BUILDING DESIGNER-FRIENDLY SOFTWARE
"MICRO COMPUTER ENERGY SIMULATION THAT GOES BEYOND USER FRIENDLY"

BRUCE T. HAGLUND

Department of Architecture, University of Idaho, Moscow, Idaho

ABSTRACT - The concept of user-friendly software is critiqued. . The critique forms a basis, for transcending user-friendliness to software that satisfies both the novice and expert. The specific method of achieving that satisfaction for this problem, the 'applications spread sheet', is discussed.

INTRODUCTION

Our concern for the topic at issue in this paper was born after the publication of the expanded SLR/LCR method in J. Douglas Balcomb's Passive Solar Design Handbook Volume III in 1982 (1). Here was a fairly accurate method for predicting performance of passive solar buildings. However, in the name of completeness it contains an overwhelming amount of data describing 219 climates and 94 passive system types. Its publication has caused waves of praise for its comprehensiveness and waves of confusion for its lack of clarity for the ordinary designer. In response to this confusion articles by its author (2) and books by others (3) attempted to clarify the inscrutable method. Of course, these attempts were not effective in facilitating access to the method for the uninitiated, potential user. They simply added more sources of information to a process that was already obscured by a glut of information. A logical next step would be to consolidate the book in a microcomputer program so that all the tedium in manipulating the parameters and the disorientation in discovering the way through the process could be resolved. The building designer and computer could have a facilitatee/facilitator relationship. Not surprisingly, a lot of other people have had the same idea and have already completed their work (4). The success of these attempts has been arguable with one critic going so far as declaring that the whole method is inappropriate (5). What we're going to attempt here is not to criticize others but to explain what we've learned about the relationship between this application and user-friendly software.

GOING BEYOND USER-FRIENDLY

Let us first explain that software development for this project is still in progress in an extravagantly leisurely academic manner. This we believe is not a normal mode for software development; it's usually done as quickly as possible in order to keep costs down or to beat the competition to the marketplace. While maintaining this comparatively slow pace, our consciousness of what we're doing has evolved, changing significantly as we've wrestled with hardware problems, system problems, self doubt, and awareness of what others have done. Somehow we've allowed the software to design itself from this assortment of inputs rather than having taken a strict approach of designing the system and then writing the software.

All this self-critiquing and other-critiquing led us to the realization that we aren't really designing a program that does calculations for passive solar buildings - that, we have concluded, is fairly trivial given the existence of the SLR method. We really are designing a means for the building designer to deal with the parameters of the method after the fashion of an enlightening guided tour. That sounded to us like a need for user-friendly software.

Our collective experience had included the use of various similar applications programs (for passive solar design) and the usual array of more generic software - spread sheets, word processors and data base managers. Oddly enough, we felt more satisfaction with the generic programs than with the applications programs, even though both types were in the realm of the user-friendly. It occurred to us, that in order to produce a satis-

ying applications program, we would have to go eyond the concept of user-friendliness.

ON TO BUILDING-DESIGNER FRIENDLY

Our problem's definition had expanded to ing a system that is not only easy to use r-friendly), but gives the user satisfaction beyond user-friendly). Since the difference in evels of satisfaction rendered by generic and pplications software was important to us, we thought it prudent to examine that difference. The distillation of our examination of the difference s that generic software offers flexibility while pplications software offers little flexibility. The result of the difference is that as users of the best generic software we feel enabled to ccomplish tasks as we see fit (the software is a ool) and that as users of the worst applications software we feel trapped by a rote procedure (the software is the master). We hope you'll agree that ven in less extreme cases the 'enabled' generic oftware users must feel more satisfaction than the 'trapped' applications software users. The lesson we learned from this line of inquiry was that we hould create a system that is as flexible as ossible, given the application and the hardware.

A second important factor in developing satisfying applications software is understanding he essence of the application and how the software an be used as a tool rather than act as a dictator. So, we also investigated the SLR/LCR method in terms of its usefulness to the building esigner. It's important to remember the reason or computerizing the method, the problems of lack of clarity and volumes of data, has little to do with its intrinsic value to designers, and therefore, solving these problems is only a secondary of the software. The primary goal of the oftware should be the same goal as that of the method itself, namely providing estimates (10%-15% ccuracy) of passive solar heating performance hich are suitable for the design process (2). On purely scientific basis, the accuracy is low and depends on numerous assumptions which can contribute unpredictably to the lack of accuracy - the ost prominent assumption being that the building's limate will duplicate the average climate for a nearby city in a given year or over a given period f time. Accordingly, we discarded accuracy as an mportant issue and are simply applying Balcomb's ormulae and data. The issue of importance is the use of the method in informing the design process. The second lesson here is that we should create a ystem that can be used as a tool by building esigners (a very specific group of users) in the design process. With this understanding we coined the term 'building designer-friendly' to describe ur software.

THE BUILDING DESIGN PROCESS

One of our assets as programmers for this ystem is our familiarity with the building design process, being building designers ourselves. The definitive term 'the building design process' is isleading in that each designer uses a process hat often differs from building to building and certainly is different from the process of any other designer. Consequently, no one process fully describes the activity of building design. You'll note that this flexible idea about the design process reinforces the importance of our earlier idea that truly satisfying software must offer flexibility. In spite of the existence of a unique process for each design there are areas of commonality in design. Each designer must make decisions about the building by evaluating an option or comparing multiple options in both subjective and objective terms.

The SLR/LCR method offers objective answers to various design options for passive solar buildings. These options could be termed 'what if's'. Typical of what if's asked by solar designers are 'what if a trombe wall were used instead of direct gain?', 'What if a four inch slab is used instead of a six inch slab?', etc... These what if's are very specific and range from large scale decisions to minute details. Although the SLR/LCR method is not an extremely accurate method, it is strong in providing information on a wide range of options (94 passive systems) and a wide range of adjustments to each of these options (over 200 sensitivity curves for mass thickness, number of glazings, etc.) These variables cater to the kinds of questions designers ask. A way to answer such questions is to give results that compare alternatives and show how changes affect performance. The best way of showing these results, especially for visually oriented people like designers, is to do it graphically with pie charts, bar graphs, interactive graphs, etc. rather than tabularly. With us it is true that 'a picture is worth a thousand words'-- or numbers.

THE APPLICATION SPREAD SHEET

With our concern for satisfying the building designer as our major criterion, we've viewed our experiences as software users in a critical light. All the conventions that make software user-friendly --- menus, help, cursor control, function keys, graphic icons, etc. --- have gone into our system design, but the main conceptual breakthrough came when we realized that the user interface could be thought of as an application spread sheet. This is similar to the generic spread sheets with which you are all familiar, but has the format 'hard-wired' for the application. This similarity in concept with the generic software offers the advantages of flexibility described above. For example, the main menu for the program that calculates SLR/LCR is a spread sheet of fixed size, four data columns by five data lines (Figure 1). Each column allows for an optional building design configuration, which allows the user to simultaneously consider up to four trial designs that are as similar or as different as desired. The cursor is operated by the arrow keys and can be manipulated to fill in data in any order. An expert user would only need to fill out this single menu/spread sheet. The function key, F5, labeled HELP provides a help menu for the data item indicated by the cursor position. The novice user would fill out the main menu by using the help function for every item. This method of facilitating the use of the software by users from rank beginners to ultimate expert provides another level of flexibility.

Similarly, each of the four major functions in the software is interfaced with a main menu that is essentially a spread sheet and subordinate help menus that relate to the items in the spread sheet.

CONCLUSIONS

The discoveries we've made while pursuing the goal of making Balcomb's Volume III accessible to many people are not of an earth shattering technical nature. Rather they are subjective lessons in designing for people and building on what is known. We've found that it's important that software designed for a specific task relates strongly to the needs of the specific users, and yet provides enough flexibility to act as an enabling tool for the users. Our solution may not be valid for any other application, but we hope that the discussion of the system development process is helpful to others with similar problems. We also have found value in the rather unique situation where we are typical users of the intended system and we are able to proceed slowly. By being both potential user and system designer we have had the advantage of being able to weigh the alternatives from two points-of-view during the decision-making process, rather than the more common scenario of writing software which the potential user must critique after the fact. We believe this is helping us make better informed decisions. Perhaps the lesson inherent in our situation is that training for programmers should include in-depth training in the field for which software is being written.

REFERENCES

1. Balcomb, J. Douglas, et al. Passive Solar Design Handbook Volume III. American Solar Energy Society, Boulder, Colo. 1982.

2. e.g. Balcomb, J. Douglas, et al. 'Expanding the SLR Method'. Passive Solar Journal Vol. 1 No. 2 Spring 1982.

3. e.g. Shurcliff, W. A. Simplifying Guide to the Los Alamos "Vol. 3 Passive Solar Design Analysis". Solar Energy Information Services 1983.

4. e.g. Cornerstones Energy Groups Inc., Passive Solar Design . John Wiley Professional Software 1984.

5. Sams, John, 'Software Review: An SLR Method for Micros'. Solar Age Jan 1985.

ACKNOWLEDGEMENTS

The 'we' referred to herein includes the author and the architecture student research assistants - Kent Hanway, Steve Mallory, Michael George, Perry Brown, and Keith Smith - whose hard work and hard questions are responsible for any progress that we've made. Also notable for their generous support are our grantors at Washington Water Power, especially Mel Ott and Blaine Petersen, whose personal interests have made the project possible.

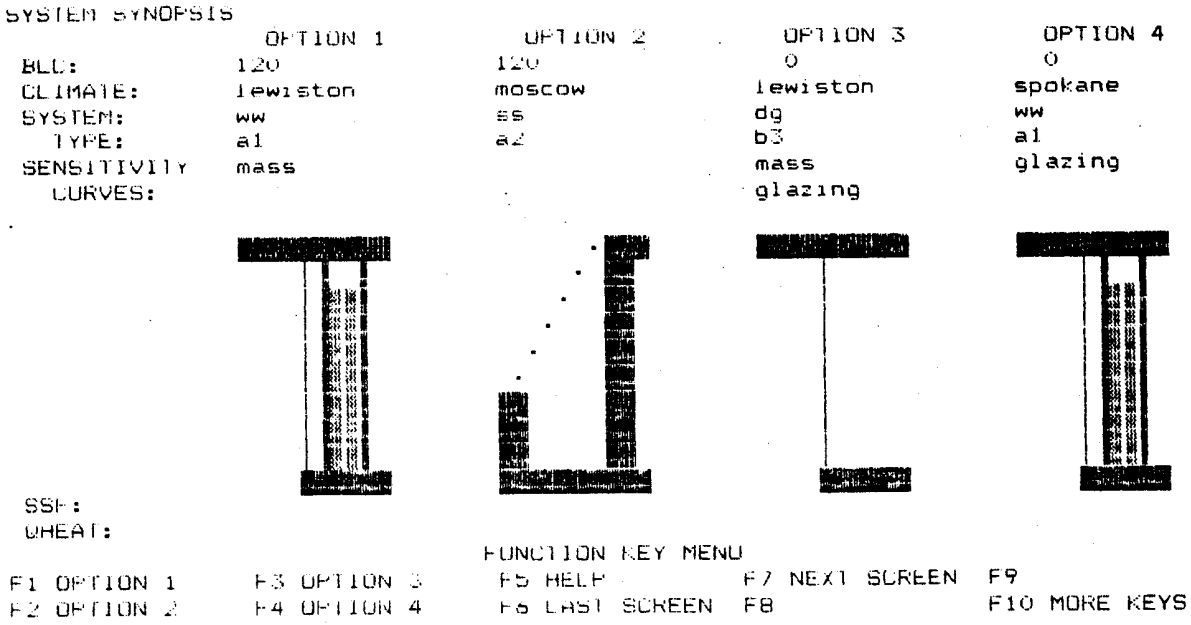

Figure 1. SLR/LCR main menu. This applications spread sheet has four columns, OPTION 1 - OPTION 4 and five lines, BLC: through SENSITIVITY CURVES:, which the operator may fill in. The graphics are icons that relate to the SYSTEM: entry and the FUNCTION KEY MENU identifies the action of the function keys.