Clients, architects, houses and computers: Experiment and reflection on new roles and relationships in design

Benjamin Cimerman
Rensselaer Polytechnic Institute, USA

Abstract
This paper reports on recent work that focused on the potential impact of standard computer technology on the relationship between client and architect in the context of residential design. A study of software applications a client could use to develop and evaluate ideas exposed the dearth of software available for the design of spatial complexity by individuals without advanced computer skills, and led to the design of a specific piece of software we call "Space Modeler." It was prototyped using off-the-shelf virtual reality technology, and tested by a group of freshmen students. The paper discusses the specificities of the software and provides analysis and reflection based on the results of the test, both in terms of design artifacts and users' comments. The paper concludes that the evolution of the interface to electronic environments is a matter of interest for those concerned with rethinking the training, role and activity of the architect. In the near future prospective homeowners may be able to experience and experiment with the space of their home before it is built. How can the profession embrace new information technology developments and appropriate them for the benefits of society at large?

Keywords
Design Software, Design Participation, Visualization, Simulation

1 Introduction

Computers have become commonplace in architectural offices, with diversity in implementation mirroring diversity in practice. In most offices computer-aided drafting has replaced the task of drafting plans and elevations by hand. In many offices computers are used for representation, allowing architects to present renderings of un-built projects or to take customers on virtual tours using fly-through animation techniques. Computers are also applied to project and information management, with the most recent emphasis on web-based systems that facilitate communication and collaboration between various parties such as designers, builders and contractors. Finally computer-aided engineering (CAE) and computer-aided manufacturing (CAM) also contribute to architecture by permitting complex buildings to be designed, simulated, prototyped and ultimately built. However, in most instances the computer is still used as a tool that may contribute to making the traditional design process easier, faster, and cheaper, but without any significant redefinition of that process. A more empowering approach to computing is to consider the computer to be a medium rather than a tool (Glanville 1995). Whereas the purpose of a tool is to facilitate a task (thereby making it faster to accomplish while reducing the risk of human error), a medium offers new avenues for experimentation and new territories for exploration. Many researchers have exploited this framing to re-think the architectural design process, and this re-thinking may manifest itself in a variety of ways. In our work, we have explored the concept of appropriation of standard computer technology to re-energize the relationship between client and architect in the context of residential design. The key concept is to empower the client and promote greater non-specialist participation in the design process. In the next two sections we explain why we believe that it is opportune to imagine new modalities of residential design. In the remaining sections we focus the discussion on software we developed for this purpose.

2 Trends in culture and society

It is usually estimated that only five percent of the new houses built in the United States every year are custom-designed by architects. Although that figure may be old news, significant and pressing issues with American housing beg for a greater involvement of the profession. Briefly, these include the interdependent problems of sprawl, environmental degradation, lack of diversity in the housing stock, and status of the house as a commodity easily bought and sold. At the same time, the 1990s have witnessed the emergence of two major cultural trends with repercussions in the field of housing: first, the development of personal computing and of the World Wide Web, and second, growing concerns regarding the protection of natural environments. The direct impact of personal computing and World Wide Web on housing may be categorized in two ways. First, housing needs are evolving as a result of cultural changes in work practices as well as lifestyle. Second, for the prospective house buyer or builder, there are now extensive resources available online, from free advice to home-design software and house plans that cost a fraction of an architect's fees. Regarding the environment, although polls indicated that environmental concerns are widespread in contemporary American society (The Gallup Organization 2000), there appears to be a discrepancy between these concerns and the current practices in terms of house design and construction. In addition to these cultural trends one should note that the computer revolution has fueled a significant increase in wealth for the middle class, and in particular for young technology professionals. This wealth probably explains the sustained growth in the average size of the house. According to the National Association of Home Builders (NAHB), new single-family homes have grown bigger over the past twenty-five years: from an average of 1,645 square feet in 1975 to 2,190 square feet in 1998 (National Association of Home Builders 2000). The NAHB also reports that the average price of a house in 1999 was $195,000 against $62,500 in 1978, but the median price in 1999 was $160,000, which suggests that the majority of houses were sold below $200,000 with a minority of houses being much more expensive. Clearly such trends should have provided architects with plenty of residential design projects. Empirically, a glance at our suburban and rural landscape suggests that this has not been the case.

Architects seem to have been unable to compete against developers on the high-end housing market. Our contention is that in order to do so architects should embrace recent technological developments to redefine the way they provide residential design services.

3 Opportunity

One of the consequences of the rapid growth of the PC market and of the Internet is that a large portion of higher-educated people in their 40s, 30s or 20s is computer-literate. In addition to mastering basic computer skills (such as knowing how to browse the World Wide Web, write with a word processor or calculate with a spreadsheet program), many students and professionals are also familiar with 3D computer graphics. One of the most widespread applications of that technology is video games, and the consequence is that many young people have become familiar at an early age with the concept of visualizing and navigating virtual space. Researchers have reported that 3D computer modeling may not be as effective as sketching or physical model manipulation in developing spatial visualization skills (Sorby 1999). However it remains a fact that a growing number of young people reach the age of college already "intuitively" skilled in the visualization and navigation of virtual space, something less than obvious for previous generations. Many professionals also had to become familiar with 3D computer graphics. This is particularly true in the field of engineering with the emergence of computer-aided design and computer-aided engineering, and in the entertainment industry. Other professions are also tapping into the technology: for example doctors are being trained in the use of 3D virtual reality systems, and financial analysts visualize complex data in 3D form. Therefore a growing pool of people master skills that are very relevant to the field of architectural design in the sense that they are familiar with computer-based spatial representation. We believe that this fact should be looked upon as an opportunity to invent new architectural services geared towards the computer-savvy clientele.

In our work we have focused on the concept of design software that encourages the client to be an active participant in exploring and proposing design ideas early in the process, as well as on the proposition that a Web-based studio can provide a central node for communication and project management. Our contention is that this new approach to working with an architect may attract people who would otherwise build a pre-designed house even if their budget can support the services of an architect, and may facilitate cost containment through better use of the architect's time, reduction in miscommunication, and efficient management of the relationship with the client. In this paper we specifically discuss our work on design software. The interested reader can consult our more extensive discussion of this work (Cimerman 2000a, 2000b) or access software and Web-based studio on our web site (www.designeasy.org).

4 Review and Analysis of Existing Design Software

Our work started with a review of existing design software, in the hope of finding an application that would satisfy our requirement for software that is easy to learn and use for computer-literate non-designers, and allows for flexibility and experimentation in form generation. There are many design software applications available on the market and each one of them exhibits strengths, limitations, and biases. The analysis focused on a simple question: what can be done and by whom? The applications that were reviewed range widely in functionality and price. They are:

- Cosmo Player by SGI
- Visual Home by Sierra Home
- DesignWorshop Lite by Artifice Inc.
- Sculptor created at ETHZ by David Kurmann
- Form-Z by Autodesys
- Softimage by Avid
- Microstation by Bentley Systems

Cosmo Player from SGI is a 3D web browser that is used to navigate Virtual Reality models. Thousands of such models have been created using the VRML standard. It is not a design software package *per se*, but it may allow an individual to review a design and comment on it, which is in itself a form of design participation. In fact various researchers have developed web applications that permits the mark-up of 3D designs (Craig and Zimring 1999, Jung and Yi-Luen Do 2000).

Visual Home by Sierra Home was selected arbitrarily as representative of the many home design software packages available on the market. Working in plan view, the user is guided through a set of well-defined steps (creating rooms, adding windows and doors, then adding the roof, etc.) that ultimately result in 3D visualization and walk-through. Rooms always exhibit right angles, and the formal and stylistic iconography of detached developer housing is ever present. A camera object that can be translated and rotated in the plan view facilitates navigation (Figure 1).

DesignWorkshop Lite is a free version of DesignWorkshop, by Artifice Inc. It provides a 3D modeling environment where simple geometric shapes can be created, mostly through a process of extrusion from 2D drawings. Its most striking features in terms of coupling advanced functionality with ease of use are the “wallify” function that turns a solid block into a room, and a cutter tool that makes punching openings very easy (Figure 2). Typological assumptions of architectural form are very obvious in the version tested and are believed to still be present in the commercial versions (the main difference seems to be that those also include extensive libraries of objects).

Sculptor was created at ETHZ by David Kurmann (Kurmann 1995) and has been used in a number of design studios. It offers a very intuitive and highly fluid 3D modeling environment where models are built from objects of simple shapes (the modeling approach is therefore very

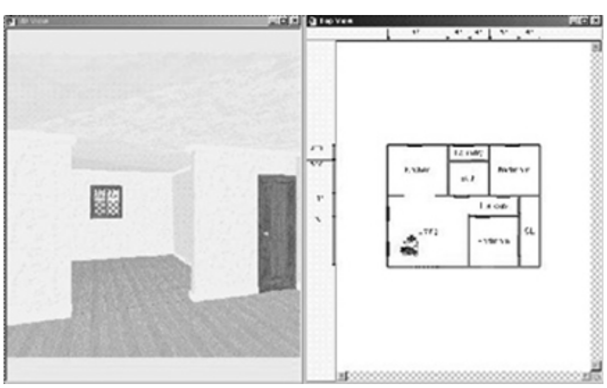

Figure 1. House design with Visual Home

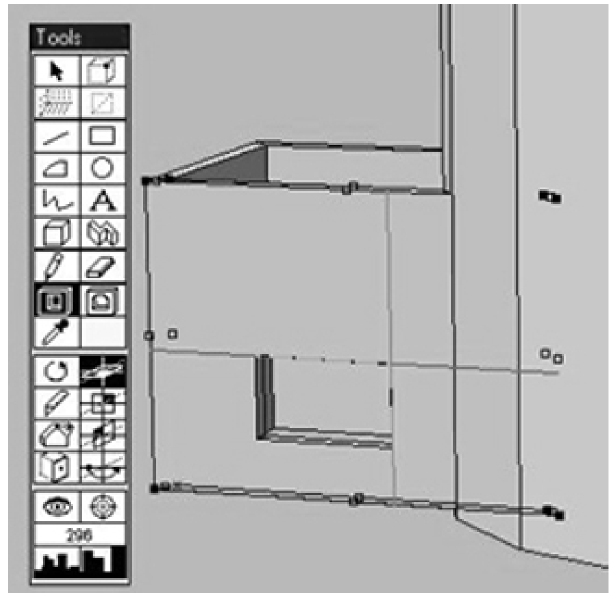

Figure 2. Cutter tool in DesignWorkshop

different from that of DesignWorkshop Lite where the form is built by extruding 2D shapes). It includes unique features such as a negative object that informs an original approach to space and void, and a deconstruct function that explodes a cube in a subset of elements. It is perhaps — of the applications reviewed here — the most influential for this work, although essentially shape-based and archetypal (Figure 3).

Form-Z by Autodesys and Softimage by Avid are high-end 3D design software applications with which complex forms may be generated. The Form-Z interface feels and behaves like that of

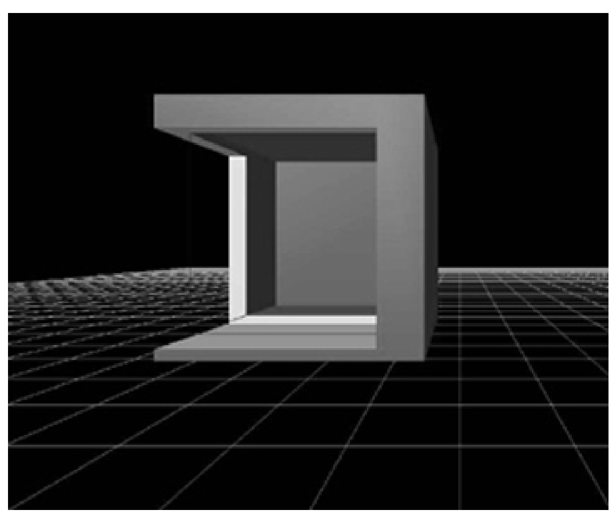

Figure 3. Intersection of positive and negative spaces in Sculptor

DesignWorkshop Lite, but in a more powerful and object-oriented fashion. While Form-Z is essentially a modeling package that includes some rendering capabilities, Softimage is designed as an advanced animation package that allows the animator to associate complex relationships and behaviors with modeled objects. These two programs allow for advanced formal experimentation beyond the capabilities of the other packages reviewed. However, the large number of tools and functions available (particularly in the case of Softimage) makes them difficult to use, unless one does so on a regular basis and commits a great number of hours to experimentation.

Finally, the Microstation suite of products by Bentley Systems is a widely used set of CAD/CAM/CAE packages with which highly specified, database-linked, parametric, and relatively intelligent models may be built. The modeling fluidity and flexibility of these packages is not as great as that of programs like Form-Z or Softimage and the latter programs are usually preferred for experimental projects. However the ability of the Microstation suite to embed non-visual representations of models (material or cost information, collision detection, performative criteria, data from Graphical Information Systems, etc.) is a substantial feature that facilitates the transfer of information through the design process, from conception to realization.

In order to gain practical insight of the pros and cons of each application, each was applied to the simple task of designing a two-volume house with very little level of detail (no inside walls and just a few windows and a door). This seemingly trivial task proved less than obvious with some of the applications and led to many observations on the nature of the modeling biases of each application. The following four general observations were perhaps the most instrumental in the development of the design software discussed in this paper:

- 3D navigation and wayfinding in virtual space can be difficult and frustrating (an iconographic representation of the camera seems helpful but its manipulation still requires abstract thinking and the development of a form of expertise).
- Placement of objects in 3D often requires alternating back and forth between different views (projections can help understand where the object is in relation to ground and other objects, but understanding such abstract information requires some spatial visualization skills).
- Creating openings in surfaces is a surprisingly complex task with most interfaces. Where boolean operations are required in Form-Z or Softimage, the act of cutting an opening is a multi-step process requiring object manipulation, view control, and object organization (deleting objects after the operation, etc.). The void object in Sculptor provides perhaps the most theoretically interesting method of doing this, but still requires a good understanding of object manipulation in 3D space. In contrast, the "cutter" tool in Design Workshop is highly intuitive and simple to use (Form-Z offers a similar tool but understanding of object topology and view control is required to use it effectively).
- Interfaces that place the emphasis on representation (such as Visual Home) are reasonably easy to use, but those that offer modeling abstraction (such as Softimage) require intense experimentation before one has acquired enough expertise to feel at ease with the software environment.

For every application tested, even this simple modeling task required an amount of trial and error to discover the proper way to model that may deter a large number of users from experimenting further.

These observations can be summarized by relating design flexibility to the level of expertise required for the accomplishment of a given task with each of the software applications. Design flexibility is understood in this context as a measure of the ability provided by the software to design objects of almost any form, simple or complex, and to do so in an experimental fashion. In order of increasing flexibility, the categories are: to visualize (i.e. zero flexibility), to create standard elements, to create simple shapes, to create complex shapes, and finally to embed non-visual representation in the designed artifact. This last category loosely groups under one heading advanced functions such as animation, parametric relationships, or embedding external data relevant to the project. The categories of expertise are computer-literate (which includes those who use computers on a somewhat regular basis and who are familiar with mainstream applications such as web browser, text editor or spreadsheet), computer-savvy (which includes those who in addition have had exposure to software requiring some form of 3D interaction and navigation such as video games, engineering software, or data visualization but who have had no formal design training), and computer-savvy design professionals (those who are at ease with complex 3D design software and who have had some formal training in creative design).

Figure 4 maps out each application's flexibility in terms of the expertise required to accomplish a given task. A linear trend between flexibility and expertise is clearly manifest. The top-left region corresponds to the area where a choice of software is available (for example a professional designer can choose between Softimage and DesignWorkshop to design with simple shapes). On the contrary, the empty area at the bottom right indicates the area currently lacking in tools. Individuals who are only computer-literate are limited to the use of a 3D browser or home design software, and are therefore highly limited in their forays into design experimentation. DesignWorkshop and Sculptor offer rich design capabilities without requiring as much expertise and training as some of the other applications, but both packages are nevertheless fairly complex to operate, and their topological limitations clearly inform the design possibilities. Softimage, Form-Z and Microstation provide a variety of tools for creating complex shapes, but only advanced users will cope with the complexity of their interfaces. The diagram also indicates that no application is available for individuals who are only computer-literate to explore even simple-shape design or for individuals who are computer-savvy but not design professionals to explore complex-shape design.

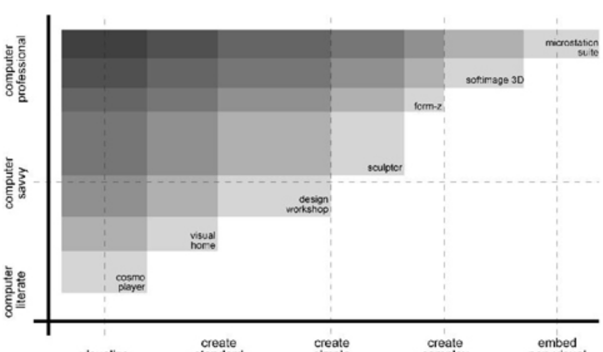

Figure 4. Flexibility vs. expertise for various software packages

This analysis highlighted the dearth of software available for the design of spatial complexity by individuals without advanced computer modeling skills. In the next section we describe Space Modeler, the software we designed and prototyped as a response to this situation.

1 Description of Space Modeler Prototype

In Space Modeler, the user is presented with the concept of space rather than that of form or object. The theoretical source for this approach can be traced back to the strain of thinking in 19th century Western architectural design which reinvigorated the conception of architecture as a spatial design problem, such as discussed by Bruno Zevi in *Architecture as Space* (Zevi 1957), and Cornelis Van de Ven in *Space in Architecture* (Van de Ven 1987). Since the objective is to enable the user to learn about design through participation and personal experimentation, it is believed that a focus on spatial modeling rather than on stylistic manipulation will help provide an environment that challenges the dominant models of popular home design.

The software presents to the user a simple modeling interface that provides tools to explore the manipulation of space via a kind of sculptural modeling of volumes bounded by surfaces (Figure 5). Modeling begins with the creation of a cuboid, where each of the six faces of the cuboid is composed of squares. In the software the cuboid is referred to as a space, and the squares that make up the faces are referred to as tiles. Spaces are assembled together without restrictions on relative position, and each space's position can be modified (rotation and translation) to study relationships within the structure and with the environment (Figure 6). In addition to transformations applied to a whole space, tiles can be rotated, translated, disconnected from other tiles or deleted (Figure 7). These operations can be performed from inside or outside the space and switching back and forth is facilitated by a function that modifies the viewpoint accordingly (Figure 8). Since the software is intended for investigations in the programmatic or early schematic design phases, there are no libraries of predefined objects such as doors, windows or stairs, and there are no texture-based material associations (al-

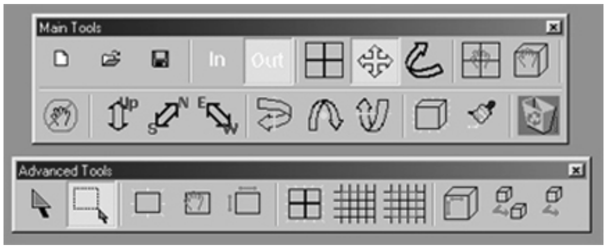

Figure 5. Icons for modeling tools

Figure 6. Combination of three spaces

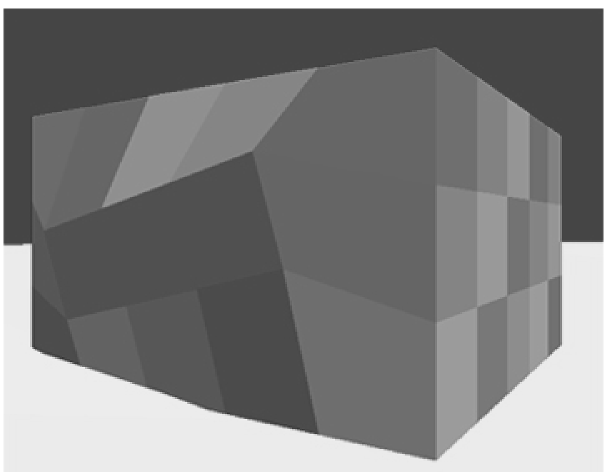

Figure 7. Single space with three tiles translated and rotated

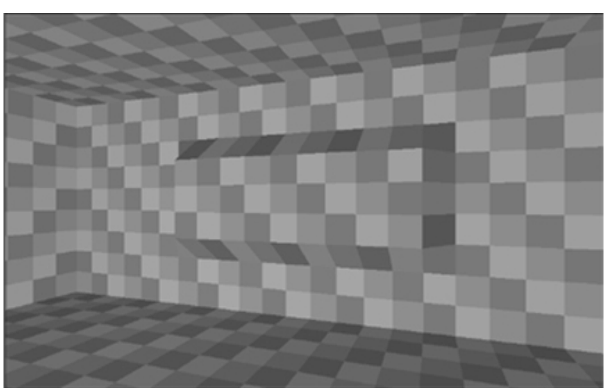

Figure 8. Inside view of a space with tiles showing

though colors or transparencies may be modified). Dimensions are suggested implicitly by the number of tiles that make up the surfaces bounding a space, but scale is intentionally indeterminate at the outset of the design session. Once the user has understood that operations can be performed at the space level or the tile level, most of the functions become self-evident and their limited number keeps the interface very simple.

The environment is represented by four components: topography, background, cardinal orientation and sun position. The topography can be modified to represent the land on which the house is to be built, and surfaces with pictures can be set up on the scene in order to simulate views of the surroundings (Figure 9). Translations and rotations are defined in terms of North-South, East-West and vertical axis rather than a Cartesian coordinate system. Getting rid of this mathematical abstraction re-enforces the connection of the building with the land and reminds the user of the importance of orientation. Finally, the sun position is calculated based on longitude, date and time.

2 Space Modeler Testing

The evaluation of the Space Modeler software was carried out to estimate the success of the implementation and the strength of the concept. At the implementation level the concern is reliability, ease of use, and functionality. At the conceptual level the aim is to evaluate if the software affords the type of design exploration it was conceived for. The evaluation period lasted over a period of more than six months. Early users' observations were very helpful in refining the interface (design of menu bar and placement of menu items, selection of functions that should be accessible through icons, design of icons and dialog boxes) and their

Figure 9. Background images in Space Modeler model

suggestions lead to the inclusion of simple but helpful functions (such as the ability to move and copy spaces or to create customized spaces with any number of tiles in length, width and height). Suggestions by sophomore and junior architecture students led to the design of functions permitting more complex design explorations. Then at the beginning of the Fall 2000 semester first year undergraduate students in Architecture and Product Design & Innovation (approximately 80 students) were asked to use Space Modeler before any other design program. After a short in-class session, they were given an assignment for the following week. Their assignment was to design a structure for the purpose of observing nature, relaxing, or meditating on a local river island. Their work often exhibits a very conventional understanding of built form, and educators may argue against the value of giving such an exercise to new students of architecture. However the insight provided by the experiment was, for the purpose of this study, extremely valuable. Particularly striking was the fact that many students used the software in ways it was not intended for. Specifically, they took advantage of the scaling functions to use the primitive volumes modeling evolves from as objects or building components (Figure 10). This suggests that there is inherent flexibility in the software, that the built-in biases and limitations are obvious and strategies for circumventing them easy to devise, and finally that the students quickly felt at ease with the software. Most observers were also surprised by the variety of forms that resulted from the exercise (Figure 10 through 16).

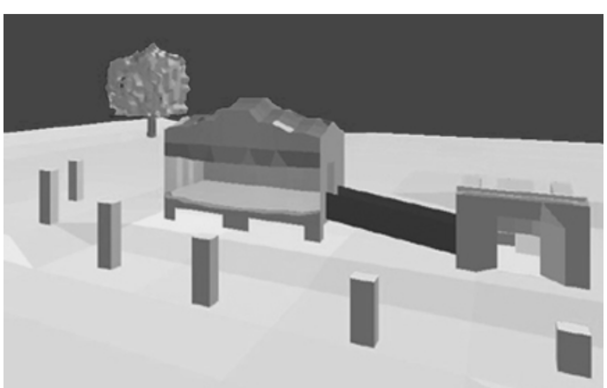

Figure 10. Design 1 - Volumes used as building objects

In addition to submitting their work, the students were surveyed about the software and provided valuable insight. Most students appreciated the fact that the software was easy to learn because of the limited number of functions, but a few reported feeling limited in what they could do. The lack of dimensioning in the software was perceived by some (often with previous AutoCAD experience) as a drawback, while others appreciated the fact that by counting tiles one could get a rough estimate of dimensions. Most students reported that moving around and in and out of spaces was helpful in getting to understand the spatial implications of their design, but some students complained that navigation was difficult while others described the experience as enjoyable. Indeed many students reported "having fun" while working on their design, which is a very positive finding of the project. The enjoyment of the students seemed to translate in the production of work far beyond what was required for this assignment.

3 Space Modeler Quantitative Analysis

In addition to user testing, one can use quantitative forms of analysis to evaluate software. In designing and implementing Space Modeler, much consideration was given to the inherent tension between ease of use and functionality. The temptation to continually add functions to satisfy a broad range of users and applications is very strong, as testified by the plethora of functions available in many commercially available software packages. This often results in cascades of menus or large palettes of superimposed icons. In Space Modeler each of the five main menu items contains a list of six or fewer items with some of them leading to a third and final level with at most four items in each list. The total number of functions in the software is only forty-four (compared to approximately one hundred for DesignWorkshop Lite and many more for Form-Z or Softimage), and most work is performed by accessing the twenty icons placed under the menu bar. An additional toolbar (which by default remains closed) provides access to eleven more advanced functions. A new user can learn how to operate the software using the basic twenty functions and only later use the remaining ones. Software complexity is also often related to the large number of

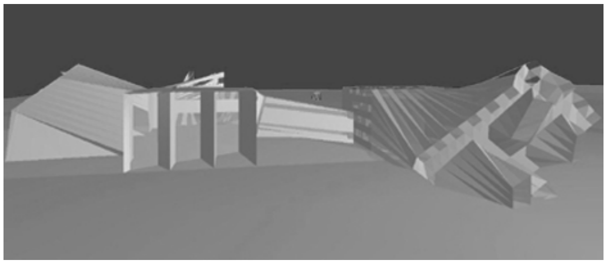

Figure 11. Design 2 – Making use of the advanced functions of the software

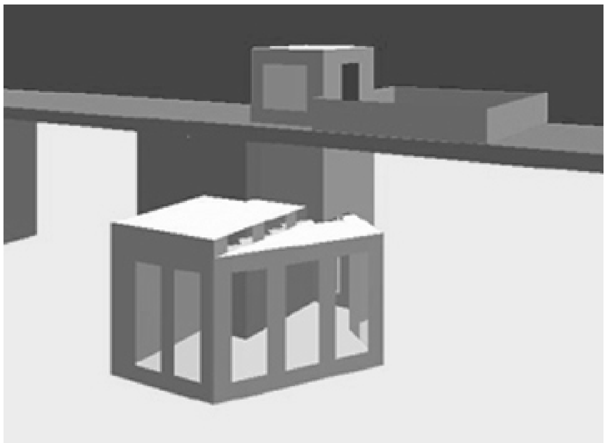

Figure 12. Design 3 - Taking advantage of the environment (Disused railway bridge)

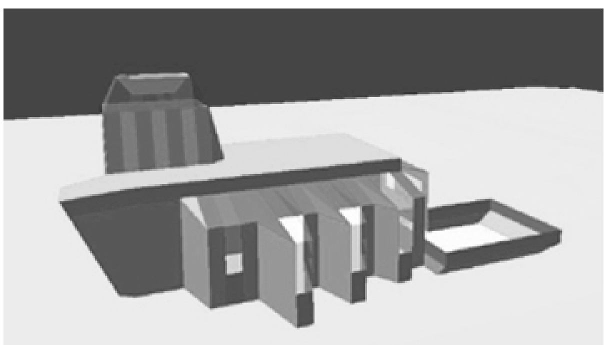

Figure 13. Design 4 – Exterior view

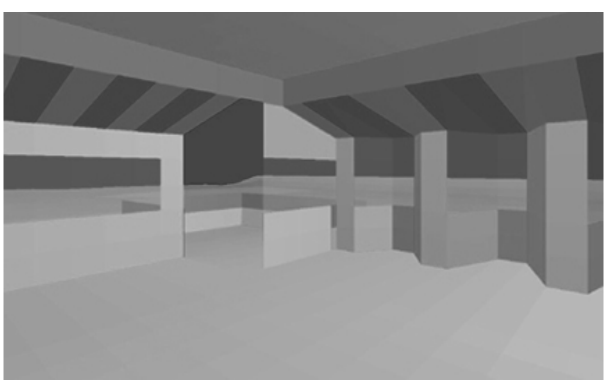

Figure 14. Design 4 - Interior view

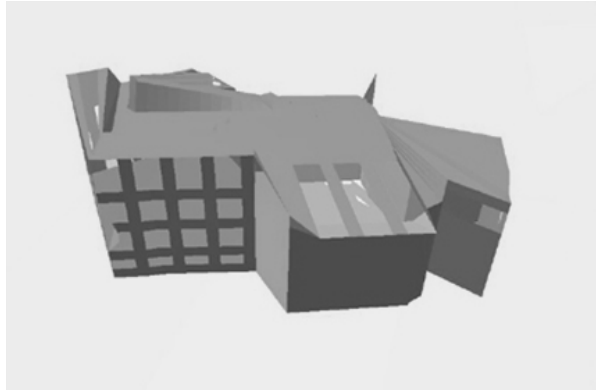

Figure 15. Design 5 - Exterior View

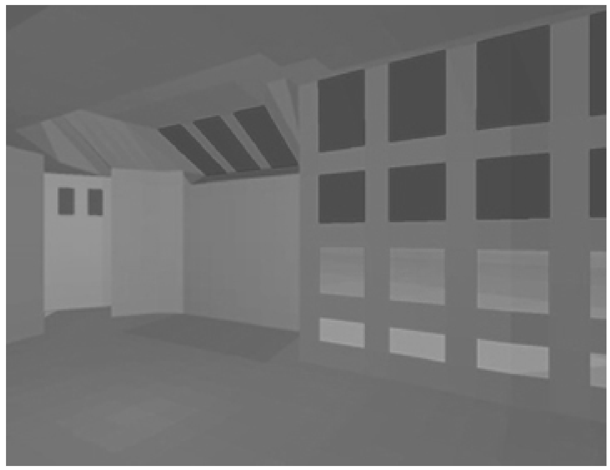

Figure 16. Design 5 - Interior view

dialog boxes. Aside from input/output and warning message dialog boxes, there are only four dialog boxes in Space Modeler. This is because most of the design decisions are taken by the user directly in the 3D environment, rather than in a text environment (the dialog box) which then affects the 3D design. The tight correlation between action and result facilitates the learning curve of the software, and so does the consistency and transparency of the underlying logic of operation. The logic for the user is first to select an object (tile, background screen, or space) or a group of objects (tiles) and then apply a transformation to the selection. The result of each action can be clearly evaluated because there are no hidden consequences to any action. Observing the new undergraduate students using the software for the first time indicated that for many of them it only took a few minutes to understand the logic of the modeling approach and to start some investigation of their own.

Overall the testing campaign indicates that the modeling approach offered by the software is well suited to the task of preliminary design investigation by non-professional designers. However minor improvements to the user interface and real case studies of people designing their house would be necessary to further this evaluation.

4 Conclusion

The impact of computer technology on the field of architecture is not limited to the use architects make of it. As the computer becomes a ubiquitous medium for communication, the way we interface with electronic data is bound to affect all areas of life. The evolution of our interface to electronic environments is a matter of interest for those concerned with rethinking the training, role and activity of the architect. Focusing on residential design we can foresee that new interfaces will empower the prospective house buyer by providing the ability to research and process information as well as assist with visualization and decision-making. The rather limiting and clumsy home design software and web sites of today may be the ancestors to more powerful and refined systems. It does not seem very far-fetched to imagine that within a few years house buyers will routinely visit virtual houses using immersive Virtual Reality systems. Through that interface with digital environments they will simulate variations to the design and select surface materials, appliances or furniture. Needless to say real estate firms and developers will most likely be the ones offering this category of services, and the ability for the client to experience the space of the home before it is built will render the architect's mental visualization skills much less crucial to the design process. It then becomes a matter of urgency for the architect to offer the client a highly individualized service and to be able to communicate the value of good design. Rather than resisting the wave of technological changes that empower client and competing professions, architects need to reevaluate their ways of practicing residential architecture. The software prototype discussed in this paper is part of a larger effort to propose a new model for the relationship between client and architect. We believe that the spatial approach we have taken to preliminary design software holds promises and should be investigated further. At the same time we view our work as simply one possible manifestation of an idea. Undoubtedly talent, experimentation and new technologies will contribute to the development of better systems.

Acknowledgments

The author wishes to thank Professor Brian Lonsway for his guidance, and is grateful for the support of the School of Architecture at Rensselaer Polytechnic Institute. Acknowledgements also go to all the faculty members and students who participated in the software testing.

References

Artifice Inc. Maybeck Studio <http://www.artifice.com/3d_gallery/Maybeck_Studio.wrl> (30 October 2000).

Cimerman, B. (2000a). *Client-Architect Design Cooperation*, Master's Thesis in Informatics and Architecture, Rensselaer Polytechnic Institute, Troy, NY.

Cimerman, B. (2000b). Participatory Design in Architecture: can computers help?, In Proceedings of PDC2000, New York City, NY: 40-48.

Craig, D. L., and C. Zimring. (1999). Support for Collaborative Design Reasoning In Shared Virtual Spaces. In *Proceedings of ACADIA 99*, eds. O. Ataman and J. Bermudez. Salt Lake City, UT: 126-137.

Glanville, R. (1995). Architecture and computing: a medium approach. In *Proceedings of ACADIA95*, eds. L. N. Kalisperis and B. Kolarevic. Seattle, Wa.: 5-20.

Jung T., and E. Yi-Luen Do. (2000). Immersive Redliner: Collaborative Design in Cyberspace. In *Proceedings of ACADIA 2000*, eds. M. J. Clayton and G. Vasquez de Velasco. Washington, DC.

Kurmann, D. (1995). Sculptor - A tool for intuitive architectural design. In *Proceedings of CAAD Futures '95*, eds. M. Tan and R. The. University of Singapore: 323-330.

National Association of Home Builders, Economic and housing data http://www.nahb.org/facts/default.htm (30 October 2000).

Sorby, S. A. (1999). Developing 3-D Spatial Visualization Skills. In *Engineering Design Graphics Journal*, vol. 63, no. 2 (Spring): 21-32.

The Gallup Organization. Environment. <http://www.gallup.com/poll/indicators/indenvironment.asp> (30 October 2000)

Van de Ven, Cornelis. (1987). *Space in Architecture*. Assen/Maastricht: Van Gorcum.

Zevi, Bruno. (1957). *Architecture as Space*. New York: Horizon Press.