EVALUATION OF A HIGHER EDUCATION SELF-LEARNING INTERFACE

A. BENNADJI
School of Art Architecture and Design, The Robert Gordon University, Garthdee Road, Aberdeen, AB10 7QD, U.K.
a.bennadji@rgu.ac.uk

AND

A. BELLAKHAL

Institut d'Architecture, Universite Mohammed Khider, Biskra, Algeria
belakehal@yahoo.fr

Abstract: This paper is a follow-up to a previous paper published in ASCAAD 2004 (A. Bennadji et al 2005). The latter reported on CASD (Computer Aided Sustainable Design) a self-learning educational interface which assists the various building's actors in their design with a particular attention to the aspect of energy saving.

This paper focuses on the importance of software evaluation and how the testing is done to achieve a better human-machine interaction.

The paper will go through the summative evaluation of CASD, presents the output of this evaluation and addresses the challenge facing software developers: how to make an interface accessible to all users and specifically students in higher education.

1. Introduction

Computational tools have the potential to provide an effective means to support design decision makers. Thereby, the most important factor is not software acquisition, but the effort needed for learning and using it. "The extent of required time and effort is believed to be one of the main hindrances toward the pervasive use of computational building performance assessment tools by designers: Currently, modelling applications are mostly used, if at all, in the later stages of design and by specialists, rather than architects" (Ardeshir Mahdavi et al, 2004).

Few studies have explicitly dealt with the ascertainment and quantification of the actual effort needed to understand, master, and apply computational building evaluation tools.

In this paper, we report on the collaborative work that took place between the School of Architecture at The Robert Gordon University in Scotland and the Institute of Architecture at The Mohammed Khider University in Algeria. The collaborative idea was to build bridges between developed and developing countries in terms of Information Technology and how to apply it in an effective and economical way.

The project's aim was to develop an (online) interface, called Computer Aided Sustainable Design (CASD) that will guide the user through a series of input screens to allow him/her to describe the building and select various environmental options. The interface is to be accessed by students at the School of Architecture to allow them to integrate environmental aspects in their designs.

A selection of building specifications from a database is used to generate numerical information that will be used by a calculation engine that uses sophisticated thermal models (Brown, J. and Palmer, J.: 1991).

One of the project's main objectives was to develop a tool that assists in reducing the number of design cycles (revisions). This fits with government policy to improve the effectiveness of the design process and to make procurement of buildings cheaper and more efficient (A. Bennadji et al ,2002).

In the past, people have resisted the use of such simulation programs for a number of different reasons. They felt that it was time consuming to input the necessary data and that the program was not user friendly enough. The team aimed to reduce the amount of input required for each building with the use of extensive, intelligent defaults. To make the program more user friendly, an implementation of a pictorial based input system was used with a minimal amount of data required to describe a building. Users also felt that programs could not be trusted, perhaps because of their complexity and the fact that they could not understand them.

The idea of this project at industrial level was developed in a previous project "EnergySave" (D.Bouchlaghem et al, 2005). At this later stage, CASD software is developed for purely educational purposes as we thought useful to allow students in architecture and building-related courses to understand the extent of environmental issues while designing a building.

As the effort and time required to learn and understand a software is extensive (A. Mahdavi, et al, 2003. and K.P. Lam, 1999), we opted for HTML as it seemed to be the most popular, not only at university level where all students are used to the Internet but also at professional level as most daily common needs are now achieved by using the Internet.

Therefore, the interface for CASD software will be in HTML format, and all the evaluation study will be based on this specific format and platform.

1.1 PROJECT'S OUTCOMES

The project attempts to produce an interface for students in architecture and tries to give guidance for interface developers in terms of assessment and user's appreciation of the implemented version. CASD was an attempt to the development of fully working software that can simulate buildings' energy consumption and CO2 emission. CASD was then used, in this collaboration with developing countries, as an educational program where students can enter their building's specification and then compare their design with a benchmark building of a similar description and specification. The comparison will be purely in terms of sustainability. The student will then be able to study in details the specifications of the benchmark building, so that he can improve his/her design in order to reach a better environmental impact. This version of CASD will not include any simulation by a third-party calculation engine, but only a comparison from a background database of buildings' performances.

1.2 PAPER'S OBJECTIVES

The paper aims to address software evaluation at its different development stages. It is essential for software designers to reach users' desires. As a scientifically recognised research method, the final product is not necessarily the same as what the designer planned it to be, but rather what the users feel more comfortable with and more adapted to their needs. The question: "why are we evaluating?" becomes more obvious after the failure of many software projects which didn't receive a great welcome from users. The simple answer to this question is: to help improve the system. The system function will not be addressed in this paper. Generally it is the IT expert who deals with the technical side of software development. Thus, the paper will focus on the software's usability.

2. CASD

CASD has been developed using HTML in the Macromedia Dreamweaver MX environment. ASP (Active Server Pages) code was used to link the Web pages with a MS Access Database containing all the data entered by the user, and other default (standard) values related to the building regulations as well as benchmark projects. ASP allows one to dynamically edit, change or add any content of a Web page, respond to user queries or data submitted from HTML forms, access any data in the database and return the results to the

web browser. CASD (see Figure 1) is comprised of a sequence of screens (Web pages) in which the user can freely move forward or backward.

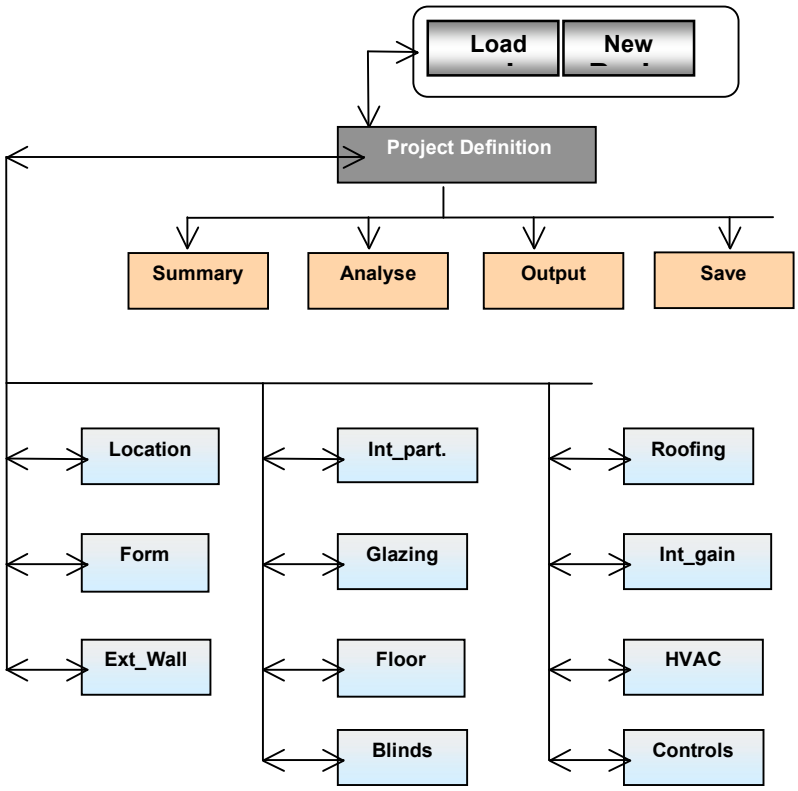

Figure 1. CASD interface

The user is free to start from any screen after completing the first one which relates to the project description. The user can stop at any stage, save his/her work and resume at some other time. Figure 2 shows an example screen.

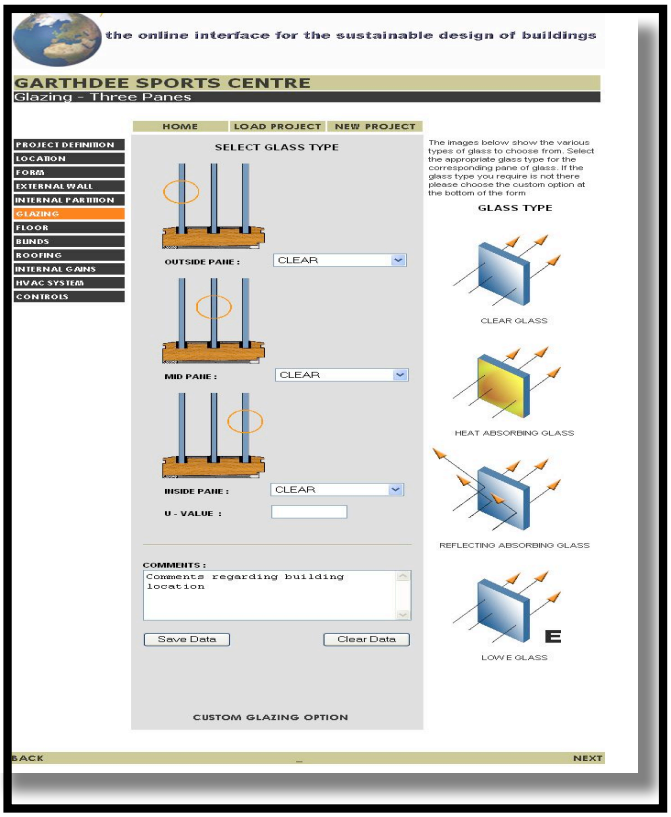

Figure 2. A typical CASD screen

3. Evaluation Methodology

Evaluation theories and methods were widely disseminated for different research areas for decades. However, not enough attention was given to software evaluation research methodologies. The fact that software development is a new exploration might be an explanation of this lack of research in this area (Molich, R., et al., 1990. Nielsen, J., et al, 1993. Nielsen, J. 1992. and Nielsen, J., 1994).

We were not deeply engaged in the evaluation while designing the software, as it is for students use only and we worked on similar software intended for professional users where a proper evaluation by professionals was done (D. Bouchlaghem., 2005). It appears after a first try by students, that we are dealing with a different type of users, who are not necessarily IT users.

Therefore, we found ourselves obliged to evaluate the software in 3 steps if we want a great welcome of CASD by the students' community.

The diagram in Figure 3 summarises the evaluation methods from which our methodology of evaluation is inspired. It gives an overview of different types and stages of evaluation. Our evaluation stages were based on the interpolation of this method to the use of CASD for teaching and learning. This strategy circumplex is drawn from (McGrath, 2000).

4. Background to the Method's Application

Due to the conference character and the attendees' backgrounds, we thought it would be wise to give an overview of systems' evaluation methods and how a specific method was applied to CASD.

Many methods for software evaluation have been developed in the last 3 decades. Three main levels are used, Formative, Summative and Integrative levels[3]:

- Formative: For the designer of the system
- Summative: For the customer of the system
- Integrative (Draper, S. W., et al. 1996) : For neither directly – intended to help embed a system effectively.

[3] The distinction between formative and Summative evaluation is due to the philosopher Michael Scriven. It is a theme which runs right through this paper.

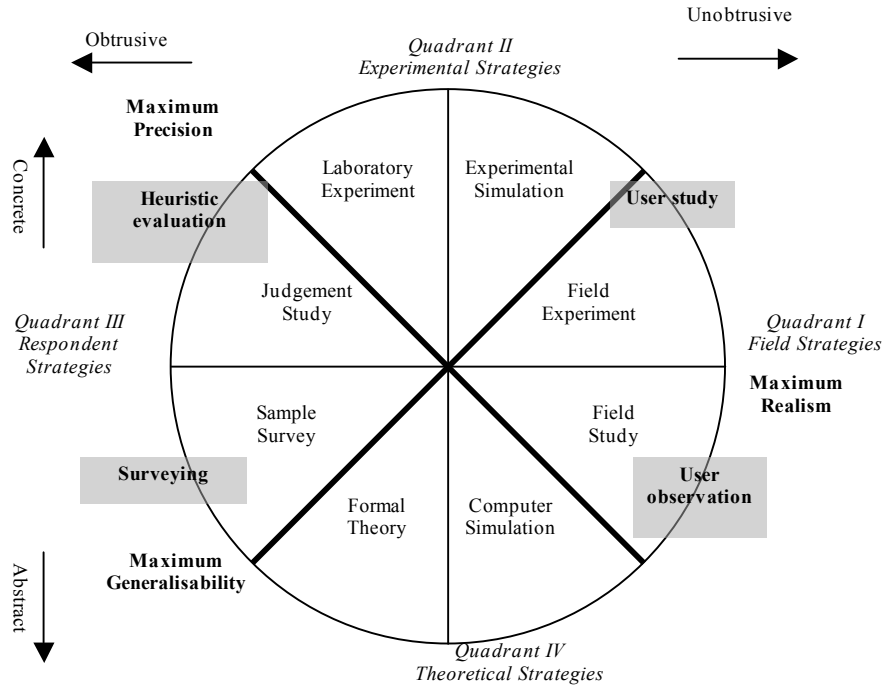

Figure 3. The strategy circumplex

Within these various levels of evaluation; different methods are available to test software's effectiveness. Due to the specific character[4] of the software to be evaluated, three methods of evaluation will be used:

- Experiments: user testing.
- Observation: protocol analysis.
- Surveying: questionnaire.

4.1 FORMATIVE EVALUATION

Formative evaluation took place at different realisation stages:

1. Before interface design: The design is iterative, and design decisions are tested at key development stages. This step will influence the final product as feedback during conceptual stage will assist design decisions.
2. Early design stages: this evaluation is based on task completion times to determine the most effective method for each operation. This early stage can be long or short depending on the software aim and utilisation and outcomes.

[4] In architecture education a huge emphasis is given to the visual representation of objects, and students refer to their visual memory to understand phenomena by analogy.

3. Later design stages: at a final design stage the evaluation focuses on identifying user difficulties with the interface and aims to address essential questions such as,
    - Are users able to control the interface?
    - Do they understand the interface's feedback?

4.2 SUMMATIVE EVALUATION

The summative evaluation takes place after the software completion. It can be considered as a final evaluation, as feedback at this stage can be positive and the software can start to be used. Unfortunately this doesn't happen very often and other steps of evaluation may be needed.
The characteristics of this evaluation step are:

1. The summative evaluation is not generally iterative and is used to verify the software success.
2. Based on usability metrics: e.g., time to complete benchmark tasks, existence of documentation.
3. Tests for proper functioning of system:
    - Often uses quantitative measures
    - Conducted after design completion

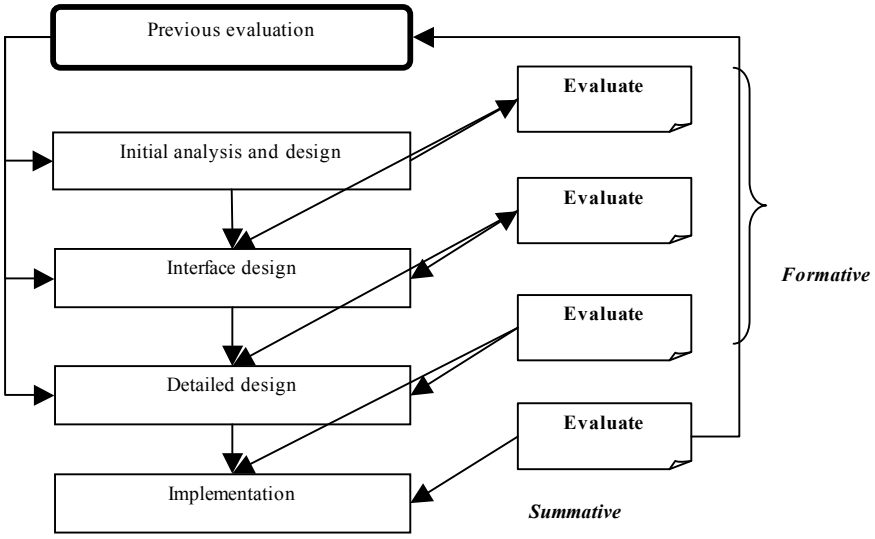

Figure 4. Evaluation as part of the design process

5. The Application of the Evaluation Method

The theoretical part above is an explanation of the long process throughout CASD's development. This paper will only address the Summative evaluation process and its results.

5.1 EXPERT EVALUATION

The expert evaluation is generally carried out using experts and professionals in the field. By analogy; and because the software is dedicated to teaching; the expert students that we have used to evaluate CASD were students from honours and Diploma year. We assume that their experience in using software packages as well as the Internet is much wider than any students in earlier stages. This professional evaluation method was achieved using heuristic evaluation.

The process of heuristic evaluation has been carried out as follows: each evaluator inspected the interface individually. They went through the interface several times, inspecting the dialogue elements and compared the dialogue with a list of usability principles (the heuristics – basic guidelines). Findings were aggregated afterwards (see Figure 5).

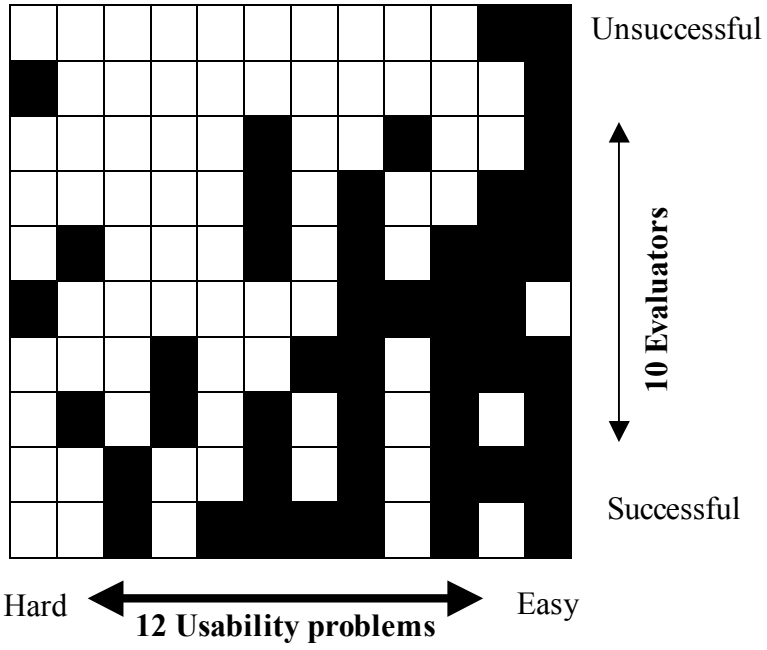

Figure 5. Evaluators' responses to found usability problems in CASD

As the heuristic evaluation method is difficult for a single individual to do because one person will never be able to find all the usability problems in an interface, we have used 10 students from higher stages to evaluate CASD. Figure 5 shows 10 evaluators who found 12 usability problems. Each of the black squares indicates the finding of one of the usability problems by one of the evaluators. The Figure clearly shows that there is a substantial amount of non overlap between the sets of usability problems found by different evaluators. It is certainly true that some usability problems are so easy to find that they are found by almost everybody, but there are also some problems that are found by very few evaluators. Furthermore, one cannot just identify the best evaluator and rely solely on that person's findings. First, it is not necessarily true that the same person will be the best evaluator every time. Second, some of the hardest-to-find usability problems (represented by the leftmost columns in Figure 5) are found by evaluators who do not otherwise find many usability problems. Therefore, it was necessary to involve multiple evaluators in this heuristic evaluation.

5.2 SURVEY

A survey was carried out, but the students were from early stages; and hence different from expert students used for the heuristic evaluation. Only students from first year were discarded due to their lack of computer use in their design. CASD was installed on the university's server as well as the questionnaire. Web-based surveying is seen to be more effective as it reduces cost and time of data entry

5.2.1 The Questionnaire

The questionnaire (see Table 1) was comprised of 17 closed questions and 1 open question. Closed questions were used in order to be analysed statistically. They were ordered in such a way that the key criteria come first. The last open question was there to allow the questionnaire-takers to give general comments on CASD's usability.

TABLE 1: Questionnaire submitted to students

Criterion		Rating
1.	Is it easy to start the program?	Yes/No
2.	Is the use of interface easy to understand? (For example, is the screen layout clear and easy to interpret?)	Yes/No
3.	Is the visual message clear enough on all pages globally?	Yes/No
4.	Is it easy to navigate through the program?	Yes/No
5.	Are the icons used to assist navigation (e.g. back to previous pages or main page, exit) clear and intelligible?	Yes/No
6.	Is it always clear to you which point you have reached in the program?	Yes/No
7.	Are the included pictures, relevant, and aid understanding?	Yes/No
8.	Are the choices given to you helpful?	Yes/No
9.	Does the interface look more serious compared to any ordinary website?	Yes/No
10.	Is it worth having feedback if you get something wrong?	Yes/No
11.	Do you like the fact that some of the spaces that you have to fill in are filled automatically for you?	Yes/No
12.	Do you like the neutral background of the interface?	Yes/No
13.	Can the user easily quit something that is beyond his/her ability?	Yes/No
14.	Do you prefer to fill only the part that you are expert on and leave the rest to other experts?	Yes/No
15.	Do you feel able to navigate it you start from other than the 1st page?	Yes/No
16.	Does the site contain what you expected, e.g. as indicated in its title or URL?	Yes/No
17.	Is the interface logo communicative enough?	Yes/No
18.	*Please write in few comments that can improve the software.*	

5.2.2 Analysis of Survey Results:

The results of the survey are summarised in Figure 6 and are self-explanatory[5]. The diagonal in Figure 6, shows a limit between the best and worst scenarios.

[5] Note that the answer yes for question 18 represents the number of users who have given comments.

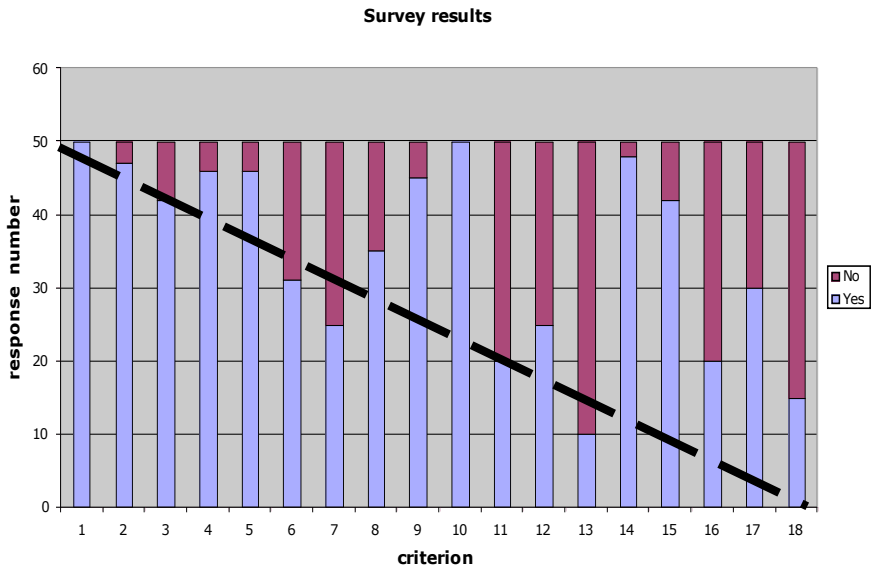

Figure 6: Questionnaire's results

Figure 6 shows that a majority of evaluators rated the key criteria (from 1 to 10) positively, as shown by the diagonal. The survey's verdict was satisfactory overall, and allowed us to pin point areas of further improvement. In that sense it was complementary to the heuristical evaluation.

6. Conclusion

Designing software interfaces has become a very challenging and competitive area. Clients do not only look for software that does the job, but they also want it to be widely accessible. The only way to reach this objective is to design a user friendly interface that maximises the user's acceptance of the software.

As such, Computer-based learning packages are no different, and extreme care should be devoted to evaluating their usability. In this paper, we focused on how such an evaluation can be conducted using well established methodologies and techniques in the field of software evaluation.

The aim is to have an impact on teaching methodologies where the computer is increasingly playing a big role. The tendency now is towards encouraging students to be more independent and to take more responsibility for their learning experience, and this can only be achieved if user-friendly self-learning computer packages are made available to them.

References:

Baecker R. M., Grudin J., William A. S. Buxton & Saul Greenberg (Editors). *Readings in Human-Computer Interaction: Toward the Year 2000* (Second Edition). Los Altos, CA: Morgan-Kaufmann Publishers, 1995. ISBN 1-55860-246-1.

Bennadji A., Ahriz H. et Alastair P, Computer Aided Sustainable Design, *Proceeding of E-Design in Architecture Conference*, 22-24 Feb. 2005, Dhahran, Kingdom of Saudi Arabia.

Bennadji.A., Bouchlaghem.N.M., Loveday.D., Holmes.M., Mitchell.T.K., van Zyl R.:2001, "Engineering Assessment of Building Design option at Sketch Design Stage-EnergySave- *", In The 18th International Conference on Passive and Low Energy Architecture, PLEA 2001*, Florianopolis- Brezil, 7 to 9th November, pp.931-933-. ISBN: 85-901332-4-9.

Bennadji.A., Bouchlaghem.N.M., Loveday.D., Holmes.M., Shaffery.J., van Zyl.R., Hallam. A.: 2002, Friendly Interface and Data Transfer for Engineering Assessment of Building Design Option at Sketch Design Stage -EnergySave-, *In Advances in Building Technology,* Vol II, -Hong Kong- China 4 to 6th December 2002, pp.1209-1215, ISBN: 0-08-044100-9.

Bouchlaghem N B, Holmes M, Loveday D and Bennadji A (2005) The engineering dimension of nD modelling:, *ITcon Vol. 10, Special Issue From 3D to nD modelling* , pg. 17-25.

Brown, J. and Palmer, J.: 1991, Occupancy profiles and incidental CIBSE guide: 1986, *The Chartered Institution of Building Services Engineers*, London.

Draper, S. W., Brown, M. I., Henderson, F. P., & McAteer, E. (1996). Integrative Evaluation: An Emerging Role for Classroom Studies of CAL. *Computers and Education*, 26(1-3), 17-32.

Jeffries, R., Miller, J. R., Wharton, C., and Uyeda, K. M. 1991. User interface evaluation in the real world: A comparison of four techniques. *Proceedings ACM CHI'91 Conference* (New Orleans, LA, April 28-May 2), 119-124.

Lam KP, Wong NH, Feriady H. A study of the use of performance-based simulation tools for building design and evaluation in Singapore. *Proceedings of the sixth IBPSA conference*, Kyoto, Japan, 1999. p. 675–82.

Mahdavi A, Feurer S, Redlein A, Suter G. An inquiry into the, building performance simulation tools usage by architects in Austria. In: *Augenbroe G, Hensen J, editors. Proceedings of the eight international IBPSA conference*, *vol. 2,* Eindhoven, Netherlands, 2003. p. 777–84, ISBN 90 386 1566 3.

Mahdavi A., El-Bellahy S., Effort and effectiveness considerations in computational design evaluation: a case study, *Building and Environment*, 40 (2005) 1651- 1664

McGrath J., Methodology Matters: Doing Research in the Behavioural and Social Sciences, *APA edition*, ISBN: 1-59147-053-6 October 2003

Molich, R., and Nielsen, J. (1990). Improving a human-computer dialogue, *Communications of the ACM* 33, 3 (March), 338-348.

Nielsen, J. (1994). Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), Usability Inspection Methods. *John Wiley & Sons*, New York, NY.

Nielsen, J. 1990. Paper versus computer implementations as mockup scenarios for heuristic evaluation. *Proc. IFIP INTERACT90 Third Intl. Conf. Human-Computer Interaction* Cambridge, U.K., August 27-31, 315-320.

Nielsen, J. 1992. Finding usability problems through heuristic evaluation. *Proceedings ACM CHI'92 Conference,* Monterey, CA, May 3-7, 373-380.

Nielsen, J., and Landauer, T. K. 1993. A mathematical model of the finding of usability problems. *Proceedings ACM/IFIP INTERCHI'93 Conference,* Amsterdam, The Netherlands, April 24-29, 206-213.

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf*. Seattle, WA, 1-5 April, 249-256.