AN INTERFACE MODEL FOR IMPROVING THE USE OF SPACE SIMULATION SOFTWARE IN ARCHITECTURAL DESIGN

A THESIS
SUBMITTED TO THE DEPARTMENT OF
INTERIOR ARCHITECTURE AND ENVIRONMENTAL DESIGN
AND THE INSTITUTE OF FINE ARTS
OF BİLKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN ART, DESIGN
AND ARCHITECTURE

By

BURCU ŞENYAPILI

March, 1998

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Bülent Özgüç (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality as a thesis for the degree of Doctor of Philosophy.

Prof. Dr. Varol Akman

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality as a thesis for the degree of Doctor of Philosophy.

Assoc. Prof. Dr. Cengiz Yener

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality as a thesis for the degree of Doctor of Philosophy.

Assist. Prof. Dr. Veysi İşler

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality as a thesis for the degree of Doctor of Philosophy.

Assist. Prof. Dr. Mahmut Mutman

Approved by the Institute of Fine Arts

Prof. Dr. Bülent Özgüç, Director of the Institute of Fine Arts

ABSTRACT

AN INTERFACE MODEL FOR IMPROVING THE USE OF SPACE SIMULATION SOFTWARE IN ARCHITECTURAL DESIGN

Burcu Şenyapılı
Ph.D. Program in Art, Design and Architecture
Supervisor: Prof. Dr. Bülent Özgüç
March, 1998

There is an ongoing debate on the success of architectural software in meeting the designers' wishes and in being *familiar* to the way designers design. One dominant belief is that as architectural software introduces a work environment closer to that of the paper-based techniques, the efficiency of the use of such software in the profession will increase. We argue that the use will increase by designing interfaces through which the users will be able to customize the digital environment according to their wishes. This thesis introduces a context-specific interface model to transform a state in the *user+need space* to a digital aid in the *virtual design space.* This model incorporates the Customization Scale Menu (CSM) to act with the menu options of the architectural space simulation software. The menu options are customized through the selections made on the CSM by the user. These selections will determine the required level of interaction between the software and the user, thus customizing the digital environment according to the user's needs.

KEY WORDS: Computer Aided Architectural Design, Virtual Design Environment, Interface Design, Architectural Space Simulation Software, Modeling, Virtual Reality.

ÖZET

MİMARİ TASARIMDA MEKAN SİMULASYONU YAZILIMLARININ KULLANIMINI İYİLEŞTİRMEK İÇİN BİR ARAYÜZ TASARIMI

Burcu Şenyapılı
Sanat, Tasarım ve Mimarlık Doktora Programı
Danışman: Prof. Dr. Bülent Özgüç
Mart, 1998

Mimari bilgisayar yazılımlarının tasarımcıların gereksinimlerini karşılamada ve tasarım yollarına yakınlık sağlamadaki başarıları tartışılagelmektedir. Tartışmadaki baskın görüşlerden biri mimari yazılımların kağıt esaslı mimari çalışma ortamına yakınlık sağladıkları oranda kullanım etkinliklerinin artacağı yolundadır. Bu çalışmada ise etkinlik artışının ancak mimari yazılımlarda kullanıcıların yazılımları isteklerine göre düzenlemelerine olanak tanıyan ara-yüzler kullanılması ile sağlanabileceği iddia edilmektedir. Çalışmada, kullanıcı+gereksinim uzayındaki bir durumu sanal tasarım uzayına aktaracak bağlam-özel bir arayüz modeli sunulmaktadır. Bu modelde yer alan Biçimlendirme Ölçüleri Menüsü (BÖM), modelin birlikte kullanılacağı mimari yazılımın menü ve menü seçenekleri üzerinde çalışacak ve kullanıcının seçimleri doğrultusunda düzenlemeler yapacaktır. Böylelikle kullanıcı ile bilgisayar arasındaki iletişim kullanıcının arzuladığı düzeyde gerçekleşecektir.

ANAHTAR SÖZCÜKLER: Bilgisayar Destekli Mimari Tasarım, Sanal Tasarım Ortamı, Ara-Yüz Tasarımı, Mimari Mekan Simulasyon Yazılımı, Modelleme, Sanal Gerçeklik.

ACKNOWLEDGMENTS

I would like to thank my advisor Prof. Dr. Bülent Özgüç not only for his invaluable guidance and advises, but for supporting me in publishing parts of this thesis during our studies. I owe many thanks to Prof. Dr. Varol Akman for his continual support, tutorship and patience which made a great deal of this thesis to be completed.

I would like to mention two special people who helped me develop this thesis during my stay and studies in the United States. I would like to thank Prof. Ardeshir Mahdavi at Carnegie Mellon University who supported my studies with great concern and appreciation and Rudi Stouffs for his friendship and the many hours he spent to explain the difficult concepts to me.

Last, but not least, my special thanks to my family, who made me go on when I had no courage to do so and for being so exceptional, loving and fun to be with. I would like to express my thanks to Suat Özcan who made me believe in myself. As always, I dedicate this thesis in the loving memory of my grandmother, a wonderful person, of whose love and care I would like to be worthy of with every step I take.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FORMULAS

LIST OF ABBREVIATIONS

A:	Audience (expectations of the audience)
AIDA:	Adaptive System for Interactive Drafting
ATC:	Actual Time Coordinate
AUI:	Adaptive User Interface
CAAD:	Computer Aided Architectural Design
CAD:	Computer Aided Design
CS:	Context Specific value
CSM:	Customization Scale Menu
D:	Default value
DAAD:	Digital Aided Architectural Design
DAD:	Digital Aided Design
E:	Experience (experience of the user in using the software)
HCI:	Human Computer Interaction
I:	Interaction (coordinate of the IRT space)
II:	Intelligent Interface
IRT:	Interactivity-Rendering-Time
P:	Purpose (purpose of making the simulation / design stage)
PAE:	Purpose-Audience-Time
R:	Rendering (coordinate of the IRT space)
T:	Time (coordinate of the IRT space)
S:	Software (references)
VDS:	Virtual Design Space
VR:	Virtual Reality
VTC:	Virtual Time Coordinate

1. INTRODUCTION

1.1. Aim

Architectural design process is concerned with the creation and representation of spaces. Architects have been using the paper-based techniques to carry out this process until recent years. Then, with the introduction of the *digital media* by Mitchell and McCullough to architecture, they were given the option of using the *digital work environment*. This environment introduced the opportunity to create, manipulate and simulate the architectural space digitally, as is discussed in detail in Section 2.2.2. However, the environment, although efficient and fast especially in the representation part of the design process, is considered to be *unfamiliar* to the way architects create. Thus, in spite of the fact that more architects and architectural firms get involved with computers everyday, a large number of them use the digital environment for representation purposes rather than creation.

Architects have not asked for an alternative design environment. They have been using the paper-based techniques for a long time and even the ability to use these techniques has become an indispensable part of the profession. As the digital work environment was made available to the architects, they were impressed by the speed and ease provided by this environment especially for presentation. This has become one of the major reasons in the fast acceptance of the digital work environment to the profession. However, as the architectural software are developed by non-architects, the architects are bound to express their wishes and complaints only after the software is produced, not during the production. Richens states the fact that the creativity shifts from the architect to the ones who write the standards, the databases and the engines to operate them (306). Therefore, shortly after the emergence of the digital media in architecture, architects chose to employ them mainly for representing and simulating what has already been created (where they were very efficient) rather than for creating (where they found the digital environment 'unfamiliar'). Thus, the use of the new media has not reached the limits of its capacity.

This situation led us to seek ways to improve the use of the architectural software within such media employed in architectural design. It is initially required to point out the problems that the architects are faced with while using the software. These problems, when overcome, will enable an efficient use of the software in the profession. Our understanding of improving the use of the architectural software in architectural design is to create a platform where architects can get a hold of the emerging possibilities of the digital media and control the development according to their wishes instead of leaving the development in the hands of other professionals.

Architects complain about not being able to be as free as they are with paper-based techniques while using the architectural software and many long for the strikes of the soft pencil (O'Connell 16). To overcome the complaint that the architectural software packages are 'unfamiliar' to the way architects design, one major tendency is to force the architectural software to offer a work environment similar to that of the paper-based techniques like Gross and Do. This is tried to be achieved by features like using pens as input devices, offering sketchy looking line quality, allowing file exchange between various software, integrating large libraries and increasing the menu choices. But then the software packages expand in such a manner that both the user trying to see the composition of two basic geometrical shapes and a second one making a lighting analysis of a space have to go through the same steps and have to input the same amount of data to perform their very different tasks. As such, new complaints arise concerning the amount of time required to design (Potter 16), amount of time required to get used to the new additions and versions (Charles 121), amount of decisions to be given in the form of data even at the initial steps of design while using the architectural software packages. The latter factor has a crucial role in the fact that the architectural space simulation software can have little impact on the early stages of design (Richens 316).

In this study, we initially intend to show that the potential of using the digital environment for creation in architectural design is more than the paper-based techniques. We argue that it is actually the paper-based techniques that serve more for representation than for creation in architectural design. As such, trying to make the way we use the architectural simulation programs similar to the way the paper-based techniques reduces the program's potential for being used for creation (Thomsen 167-88). Recent researches confirm the view held by this study that architectural simulation packages, in spite of the powerful and complete design environment they offer, are not used efficiently, effectively and widely (Ormerod and Aouad 322-28).

Therefore, unlike the studies which try to render the virtual design environment similar to the paper-based, we study on the interface, which the user is confronted with before accessing the virtual design environment. We aim at defining the properties of an ideal interface, capable of manipulating the above stated problems of the architects, to overcome the inefficient, ineffective and narrow field of use of the architectural simulation software. Based on this interface definition, we then plan to increase the efficiency and use of these software by decreasing the menu and numerous other interaction items for the designer according to the task and to the designer's profile. This approach contradicts with the current trend of the software developers who increase the menus and menu items for a rich looking simulation program, seemingly capable of doing anything. The problem in this case is the fact that such a simulation program can be used to its full capacity only if the user is very experienced in both architecture and in using the program. Otherwise, the increased menus remain untouched and untested.

Maulsby observes that what users really want is more than an intelligent interface, it is an interface adapted to their own way of working. Because of the economies of scale he states that nearly all systems have to be thought for the *generic user* (234). Within this framework, we develop an interface system which will not be adapted to each user, but will allow each user adapt the software's menu options. In other words, the model is developed to allow the designers customize any architectural space simulation software according to the way they design rather than customizing the way they design according to the software.

1.2. Object, Scope and Structure

Within the framework put forth in the first chapter, the second chapter discusses the creation and communication in architectural design as a modeling process. The properties of the mental design model and the modeling process are examined. Then, the potentials of both the paper-based and the digital media in handling this modeling process are compared. Based on this comparison, we assert that the digital media are 'familiar' to the essence of design, this essence being the mental design model. To benefit most from the digital media, instead of trying to bring its potential down to the level of paper-based techniques, architectural design must be re-defined in relation to the digital potential. We then argue that the problem faced when using the architectural software is not based on the lack of familiarity but rather on the lack of adequate interface design.

In the third chapter, we examine the architectural software mentioned in the previous chapter closely. We define the services provided by and the problems faced with the currently used architectural space simulation software and the virtual design environment formed by these software. We then define the ideal interface for the architectural simulation software to overcome these problems. This definition guides us through our interface design in the next chapter.

With the fourth chapter, we concentrate on forming an interface model for architectural space simulation software based on the properties of the architectural space simulation software and the ideal interface defined in the previous chapter. We initially re-define architectural means of communication in the Cartesian space of the digital environment, freeing it from the domain of the paper-based techniques. Thus, we obtain a space where we can determine the level of architectural communication which is applicable to the architectural software. Next, we have to allow the user to define the level within this space. But, instead of loading the user with such a burden, we form another Cartesian space to indicate the user's expectation from the architectural software. Consequently, our task of forming the interface model becomes a transformation of a given point in the user's space to the digital space. We define a transformation between the two spaces and then test the possible cases and discuss the relevant implications.

Finally, the thesis concludes with the discussion of the implications of the CSM and introduces areas of further study.

1.3. Original Discussions of the Thesis Within the Related Field of Research

Most of the current researches on human computer interaction (HCI) deal either with the human or the computer side. Studies on the human side focus on analyzing the task to be done (Shepherd 145-74) or understanding the user profile (Howes 97-119) which lead to user-centered interface systems named as adaptive user interfaces (AUI) to be built. Studies on the computer side deal with the provision of expert help by the computer that result in system-based interfaces referred to as the intelligent interfaces (II).

AUIs try to tailor the interaction of the software system according to the changing needs of the users, changing conditions (Dietrich et al. 13) or changing user profile. Some studies concentrate on the specification of the task and develop tools to analyze and build target task models. The interface's dialogue with the user then is realized on task-based platforms like in CHARADE where the user is recognized through the specified task models (Marti and Normand 39-50). Some prefer to outline the user profile according to their problem solving ways and learning capacities (Howes 97-119). However, none of the researches of either approaches declared success with the users so far. There exists no application of the AUI research on the commercially available programs, including architectural simulation packages.

This thesis claims that for the success of the AUI in architectural programs, the analysis of the task and the user are not sufficient alone, and must be combined. Architectural design process requires different tasks to be performed at different stages of design and they differ further according to the designer who executes them, unlike say a medical task where the sequence of actions are almost solid (Sherman 285-315). Therefore, ideally the interface is expected to respond or adapt not only according to one criterion, but more criteria pertaining the user.

II research seeks ways to provide the user with the relevant information and context-sensitive aid through expert systems and knowledge-based agents. Recent researches in this field deal with loading the interface with sets of information to be used in a specific domain so that the interface gives adequate response to every situation. They either utilize agents to do tasks on the user's behalf (Maes 41) or try aiding the user by an extensive run time support, answering the questions about the hows and whats of the program. However, neither the agents nor the II supported programs are yet commercially available, especially in the field of architecture. The current applications of the studies are implemented on programs created especially for demonstrating the purpose. COLLAGEN, the collaborative agent toolkit designed by Rich and Sidner to help the users problem solving process while using the software is implemented on an air travel planning system created by the researchers. The II system I-SEE toolkit by Quemeneur and Brossier, provides intelligent help for OPX2, again a software developed for research purposes.

Unlike most of the relevant research, this thesis builds (Sections 4.1, 4.2 and 4.3) and implements (Section 4.3.1) a model upon the existing software packages, rather than creating one from scratch. Within the current research, interface models are integrated within domains of their own, rather than utilizing the existing software packages like AIDA (Adaptive System for Interactive Drafting) for SIEMCAD (Cote-Munoz 225-40), an intelligent interface created to run with only the special package SIEMCAD which is a non-commercial drafting program. The only exception are the interface models developed to act with the internet browsers and e-mail programs like ActionStream (Maulsby 235) or BASAR (Thomas and Fischer 53-60).

The interface model dealt within this thesis is not a user-based model. An user-based model would either try to find out about the tasks the users have to perform and the procedure they have to follow, or would require the users to be conscious about the menu item addition and subtraction operations (Sherman 285-315). The introduced interface model requires task specification but does not limit its scope to task specification only. It also inquires about the user's background in using the software and the audience to whom the finished task will be displayed. This broadening of the scope is an unprecedented approach in interface design for architectural software in particular.

On the other hand the introduced model is not a system-based model. Although intelligent help is provided, users have the option of customizing the level of help. In other words, they have the opportunity to tailor the level of default help.

The best address for the introduced model is an area in between the sets of the user-based and system-based models, combining their capacities with the opportunity of customization. Thus, the user has the chance to customize the model to act completely as an user-based or a system-based interface.

Illich described the media whose purpose and content are specified by the user as the convivial tools. Years later, Thompson asserted that the truly convivial medium is the one which enables the users to find their way to the right information through the interface. CSM is a pace towards making the use of the architectural space simulation software packages convivial, enabling the user to specify the purpose and the content while providing intelligent help within this context.

Similar to the address of the introduced model within the existing models (combining both the intelligent help and the adaptation facilities of the commercially available architectural space simulation software packages), this thesis defines itself a research area which can be referred to as the intersection of the II and AUI research. The discussions about the introduced model are supported by the current researches of the author (Şenyapılı, "Proposal", "True Model", "Visualization"; Şenyapılı and Özgüç, "Computer Aid", "Interface").

2. THE CONCEPT OF MODELING IN ARCHITECTURAL DESIGN AND DESIGN COMMUNICATION

2.1. The Design Model in the Creation Process

We perceive, comprehend, implement and communicate with the environment via forming mental models of that environment. These models store the information about the environment and this information is referred to for purposes like evaluation, change, comparison and communication (Fig. 2.1).

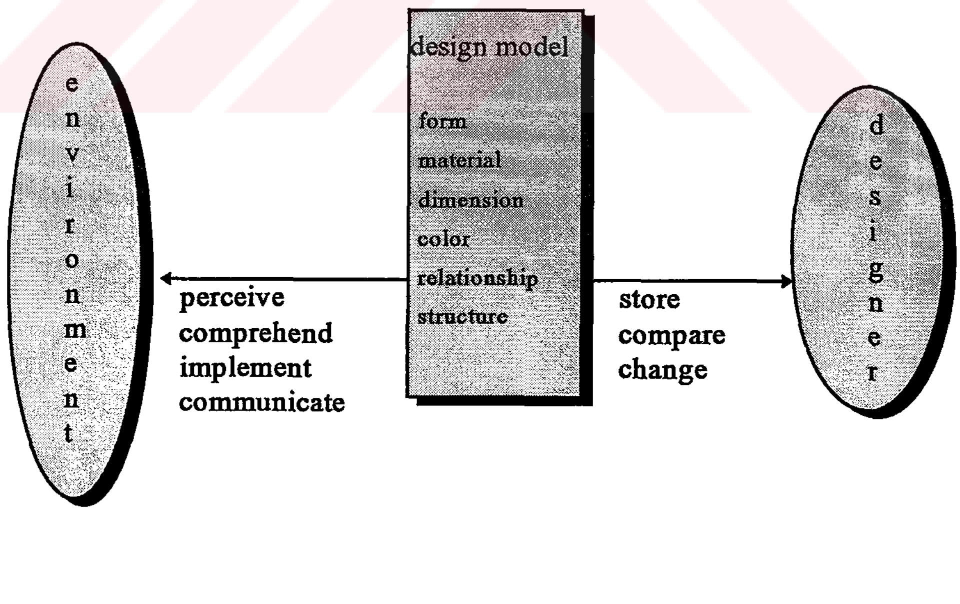

Figure 2.1. The design model

The design process, also, depends upon models loaded with various kinds of information (form, dimensions, relations, materials, colors, structure, etc. of space) about the design. The mental design model acquires three aspects. The first one is *information processing*. The mental design model is a dynamic model, meaning that it is capable of updating itself if there happens to be a change in any of the data it contains.

The second one is *interactivity*. The model allows the designer to implement, change and make associations with other models if necessary. Sumner et al. group design problem-solving as the construction of partial solutions on the understanding of the current goals and specifications and evaluation of these solutions according to various criteria and constraints. This process requires the designer to constantly manipulate the mental design model and refine it by checking the aspects of the design against each specification.

The third aspect is *time*. Each architectural mass is based on a mental design model, i.e. it is the representation of a mental design model. However, there are two major differences between the architectural product and its mental design model; the first one is the physical existence, the second is the factor of time.

Architecture can be defined in four dimensions. While the three of these make up the architectural volume, the fourth dimension, that of time, is concerned with the perception of the first three dynamically.

This latter dimension for any architectural building can be determined on a time coordinate that runs parallel to history and can be named as the *actual time coordinate* (atc). On this coordinate, the architectural space is perceived dynamically, and lives through a life span where it is faced with issues like deterioration, maintenance, changes of use, and restoration. This life span occupies a definite time period on the *atc*.

On the other hand, any design model created to carry knowledge about the future architectural building acquires two time coordinates. The first one is (again) the actual time coordinate displaying the time period when the design takes place and is generally prior to the life span of the building. The second one is the *virtual time coordinate* (vtc) offering virtual time periods for the design model to be tested, analyzed and revised, imitating the life span of the future building (Fig. 2.2). On this coordinate, not only the performance analyses of different design alternatives (thermal, structural, acoustics, lighting, etc. analyses) and maintenance analyses (deterioration, resistance to fire, earthquake, etc.) can be executed, but revisions can also be implemented based on the results.

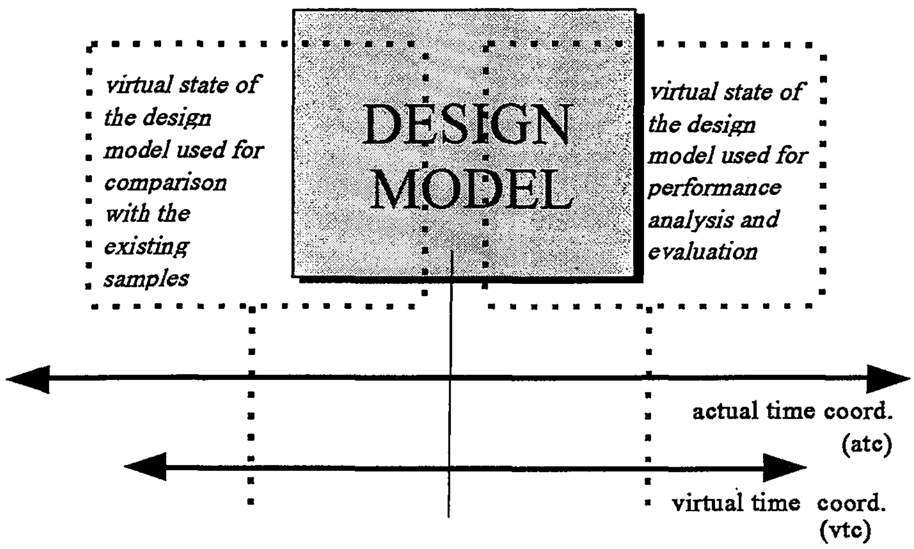

Figure 2.2. Actual and virtual time coordinates of the design model

2.2. The Design Model in Representation and Communication Process

Architectural design communication takes place between the architect and the engineer, the colleague, the customer, the critic, etc. during the process of design. During this communication they refer to the design model, or rather, to the representations of the design model. The designer seeks ways to communicate about the design through various displays of the design model. We may group the techniques for developing and displaying the design model in two; paper-based and digital media. Within the framework of the aspects of the mental design model as discussed above, we now evaluate both media.

2.2.1. Paper-based Media

In the long history of architecture, the media used extensively to display the design model have been the paper-based (drawings and mock-ups) and the verbal. Sketches, detail drawings, plans, elevations, sections, perspectives, diagrams, axonometric drawings depicting the architectural design and description of architectural designs through texts, and other written material are included in the paper-based techniques. However, paper-based media can only represent the design model partially and statically. Because, be it any kind of drawing or mock-up, it reflects the state of the design model at a certain point on the *atc*, and another on the *vtc*, the two points not corresponding to each other. Such a representation refers to a certain time on the *vtc*, and the result is a static representation displaying the design model at that virtual moment, with limited amount of information relevant to that moment only.

Therefore, in the paper-based representations of architectural design there always occurs a difference, a gap between the design model and its representation. The design model in the architect's mind is revised as he thinks, talks, and consults about the design. However, this revision can not easily be applied to the design presented with the paper-based media, unlike the mental design model.

To illustrate this, imagine the exterior perspective drawing of a building. The drawing is completed by the architect at a certain date which denotes the actual time coordinate of the drawing. The drawing depicts the building at a certain hour (determines the angle and intensity of sunlight to be shed on the building facade) during a certain season which indicates its virtual time coordinate. As this virtual time coordinate consists of one point on the *vtc*, the information covered by this drawing is limited to that hour in that season and to the materials, colors and proportions shown on that drawing. Although the architect may decide to change the proportions of the windows, it will not be possible to show the revision until a new drawing is prepared. If there will be a question about the view while looking from inside to the outside from one of the windows, the current perspective will not supply the answer, and a new drawing will have to be made.

2.2.2. Digital Media

Burden lists the digital media used in architecture to include the digital distance measuring devices, stereophotogrammetry, optical digitizing, interactive movie map, 3D computer model and - as everything that can be digitized can be simulated (Binkley 15) - all kinds of simulations made by the computer and by the digital camera.

Photomontages gather scanned/digitized photographs with the computer based design proposal displaying a still frame or photograph as if the proposed design is inserted in the frame.

Stereophotogrammetry makes use of contour data from a clay model which is mapped by a precision camera. The data obtained are then digitized to obtain a computer model.

Optical digitizing involves the use of a video camera to record and digitize the data which are then transferred to the computer for further processing.

Making of a 3D computer model is the process that takes place in digital format from the start until the end. It covers design steps from initial ideas to final design which are both input and implemented digitally. Walkthroughs and flythroughs are the animations obtained from these models, which can either be displayed on a frame buffer or can be recorded.

The converse approach of optical digitizing is called sequential simulation; this indicates the recording of a completed 3D CAD model on video format. Sequential simulation can either be based upon slides taken from a 3D CAD model and recorded sequentially, or upon a sequential mix of animated drawings with the still ones.

Interactive movie map is a video-disc based system controlled by a computer. With the use of such a map it is possible to walkthrough the spaces which are prerecorded or to overview the space from dynamic viewpoints with differing scale and perspective. Users of interactive movie maps are free to determine their own routes.

Finally, virtual reality is a digitized make-believe environment, where the user gets the feeling of having dived into the space and not only can define her own route, but can alter the environment as well.

In order to refer to the possibilities offered by the above digitally operating units to the field of design, it is suggested to employ the umbrella term 'digitally aided design' (DAD). Therefore, being more specific, digitally aided architectural design (DAAD) can be mentioned. Consequently, the long used terms of computer aided design (CAD) and computer aided architectural design (CAAD) turn out to be subsets of the sets defined above.

DAD can be defined - based on Kalay's definition of CAD - as the means to solve design problems, present the design proposals and simulate the results of various analysis on these proposals with the aid of digital media mentioned above.

The digital media provide a memory which can hold one or more algorithms and the input data, and are capable of applying the algorithm to the data and displaying the result. The trivial form of this can be seen in the digital measuring devices, whereas the most sophisticated case is the virtual reality environment.

As everything which is digitally coded is virtually real, the territory of digital operations is also virtual, corresponding to a range on the *vtc*. This range on the vtc forms the virtual work environment for the architect where the different states of the design model during various analyses can be simulated (Fig. 2.3).

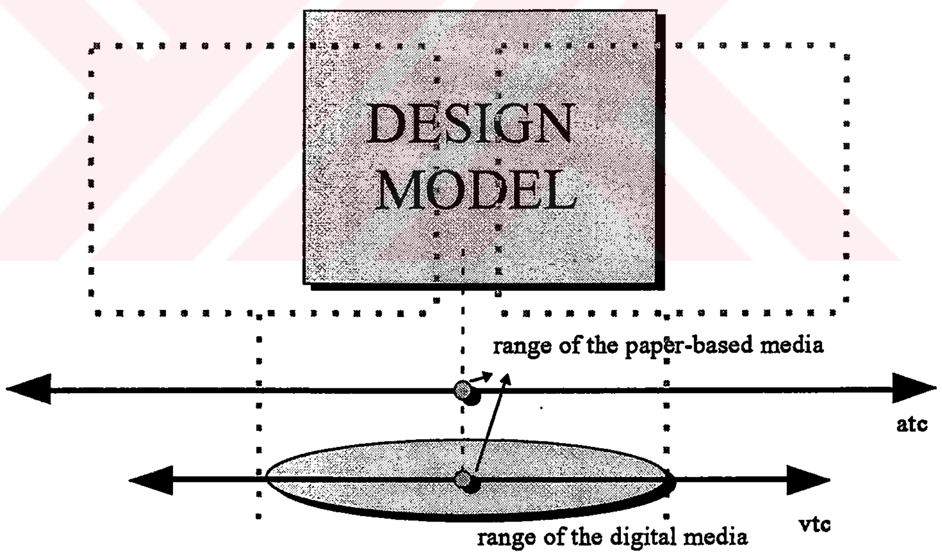

Figure 2.3. Ranges of representation of the paper-based and the digital media

The impressions gained from these *virtual experiments*, as Mahdavi calls them, that substitute the real ones result in various revisions. The representation of the design model as a digitized model then, turns out to be 'dynamic' which can continuously be updated similar to the mental design model.

A computer model then, turns out to be 'dynamic' which is subject to continuous change -since it is easier than throwing away paper drawings or hand-made models- as a result of the 'feed-back' process.

> Models are simulations of the real world. They can be static models, simulating the real world at a given point in time. An architectural plan is an example of this. Models can also be dynamic, simulating the real world seen over a period of time and allowing a study of the consequences of actions. In other words, the dynamic models give us the capability to describe changes, and unlike static models, are not rigid and can offer a great deal of flexibility. Therefore, they offer a possibility to oversee the consequences of different directions or courses of actions. (Beheshti and Monroy 154)

To illustrate the dynamism of this digitized model, we go back to our previous example and imagine the exterior perspective of the building on the screen of a computer with a high capacity. The perspective, depicting the façade at a certain hour during a certain season can quickly be altered to render the state of the same facade if the hour or the season or both were to be changed. Furthermore, changes like the proportions of the windows can be tested on the same drawing with ease.

And, inquiries about another view taken from inside looking outside or the material properties of the surface cladding can be answered within a short period of time.

2.3. The Familiar Design Environment

Based on the discussions above, we may point out that the paper-based media do not display the following three properties of the mental design model (whereas they are displayed by the digital media):

- dynamic perception of space
- performance analyses and
- instant adaptation.

These three properties depend on information processing, interactivity and dynamism in time. Although an architectural drawing made by the architect using pen and paper can be loaded aesthetically, it is very limited in providing design information, interactivity and dynamism.

Paper-based techniques alienate the architectural product from the design model. They only display parts of the design model and display them statically. Hoffman sees the representation to be at the service of the idea, not necessarily the final product, the actual building. The efforts in learning to design are directed toward the creation, development and presentation of the graphic tokens. These abstractions remain untested if the building is not constructed.

According to Hoffman "The distance between the representation of the thing and the thing is inherent in this process. Learning to design within an academic world is, in a large measure, learning to bridge this distance" (1). It is not only the distance between the design model and its representation but, the distance between the model and the actual building as well.

Paper-based representations become referents for themselves, loaded artistically but weak in providing design data. This indicates that the paper-based media fall short of displaying and processing design information carried by the design model. Thus, they introduce abstractions of selected design data.

Paper-based techniques are based largely on this abstraction. The abstraction is graphical in drawings, verbal in writing, and speech and physical in mock-up models. The value of the abstraction remains artistic most of the time in all of the above mentioned media. We enjoy the sketches made by an architect aesthetically, not caring much about the information delivered by that sketch.

However, "... through the means of three dimensional modeling programs and the emerging possibilities of virtual reality displays, the computer offers a direct way to deal with the elements of architectural design as a composition of three dimensional entities rather than a simple collection of lines" (MacLeod 55).

The architectural space in the virtual environment enables the display of the architectural entities with the properties of the architectural elements they represent, i.e. the display of a wall is an entity of a certain height, width and depth, in a certain position, of a certain material rather than a prism formed by various lines. This creates a familiarity between the design model, its representation, and the actual construction through the shared symbols. These symbols carry equivalent information in each case (model, representation and building). On the other hand, the abstracted graphic tokens of the paper-based media do not carry as much information as the model or the building.

In the virtual environment our mental images turn into visual ones loaded with design data displayed upon request. Lanier refers to the language of virtual environment as a post-symbolic one, indicating that in the physical world, we are not able to make physical changes quickly unless we form words that refer to all the possible changes wished to be made if possible. "... In a good shared virtual reality system, you can just directly make up the objective world instead of using the symbols to refer to it" (quoted in Porter 69).

It may then be argued that, a well developed virtual environment is in fact, a very familiar design environment. If designers have had the possibilities of such an environment instead of the paper-based techniques, discussions today on the familiarity of the architectural software would not be based on their similarity to the paper-based techniques.

3. ARCHITECTURE IN THE VIRTUAL ENVIRONMENT

The digitized model in the virtual environment transfers the mental design model to a medium where it can be shared and criticized by people other than the architect; where various analyses can be carried out and the results of both the critics and the analyses can be used to change or improve the design. The medium mentioned here is not the computer screen, but the virtual environment offered by the computer and other digital media.

Architecture in a virtual environment promises a powerful future, not only because of the ease of adaptation as a presentation medium, but because of its advantages regarding the ease of change and intervention to the design before actual construction. Free of physical damages it may be used as an efficient medium in construction tests both for educational and practical purposes. Seeing the results of any changes applied to the structure is especially of importance not only to architecture students learning to deal with structures, but to architects who attempt to make structural changes (like pulling down a wall, omitting several columns or adding new ones) within the space that was designed beforehand.

Moreover, all these advantages are present both for the architect and the client. Based on these possibilities, current applications and projects include virtual office layouts for office design; virtual kitchens where the purchasers may modify the layouts before buying; housing layouts; air-conditioning, lighting, and acoustical tests to be applied by customers to their own homes.

Visualization in virtual environments may not only be considered as a radical new approach to architectural design, but for design communication as well. If designers design in the presence of both consultants and the client in a 3D space, testing the results of design decisions by seeing them in full scale as if in the actual setting, is bound to change the procedure of the whole design practice.

3.1. Architectural Space Simulation Software

Within the coverage of DAD and CAD there are various software packages used for 2D drafting, 3D modeling, rendering and animation purposes. This thesis concentrates on the architectural space simulation software which enable 3D modeling, animation (simulation in motion) and information processing of the designs. Among the currently used such software are Autodesk's AutoVision, 3D Studio and 3D StudioMax, Intergraph's ModelView, and Alias/Wavefront's ArcVision (S1; 2; 3; 4; 5).

It is useful for an architect to be able to simulate architecture in motion (Greenberg 540; Amor 19-20). Greenberg states that, one of the principal concerns of architectural design is space and the architectural space can be classified as the interior space of a building and the external space of the building and/with its setting. We do not react to none of these spaces from a static position like viewing a painting, but perceive them dynamically. Consequently, Greenberg suggests: “To obtain a deeper understanding of architectural space it is necessary to move through the space, experiencing new views and discovering the sequence of complex spatial relations” (540).

Mark defines architecture in motion as the changes of visual image of a building when observed in real time. These changes may be due to:

i. changing of the observation point
ii. variation of light
iii. variation of use
iv. relocation or transformation of building parts (14).

The architectural space simulation software provide the simulation of the architectural space with respect to the above factors. The simulations realized through such software can be grouped as the walk/flythrough and the virtual reality applications.

3.1.1. Walkthrough and flythrough

Walkthroughs and flythroughs can be regarded as the digital tours within the architectural design model in the digital format (Mahoney 23). The essential difference between a walkthrough and a flythrough is that the former takes place as if one is walking in a space, i.e., space is observed from the eye height and area of movement is restricted by physical boundaries, whereas the latter creates the feeling that one is flying through the space, i.e., observing the space from bird's eye level and capable of going through every wall and window.

While walking or flying through the space, the simulations introduce the possibility to experience the proposed building from various points, study shadow effects, illumination and color scheme quality (Witte 93). The simulation may also show how the building actually works (Emmett 31) and this fulfills the task of displaying and analyzing both criteria of the fourth dimension in architecture before actual construction.

Both walkthroughs and flythroughs can be classified as architectural animations. These animations display the space in 3D, walking or flying through the space on a pre-determined path. Using the architectural space simulation software it is possible to create walk/flythrouhgs either with packages that include both drafting and animation capabilities like Nemetshek's ALLPLAN (S6) or with supplement software that work with a 2D package like EaglePoint's AutoPro (S7) for AutoCAD. There are also animation software that operates upon imported files from any 2D drafting package like Autodesk's AutoVision and 3D Studio, MicroStation's Visipix, Intergarph's Model View, Alias/Wavefront's ArcVision and Virtus' Virtus Walkthrough Pro (S1; 2; 8; 4; 5; 9) .

3.1.2. Virtual reality

As a term, virtual reality (VR) has been used from 1980's onwards. The term was put forth by Jaron Lanier whose aim was to differentiate existing types of computer simulations by then, from the digital world Lanier was working on (Porter 61). VR's first introduction to the public was in 1989 and ever since, it has been used in many fields, especially in the entertainment industry, military purposes and medical applications.

The world created in the VR environment is a space, called cyberspace, where one can enter and interact with. Using special monitors and scanning devices that give a 3D view, the user finds herself in a computer-generated world and using movement and gesture tracking devices, can move the elements within that world. The feeling of immersion, interaction with the elements and the lack of need to pre-determine the path to be followed within the space are the main differences between walk/flythroughs and the VR environment.

Gromala refers to the virtual reality system as "a highly developed multi-media environment," (6) adding that the environment in which the user enters is a sensory computer-generated space, with high degree of simulation capability. As the users are isolated from all outside stimuli, they get the feeling of having entered completely inside a computer-generated environment, where the displayed view constantly updates and changes itself with respect to the viewers' position and the modifications they have made. Thus, designing in the VR environment means creating an interactive and visual database for the design as well.

Although there exist several technical problems, VR can be considered as "the ultimate example of ideal human-computer interface" (Brill 48) where human beings not only meet with the cyberspace but the two mutually influence each other as well (Thomsen 183). Recent problems in the visualization of virtual environments include the lack of ability to import objects, integrate sound, display in more than 256 colors and communication with other applications (Von Schweber and Von Schweber 170-6). VR is believed to have a wider field of use when "...improvements in the design of interaction and display devices, user interfaces, development tools and applications are introduced" (Singh et al 35).

Until the technical problems regarding the creation of an immersive artificial environment are solved, augmented reality systems are introduced to create the artificial environment by superimposing computer graphics onto the real environment where the user stands (Tatham 348). The user then sees the real environment combined with the images of the artificial environment either through silvered-mirrors or head-mounted display units. Although the possibility of interaction is very low at this level, the feeling of immersion is achieved.

Among the currently used simulation software for creating virtual reality applications are Apple's Quick Time VR, IBM's 3 DIX Interaction Accelerator and Sense8's World Tool Kit (S10; 11; 12).

3.2. Ideal Interfaces for the Architectural Space Simulation Software Packages

The digital format is the numeric system into which the input data are converted to be executed by assorted algorithms. The input data may be analog, formed by physical entities (like electric flow, voice, etc.), or digital, in the form of electric pulses. In order for input data to be processed due to a prescribed formula (or formulas forming an algorithm), they have to be digitized. Digital format is the language of operators working with algorithms. These operators can be referred to as the digital media such as computers.

When the digital media are employed, it is not necessary for the user to make the data conversion from analog to digital, thanks to the interfaces. Interfaces are assigned to the task of converting user's analog input into digital format and converting the digital result into an analog output. It is through the interfaces that the user interacts with the digital world and uses it for her purposes. The success of the interface in enabling the user manipulate the digitized data usually indicates the level of interaction between the user and the computer. As such, the efficiency of the use of computers in any field for various purposes depends largely upon the creation of successful and efficient interfaces.

The virtual design environment poses new questions for the architects. These questions arise from the fact that this environment offers a strong potential for high-end architectural simulations yet its use and efficiency still depend on a well-designed interface enabling flexible use of the software. Laurel (quoted in Pimentel and Texiera 157) names the need of the architects as a "well-designed setting";

> We get to play 'what if' in an organic world where everything is there for a purpose. This is what virtual world designers need to be investigating, this is what's new about this media. We will have to design in cues, clues, and overviews to serve as advance organizers for travelers new to the territory. The key to a great experience is going to mean a well-designed setting.

The properties of such a well-designed setting can be grouped in the following manner, with respect to the problems of the architects in using the architectural space simulation software:

Context-Specification: The flexibility in using the architectural software package is important for the architects trying to get used to the new medium of design where definitions are changing. This flexibility can be achieved through the opportunity of *context specification* for the software. The context is determined based on the analysis of purpose, and profiles of the user and the audience.

Analysis of purpose examines the needs and the tasks to be performed by the aid of the software. With the emergence of the digital media, the concept of modeling in design broadened. This new concept indicates the whole design process (from initial diagrammatic sketches to final drawings, simulations, and representations) to be carried out digitally, the model becoming the design method itself. Designers, using digital aid, not only build models of what they design; but the whole procedure through which they reach the final design as well. Since with the digital aid in design, the design process itself has become representable, the designer must design the stages of the process accordingly, to share with and to display to the others. This is possible with a software package which recognizes the design stage and offers relevant menu options and aids.

Novak introduces the notion of liquid architecture defining architecture in the virtual environment. He says that music, which was the most temporary of all arts became permanent by technical means like recording and digitizing but, on the other hand, architecture, the most permanent art is becoming temporary by being dematerialized in VR;

> For architecture this is an immense transformation: for the first time in history the architect is called upon to design not the object but the principles by which the object is generated and varied in time. For a liquid architecture requires more than just 'variations on a theme,' it requires the invention of something equivalent to a 'grand tradition' of architecture at each step. (251)

Ideally, each step in the design process must be handled by the software with the appropriate services and operations for that stage.

Architecture is a profession can also be practiced by non-professionals by pragmatically building small gadgets, modest structures, and organizing the interiors that they live in. Consequently, in the virtual environment, architects are by no means the only designers by definition. Any participant (client in our case) may implement the environment. Lanier (quoted in Porter 4) draws a future picture where people will be able to change their environment decorated by new virtual furniture as soon as they return home and put on a pair of glasses and gloves. Therefore, the interface of such software is expected to be designed to serve not only for the architect but for the customers as well. Thus, not only the analysis of the task but, independence from the background and the experience of the user in using the software become issues that determine the context of use. The interface enabling the user to specify the context is then responsible of the provision of a relevant setting, a relevant work environment for that context..

Compatibility and Standardization: Architects complain about the problem of having to use more than one package in order to realize a complete animation. Initiating the design with a 2D drafting software, the architect then employs a 3D modeling package and an animation or rendering software to complete the presentation (Kempfer 48; Mahoney 53). The problem is further enhanced if the selected or available software within the above work chain are not compatible with each other. The ideal interface is expected to solve the dissatisfactions occurring because of the compatibility problems.

If *compatibility* between the architectural space simulation software is not achieved, the tasks executed by using different software can not be brought to a standard. *Standardization* of the computer aided design tasks is crucial for design firms and for competition purposes. If certain standards are set, a design firm can produce design alternatives in the same format enabling easy comparison in between. Also, competition submissions may be prepared in the same format for ease of evaluation. Ideally, the task of executing the standardization should be loaded on the interface of the software.

Efficiency: The amount of time required to design in the virtual environment changes with the time required to learn the software, to adapt to its ways of functioning and to make all the design decisions required by the software, as was mentioned in Section 1.1. Longer the time to design in the virtual environment gets, less the satisfaction of the user will be. Requirement of programming skills (McLaughlin 86) to operate the software and having to deal with non user-friendly interfaces (Mahoney 54) add up to the dissatisfaction.

The ideal interface is the one that reduces the learning curve, speeds up adaptation of the user, lessens the amount of design decisions required to execute a task, overcomes the requirement of programming skills and introduces a user-friendly environment. We refer to all these assets as the *efficiency* of the interface.

Familiarity / Mastery of Functions: Given the opportunity to easily master the operations and functions of the simulation software, architects may begin to regard the issue of familiarity as the ease of learning and using the software, rather than similarity to the paper-based techniques. The issue of familiarity should be discussed in relation to Sutherland's definition that the world we live in is;

> ... a physical world whose properties we have come to know well-through long familiarity. We sense an involvement with this physical world which gives us the ability to predict ... where objects will fall, how well-known

> shapes look from other angles, and how much force is required to push objects against friction. We lack corresponding familiarity with the forces on charged particles, forces in non-uniform fields, the effects of nonprojective geometric transformations, and high-inertia, low-friction motion. A display connected to a digital computer gives us a chance to gain familiarity with concepts not realizable in the physical world. (506)

To acquire such familiarity is possible through the use of a software environment which the designer can easily manipulate with the aid of the interface.

True Modeling: Regarding Mitchell's definition of architecture, a new aspect for the adequate interface design can be added. He defines architecture;

> ... to be concerned with the skin-bounded body and its immediate sensory environment - with providing shelter, warmth, and safety, with casting light on the surfaces around it... Now they must contemplate electronically augmented, reconfigurable, virtual bodies that can sense and act a distance but also remain partially anchored in their immediate surroundings. (43)

Accordingly, the idea of the *trueness* of the design model in the virtual environment can be introduced. The design model is expected to be related not only to the mental design model but to the actual setting as well. Thus, the ideal interface should aid in creating the model as true as is expected by the audience at that specific stage.

Determining the levels of 'trueness' of the virtual model is a difficult task. It requires optimization of various criteria such as the expectations of the user, expectations of the audience and technical factors. If the interface of the software can execute this optimization, the user will be relieved of a heavy burden.

However ideal the interface may be, it is still a difficult task to introduce a new design environment that requires different methods of handling, both in drafting and design, from those that the architects were accustomed to. Many architects will naturally long for the old techniques if they are faced with difficulties in understanding and manipulating the new environment and will complain that the newcomer is not familiar. Therefore, we need a transitional link, a flexible platform to enable the designers to get used to the new design environment by not adjusting the way they design according to the environment, but vice versa. The next chapter concentrates on the introduction of such a transitional link between the designer and the virtual design environment.

4. DEFINITION MODEL FOR ARCHITECTURAL DESIGN IN THE VIRTUAL DESIGN SPACE

To construct this transitional link, we have looked at both sides of the architectural software's interface. On one side there stands the user with a special profile and a unique purpose in using the package. On the other side, there is the digital environment with standard and rigid menus to offer. The problem in the meeting of the user and the digital design environment is that every architect has her own approach to design, while the software package provides a general, predetermined platform for design. The interfaces of most of the commercially available software packages are like a one-sided mirror glass that act like mirror from one side, and like glass from the other. The user can see through the interface, but the interface does not recognize the user.

In order to enable the interface to recognize the user and her intentions we need a common platform of definition. Current interface design deals with the both sides as constant scales. In other words, current interfaces respond within a linear process from wire-frame to high-end rendering to the users on a linear scale of purpose expanding from drafting to animation. We believe that both of the sides have more variables than can be placed on one scale only, that can be addressed within a space of a coordinate system. Therefore, we have taken both the user's and the digital side as two Cartesian spaces.

4.1. The virtual design space

We have converted the factors of the design model in the digital environment: time, interactivity and information processing into the following three scales (Fig. 4.1). Information processing is determined as the level of rendering of the digitized model.

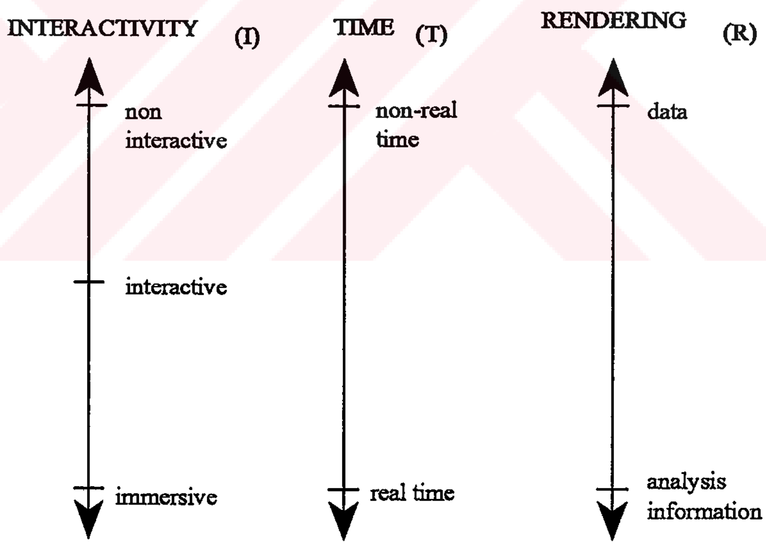

Figure 4.1. Scales of means of architectural communication

Interactivity defines the domain of manipulation of the communicator over the representation. The interactivity scale (I), meaning the possibility to define and execute tasks, expands from non-interactive (where the communicator has no possibilities of data processing or implementation) to immersive (where the communicator is capable of making instant analysis and changes). From the immersive end onwards is the realm of virtual reality experiences.

The time scale (T) expands from non-real time to real time indicating the time of the display versus the actual time required for the represented act. The values of this scale varies between 0 and 1. At the 0 end, we are confronted with the still representations of the architectural design, whereas at the 1 end we have the dynamic simulations. When 1 is reached on the time scale, the representation is real-time on the virtual time coordinate, running parallel to the life cycle of the design on the actual time coordinate.

Finally, the rendering scale (R) expands from trivial and boundary line quality to highly rendered, colored, material conscious, and illuminated versions upon which performance analyses can be applied, and information processing can be maintained. The rendering scale indicates the level of knowledge graphically displayed by the representation. Not only thermal, structural, illumination factors but even the cost factor can be displayed through rendering.

The next step is to test the scales to see whether they can be used to define all possible means of architectural design. Using the three scales mentioned above a 3D coordinate system can be formed at the center point of which paper-based media like drawings of plan, perspective, elevation, and sections can be placed (Fig. 4.2).

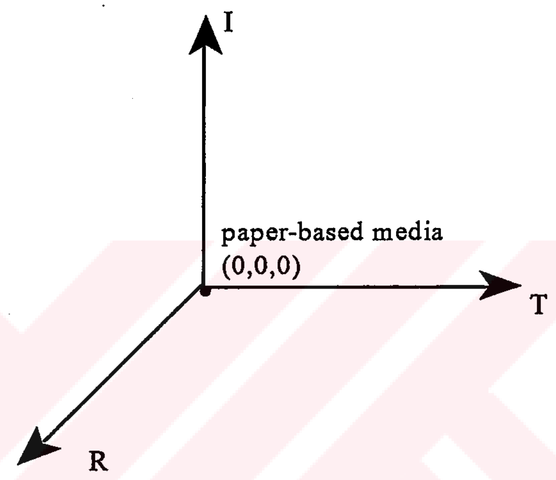

Figure 4.2. 3D coordinate system based on the scales of means of architectural communication

When projected to the three coordinate planes, the relations of the three axes turn out to be as follows. At the intersection of presently available ends of the axis of interaction and axis of time lies the verbal description of an architectural design (Fig. 4.3), since it can answer to various questions as if executing different tasks and can represent the design rapidly.

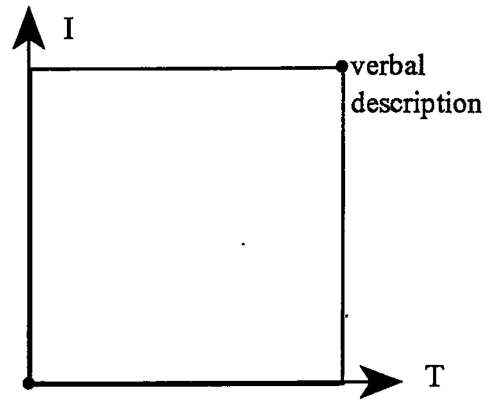

Figure 4.3. Interactivity versus Time

At the intersection of presently available ends of the axis of rendering and time lies a highly developed walkthrough recorded on video tape (Fig. 4.4). It is highly rendered, including color, lighting and displays of various performances of color, lighting and material properties due to different situations. It is also in real-time, without any time lapse.

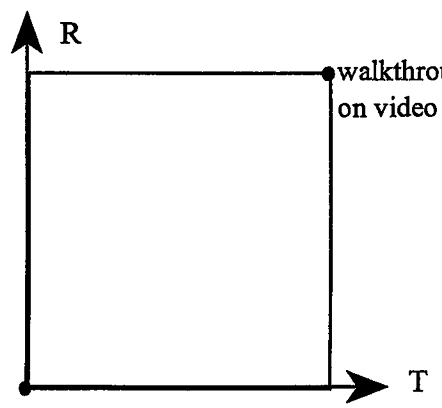

Figure 4.4. Rendering versus Time

Finally, at the intersection of the presently available ends of the axes of interaction and rendering lies the 3D CAD model (Fig. 4.5), which is capable of meeting the needs of different tasks required and is highly rendered with the possibility of performance analysis.

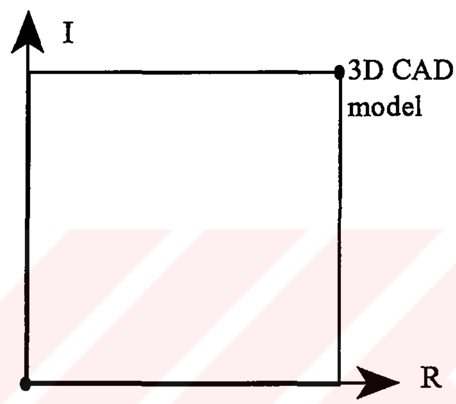

Figure 4.5. Interactivity versus Rendering

The next step is to test the scales to see whether they can be used to define all possible means of architectural design. Using the three scales mentioned above a 3D coordinate system is formed, at the center point of which paper-based media like drawings of plan, perspective, elevation, and sections can be placed. If we construct a cube in this Cartesian space, each vertex can be used to address one of the means of architectural design communication (Fig. 4.6).

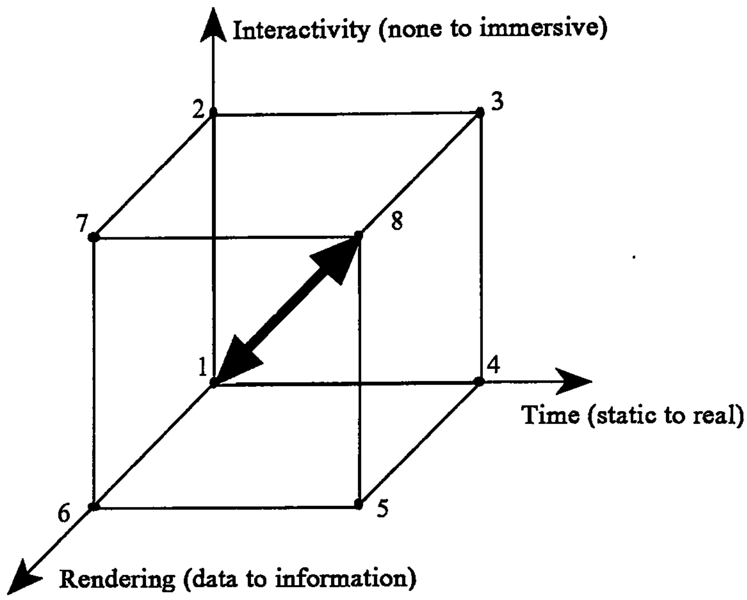

Figure 4.6. Architectural communication as a cube in 3D coordinate system

On the cube, the vertices stand for the following; 1 (paper-based drawings), 2 (verbal description), 3 (augmented wire-frame), 4 (video recording of a mock-up), 5 (walkthrough on video), 6 (still picture of photo real quality), 7 (3D CAD model), 8 (the actual building itself). Different states of the architectural simulations in the virtual environment then, can be defined on a line expanding from vertex 1 to vertex 8.

This space gives the opportunity to address forms of architectural design communication within the dictionary of the virtual environment. We name this space as the *virtual design space* (VDS).

4.2. The space of user+need

The next step is to determine which address the architect wants to go to in the VDS. To achieve this, we have to indicate the factors that make up that address, in other words, the factors in the user's domain that determine the context of simulation in the VDS. This context is unique to each user and to each performance, because the experience level of each user in using the software may differ as well as the purpose in using the software may differ for the same user at different times. The domain of the possible contexts is formed within the Cartesian space of user+need. This space is built based of the following factors acting as coordinates.

1. Purpose of making the simulation/the design stage (P): The currently employed architectural space simulation software are oriented in the same manner towards different kinds of needs, i.e. they introduce the same menu for different purposes, to users with different experience levels in using the software.

Architects complain about the amount of decisions to be given when making a simulation. The problem is not only the amount of time required for making all of these decisions, but the lack of possibility in making such precise decisions at the early stages of design. This problem makes the use of most of the commercially available software packages impossible at the initial, early stages of design. The complaint arises from the fact that the simulation software are programmed to serve in the same manner for different kinds of purposes. As a natural outcome, most of the architectural space simulations are made for the sake of having used the software and reveal little about the architectural quality and structure of the space. Then, it turns out to be a challenge as Emmett points out, for the architects to go beyond the standard applications and reveal the qualities of the architectural design (33).

Architects can get lost within the large range of menus that can lead them to miss the point in making the simulation. They can end up producing some simulation different from the one that was intended (Mahoney 24). Considerations in preparing an architectural space simulation are very closely related to those of moviemaking since both record motion. Nevertheless, it is not relevant to require architects to be educated in filmmaking, as Buday suggests (20), just for the sake of making a successful simulation. It is the task of the simulation interface to aid the user with some default assignments and suggestions.

2. Expectations of the audience (A): The architectural space simulation is always made for an audience. In some cases the audience may be the architects themselves, in others the audience may vary from a colleague, to a customer, an engineer or a design competition jury who all expect to see different aspects of the designed space. Whoever the audience is, architects try to communicate their projects so that the audience perceives it in the same way as the architect does. This is very crucial before the actual construction begins (Kempfer 51).

However, most of the currently available architectural space simulation software packages offer the same level of complexity of display independent of the required type of information to be revealed. A colleague may be interested to see the overall quality of a space while an engineer is concerned with the structural system of the same space. The task of adjusting the software to differentiate between two such displays and to emphasize the required aspect is loaded on the user. In such a case, the user tired of having gone through the complex and tedious process of altering the input parameters to achieve a display (Richens 314-5) or lacking the adequate knowledge of producing a different display for the new audience may end up showing the same display to different audience.

3. Experience level of the user in using the software (E): Since many of the currently used architectural space simulation software requires special training and expertise, we came to speak of someone called the "CAD operator" (McLauglin 86) responsible of executing visualization tasks on the behalf of the designer (Porter 4). This alone, makes the utilization of the software very difficult by students and the customers. Richens observes that computers contribute little to design after 20 years, since they "... are not used by designers, at least not when they are designing" (307). In spite of the recent additions and revisions to the currently used software, the problems of the "long learning curve" (Kempfer 49; Potter 25) and underdeveloped interfaces (Singh 35) still exist, requiring a long time and effort to learn, manipulate

and make the correct selection from the extensive menus. Contradicting the belief that this freedom can be obtained by increasing the number of menus and options, we choose to be on the camp of the *ill effects of too many choices when using products of technology* with Negroponte and assert that the freedom can be maintained by downsizing the menus and the options.

In the Cartesian space formed by PAE, the relations indicate the service that the user expects from the software according to the intention in using the architectural space simulation software. At the intersection of highest ends of the axes of purpose and axis of audience lies the high-end rendered animation with default input (Fig. 4.7). At the late phases of design, the user with no experience in using the software and having to prepare a presentation for the customer or a design jury would like to make a high-end rendered animation with as much help from the computer.

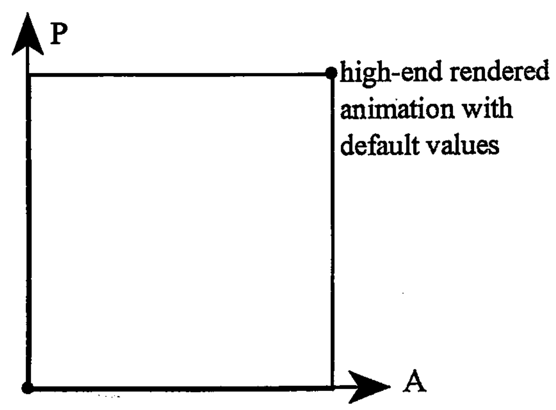

Figure 4.7. Purpose versus Audience

At the intersection of highest ends of the axes of purpose and experience lies 3D model with user input (Fig. 4.8). At the late phase of design, the designers will be content with a 3D model if they have no other intention but to examine the design themselves. They will easily provide the inputs as they are experienced in using the software.

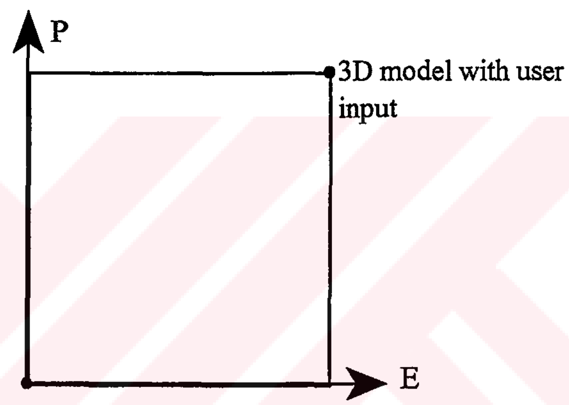

Figure 4.8. Purpose versus Experience

Finally, at the intersection of the highest ends of the axes of audience and experience lies the wire-frame volume study with the user input (Fig. 4.9). The experienced users having either the customer or design critics to examine the design at the initial phases will be satisfied with this volume study, providing necessary inputs themselves.

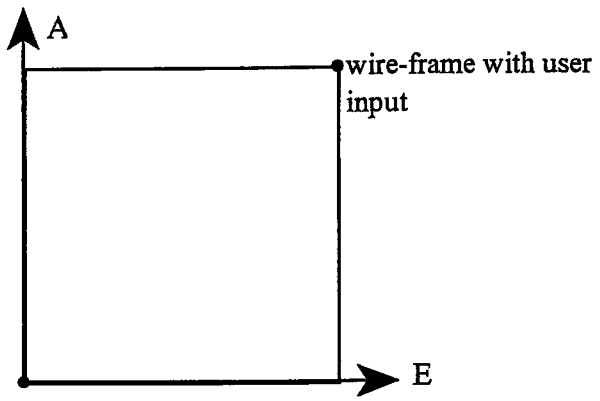

Figure 4.9. Audience versus Experience

The next step is to form the Cartesian space and study the expected services from the software at each junction formed by the purpose, audience and experience. At the center point of the space of user+need is the demo version of the software which the user does not interfere with (Fig. 4.10).

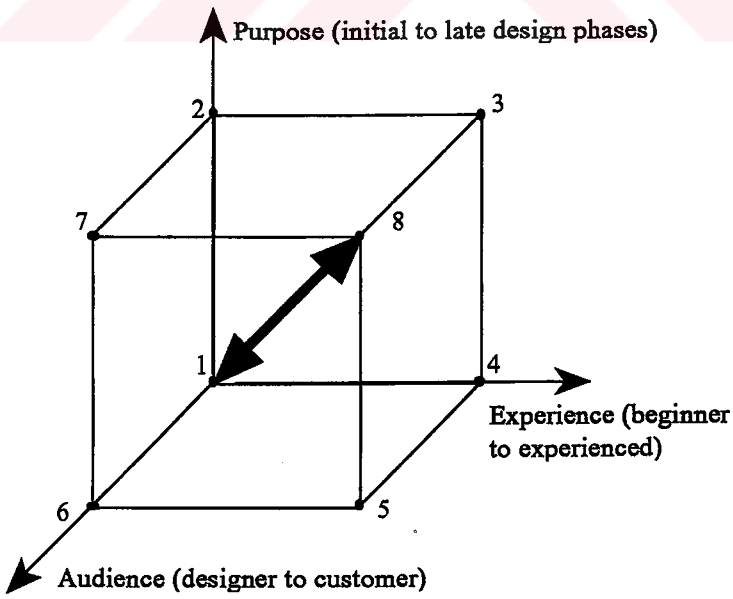

Figure 4.10. The Cartesian coordinate system of user+need

On the cube, the vertices stand for the following; 1 (demonstration version of the software), 2 (3D model with default values), 3 (3D model with user input), 4 (wire-frame with user input), 5 (volume study with user input), 6 (volume study with default values), 7 (animation with default values), 8 (software package running without customization). Different contexts of the architectural simulations in the virtual environment then, can be defined on a line expanding from vertex 1 to vertex 8. This space gives the opportunity to address the profile of the users and their specific needs. We name this space as the space of *user+need*.

The relationships between the P, A, E and the levels of I, R, T are as shown below (Fig. 4.11), though each arrow may weigh differently.

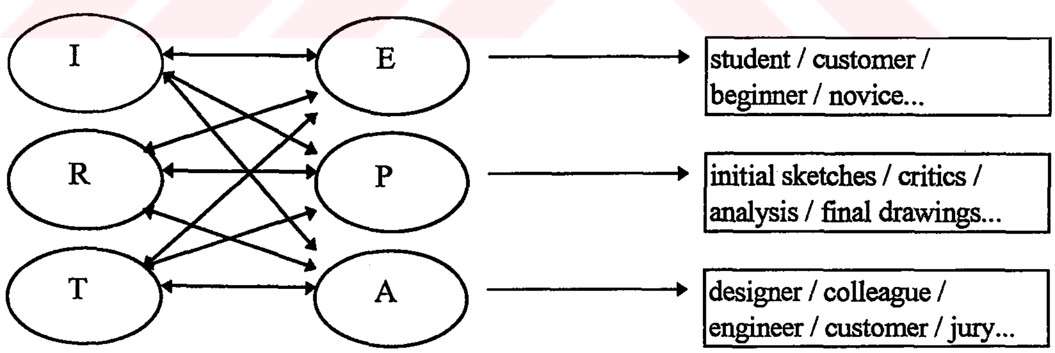

Figure 4.11. Relational scheme between scales of visualization and properties of the user

4.3. Customization scale menu (CSM)

Richens asserts that for a CAD system to gain wide acceptance, it should be based visually and freed from theory, rule and knowledge (Richens 308). The user, independent of the rules, theories and programming knowledge, should be able to indicate her aim in using the software and the software must be intelligent enough to customize itself according to the user's needs. Within this framework, the customization scale menu (CSM) is introduced to enable the user indicate her choices before using the architectural simulation software. The menu is responsible of running the program according to the indications given, thus narrowing down the choices and making some automatically. CSM is an application-independent model, the implementation of which should be adapted to each software specifically.

CSM is the model of a transformation design which transforms a state in the PAE space (the space of user + need) to a digital aid in the IRT space (the virtual design space). The relationship of the PAE space with the IRT space is shown below.

$$I = f_I(P, A, E)$$

$$R = f_R(P, A, E)$$

$$T = f_T(P, A, E)$$

Formula 4.1. Relationship of the PAE and IRT spaces

This relationship is basically the addressing of the equivalent in the I, R, T space of a point selected in the P, A, E space. This can be explained with the below transformation matrix.

$$\begin{bmatrix} I \\ R \\ T \end{bmatrix} = \begin{bmatrix} u_I & v_I & w_I \\ u_R & v_R & w_R \\ u_T & v_T & w_T \end{bmatrix} \begin{bmatrix} P \\ A \\ E \end{bmatrix}$$

Formula 4.2. The transformation matrix

In this transformation matrix P, A and E have values corresponding to the choices determined by the user; u, v and w's are the parametric coordinates that add up to 1.

$$u + v + w = 1,\ 0 <= u <= 1,\ 0 <= v <= 1,\ 0 <= w <= 1$$

Formula 4.3. The parametric coordinates

According to the choices of the user, the value of each parametric coordinate is determined. Thompson sees the governing paradigm for the virtual interface design as the potential of an infinite number of domain-specific requests versus the standardized set of metasystem requests (12). In other words, the interface must offer standardized sets of the expected, most-used, and common requests while being able to meet the specific requests of the user during each use. The above transformation matrix meets both types of requests. While PAE define the standard requests, the parametric coordinates determine the domain-specific ones. This enables two functions to be achieved by the interface. The first one is the user satisfaction, giving the user the possibility of customization according to the most important factor among P, A and E. For instance, the importance of the experience level may matter to the user more than the purpose and the audience for a specific case. The values of the parametric coordinates reflect this difference to the transformation.

The second one is the possibility of history-based operations. The parametric coordinates enable the context specification to be saved in journal files. Therefore, different users of the same software may save the context suitable to them under their names. Thus, while using the software they may always access the user-specific context. Or the user may save the state of the software for a specific purpose. For instance, students may save the state of making a jury presentation and run it every time they prepare for a jury.

The transformation from the user's requirements to the virtual environment specifies the context for the interface, indicating the global context for the software to run at. This context is formed into the intervals on the three scales of I, R and T (Fig. 4.12).

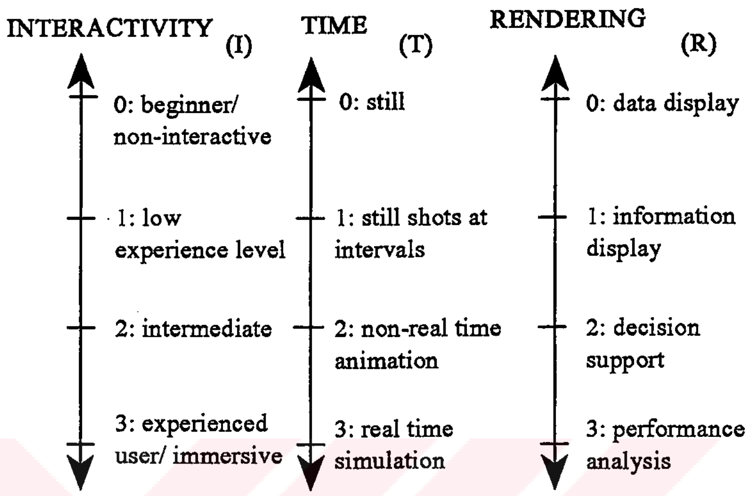

Figure 4.12. Scales of means of architectural communication

The value of I is the determinant of the default value assignments. If the addressed I is between 0 and 1, the default values will be assigned for various menu options. If the value of I is between 1 and 2, a default value will be displayed for the same menu options with the possibility of user's intervention and if it is between 2 and 3, the user will be handling the assignment of the menu option.

The value of R indicates the level of information processing on the model. If R is between 0 and 1 the display contains data only, but analysis of the data is impossible at this interval. If the value of R is between 1 and 2 , information is provided in relation to the database. If the value of R is between 2 and 3, the results of a performance analysis are displayed.

The value of T determines the speed of the display. If the value of T is between 0 and 1 the display is still. If the value of T is between 1 and 2 the display consists of snapshots in sequence and if it is between 2 and 3 the display is a motion picture.

Within this framework, each context determined by the (I, R, T) set has a corresponding assignment (default, suggested value with the possibility of intervention or user's choice) for each menu option. The possible 27 different states of this set are shown below (Fig. 4.13).

Default	Suggested	User choice
(I,R,T)	(I,R,T)	(I,R,T)
(0,0,0)	(1,0,0)	(2,0,0)
(0,0,1)	(1,1,0)	(2,1,0)
(0,0,2)	(1,2,0)	(2,2,0)
(0,1,0)	(1,0,1)	(2,0,1)
(0,2,0)	(1,0,2)	(2,0,2)
(0,1,1)	(1,1,1)	(2,1,1)
(0,2,1)	(1,1,2)	(2,1,2)
(0,1,2)	(1,2,1)	(2,2,1)
(0,2,2)	(1,2,2)	(2,2,2)

Figure 4.13. The states of the IRT set

Special states of CSM (0,0,0), (1,1,1) and (2,2,2) determine the extents of the program application. The state of (0,0,0) indicates the use of the software as a paper-based tool, not different from any of the paper-based methods. The state of (1,1,1) indicates the state at which many of the commercially available software presently run. Finally, the state (2,2,2) indicates a level of immersion in the virtual environment where the user is in control of every facility. To us, the most important states are the intermediate ones, which cannot be realized alone without the CSM.

4.3.1. Implementation

The implementation of the context-specific interface model is realized with Microsoft's Visual Basic. While Visual Basic is a procedural language, it is not in the same class as the other productivity tools. It is representative of a different approach to program development, and is used as the macro language for the various Microsoft Tools. By working with such a macro language, it is possible to develop an interface model that will easily be adapted by various software.

In the choice of Visual Basic the following criteria were considered. It provides an environment in which interfaces to programs are quickly built. It will automatically create all of the necessary code to handle user interactions. It implements an event-driven programming model -program control is done by clicking the mouse or typing characters, etc-. It is also object-oriented in that each part of the system is an object with an associated set of code and attributes.

Although the introduced interface is a model to be adapted by each software package, the guiding factors of the cognitive principles of interface design (Thompson 10-2) were still considered. The first one of these factors is the consistency between applications, i.e. a common language and structure for all the interfaces. In our model we have utilized English as the operation language and have utilized the interface elements (such as buttons, dialogue boxes and pull-down menus), and common interface operations (such as clicking, pulling down and highlighting) employed by the widely used systems like Windows 95 and Windows 98, System 7.0, Mac OS (S13; 14; 15).

The second cognitive principle indicates simple, unambiguous and shallow command structure. Therefore, we have picked up the most straight forward commands and descriptions via keywords. Finally, following the principle of offering the selections rather than expecting the user to input from scratch, we have provided lists of selections at each step. This does not require the user to remember the tasks of the previous steps to be able to proceed.

CSM is a model to be integrated to any architectural space simulation software with necessary adaptations required by the software. To illustrate a sample integration of CSM to an architectural simulation software the following screens are designed as illustrated in Appendix A:

1. The initial screen alerts the users that the particular software they have accessed introduces an in-built CSM allowing the user to customize the software (Fig. A.1).

2. The next screen allows the user either to continue accessing the software without customization or to use the CSM to customize the software (Fig. A.2).

3. The next screen welcomes the user to the customization process (Fig. A.3)

4. The next screen briefly explains the user how the customization will be done (Fig. A.4).

5. The next screen introduces CSM with three options in three categories for the user to indicate her preferences (Fig. A.5).

6. The final screen in the process informs the user that further customization can be made while using the software through the on screen customization bars (Fig. A.6).

The above described screens have the common features like the CSM logo at the upper left corner and the button for more information at the lower left (except for the CSM screen as illustrated in Figure A.4)

When the user clicks on the CSM logo on either of the displayed screens, a brief explanation about the production of the menu appears (Fig. A.7).

When the user clicks on the 'tell me more about customization' button on any of the screens, a series of explanatory screens follow. The series consists of five consequent displays, giving information about customization, customization options in general, and customization options one by one 'purpose', 'audience', 'experience' (Fig.s A.8; A.9; A.10; A11; A.12). The user has the opportunity to proceed back and forth between these displays or to return back to the customization process from any of these displays.

The user can start using the architectural space simulation software which has an in-built CSM via two ways. The first one is to select not to customize the software and click on the 'skip' button on the second display (Fig. A.2). The second one is to complete the customization process and click on the 'continue' button on the last one of the displays (Fig. A.5). In either way, the software will initiate its run time (Fig. A.13) and introduce a blank layer on the screen (Fig. A.14). The only difference of this blank layer of the software with the in-built CSM is the addition of the CSM logo.

The logo when clicked during the run time of the software, turns into CSM on screen. This scale menu allows further customization for those who are not satisfied with the current state of the customization. Further customization is based on the transformation matrix (Formula 4.2) introducing the scales of P, A, E and I, R, T as scroll bars and the coefficients as number slots (Fig. A.15).

The PAE and IRT scroll bars allow scrolling between 0 and 2. Each scroll bar can be adjusted to 0, 1 or 2, each number coinciding with a state of the scale it belongs to. Such as, 2 on the scroll bar of Purpose (P) corresponds to the late design stage or 1 on the scroll bar of Time (T) corresponds to the still views of the design. The state which the number corresponds to is displayed on a bar situated over the scroll bar.

The coefficients which determine the weight of each of the three determinants P, A, and E can be adjusted between 0 and 100, which corresponds to a value hundred times the real coefficient value. This enables fine adjustment of the coefficient values. By adjusting the coefficients, the users, for instance, can reduce the weight of purpose (P) and audience (A) in the customization process, and indicate experience (E) to be the most important determinant. They may then select the values of P and A as 22/100 (0,22) each and allow E to be the most dominant determinant by 66/100 (0,66).

While making on-screen customization, the user can utilize the 'fix' buttons to freeze the value of one or several variables and observe the effects on the customization of the software.

Information about further customization (Fig. A.16) is available through the selection of the 'tell me more about further customization' button on the last display of the customization process (Fig. A.6) and the on-screen CSM (Fig. A.15).

After the customization process is completed, the simulation program runs with a certain menu behaviour, specified for the customized context. For illustrating the context-specific behaviours of the software, the commercially available space simulation software package; Virtus Walkthrough Pro (S9) is selected.

The choice of Virtus Walkthrough Pro depends on the fact that the program gives is one of the most used software packages designed to execute simulation tasks only. In other words, Virtus Walkthough Pro is an architectural space simulation program, as was described in Chapter 3, used for simulation purposes mainly, rather than packages which combine and give importance to drafting, designing and rendering. Among the other simulation programs, Virtus Walkthrough Pro is preferred due to the chances of availability and introduction of navigation related functions adequate for the demonstration purposes of this thesis.

The following tables display the behaviour of the menus and menu items of Virtus Wlakthrough Pro with respect to the specified selections (in the space of user+need) converted into terms of IRT.

For instance, for the sub-menu item camera lens under the 'navigation' menu, when I=0 the lens will be determined as a default value x. This menu item will not be made available to the user since the interactivity level is set to 0. When I=1, the lens will still be set to x with the possibility of user's intervention this time. When I=2, the lens slot will appear empty for user's input.

The menu 'lighting editor' is off whenever I=0 and R=0 because the former constraint defines a state of inexperience to deal with the editor on the user's side and the latter indicates no need for the use of the 'lighting editor' since no rendering is required. The menu items under the editor are assigned with default values when I=0 and R=2 since this state defines an illiteracy on the user's side to manipulate the editor, yet in need of a rendered version of the design. Thus, the CSM aids the users with operating the 'lighting editor' on their behalves.

Nevertheless, the implementation of the CSM on the menus of the Virtus Walkthrough package remains as an initial stage in the customization process. The illustrated states of the menu items can be interpreted to be randomly assigned and then strictly bounded at this initial stage, but it should be noted that the customization process continues with constant evaluation and further adjustments by the user. In other words, the process involves user satisfaction as a criteria and evolves with the user. For this purpose, the users are given a small screen displaying the transformation matrix visually, rather than mathematically, on which they may further customize either the PAE or the IRT or the parametric coordinate values. Furthermore, this screen allows the user to see the effects of the changes of these values on the menu items in real time.

The menus in Virtus Walkthrough consist of 'file', 'edit', 'view', 'surface', 'walk', 'light', 'design' and 'window' operations. Some of the most important and most used menus among these are represented also with an icon on the left hand side of the screen on a 'tools pad'. To illustrate the effect of the CSM on the running of the walkthrough program the menus which are crucial to the design process are chosen. The general menus inherent in most of the commercial programs such as 'file' (with menu items like <u>open</u>, <u>close</u>, <u>save</u>, <u>print</u>, etc.), 'edit' (with menu items like <u>undo</u>, <u>cut</u>, <u>copy</u>, <u>clear</u>, etc.) and 'surface operations' (with menu items like <u>bring to front</u>, <u>send to back</u>, <u>new layer</u>, <u>zoom</u>, etc.) are not taken into consideration. Therefore, the menus 'view', 'walk', 'light', 'design' and 'windows' remain to be analyzed.

However, since from within the 'windows' menu it is possible to work with the 'view', 'light' and 'walk' editors, it is sufficient to illustrate the states of the 'windows' menu to see how the 'view', 'light' and 'walk' editors shall behave (Table 4.1).

In the following tables, ON/OFF indicate either the menu item is being used or not. For the default values assigned to the items by Virtus Walkthrough Pro, the behaviour is indicated by (d); for the assignments of CSM which are context-specific, the behaviour is indicated by (cs). Further explanations about the functions of the Virtus Walkthrough Pro menus and menu items used in the tables can be found in Appendix B.

design phase	0 initial	0 initial	0 initial	2 late	1 medium	2 late	2 late
audience	1 critics	0 designer	2 customer	2 jury	1 critics	1 critics	2 jury
experience level	0 novice	0 novice	0 novice	0 novice	1 medium	2 expert	2 expert
WINDOWS							
TOOLS WINDOW (1)	(d) ON	(d) ON	(d) ON	(cs) OFF	(d) ON	(d) ON	(cs) OFF
DEPTH WINDOW	(cs) ON	(cs) ON	(d) OFF	(d) OFF	(cs) ON	(d) OFF	(d) OFF
COORDINATES WINDOW	(d) OFF	(cs) ON	(d) OFF	(d) OFF	(d) OFF	(d) OFF	(d) OFF
TEXTURES WINDOW (2)	(d) OFF	(d) OFF	(d) OFF	(d) OFF	(cs) ON	(cs) ON	(d) OFF
WALK VIEW	(d) OFF**	(d) OFF	(d) OFF**	(cs) ON	User Input	User Input	User Input
DESIGN VIEW	(d) ON**	(d) ON	(d) ON**	(cs) OFF	User Input	User Input	User Input
(1) see Table 4.2.							
(2) see Table 4.3.							
** user intervention enabled							

Table 4.1. The behaviour of the 'Windows' menu with respect to various I, R, T sets

The menu items <u>walk view</u> and <u>design view</u> are mostly left to the user to indicate. However, even at the initial stages of design and low level of experience (such as (010) or (020)) the user is given the chance of intervening with the default view due to the properties of the audience.

The 'tools' window within the 'windows' menu allows 'design tools' and the 'lighting editor' to function. The following table displays the behaviour of the sub items and editors of the 'tools window' (Table 4.2).

design phase	0 initial	0 initial	0 initial	2 late	1 medium	2 late	2 late
audience	1 critics	0 designer	2 customer	2 jury	1 critics	1 critics	2 jury
experience level	0 novice	0 novice	0 novice	0 novice	1 medium	2 expert	2 expert
TOOLS PAD					**	**	**
DESIGN VIEW TOOLS	(d) ON	(d) ON	(d) ON	(cs) OFF	(d) ON	(d) ON	(cs) OFF
DRAWING TOOLS	(d) ON	(d) ON	(d) ON	(cs) OFF	(d) ON	(cs) OFF	(cs) OFF
LIGHTING EDITOR TOOL	(cs) OFF	(cs) OFF	(cs) ON *	(cs) ON *	(cs) ON	(d) ON	(cs) OFF
APPEARANCE MODIFIERS	(cs) OFF	(cs) OFF	(cs) ON *	(cs) ON *	(cs) ON	(d) ON	(cs) OFF
COLOR BAR	(cs) ON*	(cs) OFF	(cs) ON *	(cs) ON *	(cs) ON	(d) ON	(cs) OFF
* menu unabled to user							
** user intervention enabled							

Table 4.2. The behaviour of the 'Tools Pad' with respect to various I, R, T sets

For the states of (111), (212), (222) even though default values are suggested, the user has the chance of altering the values. This is due to the fact that the user has some experience with the software indicated by the choice of (1) and (2) as the experience level. Whereas, where experience level is set to (0), the user does not have to worry about choosing the appropriate settings.

Within the 'windows' menu, the 'texture' window deals with texture mapping and material selection in the following manner when used with CSM (Table 4.3).

design phase	0 initial	0 initial	0 initial	2 late	1 medium	2 late	2 late
audience	1 critics	0 designer	2 customer	2 jury	1 critics	1 critics	2 jury
experience level	0 novice	0 novice	0 novice	0 novice	1 medium	2 expert	2 expert
TEXTURE options				*	**		*
Name	U	U	U	default	default	User Input	User Input
Source	N	N	N	default	default	User Input	User Input
Format	A	A	A	default	default	User Input	User Input
Dimensions	B	B	B	default	default	User Input	User Input
Edit	L	L	L				
First Tile	E	E	E	default	default	User Input	User Input
Tile Pattern	D	D	D	default	default	User Input	User Input
Appearence				default	default	User Input	User Input
* menu unabled to the user							
** user intervention enabled							

Table 4.3. The behaviour of the 'Texture' menu with respect to various I, R, T sets

As observed at states (000), (010) and (020) CSM does not display the menu to the user. Thus, the user at the initial stages of design (as indicated by the selection of (0) as the design phase) does not have to bother with settings of a menu like 'texture' which is irrelevant for her purposes at that phase. At the states (220) and (111), the required values within the menu are set by CSM, still allowing the user free of dealing with finding the most appropriate settings.

The design process in the virtual environment can briefly be summarized to consist of the following acts:

1. modeling

2. rendering

3. texture mapping, selection and application of materials

4. lighting, adding details (Richens 305-25; Irikiin)

5. navigation.

So far, the tables illustrated the behaviour of the Virtus Walkthrough menus for modeling, texture and lighting purposes. The behaviour of the menus for the remaining purposes are as follows (Tables 4.4. and 4.5) when operated with CSM.

design phase	0 initial	0 initial	0 initial	2 late	1 medium	2 late	2 late
audience	1 critics	0 designer	2 customer	2 jury	1 critics	1 critics	2 jury
experience level	0 novice	0 novice	0 novice	0 novice	1 medium	2 expert	2 expert
RENDERING					**		
SHADING		U					
Shaded	(d) OFF		(d) OFF	(d) ON	(d) ON	(d) ON	(d) ON
Unshaded	(d) OFF	N	(cs) ON	(d) OFF	(d) OFF	(d) OFF	(d) OFF
White	(cs) ON		(d) OFF	(d) OFF	(d) OFF	(d) OFF	(d) OFF
DRAWING		A					
Fill and Frame	(d) OFF		(cs) ON	(cs) OFF	(d) ON	(d) ON	(cs) OFF
Fill	(d) OFF	B	(d) OFF	(cs) ON	(d) OFF	(d) OFF	(cs) ON
Frame	(cs) ON		(d) OFF	(d) OFF	(d) OFF	(d) OFF	(d) OFF
BLACK FRAME	(d) OFF	L	(cs) ON	(d) OFF	(d) OFF	(d) OFF	(d) OFF
PRINT FILL	(cs) ON		(d) OFF	(d) OFF	(d) OFF	(d) OFF	(d) OFF
OPENINGS		E					
See In	(cs) ON		(cs) ON	(cs) ON	(cs) ON	(d) OFF	(d) OFF
See Out	(cs) ON	D	(cs) ON	(cs) ON	(cs) ON	(d) OFF	(d) OFF
See Through	(d) OFF		(d) OFF	(d) OFF	(cs) ON	(cs) ON	(d) OFF
DITHERING	(cs) OFF		(d) ON	(d) ON	(d) ON	(d) ON	(d) ON
BLENDED TRANSLUCENCY	(d) OFF		(d) OFF	(cs) ON	(d) OFF	(d) OFF	(cs) ON
** user intervention enabled							

Table 4.4. The behaviour of the 'Rendering' menu with respect to various I, R, T sets

design phase	0 initial	0 initial	0 initial	2 late	1 medium	2 late	2 late
audience	1 critics	0 designer	2 customer	2 jury	1 critics	1 critics	2 jury
experience level	0 novice	0 novice	0 novice	0 novice	1 medium	2 expert	2 expert
NAVIGATION					**		
AIDS		U					
Button Down	(d) OFF		(cs) ON	(cs) ON	(cs) ON	(d) OFF	(d) OFF
Cross Hair	(d) OFF	N	(cs) ON	(cs) ON	(cs) ON	(d) OFF	(d) OFF
Velocity Detector	(d) OFF		(d) OFF	(cs) ON	(d) OFF	(d) OFF	(d) OFF
Collision Detector	(d) OFF	A	(d) ON	(cs) ON	(cs) ON	(d) OFF	(d) OFF
CAMERA							
Film	(d) 35	B	(d) 35	(d) 35	(d) 35	User Input	User Input
Lens	(d) 26		(d) 26	(d) 26	(d) 26	User Input	User Input
ASPECT RATIO		L					
35mm. Horizontal	(d) OFF		(d) OFF	(d) OFF	(d) OFF	(d) OFF	User Input
35mm. Vertical	(d) OFF	E	(d) OFF	(d) OFF	(d) OFF	(d) OFF	User Input
2.25 Square	(d) OFF		(d) OFF	(d) OFF	(d) OFF	(d) OFF	User Input
4x5 Horizontal	(d) OFF	D	(d) OFF	(d) OFF	(d) OFF	(d) OFF	User Input
4x5 Vertical	(d) OFF		(d) OFF	(d) OFF	(d) OFF	(d) OFF	User Input
Television	(d) ON		(d) ON	(d) ON	(d) ON	(d) ON	User Input
Euro Wide	(d) OFF		(d) OFF	(d) OFF	(d) OFF	(d) OFF	User Input
US Wide	(d) OFF		(d) OFF	(d) OFF	(d) OFF	(d) OFF	User Input
Anamorphic	(d) OFF		(d) OFF	(d) OFF	(d) OFF	(d) OFF	User Input
** User intervention enabled							

Table 4.5. The behaviour of the 'Navigation' menu with respect to various I, R, T sets

5. CONCLUSION

The properties of an ideal interface for the architectural space simulation software were stated in Section 3.2. The introduced interface model can be discussed within the framework of these properties to underline the original contributions of the introduced model.

Context-Specification

" The more relevant the context, higher the value and longer the life-span of the information" (Sanders 7).

This thesis introduces an application-independent tool to customize the architectural space simulation software packages according to the user's profile, goals and intentions. CSM is a menu that indicates the level for the architectural space simulation software to run at, in other words, the context for the interface to operate at.

For instance, students at the early stages of design may wish to operate at a very low interactive level. At the early stages of design, they may not be capable of determining most of the values needed to form the simulation, since many design decisions are not set yet. At the low level of interaction, the required values for some of the menu options (e.g. illumination level, camera angles, camera moves, etc.) will be automatically assigned based on the standard values. The users may be satisfied to study the space in the form of a wire-frame model only, and finally, they may want to have a non-real time simulation to be able to spend as much time as possible to comprehend the frames.

Within the framework of these needs the users are expected to select the early design stage as the purpose, themselves as the audience and beginner as the experience level on the CSM. Based on these selections CSM assigns standard values for some menu options. These standard values are already been set and used in the commercially available simulation software as default values. CSM also highlights relevant menus for the user to assign values to and hides some of the menu options which are irrelevant vis-à-vis the purpose of simulation. Thus, the user with no prior experience of the application can make use of the default schemes and standard values of sub-menu items like camera position, lighting level, etc.

An architect, at the initial phases of a design problem, may only give the global dimensions of the space and experience the design at different levels of lighting or color schemes. A design firm can display a highly rendered simulation to the client, while they show the same data in a longer and non-rendered version to the construction engineers. Moreover, the design firm may produce a series of similar simulations for different designs, allowing comparisons to be made.

Standardization

The users can benefit from the standardization with CSM in two major ways. The first one is the production of simulations in similar format. Production of such simulations is crucial for architectural competitions. CSM may be used to define the main circulation paths, cardinal views and important moves required by the competition jury (Mark 408) and the resulting simulations in the similar format may be used to evaluate different designs. Mark describes the expected results as: "Motion picture renderings of architecture could take the form of standard sequences. This would be comparable with the section, plan and elevation drawings made on paper-based media" (Mark 14-5).

Secondly, CSM helps the architects by giving a certain address to every context specified by the user. The unique address of each context is formed in terms of PAE and IRT. These addresses can be saved for further use. Then, the users can either use the context specified for them every time they access the software or may use the context created for a special task (such as drawing perspectives) every time they perform that particular task.

Compatibility

Not only these addresses but the operations performed using the software can also be saved by CSM. These history-based operations are saved in sequential files. The files do not only enable the use of the same format repetitively, but also allow the transfer of context within the same package or different packages.

Transfer of context within the same package may aid the users in dealing with different detail levels of a project. The users may easily use the same customized context for working on different parts of a project like the space, the furniture, facade, etc. For instance, they may work on a whole floor plan, then switch to the bathroom, and then to a chair, all executed in the same customized context. Still, if change of context is required while switching between levels of detail, CSM allows this by on-screen adjustments.

By being able to customize the software according to the purpose, CSM may enable software of various capabilities (2D drafting, 3D modeling, animation, rendering, etc.) to be built which will then be used partially at each phase of the design. This actually makes the use of various different packages towards one presentation unnecessary.

Yet, if the users, for some reason, have to employ more than one package to execute a task, given that both packages work with CSM, they may easily transfer the context specified in one package to another. Still, further changes may be done through the on-screen CSM logo.

Efficiency

Design changes and improves at each successive phase. The software that the architects use has to answer the different requirements of each phase and be flexible enough to accommodate the changing data with each phase. Being able to customize the software for each successive phase will therefore let the architects work with an interface which 'recognizes' their intentions and aids them accordingly.

Along with this aid, the introduced interface model does not require any programming skills on the side of the user. It offers a to-the-point and common interface environment which does not require much effort of adaptation from the user. Thus, the required time to learn, operate and get used to the software reduces. Also, architects are given the chance to direct the architectural space simulation software to offer services which are adequate to their purposes. Then, the architects can be expected to appreciate and use the medium more efficiently, and better design evaluations and sales can be obtained.

In architectural education, schools do not need to purchase different software packages for different levels. Any architectural simulation software can be used by students of different levels since it may be customized according to the requirements of different levels, projects and prior knowledge in using the software.

Mastery of Functions

The longing for the artistic expression of the paper-based architectural drawings over the digital ones has been stressed several times before. Irikiin asserts that computer graphics is not devoid of giving the personal style provided that the designer knows the necessary techniques available (25). Menu functions of the architectural space simulation software packages offer a large number of possibilities, more than one can achieve by hand indeed. The possible color combinations (with different hues, saturation, etc.) and textures, for instance, are of a level far too superior to the hand-made ones both in terms of quantity and precision. It is then quite easy to achieve personal styles within this set of numerous combinations, given that the user can manipulate the functions in a masterly fashion.

With CSM, at the states where I >=1, the users are suggested with the default values suitable for the context. They may alter them if necessary. In order to truly master one of the functions such as texture mapping, they may hold the other variables constant and observe the impacts of the alterations made through the texture function only.

The benefit here, over the currently used architectural space simulation software packages' interfaces, is that the users are supplied with the constraints/constants of the specific context they are working in. Thus, the tests with the selected function are displayed on a model relevant to the properties of that particular context.

To illustrate this, imagine that the user is trying to master the 'texture' function. Without CSM, users will input the dimensions of the space and then either determine new values or accept the default values for 'lighting' (intensity, angle, number of fixtures, etc.), 'color' (hue, saturation, brightness) and 'material'. If they determine the values themselves, there is always the risk that these values may not reflect the impacts of texture at best, i.e. the lighting angle may be in such a manner that it does not cast proper shadows on the texture. On the other hand, if they go along with the default values, these optimized values may give a wrong impression about the texture which would actually be a good choice for another context of design. However, with CSM, they may specify their context, so that the simulation program assigns default values relevant to that context. Then, while working with different choices in the 'texture' menu, other functions like 'lighting', 'color', 'material', etc., are supplied with appropriate values for that context, meaning that the results of the changes made in the 'texture' menu are displayed with proper shadows, proper lighting, color and material properties. The user then, does not only grasp the properties of the 'texture' menu, but is also given an insight about the appropriate values for other menus at that context.

True Modeling

The virtual environment is a model that is formed by the combination of different models with different scaling factors. The scale of rendering of the design model differs from that of the actual building. When the scale of time is superimposed the problem gets further complicated. CSM is expected to be an intuitive solution in mixing the different scales in one virtual modeling environment.

However detailed a model or well-made a simulation is, it can never replicate, duplicate, or simulate the real visual experience taken from the actual building. There are technical restrictions which prevent any simulation from giving a perspective as real as the direct visual experience. Human eye has a wide perspective which cannot be obtained in a computer screen where the virtual environment is displayed. In order to compensate for this, wide-angle lenses are employed which in return distort the view and give untrue information about the space. In addition, all walkthrough, flythrough, and virtual reality simulations lack the capacity of the human eye which is to see edges of the frame where the eye is directed to (Mahoney 26).

On the other hand, whatever 3D effect is observed, it is observed from a medium which is 2D by definition (Mahoney 28). Therefore, in order to give a real-looking view, architectural space simulations simulate not what is real, but what is unreal. (In other words, they tell lies in order to bring out the truth.)

At this point, it is relevant to ask to what extent it is important that the space simulation simulates the reality. The aim of the simulation should be to show frames similar to the space provided if the architectural design was to be constructed, rather than trying to simulate the real visual experience expected to be observed within that space. An approach as in the latter case is to result in a more unreal looking simulation due to the technical restrictions mentioned above. To put it more clearly, it is unlikely to produce a true simulation of a 3D space in 2D media. Moreover, the visual experience changes depending upon each person within the space and is subjective.

The optimization of CSM for the values between certain intervals is not an underestimation of the choices in the indicated intervals, rather it is an aid in using the virtual environment for creating successful displays. The provision of visual abstractions is more suitable to the designers' cognition, rather than the provision of almost the same perceptual experience of being inside the architectural space (Mark 394), which is not yet technically achieved. In this case, CSM can determine a level of optimized menu values, which is actually a very difficult task if it is left to the user.

Sanders sees the best approach to the problem as "... not to mimic the style of traditional renderings or emulate the realism of photographs, but to communicate design concepts in ways that physical models or renderings cannot." This approach naturally does not depend on realism but on "... abstraction, disassembly, and motion" (37).

It is a difficult task for both the user and the interface to measure the degree of realism or trueness of the virtual design model. Yet, CSM aids the user by taking in the measurable data (PAE: purpose, audience, experience) from the user and converting them into an optimized, an abstracted form. This form can then be measured in terms of IRT (interactivity, rendering, time). As such, CSM constitutes means to measure the trueness of the virtual model in an abstracted space, which would otherwise be impossible to be measured.

5.1. Further Studies

CSM does not only supply advantages to the architects, but can create research and implementation areas for software developers, where they are expected to collaborate with the architects. Consequently, an opportunity for architects to become involved in the development of architectural software tools will be created.

The introduced interface model in this thesis reflects the ideas of an architect towards improving the use of architectural space simulation software in architectural design. Such an improvement is found necessary upon the observation of various problems that the architects are faced with while using the software. Most of the currently available software are marketed claiming that they are capable of executing many tasks. However promising this claim may be, it does not lead to efficient use of the software. Many users, who would employ the software for initial volumetric studies, simple drawings or particular design tasks (like detailed study of a space, measurements of a space, performance analysis, etc.) are bewildered by the amount of awaiting decisions, value settings to be made in order to execute the basic task. Also, the abundance of menus and menu items requires constant reference to the manuals, help menus or experienced users for the user who wants to use the program efficiently and to its total capacity.

The thesis initially displays a theoretical framework of the two spaces on the two sides of the software interface, and then introduces the interface model based on the properties of these two spaces. Studies may continue by integrating the CSM model to one of the commercially available software and observing the results of use. That would be a joint area of study for the computer engineers and architects.

Architecture in the virtual environment promises a powerful future. It may not take a long time before we are confronted with architects, designing buildings, interiors, and settings for the virtual environments only. There awaits a whole new market of estate, environmental, interior, furniture design for virtual environments - although they may acquire new names- as new potential fields of design. With its new definitions and concepts, virtual architecture may become the antithesis of physical architecture.

Before virtual architecture attains such importance, it is initially required that the virtual design environment to gain wide acceptance of use. Buday points out the fact that: "... computer graphics in general will bring about well-designed buildings, but only for the architects willing to change their work styles" (quoted in Mahoney 27). In order to benefit from the potential of digital aid and new technologies it is required to redefine architecture, architectural design process and architectural terms with respect to the emerging digital media. Possibilities offered by the digital media do not fit into the paper-based way of architectural thinking. The new way of thinking offered by the digital media is a visual one and unless the new generation of architects are prepared to think visually, the promised potential of digital media may never be fulfilled.

This new way of thinking can only be developed on a platform that enables easy, customizable and intelligent use of the architectural software. As such, architects will not have to master the software itself, but will have the opportunity to master the ideas that can be produced by the use of the software.

LIST OF REFERENCES

Abbo, I.A. "Effectiveness of Models." 6th Conference of the European Full-Scale Modeling Association (EFA). Ed. Bob Martens. Vienna: Vienna University of Technology Iris-Isis-Publications OKK Editions, 1996. 69-78.

Amor, A. "The Real Value of 3D Rendering to Architects." Computer Graphics World June 1996: 19-20.

Beheshti, M. and Monroy, M. "Requirements for Developing an Information System for Architecture." CAAD'87 Features. Eds. T. Maver and H. Wagter. New York: Elsevier Science Publishing Company Inc., 1987. 149-69.

Binkley, T. "Digital Dilemmas." Leonardo (Digital Image-Digital Cinema Supplemental Issue 1990: 13-19.

Brill, L.M. "Facing Interface Issues." Computer Graphics World April 1992: 48-58.

Burden, E. Design Simulation. London: The Mitchell Publishing Company Limited, 1988.

Buday, R. "Architects Need to Think Like Filmmakers." Computer Graphics World April 1997: 19-20.

Charles, C. B. and Brown, K.M. "Visualization Frontiers." Architecture December 1995: 121.

Cote-Munoz, J.A. "AIDA - An Adaptive System for Interactive Drafting and CAD Applications." Adaptive User Interfaces: Principles and Practice. Amsterdam: Elsevier Science Publishers, 1993. 225-240.

Dietrich, H., et al. "State of the Art in Adaptive User Interfaces." Adaptive User Interfaces: Principles and Practice. Amsterdam: Elsevier Science Publishers, 1993. 13-48.

Emmett, A. "A New Dynamism." Computer Graphics World June 1992: 30-33.

Gross, M.D. and Do, E.Y.L. "Ambigious Intentions: a Paper-Like Interface for Creative Design." Proceedings of the ACM Symposium on User Interface Software and Technology. NewYork: The Association for Computing Machinery, 1996. 183-192.

Greenberg, D.P. "Computer Graphics in Architecture." Tutorial: Computer Graphics. Eds. J.C. Beatty and K.S. Booth. New York: Institute of Electrical and Electronic Engineers, Inc., 1982. 533-540.

Gromala, D.J. "Multi-Media in Graphic Design." Academic Computing in Macintosh Environment III. Eskişehir: Anadolu University Press, 1992. 1-7.

Hoffman, M.C. "Cyberpunk, Generation X and the Ataxia of Space." ACSA/EAAE Schools of Architecture Conference. Prag: Czech Technical University, 1993. 1-9.

Howes, A. "An Introduction to Cognitive Modelling in Human-Computer Interaction." Perspectives on HCI. Eds. A. Monk and N. Gilbert. London:

Academic Press Limited, 1995. 97-119.

Illich, I. Tools for Conviviality. NewYork: Harper and Row, 1973.

Irikiin, T. et al. Computer Graphics Architectural Rendering, Computer Graphics Creative Laboratory. Tokyo: Graphic-sha Publishing Co., Ltd., 1994.

Kalay, Y. Modeling Objects and Environments. New York: John Wiley & Sons, 1989.

Kempfer, L. "Maximizing Visualization." Computer-aided Engineering 15 (1996): 48-54.

MacLeod, D. "Computers: Virtual Reality." Progressive Architecture April 1992, 55-56.

Maes, P. "Intelligent Software." Intelligent User Interfaces IUI 97. Eds. J. Moore et.al. New York: ACM, 1997. 41-43.

Mahdavi, A. "Computational Decision Support and the Building Delivery Process: A Necessary Dialogue." Descriptive Models of Design. Eds. Ö. Akın and G. Sağlamer. İstanbul: İstanbul Technical University, 1996.

Mahoney, D.P. "Walking Through Architectural Designs." Computer Graphics World June 1994, 22-30.

Mahoney, D.P. "Virtual Reality's New Home." Computer Graphics World April 1996, 53-54.

Mark, E. "Case Studies in Moviemaking and Computer-Aided Design." The Electronic Design Studio. Eds. M. McCullough, W.J. Mitchell, and P. Purcell. Massachusetts: The MIT Press, 1990. 393-412.

Mark, E. "Architecture in Motion." Computers in Architecture: Tools for Design. Ed. F. Penz. London: Longman Group UK Ltd., 1992. 9-16.

Marti, P. and Normand, V. "Bridging Software Design and Usability Analysis through Task Modeling." Human Comfort and Security of Information Systems. Berlin: Springer-Verlag, 1997. 39-50.

Maulsby, D. "Inductive Task Modeling for User Interface Customization." Intelligent User Interfaces IUI 97. Eds. J. Moore et.al. New York: ACM, 1997. 233-237.

McLauglin, M. "There's No Such Thing as a CAD Operator." Progressive Architecture September 1995, 86-87.

Mitchell, W.J. and McCullough, M. Digital Design Media. New York: Van Nostrand Reinhold, 1991.

Mitchell, W.J. City of Bits. Massachusetts: MIT Press, 1995.

Negroponte, N. Being Digital. New York: Alfred A. Knopf, Inc., 1995.

Novak, M. "Liquid Architectures in Cyberspace." Cyberspace: First Steps. Ed. M. Benedikt. Massachusetts: MIT Press, 1991. 225-255.

O'Connell, K.R. "Computer & Architecture." School Arts 94 (1995): 16.

Ormerod, M. and Aouad, G. "The Need for Matching Visualisation Techniques to Client Understanding in the UK Construction Industry." Information Visualization 97. California: IEEE, 1997. 322-328.

Pimentel, K. and Teixeria, K. Virtual Reality. New York: Mc-Graw Hill, Inc., 1993.

Porter, S. "Interview: Jaron Lanier." Computer Graphics World April 1992, 61-70.

Porter, S. "Architects Can't Afford to Ignore 3D Any Longer." Computer Graphics World June 1997,.4.

Potter, C.D. "Architectural CAD on the Desktop." Computer Graphics World June 1995, 25-35.

Quemeneur, J.Y. and Brossier, F. "OPX2 Intelligent Help: an Example of Intelligent Interfaces." Human Comfort and Security of Information Systems. Berlin: Springer-Verlag, 1997. 256-267.

Rich, C. and Sidner, C.L. "Adding a Collaborative Agent to Graphical User Interfaces." Proceedings of the ACM Symposium on User Interface Software and Technology. NewYork: The Association for Computing Machinery, 1996. 21-30.

Rich, C. and Sidner, C.L. "Segmented Interaction History in a Collaborative Interface Agent." Intelligent User Interfaces IUI 97. Eds. J. Moore et.al. New York: ACM, 1997. 23-30.

Richens, P. "Does Knowledge Really Help? CAD Research at the Martin Centre." Knowledge-Based Computer-Aided Architectural Design. Eds.G. Carrara and Y.E. Kalay. London: Elsevier Science B.V., 1994. 305-325.

Sanders, K. The Digital Architect. New York: J. Wiley & Sons, 1996.

Scott, A. Virtus Walkthrough Pro User's Guide v.2.0. Cary: Virtus Corporation, 1994.

Shepherd, A. "Task Analysis as a Framework for Examining HCI Tasks." Perspectives on HCI. Eds.A. Monk and N. Gilbert. London: Academic Press Limited, 1995. 145-174.

Sherman, E.H. and Shortlife, E.H. "A User-Adaptible Interface to Predict User's Needs." Adaptive User Interfaces. Eds.Schneider-Hufschmidt et al. Amsterdam: Elsevier Science Publishers, 1993. 285-315.

Singh, G., et al. "Virtual Reality: Software and Technology." Communications of the ACM 39 (1996): 35.

Sumner, T. et al. "The Cognitive Ergonomics of Knowledge-Based Design Support Systems." Human Factors in Computing Systems. Ed. S. Pemberton. New York: ACM, 1997. 83-91.

Sutherland, I. "The Ultimate Display." Proceedings of the IFIP Congress. New

York: IFIP, 1965. 506-508.

Şenyapılı, Burcu. "Proposal to Improve the Use of Dynamic Simulations in Architectural Design." 4D Dynamics. Ed. A. Robertson. Leicester, DeMontfort University, 1995. 151-155.

Şenyapılı, Burcu. "The True Model Concept in Computer Simulations Used in Architectural Design." 6th Conference of the European Full-Scale Modeling Association (EFA). Ed. B. Martens. Vienna: Vienna University of Technology Iris-Isis-Publications OKK Editions, 1996. 133-140.

Şenyapılı, Burcu. "Visualization of Virtual Architecture." Information Visualization 97. California: IEEE, 1997. 260-266.

Şenyapılı, B. and Özgüç, B. "Proposal for Improving Computer Aid in Interior Architectural Design." Architectural Science Review 37 (1994): 171-179.

Şenyapılı, B. and Özgüç, B. "A Context-Specific Interface Model for Architectural Design in the Virtual Environment." Architectural Science Review (forthcoming) (1998).

Thomas, C.G. and Fischer, G. "Using Agents to Personalize the Web." Intelligent User Interfaces IUI 97. Eds. J. Moore et.al. New York: ACM, 1997. 53-60.

Tatham, E.W. "Depth Cueing for Augmented Reality." Information Visualization 97. California: IEEE, 1997. 348-349.

Thompson, J. Virtual Reality: An International Directory of Research Projects. London: Meckler, 1993.

Thomsen, C.W. Visionary Architecture. New York: Prestel-Verlag, 1994.

Von Schweber, E. and Von Schweber, L. “World Builders.” PC Magazine May 1995, 170-176.

Witte, O.R. “How The Schools are Teaching the Uses of Computers.” Architecture Aug. 1989, 91-95.

LIST OF SOFTWARE REFERENCES

(S1) AutoVision, AutoDesk. San Rafael, CA.

(S2) 3D Studio, Release 4.0, AutoDesk. San Rafael, CA.

(S3) 3D Studio Max, Release 2.0, AutoDesk. San Rafael, CA.

(S4) ModelView, Intergraph. Huntsville, AL.

(S5) ArcVision, Alias/Wavefront, Silicon Graphics Inc. Mountain View, CA.

(S6) ALLPLAN, Nemetschek. Munich, Germany.

(S7) AutoPro, Eagle Point. Palo Alto, CA.

(S8) Visipix, MicroStation, Bentley Systems Inc.

(S9) Virtus Walkthrough Pro, Release 2.0, Virtus Corporation. Cary, NC.

(S10) Quick Time VR, Release 3.0, Apple Computer Inc. Cupertino, CA.

(S11) 3DIX, Release 2.0, IBM. Raleigh, NC.

(S12) World Tool Kit, Sense8. Mill Valley, CA.

(S13) Windows 95 and 98, Microsoft Corporation. Palo Alto, CA.

(S14) System 7.0, Apple Computer Inc. Cupertino, CA.

(S15) Mac OS, Apple Computer Inc. Cupertino, CA.

APPENDIX A: DISPLAYED SCREENS DURING THE

CUSTOMIZATION PROCESS

Figure A.1. Initial screen of the customization process

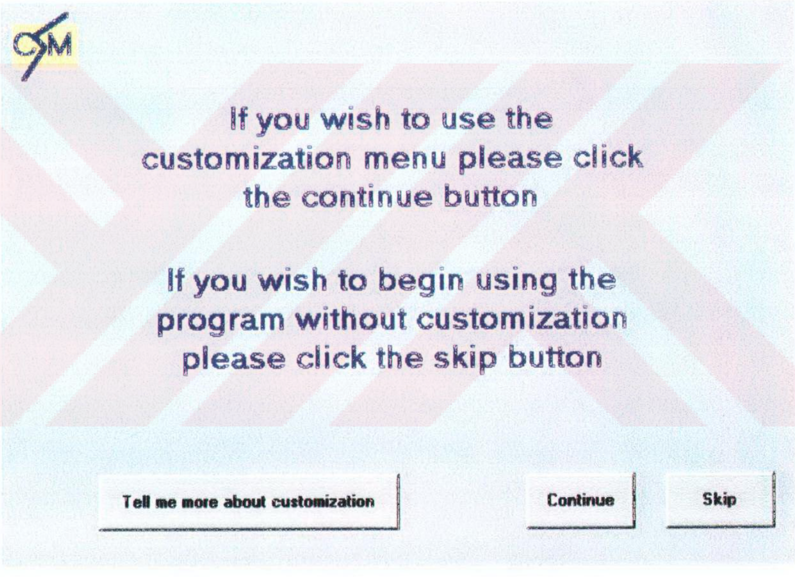

Figure A.2. Second screen of the customization process

Figure A.3. Third screen of the customization process

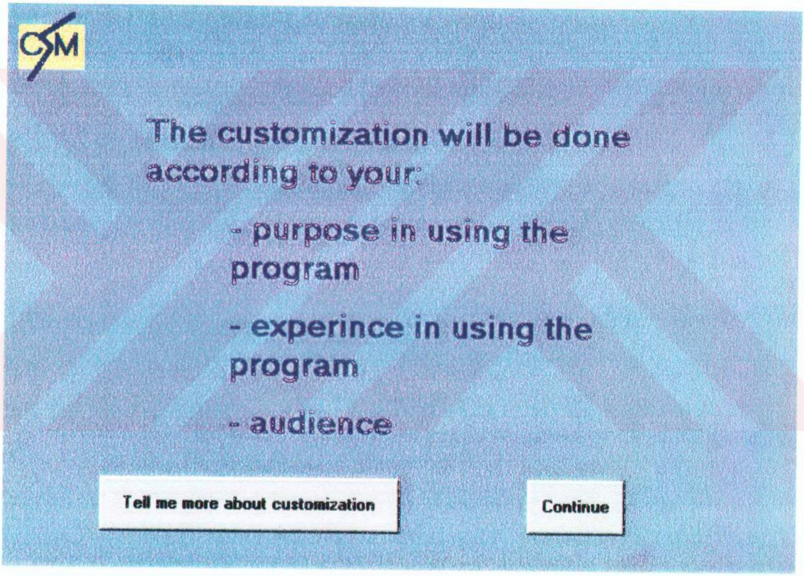

Figure A.4. Fourth screen of the customization process

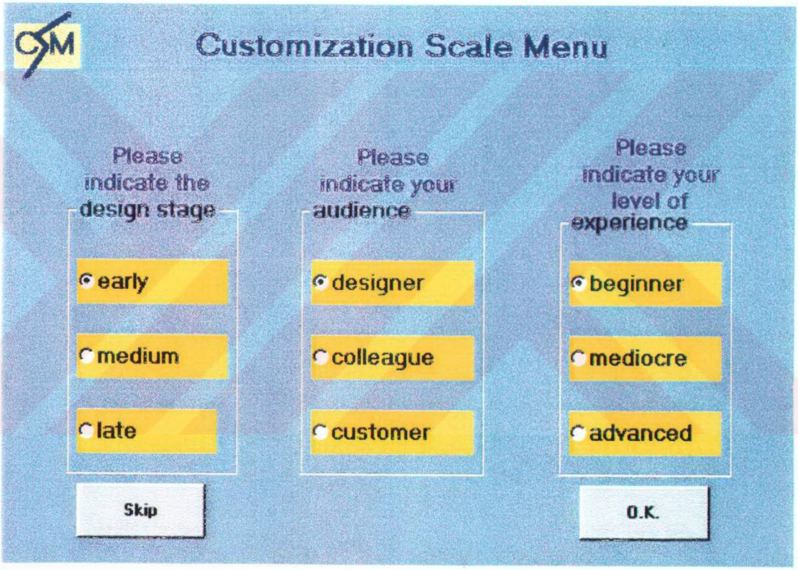

Figure A.5. Fifth screen of the customization process

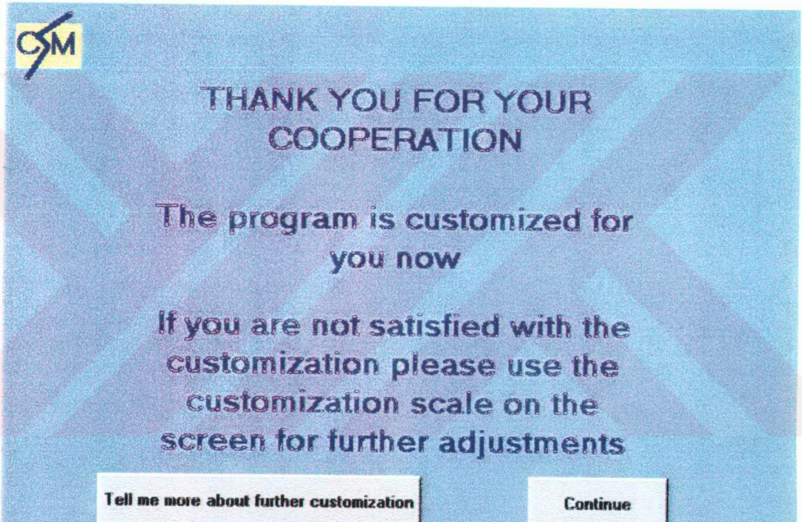

Figure A.6. Final screen of the customization process

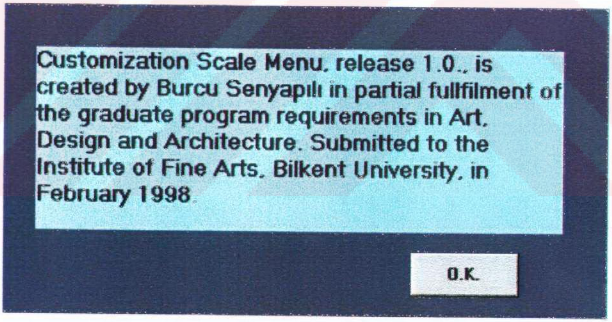

Figure A.7. Brief explanation about the production of CSM

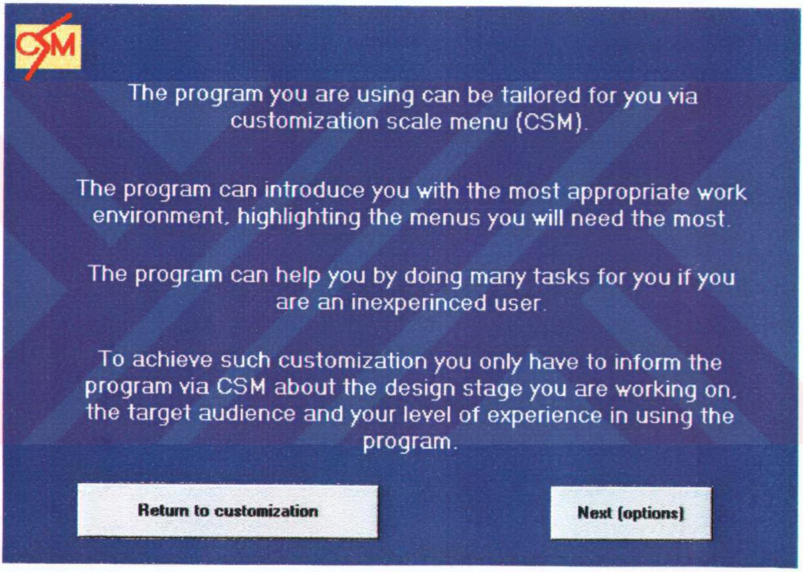

Figure A.8. Information about customization

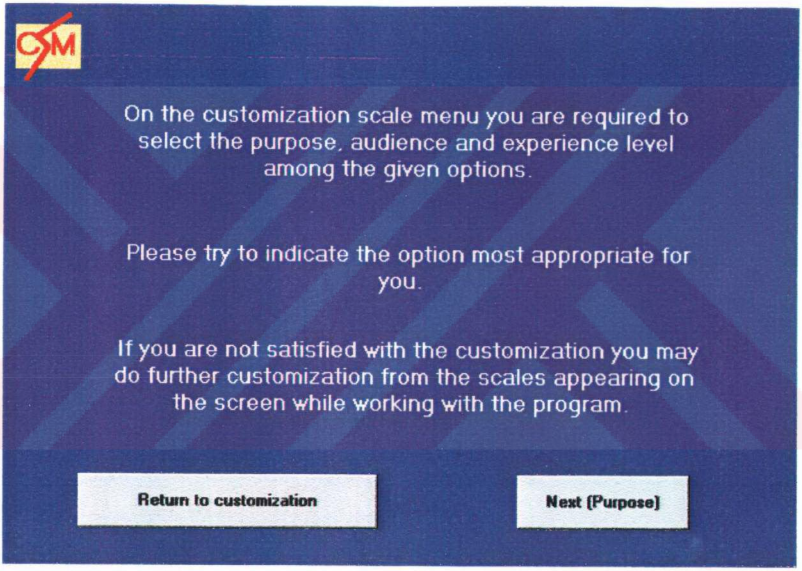

Figure A.9. Information about customization options

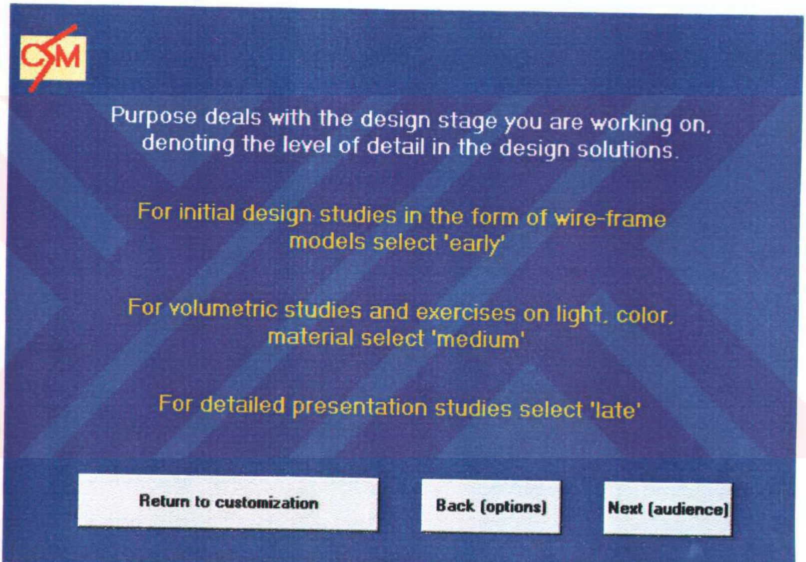

Figure A.10. Information about purpose

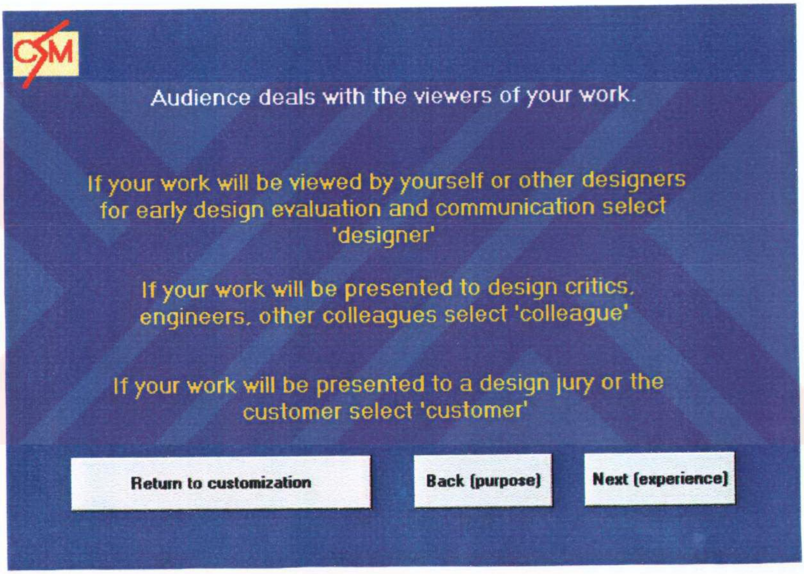

Figure A.11. Information about audience

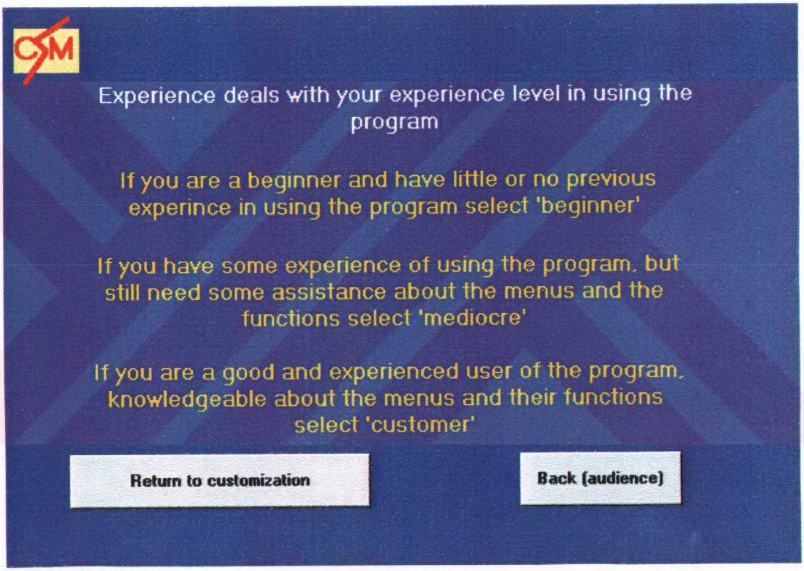

Figure A.12. Information about experience

Figure A.13. Initiation of the run time of the program

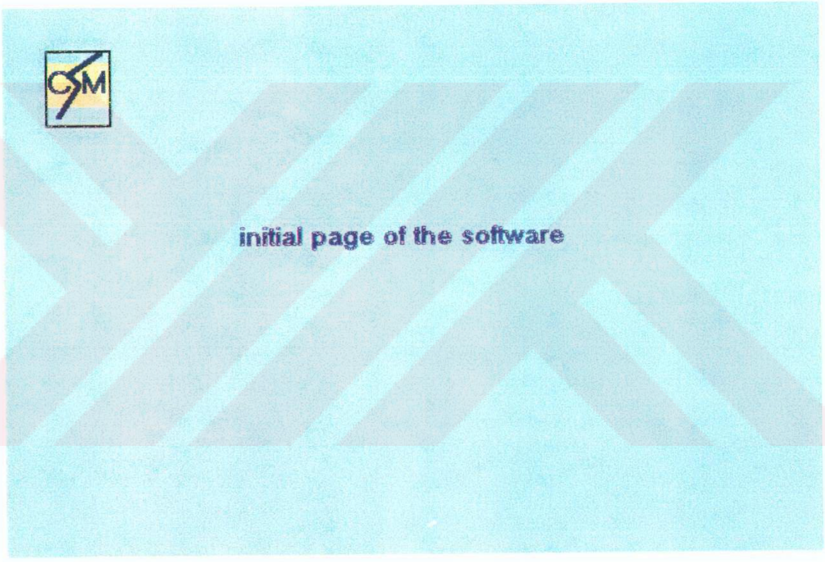

Figure A.14. Initial screen of the program

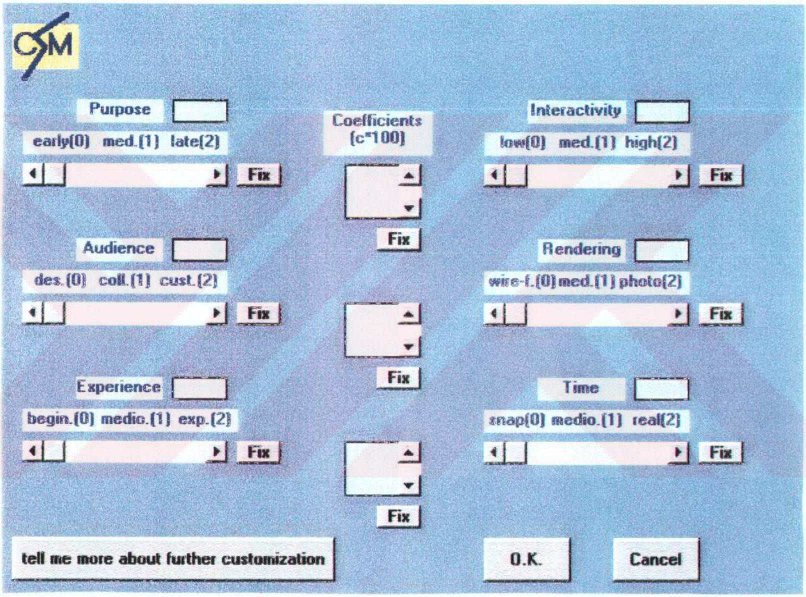

Figure A.15. On-screen CSM

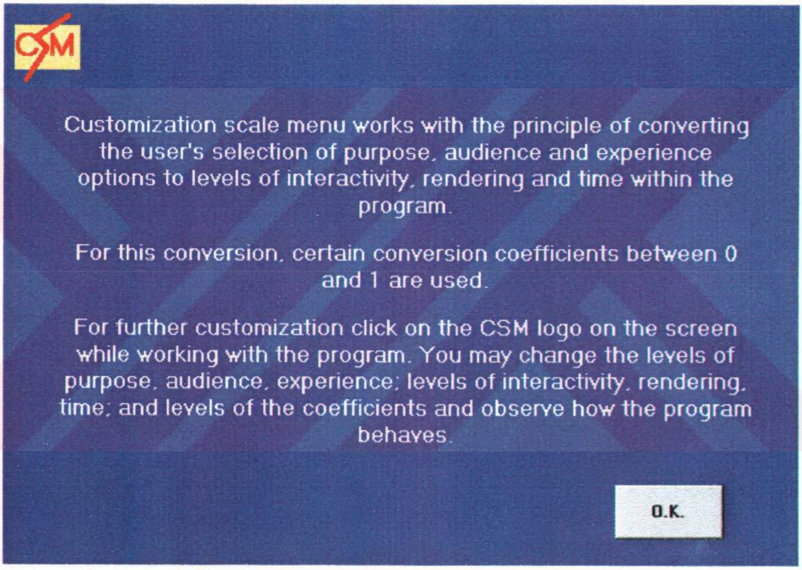

Figure A.16. Information about further customization

APPENDIX B: SELECTIONS FROM THE MENUS OF VIRTUS WALKTHROUGH PRO

Following excerpts are taken from the User's Guide of Virtus Walkthrough Pro 2.0 [Scott, 1994] in order to explain the functions of the referred menus and menu items in the tables illustrated in Section 4.

WINDOWS MENU (Table 4.1.)

The Windows menu display the name of all open windows. view and models, and it contains commands for displaying the Tools Pad, Depth window, Coordinates Window and Textures Window. The Windows menu helps you quickly navigate through open views and models.

Tools Window

Tools window displays or hides the Tools Pad. There is only one Tools Pad, though there are several sets of tools within the Tools Pad. If the Tools Pad is hidden in one view or editor, it is hidden for all views and editors.

Depth Window

Depth Window display or hides the Depth window. The Depth window is a ruler, like the Design View rulers, that contains a Depth Control Gauge. The Depth Window display the position and inflation distance for the active design view. If you change to another design view, the Depth window will display the position and inflation distance for the new view.

The measurement scale of the depth window ruler is the same as the scale of the rulers in the Design View.

Coordinates Window

Coordinates Window identifies the position of the mouse cursor in the active drawing area. The information in the Coordinates Window helps you measure and draw accurately. The Coordinates Window does not function in the Walk View.

Textures Window

The Textures window can be displayed or hidden with the Textures Window command under the Windows menu. The Textures Window is used to choose textures and apply them to selected objects and object surfaces, and to alter the display characteristics of textures.

Walk View

The Walk View is a window that displays a three-dimensional rendering of the objects that you draw in a Design View, and allows you to walk through and around the objects.

Design View

A Design View is a drawing area where you draw or view object outlines in two dimensions. There are six different Design Views: top, bottom, front, back, left, right. You may draw in any Design View. More than one Design View can be open at any time; however, only one view can be active at any time.

TOOLS PAD (Table 4.2.)

Design View Tools

The tools on the upper half of the Tools Pad allow you to create, edit and orient objects and surfaces. Some tools have other tools nested beneath them. Nested tools are indicated by a small arrowhead to the lower right of a tool.

Drawing Tools

Drawing tools allow you to draw 2-D polygonal outlines of basic shapes. When the basic shapes are combined and inflated , complex 3-D models can be created and rendered in the Walk View. Drawing tools are also used in the Surfaces Editor to draw surface features.

Lighting Editor

The Lighting Editor actually includes the Object Lighting Editor and the World Lighting Editor. Both look and function the same. The Lighting Editor lets you see the intensity and color of light sources inside a selected object.

Appearance Modifiers

Flat Shading Modifier

The Flat Shading Modifier is the default Appearance Modifier when creating a new object. With this modifier selected, objects and object surfaces are displayed with shaded surfaces controlled by default lighting and the polygonal outline of the object or surface.

Smooth Shading Modifier

The Smooth Shading Modifier is used when a more realistic image is required. This type of rendering softens edges, giving a smoother appearance to the curved surfaces of objects. Lighting still affects objects that are modified this way, but the effect is much less intense.

Color Bar

The Color Bar functions in the Design Views, and displays the default color for new objects. It also allows new colors to be selected, created and assigned to objects or surface features. Colors can be applied to translucent as well as opaque objects and surfaces.

The current color displayed in the Color Bar is the default color for all newly created objects or surface features. The default color and the color of existing objects or surface features can be changed.

TEXTURE (Table 4.3.)

Options

The Textures Options command displays a dialog with options for displaying a highlighted texture in the Textures window. The Textures Options dialog functions like a style sheet in that you set options for how textures are applied, but original texture file stored on your hard drive is not changed.

Name

The Name text field displays the logical name of the specific Texture Options settings displayed in the current dialog. The logical name is a specific name that you assign to the settings in Textures Options dialog.

Source

The text next to Source indicates the location on your hard drive of the texture file that is being referenced by the texture options that you are currently viewing.

Format

The text next to Format indicates whether the texture whose options you are editing is in PICT or Quick Time format.

Dimensions

The text next to Dimension indicates the horizontal by vertical pixel size of the texture whose options you are editing.

Edit

The pop-up menu next to Edit determines what options are displayed in the Texture Options dialog. The options in the Edit pop-up are First Tile ,Tile Pattern and Appearance.

First Tile

First Tile options affect the orientation of a placed texture.

Tile Pattern

If Tile Pattern is selected in the Edit pop-up, the Mirror and Cover options are displayed in the Texture Options dialog.

The Cover options determine how the texture repeats (tiles) across a surface. The options are the same, but there is a separate set for the horizontal (H:) dimension and the vertical (V:) dimension. The options are

Fit Tiles and Pixels/Unit. Only one option may be selected at a time. The default is Fit Tiles. Fit Tiles is the number of repeats of the texture in the horizontal or vertical dimension.

Appearance

Appearance offers the Shade and Decal options.

Shade allows a textured surfaces to be affected by lighting to the extent that a bright light makes the texture brighter, and a dim light makes the texture dimmer; textures are not affected by the color of a light. Decal allows textures to have holes through them.

RENDERING (Table 4.4.)

The Rendering options allow you to change preferences related to rendering in the Walk View.

Shading

Shaded

Displays objects with the effects of lighting.

Unshaded

Displays object colors with no lighting effects.

White

Displays objects with no color (white color fill) and no lighting effects.

Drawing

Fill & Frame

Displays both the color fill and wire frame of objects.

Fill

Displays only the color fill of objects with no wire frame.

Frame

Displays only the wire frame of objects. Frame color is a shade darker than the fill color unless *Black Frames* is selected.

Black Frames

Displays all object wire frames in black. If unselected, wire frames are displayed two shades darker than the color of the object or surface feature. Black Frames makes distinction between two adjacent surfaces more apparent, which can be helpful when printing.

Print Fill

Prints objects with black lines and white surfaces (hidden line removal). The result is a clean, black-line drawing.

Openings

See In

Allows you to see in from outside an object through translucent and transparent surfaces and surface features.

See Out

Allows you to see out from inside an object through translucent and transparent surfaces features.

See Through

Allows you to see connections between objects made with the Connect Surfaces Tool.

Dithering

Dithering is technique that allows more colors, thus more color-accurate renderings. Dithering is turned on by default. The disadvantage of Dithering is that the screen appears more grainy.

Blended Translucency

Blended Translucency offers a smoother look to translucent surfaces and, in most cases, a faster walk speed than in previous versions of Walk Through Pro. Blended Translucency applies the color of translucent surfaces to objects that you can see beyond the translucent surfaces. If Blended Translucency is turned off (not selected), a colored-dot pattern represents translucent surfaces in the rendering.

NAVIGATION (Table 4.5.)

Aids

Button Down

If this option is selected, the Observer moves when the mouse button is ressed and stops when the mouse button is released. If this option is not selected, the Observer moves when the mouse button is released and stops when the mouse button is pressed.

Cross Hair

Displays a cross hair in the Walk View that is used as a reference point for direction and walk through speed.

Velocity Grid

Displays horizontal and vertical marks at increments relative to the cross hair where walk speed changes.

Collision Detect (Collision Detection)

If this option is selected, the Observer is able to move only through doors or transparent openings on surfaces. When the Observer encounters a wall, it stops; a clicking sound confirms that the Observer ran into a wall.

Camera

Film

Allows you to specify the size, in millimeters, of the image on film.

Lens

Allows you to specify a lens focal length for the view. Film and Lens work together to determine the angle of view.

Aspect Ratio

The aspect ratio is the ratio of horizontal to vertical screen dimensions. Aspect Ratio options are: 35 mm Horz. 36:24, 35 mm Vert 24:36, 2.25 Square 1.1, 4x5 Vert 4:5, Television 133:1:00, Academy Aperture 133:100, Euro Wide Screen 166:100, US Wide Screen 185:100, Anamorphic Aperture 235:100.

To change the aspect ratio, point to the aspect ratio pop-up and mouse down. Drag to select the desired aspect ratio. Then select the Aspect Ratio check box to apply the new aspect ratio to the Walk View. If the Aspect Ratio check box is selected, the aspect ratio displayed in the pop-up menu box will override any size options set in the Snapshot dialog.