Articulating Architectural Design through Computational Media

by
Mark John Sich
Bachelor of Science, Building Science 1992
Bachelor of Architecture 1993
Rensselaer Polytechnic Institute

Submitted to the Department of Architecture
in partial fulfillment of the requirement for the degree of
Master of Science in Architectural Studies
at the
Massachusetts Institute of Technology

June 1997
© 1997 Mark J. Sich All Rights Reserved.
The author hereby grants to MIT permission to reproduce and to distribute publicly
paper and electronic copies of this thesis document in whole or in part.

Author: Mark John Sich Department of Architecture, May 9, 1997

Thesis Supervisor: William Mitchell Prof. of Architecture & Media Art and Sciences

Chairman, Dept. Committee on Graduate Students: Roy Strickland Associate Professor of Architecture
Reader: Thom Mayne  Principal of Morphosis, Santa Monica  Professor of Architecture, SCI-arc & UCLA

Reader: Takehiko Nagakura  Assistant Professor of Design and Computation
Articulating Architectural Design through Computational Media

Mark John Sich
Bachelor of Science, Building Science 1992
Bachelor of Architecture 1993
Rensselaer Polytechnic Institute

Submitted to the Department of Architecture on May 9, 1997 in partial fulfillment of the requirement for the degree of Master of Science in Architectural Studies at the Massachusetts Institute of Technology

Abstract

This thesis proposes the concept that computational tools can merge two high order cognitive operations for architectural designers. The first cognitive operation being the internalization of displacement (as defined by Kant and Piaget). The second being the construction of a mental Model (i.e. one that represents a tended external reality). The existing computational tools allow for a new external representation that has a relationship with the internal representation in the designers mind. The conceptualization of complex systems is in a direct relationship with the designers ability to visualize infopro (information processing) paradigms. The designer must be afforded the opportunity to undergo the cyclic process of conjecture, reading and evaluation.

How can an architects sensibility to physicality be integrated into computationally based modeling and representation. The larger contextual questions are; What are the inherent differences between actual physical models and computational models? What is lost and what is gained, and can they be reconciled with each other? What are the cognitive operations that are assisted through the use of these new computational models? As these questions are broken down, it becomes apparent that the difficulty of re-presenting the computational model back to the designer (and others) must be investigated. Interaction in virtual space is not new, but the investigation of an application using an architectural setting has not yet been explored.

I propose the creation of an immersive tool that will allow for a phenomenological creation of a virtual model. The application will examine the interaction that occurs between “reality” and proposed reality. The addition of an abstractive navigation system will help facilitate the conceptualization of a congruent infopro model. When the program is completed (first 1/2 of the semester), different design approaches will be tested by different end users with different physical interaction models. The output that is generated will lead to the proposal a new typology of spatial forms. The proposed implementation will incorporate the presentation model, but only for reflection, the primary purpose is a computationally centric design process.

Thesis Supervisor: William Mitchell Prof. of Architecture & Media Arts and Sciences
Acknowledgements:

I would like to extend my sincerest thanks to my Advisor, Bill Mitchell. Your insights showed me the path.

To my parents, Eugene and Jutta, Your unconditional love and sacrifices gave me the means and the vision.

To Daniel J. Brick, your support and friendship that you unselfishly gave, pulled me thru.

To Edgar Ngwenya, your skills and patient teaching, gave me the method.

To Thom Mayne, your mentoring gave me the chance to prove myself.

Last but definitely not least, to Andrea M. Stancin, your undying support and love carried me through this all. YOURS.

To all my friends at MIT, I am truly blessed to encounter so many wonderful people. I can not thank you all enough for everything that you have done. You will be in my thoughts forever.
Table of Contents

Title Page 01
reader page 02

Abstract 03

Acknowledgments 04

Chapter 1
  Introduction, overview 08
  Directions 13

Chapter 2
  Design Biases 16

Chapter 3
  Cognition 19
  Physical Vision 21
    Physical Energy 21
    The Eye 22
    Optic Nerve 25
    Visual Cortex 25
    Implications 27

Chapter 4
  Open Inventor 29
    Nodes/Fields 30
  VRML 31
  Scheme 32
  IVY 34

Chapter 5
  Media Manipulation 35
  Movement and Navigation in PHEMENOarch 38
  Manipulation in PHEMENOarch 40
Table of Contents II

Chapter 6
  Test Methods/Features 42
    SoTabBoxManip 43
    SoTransformBoxManip 44
    SoJackManip 45
    SoHandleBoxManip 46
    SoTrackBallManip 47
    SoTexture 48
    SoPointLight 48
  Testing Methods Proper 49

Chapter 7
  Physical Manipulation/Interaction 52
    The Computational Drafting Board 54
    Semi-Immersion 56
      The Brunelleschi Illusion 56
      The "Cave" 60
      The DIGIPANOMETER 63
    Virtual Reality 65

Chapter 8
  The Test Experiments 68
    Reflections on Abstractions 68
    Methodology: Experament (the Blue phase) 69
    Methodology: Case Study 75
      Edge Boundaries 81
      Light 83
      Transitional Boundaries 84
      Textures (trouble with) 85

Chapter 9
  Conclusions 86
Table of Contents III

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>89</td>
</tr>
<tr>
<td>Appendix A</td>
<td></td>
</tr>
<tr>
<td>To set up the environment for PHEMENOarch</td>
<td>101</td>
</tr>
<tr>
<td>Appendix B</td>
<td></td>
</tr>
<tr>
<td>To use PHEMENOarch</td>
<td>103</td>
</tr>
<tr>
<td>Appendix C</td>
<td></td>
</tr>
<tr>
<td>.iv Files and Alterations</td>
<td>106</td>
</tr>
<tr>
<td>Bibliography</td>
<td>115</td>
</tr>
</tbody>
</table>
INTRODUCTION

Overview:

Architecture is a means of communication. It solidifies ideologies, it takes a view. The purpose of design is to slide behind the facade of appearance and to explore the underlying truths we do not normally see, Pursuing idealized and idiosyncratic realities inherent in every project and form. It is my belief that the computer can be instrumental in these endeavors.

This thesis is aimed at a discourse into the tacit relationships between the exploration of the imagination, with all its inherent ambiguity, and the manner in which we re-represent our ideas back to ourselves for re-evaluation.

The context of Architecture is shifting. We are entering a new era in which our specialized languages of space and form can now be visualized and experienced by all and not just the few who can read and interpret plans, sections and models. For this empathic visualization is what
we are all about, we (architects) live in a virtual world of ideas. This world must be conveyed in such a precise manner that it can be brought into reality by hands other than our own. The newest and one of the most powerful tools, the computer, is emerging as a forerunner to help us with our almost insurmountable tasks. The implementation of the tool must be tacitly understood and critically applied. In my opinion, the computer is not being used correctly in most of its architectural applications. It is being used either as a simple tool of mass production or solely as a simple tool of presentation. Almost no one is using it as a design tool. The great power of the computer does not lie in the fact that it can perform 30 million recursive operations every second, but rather it lies in the fact that the computer affords us a reflection into our own minds. What are the issues in this redefinition of architecture and architectural design? These are the areas I endeavor to explore.

The scope of this thesis is to look at the inherent strengths and weaknesses of designing Architecture with in virtual environments. I argue that what I call "traditional" architecture has been constrained in regards to space and form creation, because of the inherent difficulties of abstracting oneself into an unfamiliar frame work. As a result, one found the necessity to rationalize
the environment around you and create a formal description of precedent, in order to begin to gain an understanding of the design that one is developing. In order to get back to the main issue of space creating, i.e. creating environments in which people live and work and play, I propose the using of the computer as a design tool of immersion. The conception and implementation of Architectural design has been closely aligned with the manner and technology with which it has been represented. For the first time in Architectural history, one can "build" the design proposal and then evaluate it from within, allowing for an infinite possibility of changes and iterations from a phenomenological perspective. One can achieve this and be economically viable as well. Architecture relies on an implication of space and it’s experiential qualities rely on motion through built space.

The implementation of the computer as the primary tools of design is an idea which is coming of age, schools and universities around the world are taking the first tentative steps into this "new" realm. One example of this is MIT’s virtual studios, but even here on the cutting edge, the direction of CAD development in the realm of architecture has been, on the whole, towards the later stages of design development. The machine has acquiesced into its role as a valuator of ideas.
or a communicator of established ideas across vast distances. This is obviously a very worthwhile endeavor, however I stand with the belief that the computer is not being pushed to its fullest extent.

Erich Mendelsohn sought to integrate the tensions of structure and form into a controlling image, he utilized a romanticized notion of the sketch as a tool of envision enabling. Jackson Polic used his psychoanalytic drawings. Gestalt psychology explored notions of subject/objectification. Felix Candela stated that, "a true work of art needs no justification by esoteric theories and is not necessarily the result of a process of conscious reasoning." This line of thought leads to my investigation the hypothesis that abstraction alone does not afford the designer the opportunity to fully undergo the cyclic process of conjecture, reading and evaluation. Using the limited scope of Visualization (there are other factors to architecture and its analysis), This thesis proposes the concept of designing within an immersive environment.

The utilization of these computation tools reevaluates the phenomenological approach within post-modern architecture. For this very reason
Immersion and engagement are ultimately necessary for the stimulation of a virtual architectural experience and its evaluation.

The issues of computer computation are on the forefront due to our abidance with the concepts of modernity. (What is new is better than what is old.) I take issue with this mentality, in 20 years, not a soul will be intrigued with the concept of the possibility of using the computer in the design process, it will have become a reality. This not to say that the role of models and sketches will necessarily be diminished, but there will be role shifting and the new affordances will dictate how and why a medium will be chosen.

The computer allows us another extension of our minds beyond the boundaries of the body. This ability has yet to be explored, and I suppose that the reason is due the underlying fact that the computer is acting as a mirror back into our own minds and souls. Hence, I believe that it will never be completely understood. The vigilance of computation must emphasize that the machine should not be used to quantify our existence and dictate our hands. This tool is bigger than all that. It must let us explore.

FIGURE 01.05
Description: Experiential Image of Santiago Calatrava’s ‘92 expo pavillion
Directions, the fork in the road.

One of the decisions that became prominent during the development of this thesis was the course that the development of the application should take.

The way that I saw the development of the application was in one of two directions. The first development approach was to create a tool that could stand as "THE" design development tool for architects. One would have to take advantage of the all levels of Abstraction (i.e. Plans, Sections, elevations etc...) plus have all the phenomenological feedback of an immersive environment. The interaction between these two methods of visualizing design space would have to be very carefully investigated from an architects perspective.

Initially my design strategy for the application was directed in this manner. I quickly discovered the overwhelming complexity of such a task. The physical development of the tool became but one of the challenges. The refinement required to integrate two separate information processing systems seemlessly became a tiring goal. The deeper that I progressed in my development, the more it seemed that I was shifting away from the main focus of the thesis, the investigation of designing within virtual space. At this time I had also noted...
that many of the commercial CAD packages would perform most of these tasks, to a greater or lesser extent, better than what I and this avenue of development could produce.

The second manner of application development, and for this thesis, the more fruitful, was to focus on the development of INFOPRO models, in a cognitive sense. The goal of the tool making became the development of a phenomenological 3D space manipulator that was so specialized that the designer could interact with the virtual space until the model and the designers internal representation became one. This goal lends its self to testing quite easily. If there is enough evidence that the level of error between intent and model is minimal enough, the results can be used to demonstrate other meta-cognitive operations that the designer may be embarking upon subconsciously.
The resultant application is called;

PHEMENOarch (phe•nom•o•arch)

The name derived from two parts.

The first part comes from the greek “philos”, meaning the study of phenomena. The name may also be interpreted as relating to PHEMENOlogy, the study of phenomena, which are the objects of knowledge, the only form of reality (1860), the view that all things, including human beings, consists simply of the aggregate of their observable sensory qualities (1865).

This is paired with the principals of ARCHitecture and architectural thought. Architecture, in my opinion, defies definition to a certain extent. It means so many slightly different ideals to different individuals. I am defining architecture, in this instance, as being the process of space creation and evaluation.
DESIGN biases,

Everyone is a prisoner of his own experiences. No one can eliminate prejudices—just recognize them.

-Edward R. Murrow

As with any investigation, the research will always be imperfect due to the imperfection of the researcher. There are inherent biases that all of us have developed. In order for the reader to better understand where the thesis is originating, I believe that I must disclose some of my personal design methodologies, so that the readers of this thesis can begin to untangle my bias to a certain extent.

"the designer is always faced with a decision of intention and not merely a deterministic process", -Alan Colquhoun, typology and design method

In thinking about design processes, it is difficult to isolate episodes in a design process. One can investigate and observe a series of alterations of
a singular design move using different methods. For instance, one can see medium explorations quite readily, when does a particular designer go from model to sketch to section? One can also note geometric vs. topographical paradigm shifts. However, I argue that the designer does these shifts not in relation to some predestined set of rules but rather by using his/her intuition as to when to shift paradigms. A good designer will never travel the path of creation the same way twice. The requirements of each design bring out a necessity to explore certain criteria with a specific design hierarchy.

The methodology for this investigation can be charted into 3 distinct phases. The first is conjecture, the true act of creation. One asks the question "what if?" The second stage is a re-reading of the form or decision. "what is it that I have just created?" The last phase is one of judgment. "Does this move make sense? How can it function more effectively? etc...." Once this evaluation is complete, the designer is coerced into the cycle again by trying to answer the questions that the judgment phase left unanswered. One reflects at a meta-cognitive level about operational-level moves and strategies, a realization after the fact.

I also believe that Helmholtz's constructivist perspective theory comes into fruition during design. The primary focus of constructivist theory is that
perceiving and thinking are active goal-oriented processes. One idea leads the mind and the perception to attempt a certain range of design strategies much like the eye performs saccades, a scanning over an image in order to understand the gestalt, and Schemata, which proposes where the eyes should look next in order to more rigorously investigate the object/image.

Using Don Norman’s concept of affordances. What does the tools feedback ential, encompass or mean? What does it allow you to do?

A design tool for designers and architects must afford the designer the opportunity to undergo these cyclic phases. Any tool that elevates the architects awareness of the full implications of the design moves that he/she are doing, will ultimately increase the effectiveness of the architect. There by creating a more robust and fleshed out architecture.
COGNITION

The first steps towards an immersive tool must be tempered with the understanding of how we cognitively understand the world around us.

In this chapter, I will discuss how the mind sees and understands visual input. The last few years have seen a marked increase in the science of human cognition. Human cognition is described as the branch of psychology that investigates memory, perception and thinking. I conject that with its understanding one ultimately gains a better understanding of human psychology and human design processes. This understanding is extremely relevant, especially when we investigate design inception.

To begin, Human beings have been gifted with a myriad of senses. We have the senses of audition, taste, touch, smell and vision. It is a full combination of all these stimulated senses which informs us of our placement and environment. It is this culmination that makes the implementation of immersion in a virtual setting so difficult. To truly, be "immersed" all these senses must be accounted for and stimulated in real time. However, the status of computational power is not yet ready to
adequately tackle such an endeavor. So as a result, of the different senses to investigate, the faculty of vision is the most influential in the dialogue of design and perception. This is where we will begin.

"Vision... is the big window", Robert L. Solso, 1994
PHYSICAL VISION

Vision is an analytic operation of "divide and conquer". The basic path for visual information is

physical energy->eye->visual cortex->associative cortex

These are described in further detail.

Physical Energy (Light):

The eternal world is alive with different electromagnetic radiation. One form of which is visual light. This Electromagnetic radiation is created by electrically charged particles. This radiation has definable physical characteristics, these are described in wavelengths. These waves are then reflected and/or absorbed by differing degrees when they collide with objects of varying materials. These There are two seemingly opposing theories of light, one theory states that light is a stream of particles. This theory was first proposed by Sir Issac Newton (1642-1727). The other theory of light, by Christiaan Huygens (1629-1695), was that light is a pulse or a wave, that it had no physical form. Without getting in to contemporary physics, they are both

FIGURE 03.03
Description:
Sir Isaac Newton using a prism to split sunlight into spectral colors
right. Light behaves as both a particle and as a wave, but not at the same time. The computational modeling of this phenomenon will be discussed in a latter chapter.

The Eye:

The eye is the organ which is able to detect the differing wavelengths that are reflected by the environment. Commonly, humans perceive electromagnetic wavelengths that range between 375 nanometers and 780 nanometers. The shorter wavelengths are seen as violet or blue, and the longer wavelengths are seen as reds. Greens and yellows fall between these values. The amplitude of the light also effects the perceived brightness of the light, a smaller amplitude will be registered as a dimmer light.

Light enters the eye through the pupil and is focused. The cornea is found on the surface of the eye and is followed the iris and then the lens. The lens does not actually bend the light, the aqueous humor (fluid) within the cornea performs this task. The ciliary muscle deforms the lens in order to focus on objects. A closer object is in focus when the ciliary muscle is relaxed there by reducing the curvature of the lens. Consequently objects further away focused by tightening the muscle forcing the lens into a deeper convex configuration. The mass of the eye is made up of

\[
\text{FIGURE 03.04} \\
\text{Description: wavelengths in the electromagnetic spectrum}
\]
Vitreous humor, a dense liquid that maintains the form of the eye.

Light then strikes the back of the eye. This area is called the retina (from the Latin "rete" translated as "net"). The receptor cells, of which there are two types, are sensitive to electromagnetic energy. Rods are receptor cells which are more sensitive to grays and lower intensity light. Cones are receptor cells which specialize in color differentiation. These cells translate light energy into signals that are passed to the bipolar cells and then to the ganglion cells (collectors), which in turn pass information to the optic cortex.

With in the eye proper, there are about 125,000,000 rods and 6,500,000 cones. The distribution of rods and cones across the retina is not uniform. There is a 2mm. indentation opposite of the pupil called the Fovea. The Fovea or Macula Lutea (yellow spot) is densely packed with Cones and no Rods. It is this distribution which accounts for the reason that a truly focused area that one can see is only 1-2 degrees. This area of focal vision is called Fovial Vision.

It might be said that by moving from the center of the human retina to its periphery we travel back in evolutionary time; from the most highly organized structure to a primitive eye, which does little more than detect movements of shadows. The
very edge of the human retina... gives primitive unconscious vision; and directs the highly developed foveal region to where it is likely to be needed for its high acuity.

-Richard L. Gregory

The eye can detect or perceive visual phenomenon with in a field of 180 degrees horizontal to approximately 130 degrees vertical. Fovial vision is as mentioned about 2 degrees, however parafovea, the area that is perceived as focused although less defined, is up to 30 degrees. These limitations are the cause of our necessity for constantly scanning our eyes across an image to gain an impression of the full object.

The understanding of these fundamental principals of human vision are essential for integrating a virtual design environment. The interesting computational advantage is that one must only render a small area very accurately and the rest can be approximate, if one can project the virtual environs exactly where the eye looks. This is an intriguing concept but not in the scope of this thesis.

FIGURE 03.06
Description: Visual Fields limits of monocular and binocular vision. note: distortion of symmetry caused by occlusions from the nose.
Optic nerve

The eye passed its information on to the optic nerve. These nerves (one for each eye) are merged at the optic chasm. The information from the eyes are split such that the visual fields (not what information are from which eye) are passed to the opposite visual cortex. I.E. the right half of your perspective viewpoint is transferred to the left side of your brain, even though the actual information is coming from two distinct eyes.

Visual Cortex

The visual Cortex receives the visual information, responds to specific visual stimuli. This infers that images are broken down into a specific set of stimulated neurons. If the image changes another batch of neurons will be stimulated. Each cell seems to have a "particular shape of stimulus and to one particular orientation" David Hubel (1963)

David Hubel and Torstian Wiesel shared the 1981 Nobel prize for physiology and medicine.

The implication is that we do not see a holistic object (which was once thought) but rather we place value upon certain stimuli and reassemble an image in our minds.
This is backed up by an intriguing study by Levy, Trevarthen and Sperry in 1972. They studied commissurotomized patients, these are people who have had their corpus callosum cut. Corpus callosum is the connective tissue that allows the two halves of the brain to talk with each other, if this tissue is severed the two sides of the brain can not communicate. When these patients were shown a chimeric face (in this case the left half a woman and the right half a man composted together), an interesting pattern emerged.

When the patients were asked to describe the face verbally, they picked the man to describe. This fostered the understanding that the left hemisphere the side of the brain that computes the right side of an image, is associated with verbal information processes.

These same patients were then asked to pick the face that they were just shown from a selection of whole faces. They chose the woman’s face. This seems to imply that visual and or pictorial processing is done through the right side of the brain.

Harris 1978 p463

The left hemisphere operates a more logical, analytic, computer-like fashion, analyzing stimulutus information input sequentially, abstracting out the relevant details to which it attaches.
verbal labels:

The right hemisphere is primarily a synthesizer, more concerned with the over all stimulus configuration, and organizes and processes information in terms of gestals or wholes.

IMPLICATIONS

This cognitive information would seem to imply that designers and Architects are more deeply involved with right hemisphere brain processing than left. It can also be concluded that this hemisphere of the brain should be stimulated and/or fostered to a greater extent during design charrette.

This cognitive understanding also points to the fact that almost all software that is developed is incorrectly set up for architectural thinking processes. According to the afore mentioned studies, the menu bars of software should be placed on the right hand side of the screen in order to more effectively stimulate the left hemisphere of the brain, the side which is concerned with labeling and analysis. Along with this directed consciousness concern, the main work space of a program (i.e., the view space/interaction zone) should be placed on the left side of the screen. This placement would stimulate the right side of the brain,

FIGURE 03.09
Description: AUTOCAD interface
text: The left side biased menu system and the text based command line. Both are not conducive to Right brain thinking/creativity
the side concerned with the understanding of the whole picture.

If one were to examine the popular softwares of our time, one would discover that they are all laid out in exactly the opposite way. They are all reversed. And some of the most pervasive CAD software of our time, like Autocad, have yet another difficulty. This difficulty stems from the fact that they are text based interfaces (for the most part) which are left hemisphere intensive but the work of 3 dimensional modeling and creation is more of a gestalt thought process requiring right hemisphere thinking. The two methodologies are counter intuitive and counter productive for the most part.

In the software, PHEMENOarch, that I have created, the emphasis of the interaction with the geometric forms and their generation stems from a right hemisphere stimulation approach. The designer is rarely removed from the designing process by being forced to undergo change of thinking through a menu system. The main menu items that one is required to access, are displayed in the right side of the screen in an attempt to minimize concentration shift paradigms.

FIGURE 03.10
Description: PHEMENOarch Interface
PHEMONOarch is scripted/written with Open Inventor. It makes sense to give an overview of the affordances of this language when coupled with Scheme.

**Open Inventor**

Open Inventor is an object-oriented 3D graphics toolkit. Each "object" on the screen is represented as an "object" in memory. For those of you fortunate enough to have taken 6.001, this is like the object-oriented "scheme-builder" game.

Each object is stored in the scene database, SoDB, which stores such information as the name of each part in the current scene. The SoDB then holds the objects in tree-like data structures. This fundamental data structure in Open Inventor is called the scene graph. A scene graph is a directed acyclic graph. Most of the operations in Open Inventor involve applying actions on to the scene graph proper. These actions traverse this tree from left to right, and from the top down. For example, when Open Inventor renders a scene, it does so by applying a render action to the scene graph. When a program wants to find a certain node in the scene...
graph, it does so by applying a search action to the scene graph. It was this methodology that was used in constructing the alterable selection nodes in PHEMENOarch. The scene graph paradigm that Open Inventor uses, has been shown to be a successful one. This paradigm has been adopted in many current and forthcoming 3D graphics standards.

Most of the programmatic interaction involves nodes in the scene graph with Open Inventor. Nodes often contain data elements known as fields. These Fields encapsulate information that Open Inventor uses to perform certain actions, like rendering a scene or object.

Nodes

A node is an object which stores some usable piece of information. Each node has an associated data type, or class, which defines the operations, or methods, which can be performed on that node and the data which that node can store. All nodes are derived from the base type SoBase. This means they are specializations of this data type; a node can do everything an object of type SoBase could do, and more.

Fields

Nodes usually contain data elements within
fields. Each field within a node is also a complex object, rather than a basic data type. The three main reasons for doing this are: All field types have consistent set and get functions. Hence with the use of setValue and getValue, one can retrieve values stored in a field of all field types because they are consistent. Also a node can tell when the value of a field stored within it has changed, and can automatically notify the application. The last reason is that because Fields can be connected together in chains, the manner in which Open Inventor screens it value is by automatic propagation down the chain. This mechanism is known as notification and can be used to reinforce the application of actions to be taken and/or have been taken.

VRML 2.0

VRML 2.0 is the current standard for displaying 3D graphics on the World Wide Web. Many spectacular example abound, For more indepth reading investigate Daniel J. Brick’s thesis on Virtual site interactions. VRML is based on the Open Inventor " .iv " file format with a few added features. VRML carries over the concepts of the scene graph, nodes with "fields", engines, and sensors. Most VRML browsers are implemented as Netscape plugins.

There are explicit differences between VRML and
Open Inventor. For instance, children are stored explicitly in "children" field of group nodes. Fan-outs, where a single engine create multiple nodes, are allowed, but fan-ins, where multiple fields feed a single engine, are not.

SCHEME

Scheme is a statically scoped and properly tail-recursive dialect of the Lisp programming language invented by Guy Lewis Steele Jr. and Gerald Jay Sussman. The design intent was to have an "exceptionally clear and simple semantics and few different ways to form expressions". A wide variety of useful programming paradigms (functional, imperative, and message passing styles) are made readily convenient in Scheme.

Scheme was one of the first programming languages to incorporate first class procedures as in the lambda calculus. Scheme was the first major dialect of Lisp to distinguish procedures from lambda expressions and symbols. By relying entirely on procedure calls to express iteration, Scheme emphasized the fact that tail-recursive procedure calls are essentially goto's that pass arguments. Scheme was the first widely used programming language to embrace first class escape procedures, from which all previously known sequential control structures can be synthesized. More recently, building upon the design of gener-
ic arithmetic in Common Lisp, Scheme introduced the concept of exact and inexact numbers. Scheme recently became the first programming language to support hygienic macros, which permit the syntax of a block-structured language to be extended reliably.

-Kenneth B. Russell

Due to the fact that Scheme is an interpreted language one can interactively edit Inventor programs and scene graphs. One can even include Scheme code in Inventor scene graphs. Callbacks, for instance, can execute Scheme code rather than call a C function. These callbacks can then be interactively modified at runtime.
Ivy

If you are running the application PHEMENOarch on Athena, one will have to set up Scheme with the base Inventor code. Ivy is a Scheme binding for Open Inventor.

http://www-swiss.ai.mit.edu/scheme-home.html

Ivy works by providing a consistent interface to the member functions and class variables of the Open Inventor classes. The underlying C++ functions are called from the Scheme backend. From the user’s point of view, it is a simple syntactical change.
Media Manipulation

How manipulation should behave in PHEMENOCarch

The task of manipulating objects in 3-dimensional space is a difficult one at best. The human mind is used to operating in a 4 dimensional world space (i.e. the 3 axis of x,y,z plus the dimension of time) but we rarely need to THINK in 3 dimensions, let alone 4. Attempt to solve any of the metal linked ring puzzles or Rubix cube at your local mall and you see exactly what I mean. The solutions are a simple game of geometric manipulations, however when they are coupled with the element of movement (a 4th dimensional mental manipulation, i.e. having to pass this ring through that slot, while twisting the metal angle into position etc...), these solutions become extremely difficult to conceptualize and even harder to realize.

The difficulty of visualizing true 3 dimensional space is even translated to the experienced designer/architect. Rarely, if ever, does one find an architect that designs directly in axon or model, the usual methodology is to break 3 dimensional space down into 2 dimensional

FIGURE 05.01
Description: Interlocking Metal Ring Puzzle
abstractions of plans and sections. These abstractions allow the designer to formulate relationships and a parti. This parti then develops into an ever increasing array of spatial abstractions (sections, details, built models etc...), as the designer is able to slowly internalize the implications of the space and forms that they are creating. I would not attempt to imply that this methodology of decomposing a large problem into smaller more manageable abstractions should be abandoned or dismissed. Rather, I see it as a need to be able to allow the designer more tools at his/her disposal in order to better understand what the proposed design really is. In this particular thesis, the concentration lies in the need to relate to the designer, in an intuitive manner, just how forms and spaces can be constructed in virtual space.

The manipulation of the virtual tools must be crafted as to not break the cognitive infopro model of immersion. The tools must allow for the user to be able to manipulate, change and create forms and spaces without ever having to look at a menu or other left hemisphere abstraction models like text or menus.

The manipulation tools must be intuitive to use.
This philosophy towards interface and interface design required a reevaluation of all the commercial Modeling packages. Through this research of existing modelers, the realization dawned that this methodology to spatial creation had never been implemented before.

The packages that least exemplified the concept of intuitive manipulation of spatial relationships and forms were those that required a cognitive effort to switching between creation, manipulation and evaluation nodes. The most notable of which was Autocad, in which to even move an object vertically required either typing in the coordinates or changing the ucs (user coordinate system) to another reference plane (x,y or z), shifting to "home" in order to even see the form in elevation/plan, changing modes to move the form (requiring at least 3 mouse clicks), moving the form and finally evaluating the form with "VPOINT" which does not even give you a preview of the viewpoint until you have accepted it.

The next level of intuitive use, would be the Alias/Wavefront and Form•Z packages. Alias allowed for the designer to evaluate the created forms easily and usually in smooth realtime. The form model creation paradigms (especially if you are dealing with splines) are very intuitive in that you can see your changes on the screen automatically. The menu system and other manipulators
leave a lot to be desired however. It is very easy to become lost or confused with the hap-hazard methodology of menu tools. Form •Z allows for an even more intuitive object/form creation, but its evaluation system is not as advanced as Alias. This last difficulty may stem from the power of the platforms that both software packages are written to use, Alias for high power/ high end SGI machines and Form•Z for lower end Macintosh and Windows NT machines.

PHEMENOarch does not pretend to have the modeling capabilities of even the lowest level of commercial modeling softwares, the interface, however, does bear some scrutiny.

Movement and Navigation in PHEMENOarch

When PHEMENOarch is married with a Virtual Reality system, the navigation is literally as simple as swiveling your head around to see the objects for evaluation. This exactly mimics what we do in the physical world. This “hands-off” visualization technique allows for the designer to concentrate on the tasks of conjecture, reading and judgment with in the act of creation.
When PHEMENOarch is placed on a regular display system (many of which will be discussed in a later chapter), the navigation is still intuitive. The viewer is constrained such that the designer move about the space in an analogous manner to walking. One may move forward or back in a horizon plane, turn corners or stand still and look about. The designer is also afforded the ability to elevate his/her viewpoint by consciously requesting the elevation change with a click of the mouse. There is no freeform “flying”, which is a characteristic inherent in the other browsers that have been investigated. To see an example of this “flying” just look at the proliferation of bad animations that have recently become abundant in which one shoots through the latest developer community at a height of 150 ft, then dives into an open window (or through a wall) and zooms past the kitchen. This may be flashy but it has very little to do with human inhabitation and viewpoint. PHEMENOarch allows for the operator to maneuver as if he/she were in a real spatial environment. One does not suffer the loss of placement within a space. Every move is contiguous. This is not to say that one can not get lost in a space, but the proclivity for it is decreased.
Manipulation in PHEMENOarch

The careful consideration of movement and manipulation had to be investigated. PHEMENOarch's selection system is straightforward, one places the cursor over the form to be manipulated and select. A manipulator appears attached to the form that has been selected. The manipulators can then be utilized or toggled to the next manipulator, which can then be used or toggled yet again. All your choices are immediately at the disposal of the designer. There is no need to go to a menu bar to manipulate scale, rotation and/or movement parameters. This methodology retains the immersion of the designer such that it feels as if one is able to move the walls out, the ceiling down, etc.

Open Inventor offers with in its toolset powerful selection manipulators. The strength of these "manips" are that they have an iconography built in that is readily intuitive for the designer to internalize and use. To rotate a form, for instance, is a simple act of grabbing an edge and sliding it in the desired rotation direction, the SoDB is instantaneously updated such that the screen displays the changes interactively. One can see the
updates as one works, in realtime. The investigation of the implications of these individual manipulators are discussed in chapter 6.

There was also one other concern that needed to be addressed in a serious manner. The method of primitive form creation. In the earlier versions of the program, new geometries were always created at the origin. This methodology was acceptable for small, compact models. However as the models increased in size and complexity, the designer would have to return to the origin in order to "make supplies", and then they would have to transport them to the site where they would be used. This was analogous to a bricklayer going to the palate to get more bricks. The methodology was not very conducive to a seamless creative environment. The solution was to have the SoCube create the primitives in relationship to the objects that are currently being manipulated (otherwise if no other objects were selected, it would just create the primitives at the origin). With this methodology, the designer could continue creating and not have to worry about resource management.
Testing methods;

In order to understand some of the testing methods, a few of the applications various features and versions must be explained.

Several versions of the Application PHEMENOarch were created, in order to test the effectiveness of differing abstractive barriers and functions. Especially what functions were conducive to differing designing styles and attitudes toward the design process.

The most noteworthy are the differing manipulation tools. These were tested separately to understand their affordances, before they were combined to provide a fully functional application set.
These manipulation tools are:

SoTabBoxManip:
This set of manipulation tools allows the designer to non-uniformly scale primitives. The movements are constrained to an orthogonal grid. There is no rotation afforded. The resultant forms were there for classically architectural in nature.

The Manipulator is transformed by dragging over the green highlighted edge conditions. Dragging on the mid edge segment allowed for the primitive to be scaled in one direction, this is analogous to moving a face in a perpendicular plane. Dragging on a corner segment, creates a free face transform of the three adjacent sides.

Moving the entire object required dragging the object but not on a vertices. This sliding movement is constrained to the axis that the picked face defines.
SoTransformBoxManip:
This set of manipulation tools allow for the designer to uniformly scale primitives (like cubes) and unrestrained rotation in free space. The resultant forms were more organic/plastic in nature as a result, as we shall see later.

The scaling function is activated by dragging the corner boxes. The Primitive is then scaled uniformly.

Moving the object in space is much like the SoTabBoxManip. One must click and drag on a face of the object and it will slide along that plane.

Rotation of the object is accomplished by dragging an edge. The Edge will become highlighted and will rotate around the centroid of the object as if one were pulling it from that point.
SoJackManip:
This manipulation tool set allowed for easy placement of objects in accordance with a reference axis. Height translations were facilitated by a drag handle in the center of the object to be manipulated. However, the lateral translations were exceedingly difficult, especially when one could not get hold of the edge of the primitive. It was the most encompassing but also the most counter intuitive to use.

The primitive can be scaled uniformly by dragging on the end boxes that appear off of the axial spines.

The lateral translations are actuated by dragging the edge of the superimposed symbolic box. A Transitional plane will appear along the lateral movement plane, it shows where the form will pass through.

Vertical translations are handled by a superimposed column in the center of the primitive. This column can be dragged down or up to facilitate the movement of forms when the edges are not visible.
SoHandleBoxManip
This manipulation tool set allows for the scaling of primitives from the centroid bidirectionally. It is extremely useful when a condition for center alignment between two forms is required, i.e. the pediment and its associated column.

Scaling is activated by grabbing the boxes at the ends of the axis. The two coplanar faces then move in a mirroring fashion in relation to one another.

Movement is facilitated by dragging the form from a face proper.
SoTrackballManip

This manipulator tool does not afford movement. However it allows for primitives to be rotated axially and free. It also allows for the user to alter the centroid of rotation, such that one can rotate the form around any point.

The movement of the axis point is activated by dragging on the arrows at the junction of the rings.

The axial rotation is activated by selecting the ring and the primitive will slide along the activated path.
Other Virtual environmental variants;

SoTexture
One version of the Application has texture applied to the primitives. The heightened realism seemingly removes another level of abstraction for the designer. The designer is afforded the ability to react to the materiality. The computational drawbacks of this version are large. The added memory and redraw requirements slowed the application. This redraw retardation disrupted the perceived smoothness of the manipulation and navigation. It was there for difficult to work with.

SoPointLight
The point light was present in all versions. The standard viewer is equipped with a headlight function. This function insures that all the objects are always visible by showering the scene with light as if it were emanating from the viewpoint itself. With the advent of the added point light source, the headlight can be manually switched off. This can be done to approximate a daylighting scenario. This is a testable change that would seem to hold promise in discovering variations of design methods in accordance to environmental variables. Will designers change their parti with the added/subtracted information?
The study of influences of differing versions of computational media and its ensuing feedback, will, in this thesis, be confined to observing the participating designers as they interact with the application in the design process. The examination of this media-manipulation-in-process should help inform the refinement of the application and the investigation of finding emergent typologies buried within the act of designing within differing computational environments. It is clear that there will not be one overriding typology due to the simple fact that having individual designers from different backgrounds, their styles and approaches to design will inherently be different as well. However, I propose that there are certain natural characteristics that become apparent due to cognitive psychological concerns that all humans have. It is these natural characteristics that led architects like Palladio to develop rules for designing spaces. They may not have known exactly what it was that they were after but they reasoned that certain types or forms of spaces exhibited the qualities that they attributed to "balanced" architecture. It is the use of manipulatable virtual environments that could hold the key to some of these latent characteristics.

In the course of the thesis, a set of designers will be shown the application along with its tools and
their operation. After being allowed an adequate time to get to know the program and its idiosyncracies, the test designers will be asked to use the Application.

The design problem to be solved will be left up to the individual designers, as this will clearly demonstrate the inherent aptitudes or strengths of each version of the application and how it would be naturally utilized.

The architects will be tested using the ortho-constrained version and then again with the non-ortho constrained version. The result of the separate tests will be saved and evaluated later.

One set of testers will be given the virtual drafting board version. The familiarity inherent in this type of media/interface would appear to give its usage an edge.

Another set of designers will be given the large scale projection/semi-immersion environment interface.

Another set of designers will be given the Virtual Environment in which to design.

Lastly, one small group (perhaps just one individual) will be asked to use all the versions in an ascending order of reality, descending order of
abstractions. They will begin with the drafting board and end with the VR.

The final step of the process of all of the groups is to test the designers on what they believe that they have created. Each architect will be asked to draw or draft the environment that was created, WITHOUT looking at the created file for reference. This will bring to surface the intent of the designers. Any discrepancies will there for point to a discrepancy in understanding of form (if it is very complex for instance), an underlying/sub-consiencious emphatic relationship to the created space or perhaps a faulty visual feedback mechanism. All of which are important factors in fully understanding what it means to create in virtual environments.

The files that are created will then be investigated. emergent typologies will be sought. Some of the questions that will be investigated are: Does one version of the application allow for true space making? Does one version foster object making? Is there a particular level of complexity that is discovered? Are there inherent restraining angles and lines of force that can be abstracted and utilized in differing media?
Physical Manipulation/Interaction

Architects are familiar with the extremes of abstraction/perception, they understand large/full scale interactions. This familiarity comes from experiences with built forms and their cultivated awareness of true space. Most computational interfaces are limited to a 15-17" monitor. This simple limitation does not allow for the architect/designer to adequately examine the finer expressions of scale and size.

In this exercise, the problematic difficulties relating to scale will be addressed in three stages. One being the introduction of the subject/designer into a semi-immersive environment. The second being an illusory displacement created by an approximation of Brunelleschi’s peepshow, this can be accomplished in two manners, one the recreation of the peepshow, the other a large scale simulation of the projected environment. The third being the introduction of a fully immersive Virtual Reality environment.

I discovered very early in the development of the application, that without a discourse on the physical interactive relationship between the designer and the virtual model, the infopro relationships would be
lost. Basically invalidating the experiment. So after careful consideration, stages of designer interaction needed to be addressed. The levels of manipulation became 4 fold. The first starting from the perspective of the present day architect that is familiar with 2-d representations presented in a drafting board environment. The next being the introduction of a "cave" environment, that allowed for large scale projections of the virtual environment. The third stage is the introduction of perspective point matching illusory space. The final level being the introduction of a Virtual Reality environment to the designer.

FIGURE 07.02
Description: PHEMENOarch using traditional Architectural Media
The Computational Drafting Board

Architects are familiar with the abstractions that are inherent with direct media manipulation. The most direct example of this is the simple act of putting pen to paper to sketch or draft. In some ways the phenomenology of tactile feedback has been a part of architecture since the beginning. The German term fingerspitzgefühl, or "finger tip feeling" adequately describes the relationship that designers foster with their media in order to communicate their ideas and desires.

The feedback that the fingers return during the act of drawing has been mostly lost in the computer era. The principal reason for this disparaging discrepancy is that with most computer systems, the display is removed. The act of having the display remote creates a disjunction with the designers' ability to manipulate the media.

With some simple display projectors and a calibrated image reflector, one can project a screen upon a surface. If this imaged surface is calibrated with a stylus pen (a WACOM tablet in this instance), one can return some of the familiar tactility of direct media manipulation to the user. One of the side effects that I discovered with my particular setup was that due to the nature of projection, one had to calibrated the cursor tip approximately 1/4" above the actual tip of the
stylus pen. This was done to compensate for the occlusion/shadowing created by the hand and pen. The best solution would have been a back projection but due to the nature of the input devices involved in this case this is impossible. This small shift in proposed direct media manipulation, created difficulty for some of the test subjects at first. Test designers were eager to engage with this medium immediately, sometimes even sitting down with it and experimenting without being asked. The interface was very similar to what they already were intuitively comfortable with, except for the one minor exception (the 1/4" shift) and as a result it took some real paradigm re-evaluation. After 5 minutes all the subjects were comfortable with the interface.

FIGURE 07.05
Description:
Viewpoint of Drafting board user

FIGURE 07.06
Description:
Drafting Board Workstation and Environment
The Semi-Immersion

The application of semi immersive environments is one that most computational and CAD designers are very familiar with. The projection of perspective images upon a 2-Dimensional screen is far from novel. The difficulty lies in the fact that most 2-Dimensional display media have no intention of providing any kind of immersive feedback mechanisms, i.e. they are predicated upon just displaying very basic images and relationships of the virtual space. For most applications, be that engineering, industrial design, urban planning, this is more than adequate, but for creating built habitable environments, like those that architecture proposes, this is simply not acceptable. In order to investigate this discrepancy I have broken this particular problem into two separate instruments.

The Brunelleschi illusion

Fillippo di Ser Brunellesco (Brunelleschi, 1377-1446) embarked on an experiment that "marked an event which ultimately was to change the modes, if not the course of Western history"
Brunelleschi had painted a panel that was the first painting to embody the use of linear perspective. In order to illustrate this point, Brunelleschi had drilled a hole in the painting, such that the viewer's eye would be constrained to the center of projection. The painting was placed such that it faced the building that it represented, and as the observer looked through the back of the painting, they could see the actual scene. Then a mirror was placed in front of the painting, reflecting the image back through the hole to the viewer. It seemed that the real thing was being seen.

As it turns out, Brunelleschi's method was very effective for creating the illusion of depth. This illusion was very compelling for two reasons: First, it forced the viewer to see from the very point of projection, making the picture a projective surrogate for the scene and at the same time, it reduced the eyes information as to the limited depth of the image plane. There is at least one more method of enhancing monocularly viewed images. If a lens is inserted between the subject's eye and the image, and if the lens approximates the focal length of the lens that the image is taken with (if it is a photograph that is), then the "plastic" depth that can be obtained is quite similar to binocular visions depth of field. These techniques
can be used with computational imagery as well. The image could be that of the virtual space and the focal length can be approximated in accordance with FOV (field of vision).

It has been demonstrated that a proper central projection can be mistaken for a real environment. Smith and Smith (1961) asked subjects to throw a ball at a target in a room that they could only see through a peephole. Half the study was shown the actual room in the view piece, the other half was shown a photograph taken at exactly the same position. There seemed to be more variation in the throws done by the subjects that were looking at the photograph but this can be accounted for by the direct view affording some monocular parallax. The most interesting aspect of the study was that the subjects had no awareness that they were seeing a photograph and not the actual room.

This pushed the thesis to explore the plastic effect that was created using this methodology. A relatively simple buffer was set up using an optical convex lens. The buffer constrained the possible view points to just one. This view/eye point being at the center of the computationally projected perspective. While the imagery seemed to display the "plastic" depth that aided in the per-
ception of sizes and scales, some of the test subjects complained that they were getting headaches from viewing the monitor in such a constrained position, and with the varying intensity caused by the interaction between the convex lens and the CRT monitor display.

**FIGURE 07.12**
Description: Placement and use of eyepiece

**FIGURE 07.13**
Description: The Eyepiece
The "cave"

The second stage of semi-immersion is accomplished by placing the designer at the perspective focus point of the computer model. It should be noted that this is just a slight altering of the Brunelleschi method, instead of placing just the eye of the observer at the nexus of the image, you place the entire designer in the nexus of the image, thereby surrounding him/her.

The observers field of vision will almost be totally dominated through the use of a very large projection of the virtual environment. This type of environment has often been called a "cave". The concept of the cave is very simple, and relatively convincing. The typical cave uses three wall projection systems, with each being directed at a different face (usually one roof and two side walls). In the environment that I have set up, the projection will be directed towards a primary wall/screen, and the spillover/periphery will be projected on the ceiling, two adjacent walls and the floor.

The subject was to use the stylus as seen in the afore mentioned version to navigate and interact with the virtual space, however due to technical difficulties, the stylus was reduced to a typical mouse and keyboard. There were many technical difficulties to overcome with projecting from an
SGI system. The most notable was the fact that as true SGI projector cost in the vicinity of +$15,000, outside of my budget. The remedy was to use an NEC MultiSync MT800 at a lower "video" (640x480) resolution and kick the image through a Sirus Video board system attached to an ONYX. This is of course analogous to swatting a fly with a sludge hammer.

Some comments from test designers have been that the illusion is very convincing as long as one does not move one's head around. The subtle shifts of observation points are enough to make the subject aware of the fact that they are still working with a monitor and not a truly immersive virtual space. The stylus was also a cause for immersion breaking. The act of looking down and/or manipulating the media of the screen projection through a removed connection also seemed to create a disharmony with the phenomenological tactility that the space was projecting.

An interesting side effect of this methodology was quite unexpected. It seemed that the test designers were hesitant to engage in the media as they had with the previous physical medias. I attribute this to a little bit of stage fright. The simple act of placing your exploratory design soul on screen 10-20 feet tall is a major inhibiting factor. I suppose the thought of having someone looking over their shoulder in the dark projection hall as they
explored their design rules and strategies was to
great for some. So as a result, in most cases the
designers produced far less "complete" work
given the same amount of time.
The DIGIPANOMETER

As a related aside note to this thesis, another attempt at creating a physical manipulator for virtual environments was also tried and was quite successful.

While deliberating upon display methodologies for a research grant, The unbuilt, for MOCA, Daniel Brick, Dr. Takehiko Nagakura and myself, were discussing the infusion of a new technology called QuickTime VR with the phenomenological characteristics of how does one experience space. We had become very excited about the idea that a person should interact with virtual space not only with his eyes but also with his entire body. Before I continue, it may be helpful for me to explain what QuickTime VR is.

QuickTime VR (QTVR) is a tool for creating and viewing photo-realistic environments (panoramas) and real-world objects. Users interact with QuickTime VR content with a complete 360 degree perspective and control their viewpoint through the mouse, keyboard, trackpad, or trackball. Panoramas and objects to be viewed are 'stitched' together from digitized photographs or 3D renderings to create a realistic visual perspective.

One of the results of our conversation was the inception of a device that would project the
QTVR on a movable screen. This screen would move in accordance with the movement of the viewer. As the viewer slid to the right, the screen would slide to the left, thereby always being diametrically opposed from the viewer and allowing the viewer to peer into the QTVR space as if it were real space. The advantage of this methodology is obvious, by engaging the viewer in actual body movement in order to see more of the space, the viewer begins to gain a sense of the dynamics of the actual space.

Dr. Takehiko Nagakura set up a team to construct a prototype of the machine. Due to some unfortunate time constraints, I was unable to continue with this project, even though it was shaping up to be my thesis at the time. The project ultimately was realized by Dr. Takehiko Nagakura with Haldane Liew and Hyun-Joon Yoo. They deserve the credit and honor for achieving such an amazing technical feat and for their technical mastery and craftsmanship.
Virtual REALity

The third stage is the introduction of a fully immersive environment. PHEMENOarch was written primarily for use with HMD (Head Mounted Display) and Virtual Reality.

The key commands are easily transferred to voice commands that can be recognized with even primitive voice recognition software (often times built into the VR system). This simple shift in paradigms allows for the designer to be unencumbered with the minimal menu system of PHEMENOarch.

The Navigation and view perspectives are routed though the electromagnetic or sonic resonance of the HMD. The VR helmet allows for tracking of subtle head movements and hand’s off navigation of the virtual site. It also allows for bioptic projections that give the designer the impression of 3 dimensions.

The selections of primitives can now be relayed in true space with direct spatial input from the designer. 6DOF mice and /or cyber-gloves provide 3 dimensional coordinates to PHEMENOarch. When 3 dimensional freedom is compared with the other physical methodologies used in this chapter [all of which were constrained to 2 dimensional planes], the phenomenon...
nological feedback is phenomenal. One is adding a whole other dimension (if you will pardon the pun). At this point the designer is allowed to design with his/her whole body, a full figural gesture. One does not need to rely upon Le Corbusier’s modular man, nor the golden rectangle or any of the other ingenious architectural rules that experienced architects have come up with over the centuries (that is unless you want to), because from this vantage point you are able to "feel" the spatial implications of composition and form. This is the strength of PHEMENOarch.

The application could be linked with a haptic device, one would then be able to add other constraints to the modeler. For instance, if one were pushing a cube through another cube, one would feel the resistance of the original cube resisting this intrusion. The ramifications of this simple act is that the designer could begin to understand the whole object and its interaction with other existing objects without having to be at a vantage point to "see" the interactions. Of course there could be a myriad of other links that could be manifest in such a device, like feeling the forces which a certain structural configuration could resist, the simulation of air flow patterns through a space which could be felt, the actual molding of forms as though they were clay, reinventing the "fingerspitzgefühl” tactility of true physical building.
All these factors aid in the intent of creating an application (PHEMENOarch) which would stimulate right brain infopro. The designer is allowed to concentrate upon the task of creation and is not pulled out of the immersion by having to switch cognitive paradigms to perform different tasks.

PHEMENOarch is ideally suited for the VR environment for the previous reasons and because it is written in the native language of Open Inventor, of which VRML (virtual reality modeling language) is based. However due to management and proprietary restraints, I was unable to fully test out the application with this equipment. My preliminary research was quite encouraging and is an avenue that I believe should be explored more in-depth.
The Test Experiments

The thesis became divided halfway through as to which research methodology should be employed. There are several different research strategies that could have been employed. Experiment, survey, archival analysis, history and Case study are the most commonly used research methodologies. In the beginning, the experiment methodology seemed to produce the greatest results. However, as the testing progressed, it became apparent that the research methodology which this thesis should employ was one of Case Study.

reflections on Abstractions

While the experimental evidence would require more empirical testing to validate my findings, there are some interesting preliminary implications.
Methodology; Experiment (the Blue phase)

Early in the testing phase of this thesis, the tests were geared for an experiment research strategy. The test subjects were to be given limited versions of PHEMENOarch. The "blue" model is a perfect example of this stratagem. In the preliminary versions of PHEMENOarch, the primitive forms were all a shade of blue due to the added depth that they seemed to convey on the SGI monitors.

In this experiment, the intent was to discover whether or not an orthogonal constrain system would facilitate orthogonal spaces. The hypothesis was that the spaces would have orthogonal elements, by the very nature of the constrained program but the spaces that were created could be abstracted to reveal a "plastic" spatial form. This form could then be reinterpreted and demonstrated as a series of abstractions that could be in-turn brought back to traditional methods of architectural visualization. I.E. one could interpret a new set of basic cognitive rule systems and forms like the "golden section".

The test subject was asked to design two different spaces. One was to be a public space the other was to be private in nature.

Out of the four test subjects, three of them began with the private space first. The construction of the
floor plane and ceiling plane seemed to be paramount to their subsequent relationship building.

The walls were manipulated with an emphasis on the geometry at the apex or junction points of horizontal and vertical members. If a window were to be placed in the wall, there was great attention placed upon the articulation of the surrounding pieces. In a few cases, the concept of built ruins (aka Louis Kahn) seemed to emerge.

The significant articulation of depth of the width of the "private" space walls seemed to be a derivative of the belief that this environment should seem, as one tester put it, "safe and protected". This may seem like an intuitive notion of the phenomenology of space, but it is exciting to see it articulated in such a precise manner within the test environment.

The careful elaboration of the transitional gate or passage way between the public and private space seemed to solidify architectural tactility. In almost all the cases (3 of 4) the passage between two spaces were crafted such that one could not see from the public space into the private. This was usually accomplished by making the passageway a corner of the public space, and then by making the passageway relatively deep. This made it difficult for an individual that is residing within the public space to gain a
view access point into the private space unless he/she were entering the room. In one instance, a designer/tester created a "window" from the private space to the public space that, due to its elevation, allowed for a voyeuristic one way view over the public area.

Another interesting phenomenon which occurred was completely unexpected. The interior of the private space was not rectangular. The wall of the interior private space was angled back from the wall at (depending on the subject) 12 -18 degrees. This effect may be the result of some error in the manner that the computer displays the virtual environment back to the designer. The other possibility is that there is some cognitive need to want to tweak the perspective of the private space such that it displays some as yet undiscovered characteristic. Both of these hypotheses will need further investigation in order to unveil the true reason behind this interesting construction.

In listening to the designers discuss what they were doing and why, it was noteworthy that their conversations seemed to revolve about the subtle play of view obscuring and demonstrating, and what the space should "feel" like.
The public spaces provided some what less definable results.

The definition of "public space" seems to be too encompassing for there to be enough coordination between the relatively small sampling that I was able to achieve. There were some startling results though.

The relationship between all the public and private spaces seemed to be one of a constant ratio. The private space, regardless of its placement in the overall context, was always between 1/3rd and 1/2 of the size of the public space. This seems to validate the architectural "golden section". This ratio is an even more amazing since the "true" scale of one subject's project varied greatly with the scale of the others. When a designer first enters the virtual space, the tester is confronted with a singular cube. Some testers took this cube as a unit scale and they would "back away" from the cube until it was of an "appropriate scale" to be used. Other testers, immediately scaled the block to an "appropriate scale" from their current "home" viewpoint. This second group of testers created very "small" models in relationship to the first group, because they were constantly scaling the primitives down to their perceived level. One tester, even complained that she wished that "the cubes would come in at a smaller scale".
There were some expected yields from the public space experiments.

In some of my earlier works and investigations (before thesis), some of the preliminary discussions of designing within a perspectival space led me to investigate certain aspects of spatial relations. Many of the same concerns manifested themselves here. For example, if one is designing a colonnade in a singular perspectival space. The typical perceptual mistake is to believe that the columns are equally spaced and are of equivalent sizes, when in reality the sizes are usually increasing (dramatically) and the spacing is also increasing (dramatically) as the columns move farther away from the creation camera/designer viewpoint. This is an inherent danger with a one point perspective, as it is very difficult to gauge depth accurately with this methodology. This perceptual mistake is also possible in PHEMENOarch, if the designer does not move his/her vantage point in order to reevaluate the space. In some instances, this perceptual vagueness can be utilized by the designer to great effect. One designer created a rhythm by making the blocks sit on each other all the same size, but as he stacked the blocks closer to the ceiling of his enclosure, he specifically made the blocks smaller and thinner such that the observer from the ground would feel that the space was far greater than it actually was. When asked about
this later, he replied that he wanted to see if "anyone noticed". Another designer wanted to create a set of stairs, as he finished creating them from one vantage point, he moved to another corner of his construction. He was amazed to note that the stairs were "not even close" to where he had believed that they were. He became excited by the development and began to experiment with crating forms that appeared to be one thing from one side and then changed dramatically when viewed from another (unfortunately this model was lost due to an unexpected core dump).
Methodology: Case Study

I, as researcher, was looking for certain characteristics which I had originally postulated, a hypothesis mentality. The difficulty in bringing these hypotheses to fruition became apparent halfway through the testing phase. The preconceived notions that I, as researcher, designer and developer had come up with seemed to impede the exploration of what was this program and physical interface affording the designer. The decision of following an experimental methodology seemed to produce decent results, however I was not convinced that they were demonstrating a "true" usage of the provided medias. It was also evident that the inherent control over behavioral events that was required with the experimental method, was in a way impeding the larger picture of how, why and what do designers create in Virtual free space.

The range of testers was broadened to include several varying backgrounds. PHEMENOarch was tested with different end-users, they ranged from Investment Bankers, to Engineers, from Students of Architecture, to accredited Architects, from programmers, to artists.

It was at this point that the test/designers were given a 10 minute demonstration of the capabilities of the program and then they were told to
create anything that they wished. The intent was to see if the "immersive qualities" of the software would entice the testers into using it as an architectural space former, or would they utilize it as an object creator. The results were not as expected. A major discrepancy emerged that is difficult to understand.

The testers that had little or no CAD (computer aided design) experience, seemed to pick up the program basics very easily, they had a little difficulty adjusting to the differing mindset that was required, but after 15-20 minutes of working with PHEMENOarch, they were able to create some very architectural readable pieces and forms. Many of these subjects did not even have an architectural background and yet even with a very limited design vocabulary, they were navigating through their sites to "envision" how one would approach their "buildings". This was the key, in some manner, they were navigating through the built forms to evaluate them rather than navigating around the built forms for evaluation.

... this is how you approach the entrance.... like that,... I wanted this to be the grand "facade",... god help me, I feel like an archy...

subject "A", engineer
We live in a 3 dimensional world, and yet we have great difficulties thinking in true 3 dimensions. This application forces you to think in 3 dimensions. ...It is definitely a mind shift, if you will.... It takes some getting use too, but after 5 min. I was really having fun with it! You should really think about using this to teach architects about space and relationships..... It really opened up my eyes as to how to visualize distances, I never thought about it that way......

subject "T", an investment banker

The test designers that had "some to moderate" experience with CAD systems and design, seemed to have fundamental difficulties with manipulating the built space. In most cases, this group consisted of experienced architects that were used too envisioning spatial relationships.
Here is one sample case studies notes;

Subject "J", architect and architectural graduate student

- Subjects first inclination is to create "planes" these thin forms are being created at a distance such that fine control can not be achieved, the program crashes several times as forms are force to infinity (infinitely large, or infinitely small, I'll have to fix that bug)

- The manner of spatial manipulation is not one of moving though the site but rather by maintaining one perspective and "finishing the form", "J" then moves to see if the forms are correct.

- "J"s" movement characteristics are that of more vertical moves than transversal movement, approximately 4 to 1

- "J" uses the auxiliary horizontal and tilt movement manipulators around the edge of PHEMENOarch rather than the main ones provided by the implementation of the SoWalkViewer.

aside note: "J" wanted to "walk" through the space after it was completed.

- Requested plan and section views as, subject "was used to it" as an architect.

- Wanted to "jump" from viewpoint to viewpoint (much like looking at different elevations)

- Subject elevates the viewpoint position, and tilts down, there by recreating a plan view.

- Crashes software again (more infinity difficul-
ties), I recovered the work.

When asked about problems and comments:

subject believed that the navigation tools needed work.

"architects are used to thinking in space and drawing in plan, with this... you have to think in plan and draw in space."

subject felt that it would be "easier" if immersed in VR, "would help a lot"

This is an interesting phenomenon that I would have liked to pursue further given more time.

The "experienced" CAD group had very little difficulties with adapting to the new environment offered by PHEMENOarch. It was a shift in viewpoint paradigms, and still required a few minutes of acclimation. Once this stage was passed, they were able to use the application to its fullest extent. The were the true space builders. 3 of the test designers requested copies of the application to continue with their spatial explorations.
Some of their comments;

*Boy is this easier than AutoCAD..... I like it!*  
-subject "K", practicing architect, avid AutoCAD user

... the subtitle in the manner with which one manipulates the objects is very much the quintessential heart of the matter. There are no dialogue boxes,... nothing pops out at you.... it is tactile and it beckons exploration.... I believe that it is a piece is about experience and not computation.  
-subject "S", architect and Architecture graduate student/teacher

Many of the same discussions and abstractions appeared with this research methodology and it's accompanying group of designers, as did with those from the earlier "experimental Blue group". The most relevant are as follows:

---

**FIGURE 08.13**  
Description:  
Subject "K"  
AutoCAD Architect

**FIGURE 08.14**  
Description:  
Subject "S"  
CAD experienced Architect/Grad.
Edge Boundaries

In the inception of boundary conditions, i.e. walls and enclosures, the tendency or inclination is to break up edge conditions. This may be due to some of our architectural training, the interest of exploring the dislocating of boundaries. Given this reasoning it is not unexpected for this condition to occur but what is unexpected is the regularity with which this condition occurs. The basic wall geometries can be decomposed into a simple boundary spine, this spine becomes exaggerated when a perpendicular wall intersect the original wall.
The spine is apparently based upon eye height/view point. This can be approximated by creating a cylinder that runs along the boundary, with the centroid of the cylinder at eye height.

---

**FIGURE 08.17**

Description: Sectional Diagrams of spatial grammar
Light

The addition of other factors has certain effects on the creation of the space. When a uni-directional light source was added to the design environment, the designers had a tendency to alter their design moves. If they were facing a wall that was washed with light, their reactions were for the most part, to create subtle juxtapositions of forms in order to create the sought dynamic. This was drastically different from the opposing situation. The same designers, when they were facing forms that were backlit (not in direct light), the formal moves became more aggressive. The relative juxtapositions were on a larger scale. This can be explained I believe by the phenomenological reaction to the limited light levels, in order for the designers internal infopro model to be realized the forms had to take on different proportions and scale to develop the same visceral reaction as the omni-lit model environment. In this application, the modeling of light was very basic, but I believe that it is obvious that the necessity of modeling light in a more accurate fashion has been demonstrated and should be investigated further.
Transitional Boundaries

The boundary between formal spaces were exaggerated by almost every designer. The entrance usually became an event unto itself. The transitional spaces were often created with a depth relationship that had a proportional systems.

A specific condition that produces measurable results is an entrance on the corner of an enclosed space. The rectangularity of the space is often forced into a trapezoidal form. The reoccurring angle of this final side seems to be between 12 and 18 degrees. This time the designers were asked to draw their intentions, and a quarter of them did not realize that they had done it.
Trouble with Textures

When the test designers were on a machine that had enough power to deal with the texture redraws, the amount of primitives and/or the completeness of the designs actually was decreased compared with the designers that were not given textures. The designers seemed to "work" the visual representations of materiality to such an extent that it seemed to the observer that they had lost the direction of the overall space. This seems to leads to a verification of the concept that design must begin in stages of abstraction, with verisimilitude being the last abstractive barrier/stage. It may also be a question of the test designers having never been exposed to this type of virtual media creation and that they simply want to experiment with it. More testing has to be done to see conclusively whether this tendency will continue with more media mature/savvy subjects.
Conclusions:

I believe that this thesis is a small tentative step into a new direction for architectural design methodology. This computational medium has potential and must be explored further. Its impact may be as great as the impact that perspectival drawing had on Renaissance architecture.

The application has demonstrated great promise as a tool for space manipulation and space building. I would have liked to see more experimentation done in the field of Virtual Reality, to truly push the phenomenological concerns to their ultimate limits. There are a few small functions that I would have liked to add to the program, the most notable of which is Booleans. With a complete tool set, one could begin to push all the avenues of architectural spatial thought. One could think in terms of subtractive formal gestures as well as the additive ones that are currently provided. In order to make PHEMENOarch a true design tool for architectural creation and not just an experimental tool of cognitive spatial exploration, a number functions must be added. Probably the most notable of which are copy and paste tools, and more importantly an "undo". The addition of traditional abstractions like plans and sections would also have to be carefully integrated. The color palate in PHEMENOarch was for me the most problematic and least successful of the tools. The ultimate intent was to have the architect never be presented a "menu". Though I ran out of time, the color palette manipulator was to be a push and pull manipulator that would

**FIGURE 09.01**

Description: Examples of Subtractive Boolean functions

09. 86
slide you through the color chart, in 3 dimensions. I know that this can be accomplished with a simple 3vect.

The manipulation and understanding of scale is another issue that needs to be addressed. I was working on a visual displacement ruler that would demonstrate scale distances between two chosen objects. The necessity for spatial scale was demonstrated by the designers that made their models at a super small scale. Perhaps the addition of a texture mapped “Modular Man” would be all that was needed.

The method of immersive form manipulation has proven to be a valid one. The relative ease that individuals with little to no architectural or artistic background were able to grasp and embrace the complex concepts of 3 dimensional modeling is astounding. A result that was totally unexpected. There is still some concern and confusion over why trained architects and average CAD modelers had such difficulty with the concept of the software. It is an avenue that will require more thought. It seems strange that spatial literate designers can not change their mind set to visualize space in a different manner.

The area of architectural design and computation is just developing. The research into shape grammars and parametric design is very exciting and it holds some very economically lucrative and theoretically challenging underpinnings. The research into these areas are necessary. I would believe that the next level of this program would have parametric constraints built into it. For example, if one moved the walls apart further than would be structurally feasible, the program would not allow the move to continue. The architect could then specify a stiffener or some other structural rein-

\[ \text{FIGURE 09.02} \]
Description:
Le Corbusier
Modular Man

09. 87
forcement and be able to continue. For a "real world" architect, it would also be interesting if the program returned a cost analysis while the changes were being made, such that the architect could be able to balance the design to the given budget. With the option of turning it off.

Another avenue for exploration with these tools, is that of architect training. This tool as it stands now, affords a mentor or professor a manner with which to demonstrate spatial ideas and forms, as well as forcing the student user to visualize forms in 3 dimensional space. It would also be exciting to study how children might react to this empowering environment building.

The architect is the creator of ideas, the challenger of normative mass mentalities and thinking. The architect is the mirror of society. The search for a "perfect" architecture is fruitless, because architecture is about humanity and must there for change and grow with us. It must not be complacent. We must continue to develop tools that afford the architect a new vista to see and explore.
This is the PHEMENOarch program.

It is intended to be used to build architectural spaces FROM the INSIDE, i.e. as an inhabitant of the space.

```
(define viewer (new-SoXtWalkViewer))
(-> viewer 'show)
(-> viewer 'setViewing 0)

(define scene-root (new-SoSeparator))
(-> scene-root 'ref)

(define material-editor (new-SoXTMaterialEditor))
(-> material-editor 'setTitle "Material Editor")
(-> material-editor 'show)
```

the sun light is approximated with this function
(define side-light-light (new-sopointlight))
(-> scene-root 'addchild side-light-light)
(-> (-> side-light-light 'on) 'setvalue 1)
(-> (-> side-light-light 'location) 'setvalue 20 20 101)

(define (selection-cb user-data path)
  (let ((object (-> path 'getTail))
      (sep (SoSeparator-cast (-> path 'getNodeFromTail 1))))
    (display "Selection Callback: Object ")
    (display object)
    (display " was selected.")
    (newline)
    (display "Selection Callback: Object Type is ")
    (display (-> (-> (-> object 'getTypelD) 'getName) 'getString))
    (newline)
    (if (= 1 (-> object 'isOfType (SoCube::getClassTypeld)))
      (let ((the-sphere (SoCube-cast object)))
        (display "Selection Callback: Picked a cube.")
        (newline)
        ;; additional operations with sphere
      ))
    (if (= 1 (-> object 'isOfType (SoTransform::getClassTypeld)))
      (let ((the-sphere (SoTransform-cast object)))
        (display "Selection Callback: Picked a transform.")
      ))
(define new-manip (new-sotabboxmanip))
(-> new-manip 'replaceNode path)

(display "Selection Callback: Object Type Of Sep's Child 1 is ")
(display (-> (-> (-> (-> (-> (-> (-> (-> object 'getTypelD) 'getName) 'getString)))

(define (deselection-cb user-data path)
  (let ((object (-> path 'getTail)))
    (sep (SoSeparator-cast (-> path 'getNodeFromTail 1)))
    (display "Deselection Callback: Object ")
    (display object)
    (display " was deselected.")
    (newline)

    (display "Deselection Callback: Object Type is ")
    (display (-> (-> (-> object 'getTypelD) 'getName) 'getString)))
    (newline)
(if (= 1 (object 'isOfType (SoSphere::getClassTypeId)))
  (let ((the-sphere (SoSphere-cast object))
        (manip (SoTransformManip-cast (separation 'getChild 0)))
        (display "Deselect Callback: Picked a sphere.")
        (newline))
    (define new-trans (new-sotransform))
    (separation 'replaceChild 0 new-trans)
    (separation (separation 'translation) 'setValue (manip 'translation))
    (separation (separation 'rotation) 'setValue (manip 'rotation))
    (separation (separation 'scalefactor) 'setValue (manip 'scalefactor))
    (separation (separation 'scaleorientation) 'setValue (manip 'scaleorientation))
    (separation (separation 'center) 'setValue (manip 'center))
)

(if (= 1 (object 'isOfType (SoTransformManip::getClassTypeId)))
  (let ((manip (SoTransformManip-cast object)))
    (define new-trans (new-sotransform))
    (manip 'replaceManip path new-trans)
  )
)

(define (pick-filter-cb user-data picked-point)
  (let ((path (picked-point 'getPath))
        (node (path 'getTail))
        (if (= 0 (node 'isOfType [sotransform::getclasstypeid])))
          (let ([new-path (copy path 0 (- (path 'getLength) 1))]
                (sep (SoSeparator-cast path 'getNodeFromTail 1)))
            (display "Pick Filter: Non-Transform at the end of the path")
            (newline)))
)
(define root (new-SoSelection))
(-> scene-root 'addchild root)
(-> root 'policy) 'setValue SoSelection::SHIFT)
(-> root 'addSelectionCallback
 (get-scheme-selection-path-cb)
 (void-cast (callback-info selection-cb)))
(-> root 'addDeselectionCallback
 (get-scheme-selection-path-cb)
 (void-cast (callback-info deselection-cb)))
(-> root 'setPickFilterCallback
 (get-scheme-selection-pick-cb)
 (void-cast (callback-info pick-filter-cb)))

;; 00000000000000000000
;; 00000000000000000000
;; 00000000000000000000
;; KEYBOARD EVENT STUFF
;; 00000000000000000000
;; 00000000000000000000
;; 00000000000000000000
;; 00000000000000000000

(define (keypress-cb user-data event-callback)
  (let ([event (send event-callback 'getEvent)])
    (cond ([= 1 (SO_KEY_PRESS_EVENT event c)]
          (display user-data)
          (create-new-eye root)
          )))

cline 93
(send event-callback 'setHandled))

((= 1 (SO_KEY_PRESS_EVENT event d))
 (display user-data)
 (create-new-eye2 root)
 (send event-callback 'setHandled))

((= 1 (SO_KEY_PRESS_EVENT event g))
 (display user-data)
 (create-new-eye3 root)
 (send event-callback 'setHandled))

((= 1 (SO_KEY_PRESS_EVENT event s))
 (display user-data)
 (print-scene "newfile.iv" scene-root))

((= 1 (SO_KEY_PRESS_EVENT event b))
 (display user-data)
 (newline)
 (display "Here's where I autocenter the centerball dragger.")
 (newline)

;; 'root' is the SoSelection Node
 (display "Number of selected paths is ")
 (display (-> root 'getNumSelected))
 (newline)

(if (< 0 (-> root 'getNumSelected))
 (let ((path (-> root 'getPath 0))
       (let ((object (SoTransformManip-cast (-> path 'getTail))))
         (display "Autocenter.")
         (newline)
         (cond
          ((= 1 (-> object 'isOfType (SoCenterBallManip::getClassTypeld)))
           (display "CenterBallManip - Center it.")
           (newline)
           (display "This doesn't work. Disabled it.")
           (newline)

  94
(let ((path (-> root 'getPath 0))
         (object (-> path 'getTail)))
    (display "DO IT!"
    (newline)
    (display "Switch the Manip: Object Type is ")
    (display (-> (-> (object 'getTypelD) 'getName) 'getString)
    (newline)

    (cond
      (let ((path (-> root 'getPath 0))
                (object (-> path 'getTail)))
          (display "DO IT!"
          (newline)
          (display "Switch the Manip: Object Type is ")
          (display (-> (-> (object 'getTypelD) 'getName) 'getString)
          (newline)

          (let ((path (-> root 'getPath 0))
                    (object (-> path 'getTail)))
                  (define new-manip (new-sotransformboxmanip))
                  (-> new-manip 'replacenode path))
    (newline)

    (cond
      (let ((path (-> root 'getPath 0))
                (object (-> path 'getTail)))
          (define new-manip (new-socenterballmanip))
          (-> new-manip 'replacenode path))
    (newline)

    (cond
      (let ((path (-> root 'getPath 0))
                (object (-> path 'getTail)))
          (define new-manip (new-sotransformboxmanip))
          (-> new-manip 'replacenode path))
    (newline)

    (cond
      (let ((path (-> root 'getPath 0))
                (object (-> path 'getTail)))
          (define new-manip (new-socenterballmanip))
          (-> new-manip 'replacenode path)))
((= 1 (-> object 'isOfType
(SoCenterBallManip::getClassTypeld)))
  (display "SoJackManip")
  (newline)
  (define new-manip (new-SoJackManip))
  (-> new-manip 'replacenode path))

((= 1 (-> object 'isOfType
(SoJackManip::getClassTypeld)))
  (display "CenterBallManip")
  (newline)
  (define new-manip (new-sohandleboxmanip))
  (-> new-manip 'replacenode path))

((= 1 (-> object 'isOfType
(SoHandleBoxManip::getClassTypeld)))
  (display "HandleBoxManip")
  (newline)
  (define new-manip (new-sotabboxmanip))
  (-> new-manip 'replacenode path)))
)

(begine
  (display "Nothing is selected. Do nothing.")
  (newline)
)

(send event-callback 'setHandled))

)
)
)

(define ev-cb (new-SoEventCallback))
(-> ev-cb 'addEventCallback
  (SoKeyboardEvent::getClassTypeld)
  (get-scheme-event-callback-cb)
  (void-cast (callback-info keypress-cb)))
(define (create-new-eye geometry-root)
  (define eye (new-soseparator))
  (-> geometry-root 'addchild eye)
  (define eye-trans (new-sotransform))
  (-> eye 'addchild eye-trans)
  (define eye-mat (new-somaterial))
  (-> eye 'addchild eye-mat)
  (-> (-> eye-mat 'diffusecolor) 'setvalue 1 1 1)

  (define texture (new-SoTexture2))
  (-> (-> texture 'filename) 'setvalue "img.rgb")
  ;; filename.rgb should be replaced with the appropriate filename
  ;; I'm not sure but OI may only understand SGI rgb format images.
  (-> eye 'addchild texture)

  (define eye-sphere (new-Socube))
  (-> eye 'addchild eye-sphere)

  (if (< 0 (-> root 'getnumselected))
    (let ([(path (-> root 'getpath 0))])
      (let ([(object (SoTransformManip-cast (-> path 'getTail)))]
        [display "Create-New-Eye: Setting the transformation."]
        [newline])
  )
(define vec (-> (object 'translation) 'getValue))
(define vec1 (new-SbVec3f 1 1 1))
(-> (eye-trans 'translation) 'setValue
    (-> vec 'operator+ vec1)))))

eye
)

(define (create-new-eye geometry-root)
    (define eye2 (new-soseparator))
    (-> geometry-root 'addchild eye2)
    (define eye-trans2 (new-sotransform))
    (-> eye2 'addchild eye-trans2)
    (define eye-mat2 (new-somaterial))
    (-> eye2 'addchild eye-mat2)
    (-> (eye-mat2 'diffusecolor) 'setValue 0.5 0.5 0.5)
    (define texture2 (new-SoTexture2))
    (-> (texture2 'filename) 'setValue "Conc.rgb")
    ;; *.rgb should be replaced with the appropriate filename
    ;; I'm not sure but OI may only understand SGI rgb format images.
    (-> eye2 'addchild texture2)
    (define eye-sphere2 (new-Socube))
    (-> eye2 'addchild eye-sphere2)
    (if (< 0 (root 'getnumselected))
        (let ((path (root 'getPath 0)))
            (let ((object (SoTransformManip-cast (-> path 'getTail))))
                (display "Create-New-Eye: Setting the transformation.")
                (newline)
                (define vec (-> (object 'translation) 'getValue))
                (define vec1 (new-SbVec3f 0 2.1 0))
                (-> (eye-trans2 'translation) 'setValue
                    (-> vec 'operator+ vec1)))))))

cl. 98
(define (create-new-eye3 geometry-root)
  (define eye3 (new-soseparator))
  (-> geometry-root 'addchild eye3)
  
  (define eye-trans3 (new-sotransform))
  (-> eye3 'addchild eye-trans3)
  
  (define eye-mat3 (new-somaterial))
  (-> eye3 'addchild eye-mat3)
  (-> (eye-mat3 'diffusecolor) 'setvalue 0.2 0)
  (-> (eye-mat3 'transparency) 'setvalue 0.6)
  (-> (eye-mat3 'specularcolor) 'setvalue 0.1 0.2)
  (-> (eye-mat3 'shininess) 'setvalue 0.1)
  
  (define texture3 (new-SoTexture2))
  (-> (texture3 'filename) 'setvalue "img.rgb")
  ;; filename.rgb should be replaced with the appropriate filename
  ;; I'm not sure but OI may only understand SGI rgb format images.
  (-> eye3 'addchild texture3)
  
  (define eye-sphere3 (new-Socube))
  (-> eye3 'addchild eye-sphere3)
  
  (if (< 0 (root 'getnumselected))
    (let ([path (root 'getpath 0)])
      (let ((object (SoTransformManip-cast (path 'getTail))))
        (display "Create-New-Eye: Setting the transformation.")
        (newline)
        (define vec (object 'translation) 'getvalue)
        (define vec1 (new-SbVec3f 1 1 1))
        (-> eye-trans3 'translation 'setvalue
            (vec 'operator+ vec vec1)))
    eye3)
  )
(create-new-eye root)

;; 00000000000000000
;; 00000000000000000
;; PROCEDURE TO WRITE SCENE FILE
;; 00000000000000000
;; 00000000000000000

(define (print-scene filename scene-root)
  (define write (new-SoWriteAction))
  (-> (-> write 'getOutput) 'openFile filename)
  (-> (-> write 'getOutput) 'setBinary 0)
  (-> write 'apply scene-root)
  (-> (-> write 'getOutput) 'closeFile)
)

(-> viewer 'setscenegraph scene-root)

(gc)

;; 00000000000000000
;; 00000000000000000
;; 00000000000000000
;; To open an .iv file named "filein.iv" un-comment this!
;; 00000000000000000
;; 00000000000000000
;; (define root-in (read-from-inventor-file "filein.iv"))
;; (-> root 'addchild root-in)

Cl 1.100
TO SET UP THE ENVIRONMENT FOR PHEMENOnarch

The following set up is taken from:

http://vismod.www.media.mit.edu/courses/cgw97/

This is definatly a web page to investigate. It taught me a lot, and I can not recommend it highly enough.

The development environment we will use on Athena is a version of Scheme (SCM) which has an Open Inventor binding. This allows you to write Scheme programs which use the Open Inventor 3D graphics toolkit.

Note that currently Open Inventor is only available on Athena SGIs. Therefore, you will have to find an Indy in order to be able to work with Inventor.

To get started, add the following line to the end of your ~/.environment file:

```
add iap-cgw  # Current location of IvySCM
```

Add the following lines to the end of your ~/.cshrc.mine file:

```
if ($hosttype == sgi) then
    limit coredumpsize 0  # Don't let anything dump core
    setenv SCHEME_INIT_PATH /afs/athena.mit.edu/course/other/iap
cgw/lib/scm/
    setenv SCHEME_LIBRARY_PATH /afs/athena.mit.edu/course/other/iap-
cgw/lib/slib/
```
Note: you will need to log out and log back in for the above changes to take effect.

Although you can run the Scheme interpreter, ivyscm, directly from an Athena prompt, it is strongly recommend that you run it from within emacs. To set up ivyscm within emacs, add the following lines to the end of your ~/.emacs file (or create one if you don't have one):

```
(setq load-path
  (cons "/afs/athena.mit.edu/course/other/iap-cgw/elisp" load-path))

; Use scheme-mode for files with .scm suffix
(setq auto-mode-alist
  (append '((".*\.scm" . scheme-mode)
           )
           auto-mode-alist))

; Autoload run-scheme from file cmuscheme.el
(setq scheme-mode-hook
  '((lambda () (autoload 'run-scheme "cmuscheme"
                         "Run an inferior Scheme" t))))

(autoload 'run-scheme "cmuscheme"
"Run an inferior Scheme process." t)

(setq scheme-program-name "ivyscm")
```

This will set up emacs with IVY and Scheme.
TO USE PHEMENOarch

I would like to add just a few step by step commands that will be helpful for one to evaluate and run PHEMENOarch, especially if one is new or unfamiliar with the computation environment.

Once you have the emacs window open, type with in the buffer:

\texttt{M-x run-scheme}

\texttt{(M stands for meta, this is the "Alt" key)}

While you can work by typing directly into the interpreter's buffer, it is much more convenient to work in a separate buffer. To create two windows, one for the evaluating scheme buffer and one for the evaluated PHEMENOarch application, just type:

\texttt{<Ctrl> x 2}

Move your cursor to the upper window and click once to activate the window. Then load the file PHEMENOarch.scm by typing:

\texttt{C-c C-l scheme-load-file, then PHEMENOarch.scm}

Or One can also go to the menu bar and "open file" like a traditional program.

Evaluate the file by going to "home" (the top) and type:
<Ctrl>  <space bar>
(mark set)

Then go to the end and type:

<Ctrl> c <Ctrl> r
(evaluate region)

If you make a mistake type

<Ctrl> g

to stop any errant processes.

This should get you up and running with the application. I would suggest taking some Emacs courses from Athena if you want to get more in depth.

You should see the Walkviewer appear with a cube in the middle.

The viewer will have certain buttons down it left side. The pointer refers to a selection pointer, with this selected, you can move and alter objects.

useful commands with in PHEMENOarch.scm in this mode are:

a = to toggle methods of manipulating the primitives

c = to add a "sandstone" cube, please note that if a cube is selected already it will place the new cube in relation to the selected one. If there is no cube selected, the cube will appear at the origin.

d = to add a different cube (I know "real original"), this one will have a concrete texture. Again if a primitive is already selected, it will place the
new one above the selected one.

g = adds a glass cube.

<shift> = for multiple selections.

s = Save, a file called "newfile.iv" will be created in the directory that
emacs was launched from.

Selecting the "hand" icon will allow one to move and pan through the virtu-
al space. The movements have been constrained to "walking" in a plane,
one may "pan" up but this is a conscious decision. There is no "flying", I
hope.
.iv FILES AND ALTERATIONS

This section is necessary to get PHEMENOarch to read in its own .iv files and .iv files from other programs like Form•z, architrinon, Alias, Softimage, ProEngineer, Catia, and autocad. It also will explain how to get other programs (like ivview) to read the .iv files produced with PHEMENOarch.

PHEMENOarch writes out .iv files in a very particular manner. This is mainly due to the way the SoDB and Scene graphs had to be written to allow for the manipulation of primitives in free space.

Here is an example of the Open Inventor format .iv file that is generated directly from the scheme interpreter:

```
#Inventor V2.1 ascii

Separator {
  PointLight {
    on       TRUE
    location 20 20 101
  }
  Selection {
```
policy SHIFT
EventCallback {
}
Separator {
  Transform {
    translation -1.42036 1.72645 2.48729e-08
    rotation 0.892297 -0.429779 0.13819 0.691726
    scaleFactor 0.832692 0.832692 0.832691
    scaleOrientation -0.647332 0.248573 -0.720537 0.663988
  }
  Material {
    diffuseColor 1 1 1
  }
  Texture2 {
    filename "img.rgb"
  }
  Cube {
  }
}
Separator {
  TabBoxManip {
    translation 0.267307 0.8781 8 0
  }
  Material {
    diffuseColor 1 1 1
  }
  Texture2 {
    filename "img.rgb"
  }
  Cube {
  }
}
Separator {
  Transform {
    translation 1.54463 1.15541 1.14504
    rotation 0 0 1 0
    scaleFactor 1.13823 1.10481 1
  }
  Material {
    diffuseColor 0.5 0.5 0.5
  }

  Oc. 1 07
Note that if the file was saved while a primitive was selected. The Manipulator will still be attached to the object. This needs to be altered to be read into other programs, including itself. The easiest way to avoid this, is to make sure that nothing is selected before saving. It can still be altered afterward through a text editor like Jot or Emacs.
In order for PHEMENOarch to be able to read in an .iv file (regardless of the source of the .iv), it must first be altered to look like this:

```
#Inventor V2.1 ascii

Separator {
    Transform {
        translation -1.42036 1.72645 2.48729e-08
        rotation 0.892297 -0.429779 0.13819 0.691726
        scaleFactor 0.832692 0.832692 0.832691
        scaleOrientation 0.691726 -0.647332 0.248573 -0.720537 0.663988
    }
    Material {
        diffuseColor 1 1 1
    }
    Texture2 {
        filename "img.rgb"
    }
}

Cube {
}

Separator {
    Transform {
        translation 0.267307 0.8781 8 0
    }
    Material {
        diffuseColor 1 1 1
    }
    Texture2 {
```
Cube {
    Separator {
        Transform {
            translation 1.54463 1.15541 1.14504
            rotation 0 0 1 0
            scaleFactor 1.13823 1.10481 1
        }
        Material {
            diffuseColor 0.5 0.5 0.5
        }
        Texture2 {
            filename "Conc.rgb"
        }
        Cube {
        }
    }
    Separator {
        Transform {
            translation -0.479954 0.908418 1.57299
            scaleFactor 1.88592 0.27275 1
        }
        Material {
            diffuseColor 0.0 0.2 0
            specularColor 0 1 0.2
            shininess 0.1
            transparency 0.6
        }
    }
}

filename "img.rgb"

Oc. I I O
Texture2 {
    filename "img.rgb"
}
Cube {
}

The SoSeparator of the .iv file must be placed at the top of the Scene graph. The transforms must be placed right underneath the separator. A failure to do so will result in the model not being manipulatable within PHEMENOarch, you will be able to add new geometry and this new geometry will be manipulatable but the imported geometry will NOT.

In order to read the resultant PHEMENOarch .iv files in other programs and viewers, one must make a few changes. For instance in writing for "iview" a standard Open Inventor viewer for the SGI, one must first find out the correct version of Inventor ascii that the viewer is looking for. In this case the current "iview" wanted version 2.0 and not 2.1 which PHEMENOarch was writing.
Here is an example of an .iv file readable by "iview":

```
#Inventor V2.0 ascii

Separator {
    PointLight {
        on    TRUE
        location  20  20  101
    }
    Separator {
        Transform {
            translation  -1.42036  1.72645  2.48729e-08
            rotation    0.892297  -0.429779  0.13819  0.691726
            scaleFactor  0.832692  0.832692  0.832691
            scaleOrientation -0.647332  0.248573 -0.720537  0.663988
        }
        Material {
            diffuseColor 1 1 1
        }
        Texture2 {
            filename   "img.rgb"
        }
        Cube {
        }
    }
    Separator {
        Transform {
            translation  0.267307  0.87818  0
        }
        Material {
            diffuseColor 1 1 1
        }
        Texture2 {
            filename   "img.rgb"
        }
        Cube {
        }
    }
```

Oc.112
Separator {
    Transform {
        translation 1.54463 1.15541 1.14504
        rotation 0 0 1 0
        scaleFactor 1.13823 1.10481 1
    }
    Material {
        diffuseColor 0.5 0.5 0.5
    }
    Texture2 {
        filename "Conc.rgb"
    }
    Cube {
    }
}

Separator {
    Transform {
        translation -0.479954 0.908418 1.57299
        scaleFactor 1.88592 0.27275 1
    }
    Material {
        diffuseColor 0 0.2 0
        specularColor 0 1 0.2
        shininess 0.1
        transparency 0.6
    }
    Texture2 {
        filename "img.rgb"
    }
    Cube {
    }
}

Oc. 113
The selection callback node that PHEMENOarch produces was simply replaced with another separator. I.E.

```
Selection {
    policy SHIFT
    EventCallback {
}
```

Changed to just:

```
Separator {
```

With these nuances in mind, PHEMENOarch will work with any .iv files that can be created with any other program.
BIBLIOGRAPHY


Morehead, James, A Handbook of Perspective Drawing, Florida: H & W. B. Drew Co. 1941


