unfortunate feature of much of the research on computer support for design is the lack of any form of evaluation or assessment of fitness for purpose. Without this, it is difficult to assess objectively the value of a given approach. COSTEST was developed in conjunction with practising designers and many of the features in the system were developed because of their advice. The resulting system was then evaluated by these designers and some others who had not participated in the development process. Details can be found in Foley (2000). Generally the feedback from the evaluators was positive although one evaluator was very critical.

So far as the style of the user interface was concerned, the evaluators' reaction was interesting. In previous evaluations (e.g. Moore, 1991) the bulk of the comments made by the evaluators have been about the user interface and have been critical. With COSTEST, the absence of comments about the interface was marked. Users just seemed to accept it and find it intuitively easy to use. Admittedly there is no positive proof of this, only a lack of comments but experience of past evaluations has shown that evaluators are quick to focus on unsatisfactory aspects because they are being asked to critique the software. The evaluators also found the costing model useful and appreciated the benefits of the greater accuracy offered.

6. CONCLUSIONS

It is generally accepted that the use of metaphors and direct manipulation interfaces allows users to interact with software in a manner which requires less cognitive effort than other styles of interface. A user interface for a conceptual design system has been described which is based on the metaphor of the design office. This format has been chosen to help designers navigate through the various features of the system. It is postulated that in future, designers will posses such a multiplicity of software tools that training will be uneconomic and the format of the tools will therefore have to be such that they can be used in an intuitive manner.

Also the software described improves the accuracy of the conceptual design process by increasing the level of accuracy and presenting a more rigorous form of costing model than that which is currently used.

The evaluation of the system indicates that the above features are acceptable to potential users.

7. ACKNOWLEDGEMENTS

The authors are grateful for the input to this project from a number of practising bridge designers, in particular; Dr R. Francis of James & Nicholas; Sir Alexander Gibbs & Partners; Mr P. Harris of W.S. Atkins & Partners; Mr P. Jenkins of Hyder Consulting and Mr M Craddy of Sir William Halcrow and Partners.

8. REFERENCES

Cadogan J. (1998). Constraint based decision support for conceptual design, PhD thesis, Cardiff School of Engineering, Cardiff University.

Card S.K., Moran. T.P. & Newell A. (1983). The psychology of human-computer interaction. Hillsdale , NJ: Lawrence Erlbaum Associates.

Flemming U, Bhavnani S.K. & Bonnie E.J. (1997). Mismatched metaphor: user vs system model in computer-aided drafting, Design Studies, 18, 349-368.

Foley A. (2000). Decision Support for Conceptual Design and Costing, PhD thesis, Cardiff School of Engineering, Cardiff University.

Khajepour S. & Grierson D.E. (1999). Filtering of Pareto-Optimal trade-off surfaces for building conceptual design, in Topping B.H.V. & Kumar, B. (eds) Optimisation and Control in Civil & Structural Engineering, Civil-Comp Press, Edinburgh (UK), 63-70.

Kim J. (1999). An empirical study of navigation aids in customer interfaces, Behaviour & Information Technology, 18 (3), 213-224.

Meyer B. (1988). Object-oriented software construction Prentice-Hall, Englewood Cliffs, NJ, USA.

Miles J C, Moore C J, Price G, (1995). Design Costing Models: An Application of Heuristic Substitution, Computing Systems in Engineering, 6(6), 521 - 531.

Miles J.C., Sisk G.M. & Moore C.J. (1999). The conceptual design of commercial buildings using a genetic algorithm in Topping B.H.V. & Kumar, B. (eds) Optimisation and Control in Civil & Structural Engineering, Civil-Comp Press, Edinburgh (UK), 17-24.

Miller G.A., Galanter E. & Pribram K. (1960). Plans and the structure of behaviour. Holt, Rinehart, & Winston, New York:, USA.

Moore C.J. (1991). An expert system for the conceptual design of bridges, PhD thesis, Cardiff School of Engineering, Cardiff University, 326pp

Norman D.A. (1986). Cognitive engineering, in Norman, D.A. and Draper, S.W. (eds), User Centred System Design: New perspectives on human-computer interaction, Hillsdale , NJ: Lawrence Erlbaum Associates, 31-61

Poh P. S. H. & Horner M. W. (1995). Cost Significant Modelling – its potential for use in south-east Asia. Engineering Construction and Architectural Management. 2(2), 121-139

Powers W.T. (1973). Behaviour: The control of perception. Aldine, Chicago, USA.

Rees D.W.G., Miles J.C. & Moore C.J. (1995). A second generation KBS for the conceptual design of bridges, in Civil-Comp (1995), ed Topping BHV, Civil-Comp Press,17-24.

Smith I.F.C. (1996). Interactive design- Time to bite the bullet, in Kumar, B (ed) Information processing in Civil & Structural Engineering design, Civil-Comp Press, Edinburgh (UK), 23-30.

Stubblefield W.A. (1998). Patterns of change in design metaphor: A case study, in Proc Human Factors in Computer Systems '98, 73-80.