Reasonable Computing for Architectural Fabrication

by

Rachelle B. Villalon

B.Arch
Woodbury University, 2006

Submitted to the Department of Architecture in partial
fulfillment of the requirements for the degree of

Master of Science in Architecture Studies

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2008

Signature of Author: ____

Rachelle Villalon
Department of Architecture, Computation
May 19, 2008

Certified by: ____

Lawrence Sass
Assistant Professor of Computation and Design
Thesis Supervisor

Accepted by: ____

Julian Beinart
Professor of Architecture
Chair of the Department Committee on Graduate Students

THESIS COMMITTEE

Dr. Lawrence Sass
Assistant Professor of Computation and Design
Thesis Supervisor

Dr. Terry Knight
Professor of Design and Computation
Thesis Reader

Dr. Henry Lieberman
Research Scientist
Software Agents Group, MIT Media Lab
Thesis Reader

Reasonable Computing for Architectural Fabrication

by

Rachelle Villalon

Submitted to the Department of Architecture
on May 26, 2008 in Partial Fulfillment of the
requirements for the Degree of Master of Science in
Architecture Studies

ABSTRACT

The use of digital fabrication tools in the architecture industry serve a particular group of individuals whose familiarity of the tools are by trade skill. Machines lack the understanding of people in its ability to objectify common knowledge. This thesis describes the Adeon system, a software agent for implementation at the initial design stages whereupon the designer conceptualizes about a particular building component while the Adeon system serves to bridge the gap between design and construction knowledge, fusing the two regions to ultimately inform the designer about particular construction, design, and cost analysis constraints and opportunities upon the design proposal. The system adapts to different design situations and delivers pertinent information lending to its capacity to understand design moves and translate it to machine readable instructions for direct digital fabrication.

Thesis Supervisor: Lawrence Sass
Title: Assistant Professor of Computation and Design

Acknowledgments

When I asked Associate Professor Larry Sass to serve as the thesis supervisor for this project, I knew what a good deal I was getting, especially with a similar area of research interest. Larry enthusiastically joined in, offering helpful comments and strategies on overseeing a thesis. His vision, patience, and support for what this research has become, has been very inspiring, often provoking a new rush of enthusiasm to delve deeper into the work, reformulate my thoughts, and ultimately stand back and see the great joy in work worth doing. Larry, I can't thank you enough. You have taught me invaluable techniques for not only structuring my work and evaluating the details, but had more importantly left me with the vision to see the future of architecture and the development of the physical construction process.

After exploring an academic course under Professor Henry Lieberman of the Software Agents Group at the MIT Media Lab, I was given the opportunity to learn about common sense reasoning and A.I. techniques for building remarkable applications using the group's technology. I had decided to follow through with Henry once the course had ended, under the realization that what I was learning from Henry and the Software Agents Group would prove tremendously helpful in the architecture field, as well as to tackle and find solutions to the architectural problems I care about, especially the issues encountered when using machines for architecture and design. Despite his demanding schedule, Henry would always invite me to discuss the progress of my work at a drop-in basis. Henry, I am grateful for your support, guidance, interest, stark vision, and other tremendous contributions to the development of this research. Most importantly, you have taught me to see and approach architectural problems from an array of perspectives. Your support and inspiration continues to give me the creative impulse to keep working and further develop this research for the future.

I would like to thank Professor Terry Knight, my thesis reader and academic advisor, for her helpful comments during the process of formulating my thesis, as well as for her exceptional guidance throughout my experience at MIT. Your encouragement to pursue this research has proven invaluable.

I am indebted to Associate Professor, Martin Bechthold of Harvard University, for presenting an outstanding and progressive outlook on construction automation. His dedication to this area of study continually captures my fascination with fusing advanced automated technology and architecture. I would also like to thank the staff in the machine shop at Harvard University for appointing their time and equipment available for this research.

I would like to particularly acknowledge Si Li, Lu Yi, and Raphael Rush in contributing and collaborating greatly on the development of the system application for this research. Their patience, resource recommendations, and the time taken to explore strategies and concepts with me had opened doors for a wonderful learning experience.

Thank you to all of my friends and SMArchS Computation colleagues, they have contributed greatly to my educational and student life experience with their unending curiosity about the world, sense of humor, and similar attitudes toward research and exploration for all things that catch the eye.

Lastly, to my family, my grandparents Donato and Avelina Bentajado, and Flordelis Villalon, for their sacrifices and courage to voyage across the world in the 1970s and begin a newfound life in America, paving the way for their children and future progeny to mine the opportunities that this country has to offer. I extend my earnest appreciation to my parents Olivier and Marie Jane Villalon for their unending support, optimism, and affection throughout my life. Mom and Dad, your generous dedication and hard work never go unnoticed by me. Thank you from the bottom of my heart.

CONTENTS

PART I   RESEARCH PERSPECTIVE AND ORIGINS

The first chapter in Part I contains an overview of the research exploration and approach to the thesis, namely, by addressing what is the thesis and how is it important to design? The subsequent chapters present background topics, tracing related work of researchers and individuals who contribute and place this thesis work into a larger perspective.

1 Introduction

Once upon a time, when it was still of some use to wish for what one wanted…there lived a King and Queen who had daughter who was lovely to behold, but who never laughed.
Or perhaps:
…there lived an old fisherman by the side of a sea that had hardly any fishes in it…
If you are like me, you're already hooked. You are ready to abandon all talk of computers and electronic technology and professorships, and settle in to hear a fairy story. Their attraction reaches almost all of us…fairy stories seem to me to have a close connection to technology.

—Allen Newell in *Fairy Stories*, Talks on Computer Science.

1.1 Trouble with the Machine

Machines break down. Computers tend to crash. Cellular phones ring in the middle of a class lecture, shouldn't devices know better? The machine ought to quietly fix itself. The computer ought to recognize and save valuable data before it decides to crash, and the cellular phone should know not to ring during class. People tend to adapt themselves to the devices they use instead of the device adapting itself to the user. The situation is no different in a disciplinary field such as architecture. Most design firms rely on computers to render professional services to clients with accuracy and time efficiency. With the recent availability of desktop fabrication, particularly CNC devices and more recently robotic devices used for design and construction, new design vocabularies and processes emerge thus allowing a proliferation of exploratory design aesthetics and methods. Design tools such as CAD/CAM technology, however, is a multi-stepped and often cyclical process where in most cases the technology presents difficult user handling instead of a simplified one. How then, can designers teach their tools to think like them? What are some of the ways design tools can exhibit common sense? Or exhibit common knowledge shared by designers? How can design tools contain some understanding about the design of an architectural artifact and communicate relevant information about that artifact?

The existence of today's tools (i.e. manufacturing) used by designers usually have the designer having to learn a great deal about the machine in order to conclude with a satisfactory end-product. This process typically entails confronting different user interfaces and machine behavior. The lengthy process it takes to finally arrive at the end-product is one of trial and error, or sometimes luck. If the machines could think with the user's design intent and increase its knowledge about the designer over time, then the designer can experiment with prototypes seamlessly. This is also important in the context of enabling non-expert users the full independence to fabricate their own objects without error-prone interfaces.

1.2 Machines That Know You

Design applications and other tools will become more intelligent by learning from the user's physical design moves such as the process of building an architectural model, drawing on paper, drawing on the computer using design software, and a digital fabrication machine will learn and observe the design and assembly of an artifact so that it too can participate constructively. If design applications had "common sense," or rather, shared knowledge amongst a group of designers, then designing with machines would cut down on production time, costs, increase efficiency, and enable a different dialogue with machines. The machine is no longer the routine servant of commands, but can also form a critical and constructive role.

1.3 Using Adeon: A Vision toward Digital Design Fabrication Processes

In this thesis, we present ADEON, a brick wall design editing application that interfaces with a pick and place robot articulating arm, the IRB-140. The global objective of ADEON looks at bridging the gap between design and construction knowledge by creating and using a repository of an architectural knowledgebase which can be useful towards making an 'intelligent' design application.

The goals of ADEON include:

a) A simple to use graphical user interface.

b) Display design, construction, and machine information relevant to the design being made on the draw editor.

c) Provide cost estimates of the design drawn onto the editor and maintain the sum of brick material usage in the design.

d) Translate user drawn information on the design editor for direct output to machine readable code for the digital fabrication machine.

The machine explored in this thesis is a five axis robotic articulating arm, the IRB-140, used primarily for cutting and assembling blocks of material. In this research, the robotic arm picks and stacks blocks for a brick wall assembly. Software design tools do not exhibit knowledge of the design situation at hand. If a user is confronted with designing a wall system in a two-dimensional CAD environment for example, CAD does not know that the design intention is perhaps a wall of linear dimensionality, placed in the context of a residence or maybe a museum. The proposed system reduces the ineffective process by assisting the user to design based on live textual and graphical feedback about the feasibility of construction and machine fabrication technique. The system bridges the gap between design and construction knowledge.

1.4 Research Approach

The thesis investigation is structured upon the goal of creating and implementing a two-dimensional graphical software design tool used in accompaniment with an industrial machine, or a digital fabrication instrument (Figure 1).

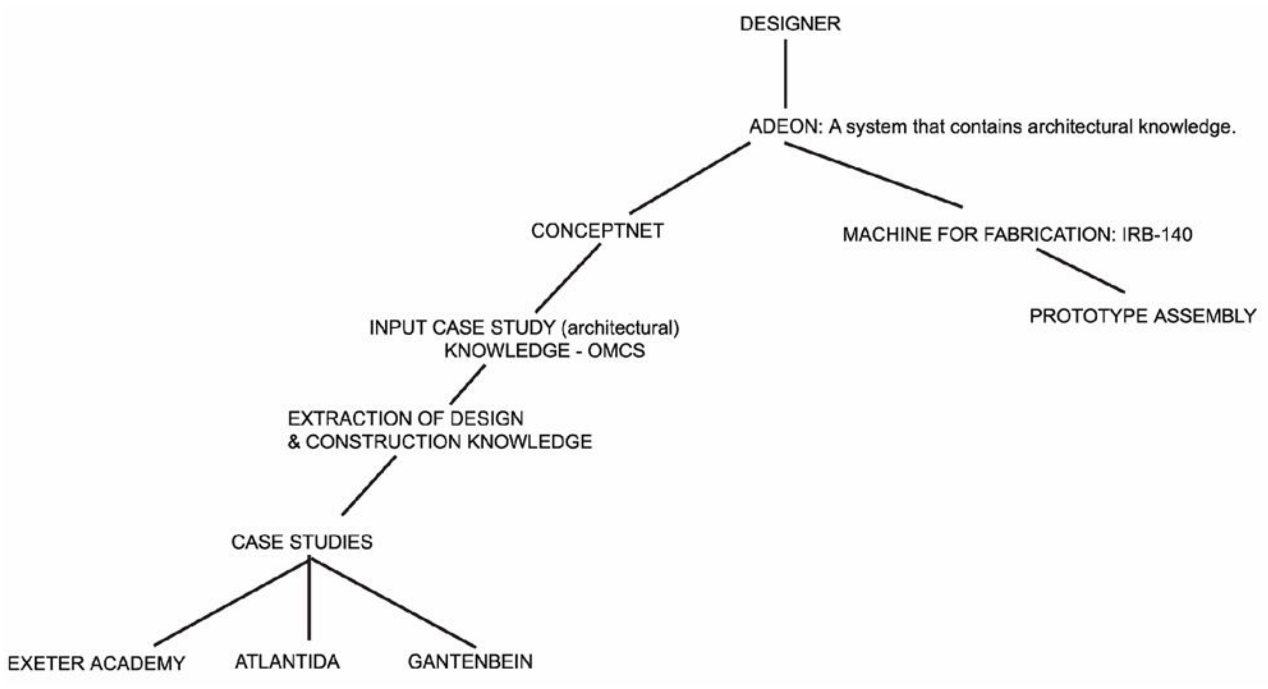

DESIGNER
ADEON: A system that contains architectural knowledge.
CONCEPTNET
MACHINE FOR FABRICATION: IRB-140
PROTOTYPE ASSEMBLY
INPUT CASE STUDY (architectural) KNOWLEDGE - OMCS
EXTRACTION OF DESIGN & CONSTRUCTION KNOWLEDGE
CASE STUDIES
EXETER ACADEMY
ATLANTIDA
GANTENBEIN

Figure 1. Thesis structure tree.

Atop the thesis structure tree is the 'Designer', where s/he serves as the user of the proposed system. Directly below the 'Designer' is the system called, Adeon, taken from the Latin root word *to go,* or *move forward* (Casselman). Adeon is a graphical user interface that enables a user to draw and design a brick wall system in elevation while receiving updated information on constructability, cost effectiveness, and material usage based on real world architectural knowledge. Adeon serves as a system that contains architectural knowledge in natural language, capable of describing design knowledge such as, "architect Louis Kahn exhibits rectilinear wall design in the example of the Library at Phillips Exeter Academy" to construction knowledge such as, "the cross-section of a wall is the depth of the brick", and lastly, to machine knowledge such as, "to place the next brick on a brick wall, I line up the new brick with the bricks already there". The systems storage of design, construction, and machine knowledge brings forth a design tool capable of informing and suggesting design alternatives for the user in the design process as well as provide price estimates of the undergoing design. Ultimately, the system exhibits inferred knowledge about design and the design situation at hand.

1.5 Method of Investigation

Below the proposed system, Adeon, in the thesis structure tree (in Figure 1) is the branching of two separate regions that continue to stem further into other investigations staying within context of the thesis' objective. To the right of the tree stems a physical, digital design experiment using an industrial robotic arm (Figure 2) whose application directly relates to the processes to the left of the tree.

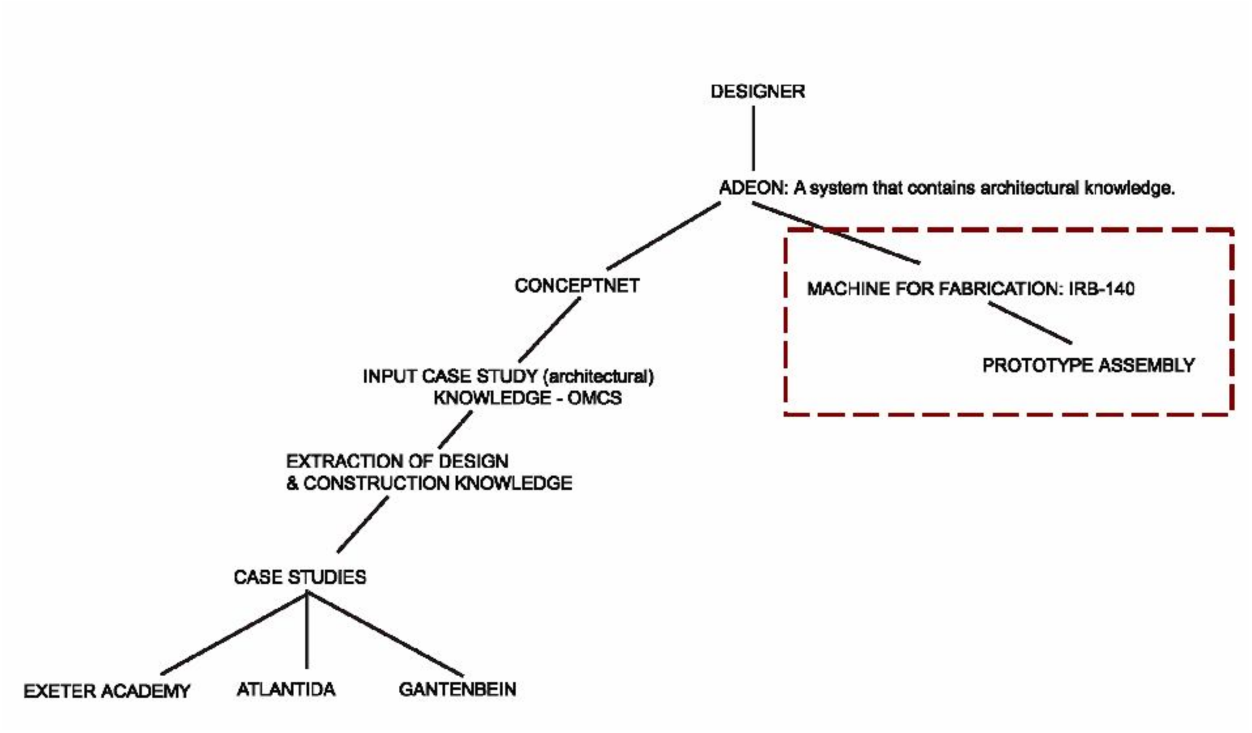
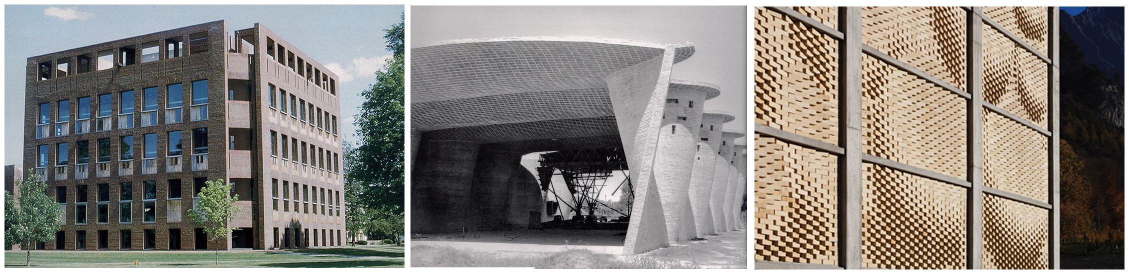

DESIGNER
ADEON: A system that contains architectural knowledge.
CONCEPTNET
MACHINE FOR FABRICATION: IRB-140
PROTOTYPE ASSEMBLY
INPUT CASE STUDY (architectural) KNOWLEDGE - OMCS
EXTRACTION OF DESIGN & CONSTRUCTION KNOWLEDGE
CASE STUDIES
EXETER ACADEMY
ATLANTIDA
GANTENBEIN

Figure 2. Use of a digital fabrication machine (IRB-140) in the thesis process.

It is the hardware explored and queried in the thesis. The left portion of the tree describe an investigative process into design and construction knowledge whereupon the extraction of such knowledge serve as the key component for creating an intelligent system containing and inferring the design of an architectural product, such as a wall system. The furtherance of stemming along the thesis structure tree describe areas of a physical process of data gathering and input for the creation of Adeon, the proposed system. The following descriptions below describe and traverse the thesis structure tree to briefly introduce what each region incurs upon the thesis investigation.

1.5.1 Case Studies on Materials and Methods: Traditional and Non-traditional Uses of Brick

This thesis also examines the use of brick as a building and assembly material to construct traditional and non-traditional surface geometrical forms. The use of blocks serves as an integral component to the fabrication machinery used in this research, the robot articulating arm, IRB-140. Three works of architecture in traditional and non-traditional uses of brick serve the purpose for the case study analysis in extracting the terms used to describe the building in design knowledge to construction knowledge in natural language. The two domains of knowledge are an extremity of one another, but are in constant flux when designing a building since knowledge of the two domains are what bring forth a physical building on site. The three case studies analysis include a) The Library at Philips Exeter Academy by Louis Kahn, b) Atlantida Church by Eladio Dieste, and c) Gantenbein Winery by Gramazio and Kohler (Figure 3).



(A) (B) (C)

Figure 3. a) Library at Philips Exeter Academy by Louis Kahn, b) Atlantida Church by Eladio Dieste, c) Gantenbein Winery by Gramazio and Kohler.

1.5.2 Case Study Extraction of Design and Construction Knowledge

The analysis of the three architectural buildings serves in creating and exhaustively articulating design, construction, and machine knowledge in natural language text. The use of words in architecture creates a vocabulary essential to describing design intentions for crafting the visual form of a building as well as to articulate building construction methods and terminology. Expressing architectural form and building information in natural language serve as supplementary dialogues to graphical images and physical model representations for visual communication. Verbal and visual communication remains complementary in architecture.

1.5.3 Core Technology

The project deliverable explores the creation of an intelligent software agent for use and in conjunction with a digital fabrication machine, the IRB-140 pick and place robotic arm. The software system proposes the implementation of The Open Mind Common Sense knowledge database and the use of ConceptNet, the natural language processing toolkit necessary to create intelligent applications developed by the Software Agents Group of the MIT Media Lab.

OPEN MIND COMMON SENSE

The Open Mind Common Sense (OMCS) Project is a web-based interface that allows contributors over the web to input common sense, or shared knowledge amongst individuals, into a database to teach machines what people would normally accept as common knowledge. To date, there had been over 13,000 people contributing to OMCS generating numerous common sense concepts since September 2000 (MIT Common Sense Computing Group 2008)

CONCEPTNET

ConceptNet is a natural language processing toolkit (MIT Common Sense Computing Group 2008) that "encompasses the spatial, physical, social, temporal, and psychological aspects of everyday life through its knowledge base of a semantic network that contain 1.6 million assertions" (Liu and Singh 2004). ConceptNet captures commonsense concepts that are automatically produced in the Open Mind Common Sense database and is a resource for creating intelligent computer programs.

IRB-140: ROBOTIC ARTICULATING ARM

Used primarily in the automotive industry for welding and assembling manufactured parts, the IRB-140 is an articulated robot arm with five axis range of motion. The robot is designed for use with a hand held pendant controller where the operator can upload, edit, and observe coded automated tasks while under operation.

Chapter One Recap

This chapter presents an overview of the thesis research, question, and strategies for analysis. It defines a problem found in digital design fabrication and presents the thesis approach, design context and scenario, and as well as to briefly explore the various types of technologies that serve to supplement and contribute to the research.

2 Architectural Design Processes and Computing

Not only was the machine unfamiliar to the person involved but also somewhat awesome in size and expense.

—Ivan Sutherland in *Structure in Drawings and the Hidden-Surface Problem*

2.1 Design, Designer, and the Designed

Before delving into background information relative to the relevance of this thesis, it is important to clarify the term ‘designer’ as used generously to describe the roles of the designer in the thesis exploration. The role of ‘designer’ and the process of ‘design’ for which tools can exhibit ‘common sense’ contain multifaceted interpretations. An artist sculpting a block of clay and an engineer calculating vector forces of a conceived bridge are understood as participants in the act of designing. The role of designer is approached as one who participates in the role of exploration to solve a particular design problem towards well developed technical and aesthetic solutions. In Bryan Lawson’s, *How Designers Think* he points out, however, that the spectrum for design definitions derive from a variety of disciplines. “The definition of design [ought to be refocused toward distinguishing]… the variety and complexity of the designer’s role” (Lawson 1980). In other words, Lawson acknowledges that there are attempts to define design, but would rather redirect the effort into defining the designer’s role in varied situations. The situations can originate from the fields of art, engineering, architecture, or any other field, such that the design process share similar characteristics as described in the RIBA handbook:

Phase I assimilation

The accumulation and ordering of general information and information specifically related to the problem in hand.

Phase II general study

The investigation of the nature of the problem.

The investigation of possible solutions or means of solution.

Phase III development

The development and refinement of one or more of the tentative solutions isolated during phase 2.

Phase IV communication

The communication of one or more solutions to people inside or outside the design team (Lawson 24).

From this, a map of the design process explains the specifications involved. It is important to note that the map above demonstrates a situation that must precede a design problem. Each phase directs itself towards the problem at hand in which a solution is the typical result of such exploration. The four phases appear to present clear steps along the processes of design. In *Educating the Reflective Practitioner*, Donald Schon however, would perhaps add that in any of the four phases as described by Lawson, a "reflection-in-action" is likely to be engaged by the designer in the problems set forth as part of the underlying process.

The reflection-in-action involves a designer to form a conversation and apprehension of the problem where s/he can "evaluate [their]…moves in threefold way: in terms of the desirability of their consequences judged in the categories drawn from the normative design domains, in terms of their conformity to or violation of implications set up by earlier motives, and in terms of…appreciation…of the new problems or potentials they have created" (Schon 63). Schon realizes that professional knowledge (derived from a professional education) does not meet "real world" problems where such answers to a problem can be found in textbook knowledge (he notes "professional" as lawyers, engineers, doctors, architects, and so forth). In other words, the assumption that academic research is professional knowledge, is false, "university-based schools of the professions are becoming increasingly aware of troubles in certain foundational assumptions on which they have traditionally depended for there credibility and legitimacy" (Schon 9). These "troubles" become apparent when students enter real-world practice which in turn compels Schon to question if professional education can ever produce a curriculum adequate to the "complex, unstable, uncertain, and conflictual worlds of practice" (Schon 12) that cannot be found in prevailing professional education.

His solution to the problem involves looking to the ways in which artists and architects learn in a design studio, "we ought, then, to study the experience of learning by doing and the artistry of good coaching…and we ought to search for examples wherever we can find them…[such as within] the apprenticeships and practicums…[in the] deviant traditions of studio and conservatory" (Schon 17). Learning by doing demonstrates a process in which the architecture student, for instance, solves problems by not only engaging in analysis and inquiry, but does so with the instructor (or coach) who oversees the student's process and is able to offer solutions based upon a repertoire of professional experience in the field.

This view of design differs from Herbert Simon in the <u>Sciences of the Artificial</u>, where Simon saw design as a process optimization, where a human being or computer, could create solutions based on a series of alternatives (that fit a design criteria, in this case, for the architect), "a series of generators may generate one or more possible outlines and schemes of fenestration for a building…" (Simon 129). Such a view would thus generate optimal solutions to a problem, but Schon is not convinced that process optimization can contribute to real-world problem solving faced by prevailing graduates from a professional education since designing contains situations of uncertainty, uniqueness, and conflict where instrumental problem solving—occupy a secondary place." Designing, to Schon, is a learn-by-doing process, or a reflection in action.

The reflection-in-action involves a designer to form a conversation and apprehension of the problem where s/he can "evaluate [their]…moves in threefold way: in terms of the desirability of their consequences judged in the categories drawn from the normative design domains, in terms of their conformity to or violation of implications set up by earlier movies, and in terms of…appreciation…of the new problems or potentials they have created" (Schon 63).

2.2 Computing Machines

The notion of tools exhibiting common knowledge shared by designers presumes the form of artificial intelligence to take place. The idea of bringing life or some life form that exhibits human intelligence to inanimate objects dates back to the time before substantial groundwork in the field of artificial intelligence was established. In the *Iliad*, Homer tells us that the golden female attendants walked with tripod legs as conjured by the blacksmith Hephaestus. The literary automaton creatures were most likely inspired by actual engineers in the Greek city of Alexandria, where automatic devices were made for amusement (Crevier 1993). The impression of machines with cognitive ability has piqued the interest of many individuals since the ancient times of civilization and has progressed into formal scientific reasoning and research.

Foundational contributions to the field of artificial intelligence took place in the 1950s with John McCarthy, Marvin Minksy, Allen Newell, and Herbert Simon (Poole 1998). Their demonstration of a computer program that solves algebraic word problems and recognizes the English language gave rise to purport that "machines will be capable [of]…doing any work…a [human] can do" (Simon 1965). Although Simon's assertion does not clearly define what type of work a machine could correspond to human activity, the affirmation raises the possibility to discover and develop machine intelligence applicable to other fields like architecture and design for example.

Computer programs excel at solving 'hard problems' in mathematical logic (such as computer chess programs that are capable of beating a human being), or a computer program that can solve calculus problems. Yet it was not until the 1970s that programmable robots were able to visually assemble a stack of children's blocks (Minsky 1986). A computer program that performs child-like physical activities present more complicated programming than computer programs that solve college level math. Marvin Minsky attributes the former such that "to be considered an 'expert' one needs a large amount of knowledge of only a relatively few varieties. In contrast, an ordinary person's 'common sense' involves a much larger variety of different types of knowledge -- and this requires more complicated management systems" (Minsky 1986).

Chapter Two Recap

The two sections described in this chapter address the role and definition of the designer in a context where "learning by doing," ubiquitously presents itself in the act of designing. Moreover, the section asks, how do designers resolve design problems? The chapter looks to different fields of thought and presents their views relevant to design as process, along with the use of computing machines.

3 Characteristics of Digital Design Fabrication

3.1 Uses of Machine Tools

Engaging with digital fabrication machinery requires knowledge and experience to gain expertise since several factors contribute to a successful end product. “The shaping of the design process and product by the design tools is a hallmark of contemporary architectural design practice. Contemporary CAD/CAM technology requires the ‘operator’ to be versed in the processes and procedures necessary to operate each particular tool (both software and hardware). This immediately introduces a new role in the design process whereby the designer, who would typically be responsible for creating the design documentation, must now do so through the intermediary of the operator” (Lobel and Villalon 2007). Take for instance, fabricating an architectural artifact using a Computer Numerical Controlled (CNC) machine (Figure 4).

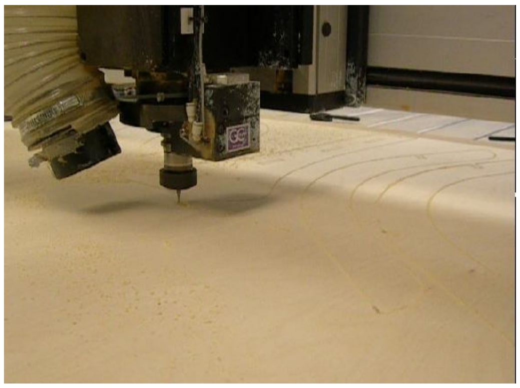


Figure 4. A three-axis Computer Numerical Control (CNC) machine in operation.

Twenty-seven steps approximately represent the routines necessary to fabricate an artifact as follows:

<u>In CAD software environment</u>

1. Flatten 3D object to 2D, save to .dwg file (confirm that origin of drawing [0,0] is set to reflect origin of the CNC machine.
2. Open .dwg file in EZ-CAM, create Tool Path (must make sure geometry is closed and dwg is in imperial units).
3. Material physical limitation 4x8x2" max. constrained by machine's physical milling area.

<u>In EZ-CAM (propriety CAM software environment)</u>

4. Import .dwg
5. Create New Curves
6. Select drawing extreme endpoint to endpoint
7. Setup workstep data
8. Select tool, load tool #5, 1/8" dia.
9. If Pocket, select Mill Left/If Mill, select Mill Right
10. Workstep data > Post (G-code)

<u>CNC Machine Preparation</u>

11. Drop oil lubricant in compressor tube
12. (control box)Switch speed controller ON
13. (control box)Switch compressor ON
14. Switch pressure valve ON
15. OPEN/CLOSE essential suction valves for material
16. Flip speed controller on pc box ON
17. Flip compressor on pc box ON

<u>In TECHNO CNC (software interface to physically control the machine)</u>

Home ALL – reset tool to its home origin (0,0,0)

18. Send machine to pick up tool – Tool Change = 5 (equivalent to 1/8" router drill bit)
19. *Optional (often mandatory) – Physical simulation, set tool head Z-axis above material and trace milling path.*
20. Control axis of CNC machine to user origin using arrow keys.
21. Set drill bit head on stock material using +/- keys.
22. On Techno Interface – set user origin, Zero>ALL
23. Open G-Code file > Preprocess
24. Hit Start 3 times.
25. End milling process (Did product mill correctly? i.e. Did it mill according to desired outcome? If not, proceed from Step 1.)
26. Send machine to put back tool in carriage –Tool Change = 0.
27. Send machine to origin –Home>ALL

A linear manufacturing process would prove as an ideal design workflow, where all approximately twenty-seven routine actions perform smoothly. In a digital design fabrication experiment in *Materializing Design: Contemporary Issues in the Use of CAD/CAM Technology in the Architectural Design and Fabrication Process*[1], a linear manufacturing process does not appear close to actual practice where several machine and software factors lend to variable and often unstable product results. The unpredictability in using digital design fabrication tools are attributed to the variability in stock materials, maintenance of machining tools, as well as humidity and temperature of the work environment. The inconsistency in the end product can either prove positive or detrimental depending on the working environment.

Instead of a linear manufacturing process, designers find themselves in a cyclical fabrication process, often revisiting previous routines in order to resolve certain production issues as seen in Figure 5.

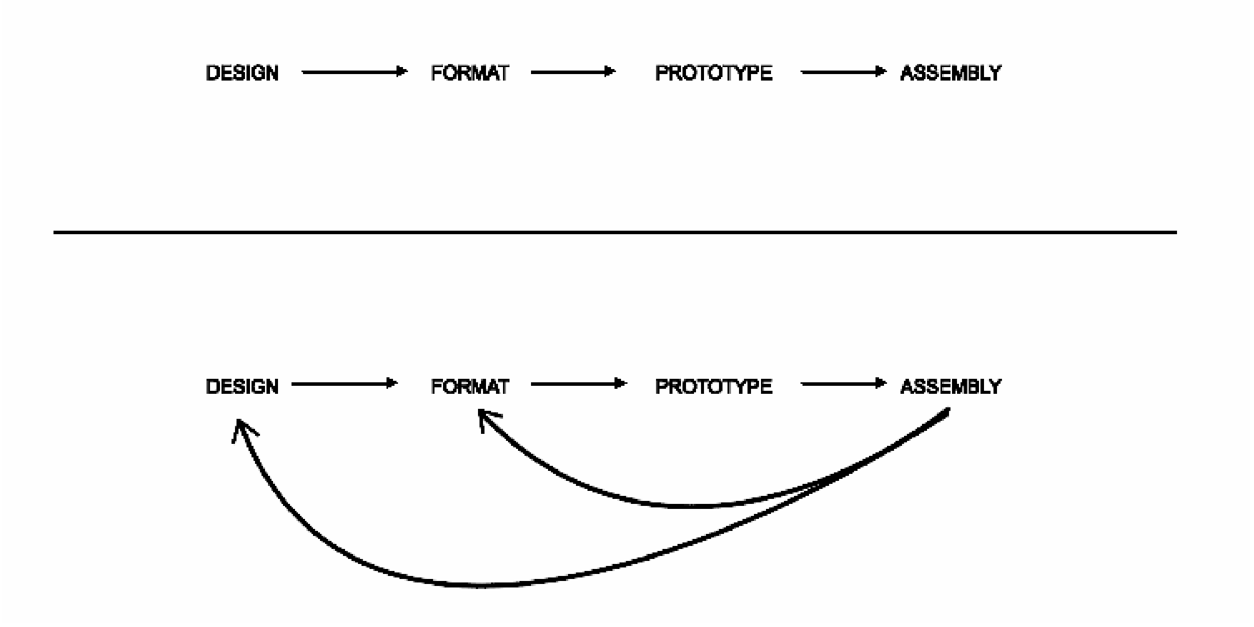


Figure 5. (Top) An ideal design workflow. (Bottom) A recursive approach to digital design fabrication processes.

[1] ASCAAD Conference Proceedings, Nov. 2007, by Josh Lobel and Rachelle Villalon

The description of the digital fabrication experiment in *Materializing Design: Contemporary Issues in the Use of CAD/CAM Technology in the Architectural Design and Fabrication Process* relate the following results when attempting to create aluminum press-fit assemblies (Figure 6):

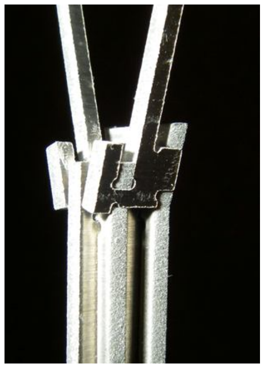


Figure 6. Waterjet cut aluminum mullion.

Empirical testing reveals that press-fit assemblies rely on specific material properties and precise fabrication tolerances in order to sustain a rigid connection between elements (Cardoso and Michaud, 2006). However, different materials displayed varying behavior at different scales, making it difficult if not impossible to predict possible changes in material behavior at different model scales and what that would mean in terms of model assembly and the efficacy of the friction joints. Ideal scenarios were achieved with cellulose-based materials (plywood and masonite) that allowed for slight material deformation during assembly. Attempts at similar assemblies with aluminum proved less successful due to the inelasticity of the material. The most successful press-fit joints that were achieved in aluminum were “slot” connections which provided an equal amount of supporting material on each of the mating pieces (Lobel and Villalon 2006).

The description above inform us that dissimilar materials at different scales makes it “difficult if not impossible to predict possible changes” in the model of assembly and friction joints. With the variability in material, recursive steps contribute a long and arduous process to conclude with a satisfactory product. The steps required to reduce this process is not mentioned, but a need for closing the distances between the designer and machines are of necessity. The next section briefly discusses the use of robotic applications for architectural design and building construction, offering a glimpse of an alternate manufacturing technique at a large scale.

3.2 Automated Building Construction

The use of robotic technologies for building construction has gained prominent use in Japan. The use of robots in construction presents a not so distant future of automation and the potential advancement of robotic “common sense reasoning” approaches to the design and building construction process. The SMART System by Shimzu Co. located in Japan, uses advanced computer automation at a grand scale in the construction processes, unifying underground work, steel erection, finishing, and equipment work to erect multi-story buildings. The SMART system utilizes prefabrication, computer, and robotic technologies:

> The *SMART* system uses an automated erection system, which provides shelter for the work environment, to construct individual floors. The erection system uses overhead gantry cranes to install steel columns and beams on a floor, and then uses hydraulic jacks to move up to the next floor. The erection system grips previously installed columns in order to lift itself to the next floor. Shimizu expects the *SMART* system will decrease labor dependence and shorten construction duration (M.J. Skibniewski, S.C. Wooldridge 1992).

The theatrical description of robots completing the construction of a building at a massive scale not only demonstrates an alternative construction automation process, but exemplifies how traditional on-site construction methods can be potentially outperformed by the advances in robotic automated technologies (Hoang 2005). The development of ‘common sense reasoning’ for robots serve as an essential component for the creation and improvement of intelligent robots in context based reasoning.

Chapter Three Recap

This chapter demonstrates the uses of a digital design fabrication machine and presents the issues and concerns when engaging with such technology, particularly the amount of time and steps necessary to operate the machine for design production. The chapter looks at system idiosyncrasies, processes for resolution, and provides an example of a design scenario. A brief example was also shown to demonstrate how robotics in construction is not a novel implementation. Instead they are already being used in the building construction industry in some parts of the world where robots facilitate the construction process at a large scale.

PART II AN EXPLORATION IN CORE TECHNOLOGY

The following chapters present the technology under investigation for this research, The first chapter introduces the Open Mind Common Sense Project and its impact on creating intelligent software applications. The second chapter illustrates the use of the IRB-140, the robot arm used in this thesis for digital design fabrication.

4 Commonsense Computing

4.1 Open Mind Common Sense

Computers ought to contain shared knowledge that people already know, for example, objects fall down and not up due to the earth's gravitational pull. Common sense, however, is a term that infers that the something "common" belongs to a shared set of understanding of the world, and "sense" belongs to a cognitive understanding of that shared set. Marvin Minsky, however, asserts that common sense can prove challenging to define:

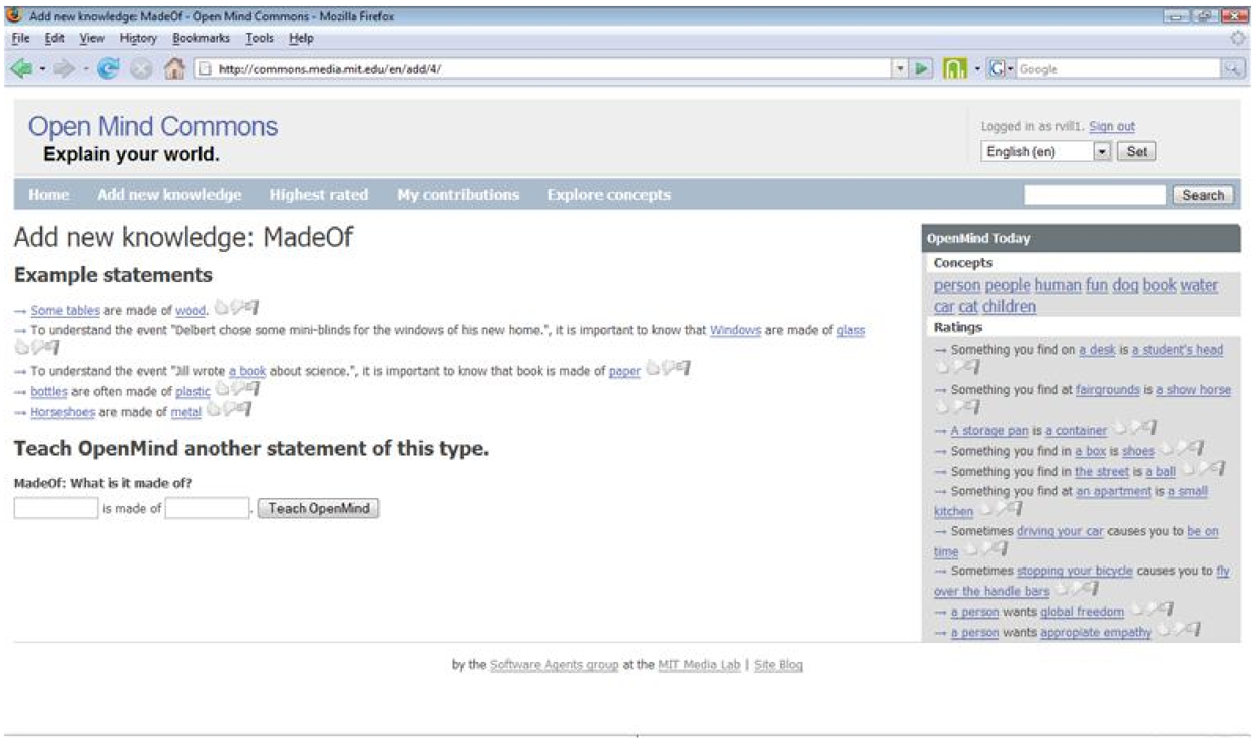
> Common sense is not a simple thing. Instead, it is an immense society of hard-earned practical ideas—of multitudes of life-learned rules and exceptions, dispositions and tendencies, balances and checks. If common sense is so diverse and intricate, what makes it seem so obvious and natural? This illusion of simplicity comes from losing touch with what happened during infancy, when we formed our first abilities (Minsky 1988).

Common sense according to Minsky, shows that common sense is indeed not so common after all. Common sense's obvious and naturalness comes from being ingrained in common concepts since 'infancy' and that which is natural tends to become overlooked. To build intelligent software that can understand some aspect the roles designers take and use in their repository of design knowledge during the design process, then "we require tools that can give machines the capacity to learn and reason" (Singh, Barry, et all 2004) about design. In an article titled, "Teaching Machines about Everyday Life" by MIT Software Agents Group researchers at the MIT Media Lab, they are convinced that in order to build intelligent software, an attempt at documenting "common" knowledge of everyday life proves necessary to move into that direction:

> Can we build a new breed of software with enough 'commonsense' to reason in useful ways about ordinary human life? Imagine if your cell-phone were smart enough to switch to silent

> mode when you entered a movie theatre but alerted you during the film if a relative were to call from the hospital, but not when your friend called from the pub…Such abilities are beyond today's machines largely because they lack even the most rudimentary understanding of people and structure of ordinary human life. They do not know any about, for example:
> the kinds of things we typically do,
> the objects we interact with and why,
> the consequences of actions in different situations,
> the things we like and things we do not like,
> the places we find familiar and the things we do there…
> Instead, our machines today are mindless tools that possess no understanding of why a typical person would need to use them. As a result they are inflexible, unfriendly, and often unnecessarily complicated – they have no ability to adapt to new circumstances, understand the context of their use, or make good guesses at what we wish for them to do, regardless of how obvious it may seem to us (Singh, Barry, et all 2004).

As described earlier in the previous chapter, The Common Sense Computing Group (Software Agents Group), are responsible for creating the Open Mind Common Sense (OMCS) database, a web user interface that allows anybody with internet access to teach machines about what people generally know. OMCS launched in September 2000 and gained 14,000 registered users who contributed nearly 700,000 items of knowledge (Singh, Barry, et all 2004). A typical knowledge browser appears as in Figure 7.


Add new knowledge: MadeOf - Open Mind Commons - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://commons.media.mit.edu/en/add/4/
Google
Open Mind Commons
Explain your world.
Logged in as rvill1. Sign out
English (en) Set
Home Add new knowledge Highest rated My contributions Explore concepts
Search

Add new knowledge: MadeOf

Example statements

→ Some tables are made of wood.
→ To understand the event "Delbert chose some mini-blinds for the windows of his new home.", it is important to know that Windows are made of glass
→ To understand the event "Jill wrote a book about science.", it is important to know that book is made of paper
→ bottles are often made of plastic
→ Horseshoes are made of metal

Teach OpenMind another statement of this type.

MadeOf: What is it made of?

is made of . Teach OpenMind

OpenMind Today

Concepts

person people human fun dog book water car cat children

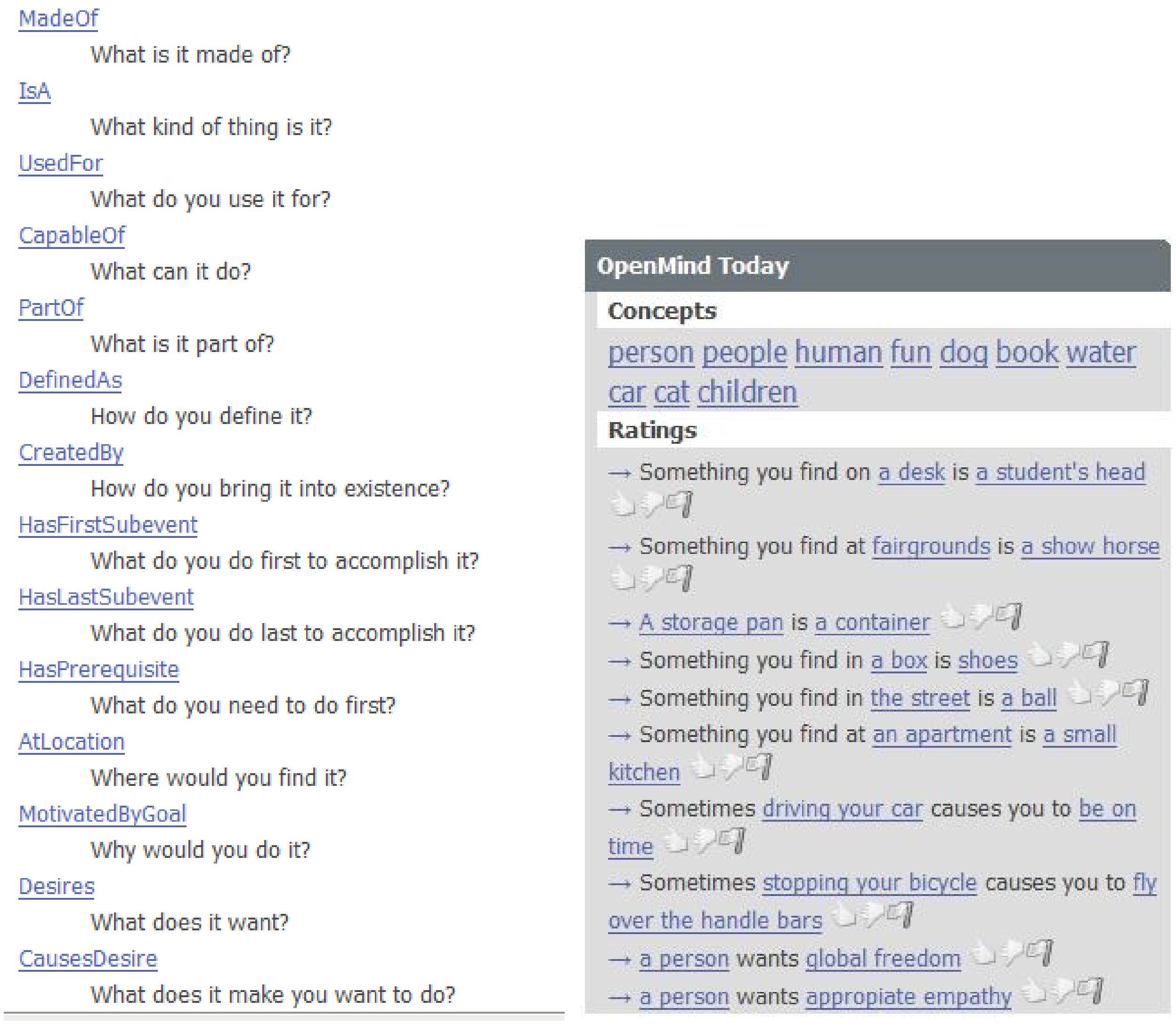
Ratings

→ Something you find on a desk is a student's head
→ Something you find at fairgrounds is a show horse
→ A storage pan is a container
→ Something you find in a box is shoes
→ Something you find in the street is a ball
→ Something you find at an apartment is a small kitchen
→ Sometimes driving your car causes you to be on time
→ Sometimes stopping your bicycle causes you to fly over the handle bars
→ a person wants global freedom
→ a person wants appropiate empathy

by the Software Agents group at the MIT Media Lab | Site Blog

Figure 7. Web user interface of the Open Mind Common Sense Project.

A web user “teaches” commonsense to a machine in the form of simple English statements that builds into a commonsense knowledge base. Semi-structured sentences are provided to extract knowledge given by simple templates such as “flashlights can be used to…” or “the effect of drinking coffee is…” (Singh, Barry, et all 2004) as in Figure 8. The OMCS corpus “showed that 90% of the contributed statements were rated 3 or higher...and about 85% of the statements were rated as things anyone with a high school education or more would expected to know” (Singh, Barry, et all 230).



MadeOf
What is it made of?
IsA
What kind of thing is it?
UsedFor
What do you use it for?
CapableOf
What can it do?
PartOf
What is it part of?
DefinedAs
How do you define it?
CreatedBy
How do you bring it into existence?
HasFirstSubevent
What do you do first to accomplish it?
HasLastSubevent
What do you do last to accomplish it?
HasPrerequisite
What do you need to do first?
AtLocation
Where would you find it?
MotivatedByGoal
Why would you do it?
Desires
What does it want?
CausesDesire
What does it make you want to do?

OpenMind Today

Concepts

person people human fun dog book water car cat children

Ratings

→ Something you find on a desk is a student's head
→ Something you find at fairgrounds is a show horse
→ A storage pan is a container
→ Something you find in a box is shoes
→ Something you find in the street is a ball
→ Something you find at an apartment is a small kitchen
→ Sometimes driving your car causes you to be on time
→ Sometimes stopping your bicycle causes you to fly over the handle bars
→ a person wants global freedom
→ a person wants appropiate empathy

Figure 8. Fill-in-the-blank forms to create semi-structured OMCS sentences.

Presently, if a user were to search concepts about an "architect", OMCS returns its current state of knowledge such as "An architect designs buildings", "architects can plan buildings", "Gaudi was an architect", and so forth. The database does not know what type of buildings an architect can construct or how it is made. Efforts to create a specialized common sense knowledge database useful for a professional field like architecture would mark a new avenue of professional resource using OMCS and ConceptNet. Below is a sample of the information collected from a consensus of architectural knowledge that the computer (for use with a digital fabrication machine) would need to know about the physical world. The sample begins with a base concept on a physical material such as plywood:

- *Plywood is manufactured primarily of softwoods.*
- *Softwoods include Douglas Fir.*
- *Douglas Fir is a type of tree.*
- *Plywood is an engineered panel of wood.*
- *Plywood is made up of a continuous veneer of wood.*
- *Plywood is several layers of wood veneer.*
- *Plywood is bonded with glue.*
- *Plywood is sold in standard sheets almost always 4 feet by 8 feet.*
- *Plywood is subject to dimensional changes due to changes in temperature.*
- *Plywood undergoes preservative treatment to prevent rot.*
- *Plywood prevents air leaks and drafts when used as a wall.*
- *¾" plywood is not easy to split with nails.*
- *Plywood is flat*
- *Plywood warps*
- *Plywood is laminated*

With the sample information above entered into Open Mind Common Sense (OMCS), extracting information from the natural language processing toolkit using extraction rules in an XML-RPC interface, the results are as follows:

```
stem id of "plywood" is 1126595
Top 5 forward relations:
HasProperty: an engineered panel of wood
HasProperty: bonded with glue
IsA: wood veneer
ReceivesAction: bonded with glue
ReceivesAction: used for paneling
Top 5 reverse PartOf relations:
all forms of stem 1126595:
['Plywood', 'plywood', 'Plywood ']
```

The term “plywood” is designated a stem id which is the stem object that corresponds to the natural language and necessary for linking and weighing the verity of the word. The forward relations, “HasProperty”, “IsA”, “ReceivesAction” are binary relations the user selects to structure a concept, or word, like plywood, in order to create associations to that word. There are twenty-two other binary relationships such as MadeOf (what is it made of?), CapableOf(What can it do?), to HasPrerequisite(What do you first to accomplish it?). MadeOf, for example, would require the input from a user *“x”* is made of *“y”*, or “*All matter* is made of *atoms*”.

Knowledge other than building materials like plywood is in the process of collection, such as necessary hardware (nails, screws, bolts, etc.) for construction. Unlike other professional fields such as aerospace, medicine, or mechanical engineering, the building industry is notably slow to update its standard knowledge of architecture and construction. Discoveries of new construction techniques and methodologies takes long for approval due to liability and established building codes for each U.S. region. Since architectural construction knowledge and technology is slow to progress, collecting a wide array of common-sense building information would amount to being a valuable resource for architects and non-experts in the long term.

4.2 ConceptNet

As mentioned earlier, ConceptNet is a natural language processing toolkit composed of a semantic network that contains 1.6 million assertions (Liu and Singh 2004). ConceptNet extracts sentences from the Open Mind Common Sense database into nodes containing four structures: nouns, verbs, adjectives, and prepositional phrases. The nodes create normalized assertions such as to correct grammatical errors, like spelling. The process further reduces the phrases towards lexical generalizations, vocabulary discrepancies, and to generate inference procedures. ConceptNet exceeds at contextual reasoning and outputs concepts into components (Figure 9), the toolkit “represents a new direction for the development of commonsense Artificial Intelligence systems” (Liu and Singh 2004).

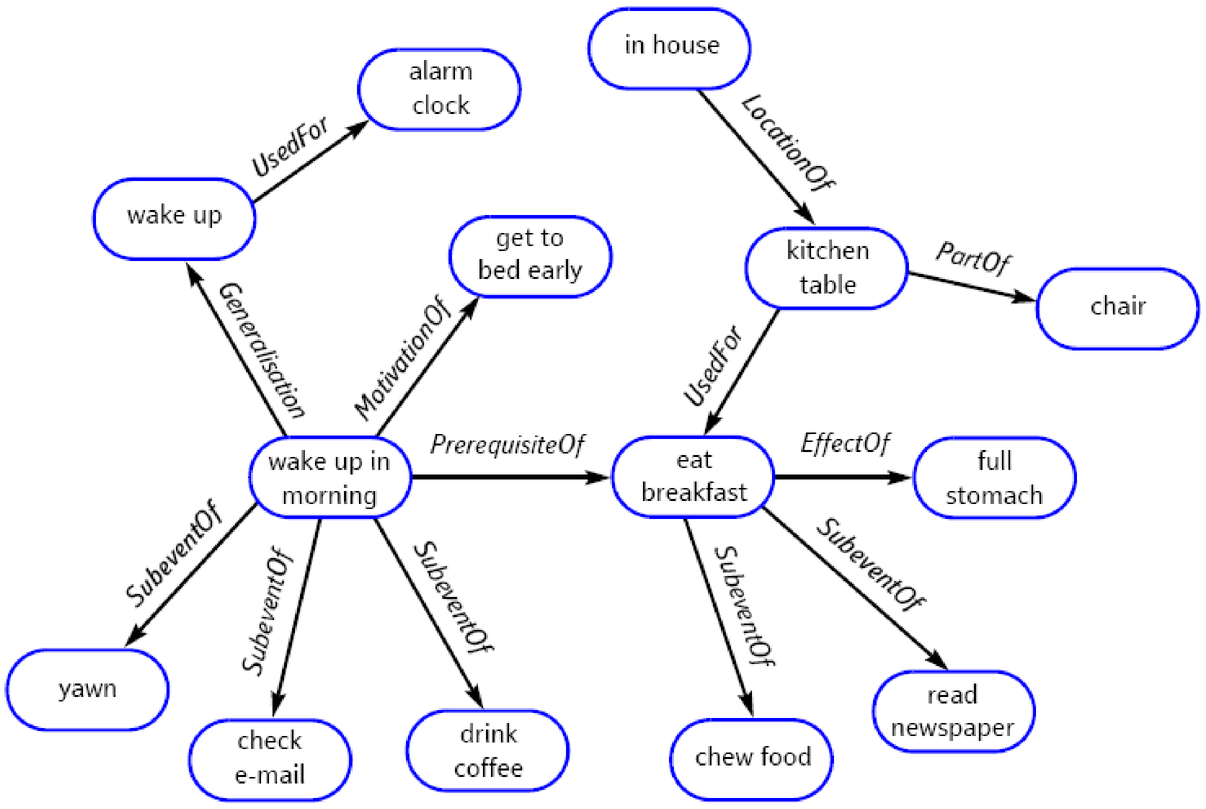


Figure 9. A semantic network of everyday concepts. Image from "ConceptNet: A Practical Commonsense Reasoning Toolkit" (Liu and Singh 2004).

ConceptNet's natural processing engine supports textual-reasoning tasks and provides advanced semantic relations in comparison to traditional semantic processing. The use of ConceptNet has brought upon a variety of novel applications without requiring specialized knowledge for input into a database.

<u>Chapter Four Recap</u>

This chapter addresses the uses and definition of common sense computing, briefly tracing its theoretical origins to thinkers like Marvin Minsky, and looks at how common sense reasoning has been recently applied as a software agent for building intelligent systems as demonstrated by the Open Mind Common Sense Project and the use of ConceptNet, the natural language processing engine. A walk through on how OMCS functions with a given term demonstrates how terms are given value and weight in the context of inserting data.

5 IRB-140

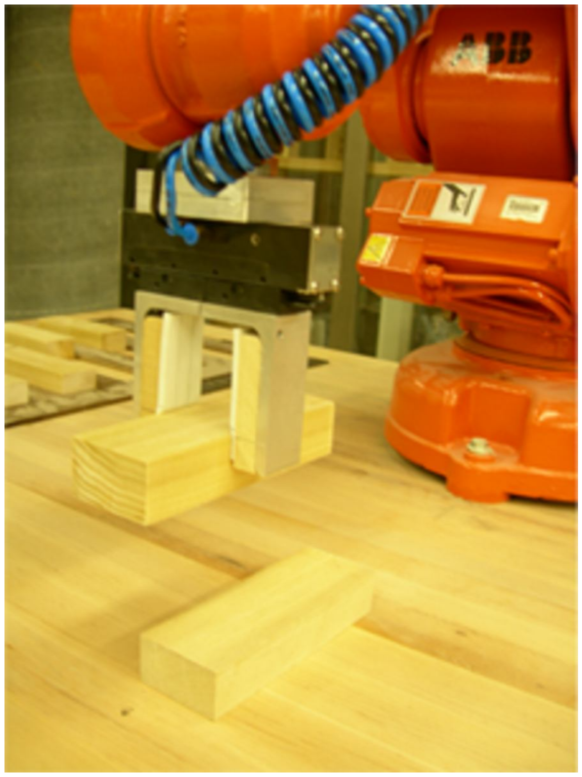


Figure 10a. The IRB-140, lifting and stacking wooden blocks.

5.1 Using the Robot

Unlike two dimensional digital fabrication cutting machines such as the Computer Numerical Controlled (CNC) machine, where two axis of motion are its range, the IRB 140 can accommodate the cutting of parts in two and three dimensions, and to finalize with parts assembly (Figure 10a). This aspect of fabricating and assembling for one machine is not new, advanced technology in automated construction techniques such as contour crafting and three-dimensional printing, are some of the technologies that employ versatility in complex surface geometry creation.

Using the IRB-140, on the other hand, provides greater flexibility in the process of fabrication due to its range of motions and static payload for lifting construction materials. The IRB-140 is a robot with five degrees of freedom whose parameter values are changeable. The robot demonstrates a potential use on the construction site, capable of repetitive automated tasks and specific operation function.

The IRB-140 recognizes objects and paths as vector data placed as target points in three-dimensional space (Figure 10b). The IRB-140 can implement the use of vision-based processing to execute routine tasks. The IRB-140, however, used in this project does not implement that capability. Instead, the IRB-140 operates under a series of executable code, called Rapid Code. Like the CAD/CAM's output of G-Code for digital fabrication, Rapid Code also contains target paths and target points. Without vision-based processing, the IRB-140 does not recognize whether an object has picked, grasped, and placed onto a specified location, nor does it recognize whether the blocks for picking and placing stack and accumulate in preferential order.

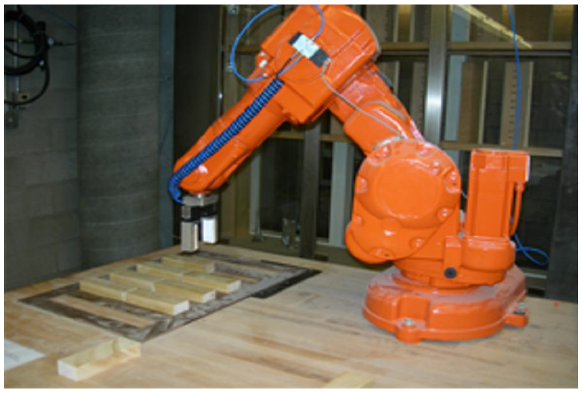


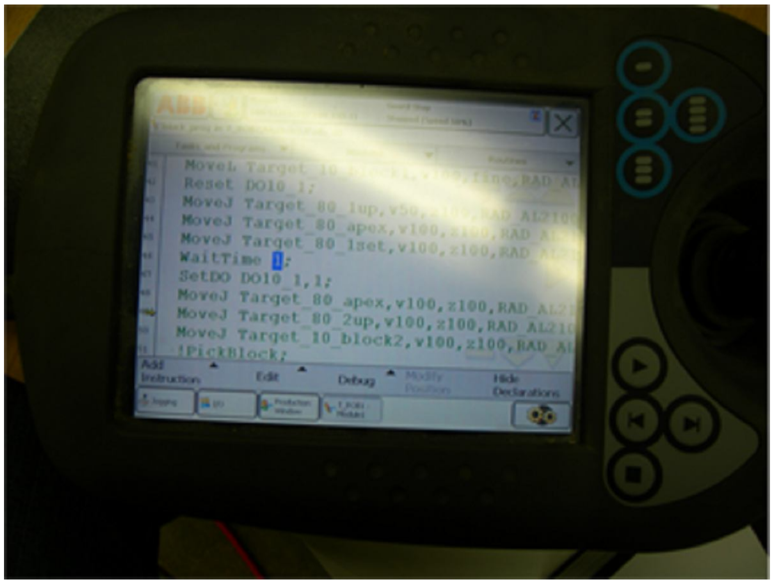
Figure 10b. IRB-140 in initial execution.

5.2 A Look into Tasks and Code

The following information is a brief overview of IRB-140's machine readable code:

```
PERS tooldata RAD_AL2100:=[TRUE,[[0,0,121],[1,0,0,0]],[0.17,[0,0,1],[1,0,0,0],0,0,0]];
```

CONST robtarget b0u:=[[-96.4,-550,80],[-8.65927457071935E-17,0.707106781186548,0.707106781186547,8.71576399210524E-32],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
CONST robtarget b0 :=[[-96.4,-550,30],[4.32963728535968E-17,0.707106781186547,0.707106781186548,4.32963728535968E-17],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];



```
MoveL Target_10_block1,v100,fine,RAD_AL
Reset DO10_1;
MoveJ Target_80_1up,v50,z100,RAD_AL210
MoveJ Target_80_apex,v100,z100,RAD_AL
MoveJ Target_80_1set,v100,z100,RAD_AL
WaitTime 1;
SetDO DO10_1,1;
MoveJ Target_80_apex,v100,z100,RAD_AL2
MoveJ Target_80_2up,v100,z100,RAD_AL210
MoveJ Target_10_block2,v100,z100,RAD_AL
!PickBlock;
```

Figure 11. The flex pendant used for operating the IRB-140, displaying Rapid Code on screen.

The code above displays vector data of each object defined as a target point. For instance,

CONST robtarget b0u:=[[-96.4,-550,80],[-8.65927457071935E-17,0.707106781186548,0.707106781186547,8.71576399210524E-32],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

describes the initial target point of a block located at -96, -550, 80 on a Cartesian grid as a constant routine. The set of numerical values subsequent to the first describe the path at which the robot will follow en suite until the next 'robtarget' value is encountered. Figure 11 demonstrates the use of the code within the flex pendant, a hand device requirement when operating the robot. The pendant device offer direct rapid code line manipulation/editing, file upload, and manual control of the robots range of arm and wrist motion.

Chapter Five Recap

This chapter presents an overview of the digital fabrication machine used as a test-bed for the thesis exploration. The IRB-140 functions according to a set of routine instructions written in Rapid Code, direct for upload by USB device to the main controller and also requires a pendant device for use while under operation.

PART III A LOOK INTO ARCHITECTURAL CASE STUDIES: EXTRACTING DESIGN, MACHINE, AND ROBOT KNOWLEDGE

The chapters contained herein investigate the processes required for architectural knowledge extraction for input into an intelligent system application.

A LOOK INTO ARCHITECTURAL CASE STUDIES…

6 Extraction of Case Studies

"If you talk to a brick and ask it what it likes, it'll say it likes an arch. And you say to it, look arches are expensive and you can always use a concrete lintel to take the place of an arch. And the brick says, I know it's expensive and I'm afraid it probably cannot be built these days, but if you ask me what I like it's still an arch."
--Kahn, Louis "An Architect Speaks his Mind" House and Garden 142, no. 4. October 1972

Taken in the light of architect, Louis Kahn's words, he describes a building and his design for them as a matter of intention towards high level design concepts. In his own words on the design of the Library at Phillips Exeter Academy, "I see a library as a place where the libraries can lay out the books, open especially to selected pages to seduce the readers. There should be a place with great tables on which the librarian can put the books, and the readers should be able to take the books and go to the light" (Wiggins 11). Kahn describes his approach in designing the library from a perspective that examines the functional and conceptual qualities of an architectural space as in to "seduce the readers" or "take the books and go to the light." The use of language as a means to describe the purpose of a built environment is an essential aspect of communication in the design process. For that particular reason, the words used for describing design are an integral component to Adeon, the intelligent design system.

6.1 Collecting Specialized Common Sense

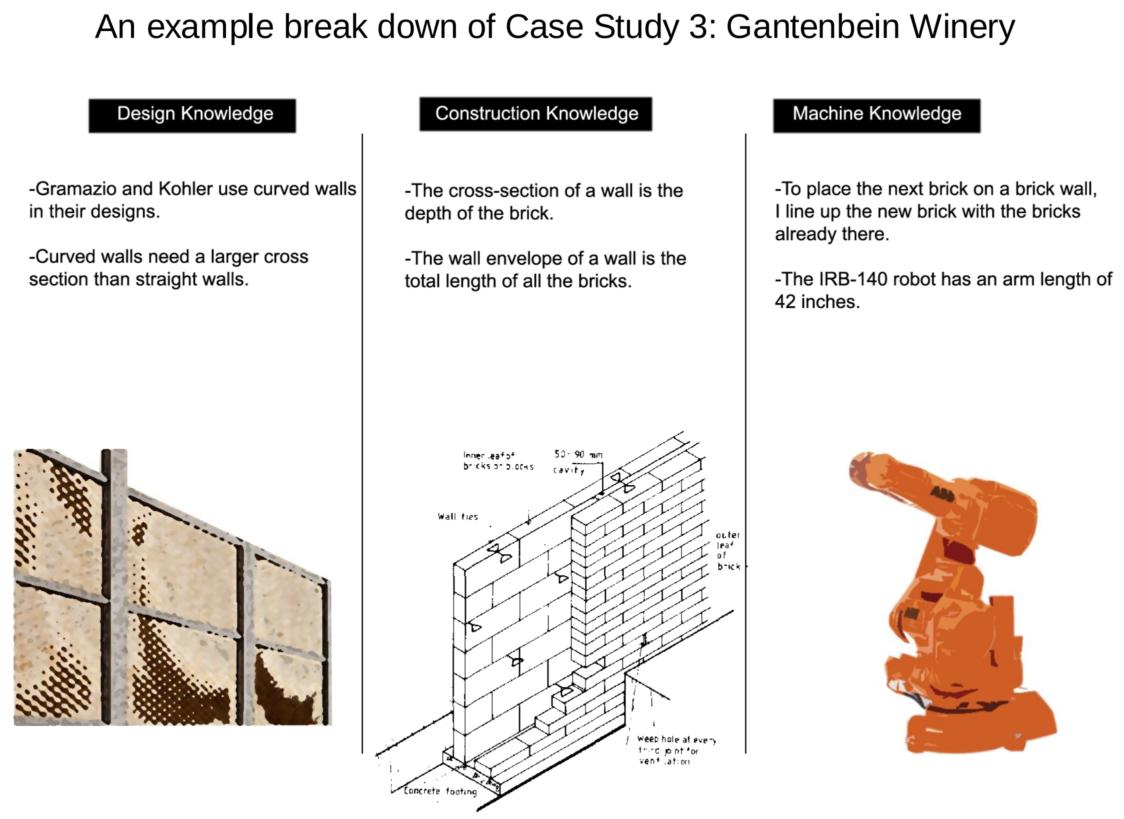
OMCS currently does not contain specialized common sense knowledge such as information that architects and experienced digital fabrication machine users may know. The integration of architectural begins by gathering a corpus of architectural "common sense" that a computer would need to know about designing, constructing, and assembling objects with the IRB-140. An attempt to do so begins by analyzing the three case studies of traditional and non-traditional uses of brick in architecture. The buildings that serve as case studies are 1) Louis Kahn's Philips Exeter Library, 2) Atlantida by Eladio Dieste, and 3) Gramazio and Kohler's Gantebein Winery. Table 1 illustrates the process of analysis for

Philips Exeter Library. Table 1 illustrates the process of analysis and comparison of design and construction knowledge for the Phillips Exeter Library. The table illustrates a phase where the case study observation undergoes a reformulation to befit the Open Mind Commons form. The initial case study observation records can be seen in Appendix D. exist

TABLE 1. LIBRARY AT PHILLIPS EXETER ACADEMY
ARCHITECT: LOUIS KAHN
TRADITIONAL BRICK ARCHITECTURE

Design Knowledge	Construction Knowledge
Louis Kahn **can** design brick walls. (CapableOf)	**An** exterior brick wall **wants to** be designed for stopping water ingress from melted snow, rain, and ice. (Desires)
Brick **is a type of** load-bearing material. (IsA)	**Something you need to do before** you lay bricks **is** to mix mortar. (HasPrerequisite)
Philips Exeter Library **is** designed in brick to keep with the traditional New England campus. (HasProperty)	Hand mixing mortar **requires** masonry cement, sand, clean water, mortar box, mortar hoe, square point shovel, and a bucket. (HasPrerequisite)
Philips Exeter Library **is a** brick building designed to keep with the traditional New England campus. (IsA)	Mortar **is created by** mixing masonry cement, sand, and clean water. (CreatedBy)
The Philips Exeter Library **is the** building designed by architect, Louis Kahn. (DefinedAs)	**The first thing you do when you** prepare mortar **is** place the bag of mortar inside a mortar box. (HasFirstSubevent)
Philips Exeter Library's brick piers **are** wider nearer to the ground where their loads are greater. (HasProperty)	**One of the things you do when you** have mortar in a mortar box **is** to divide the mix in half. (HasSubevent)
The brick wall **is** interspersed with apertures for windows. (HasProperty)	**One of the things you do when you** divide mortar mix in half **is** spread sand on top of the mortar cement. (HasSubevent)
You would put openings in a brick walls **because you want** a door or window.	

As mentioned before, the table demonstrates the use of the eighteen OMCS relations, highlighted in bold, necessary to form sentence structures for the OMCS database. For example, “Brick *IsA* type of material” which falls under the relation “Is A”; the eighteen relations function as a textual form for web users to formulate concepts to teach the computer. In essence, the extraction of design, construction, and machine knowledge are broken down into initial base descriptions. Figure 12 illustrates the filtering process of architectural knowledge for the Gantenbein Winery by Gramazio and Kohler. The sentences undergo restructuring to befit the eighteen relations necessary for input into the Open Mind Common Sense Project.


An example break down of Case Study 3: Gantenbein Winery

Design Knowledge

-Gramazio and Kohler use curved walls in their designs.

-Curved walls need a larger cross section than straight walls.

Construction Knowledge

-The cross-section of a wall is the depth of the brick.

-The wall envelope of a wall is the total length of all the bricks.

Machine Knowledge

-To place the next brick on a brick wall, I line up the new brick with the bricks already there.

-The IRB-140 robot has an arm length of 42 inches.

Inner leaf of bricks or blocks
50 - 90 mm cavity
Wall ties
outer leaf of brick
Weep hole at every third joint for ventilation
Concrete footing

Figure 12. Distinguishing between Design, Construction, and Machine Knowledge for the Gantenbein Winery case study.

The three types of knowledge, once inserted into the Open Mind Commons database, form a semantic network from which ConceptNet, the natural language text processing engine, can extract the assertions for creating adaptations toward intelligent applications. The semantic network (Figure 13) for Case Study Three, illustrates how concepts such as “curved walls” or “IRB-140” form nodes and connect to other nodes via OMCS relations (i.e. CapableOf, HasProperty, and so forth).

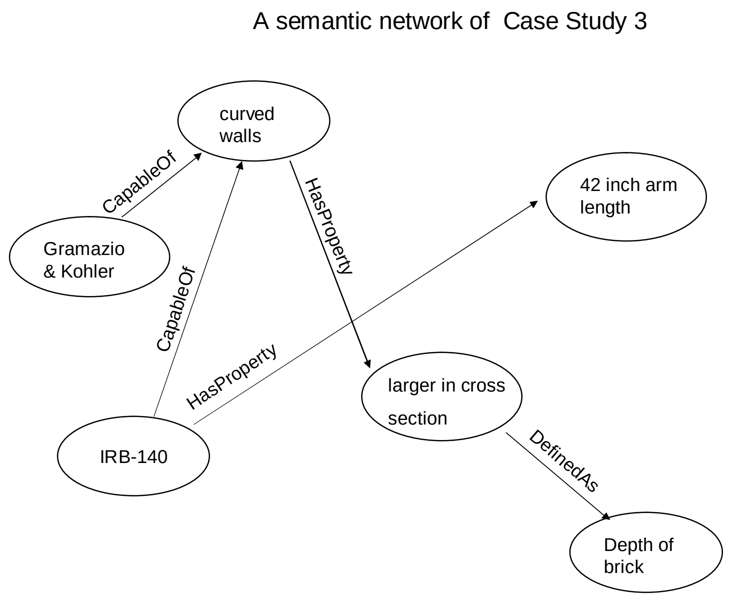


Figure 13. A semantic network of the Gramazio & Kohler wall.

<u>Chapter Six Recap</u>

Chapter Six presents an overview of case studies for analysis and extracts specific information regarding design, construction, and machine knowledge in natural language text. Moreover, the data undergoes retranslation for input into the Open Mind Common Sense database to create a relational ontology of relevant architectural concepts.

PART IV ADEON: SYSTEM AND DESIGN CONTEXT

This section describes the system architecture of Adeon as an application in an architectural design context. An illustrated walkthrough of some design scenarios exemplify the advantages of using and adapting Adeon as part of the digital design process.

7 ADEON

7.1 The System Architecture of ADEON

Adeon is designed to operate with a digital fabrication device. This research explores the IRB-140, a robotic articulating arm to be used with Adeon. The IRB-140, as mentioned earlier, functions as a pick and place robot, picking and placing 4” x 6” blocks while situated in a stationary base. A user instantiates Adeon through a graphical user interface containing a draw editor for creating brick walls with the option of three different geometrical configurations: vertical, inclined, and curved walls. The three geometric wall selections serve as initial test options for the user to engage with, leaving an opportunity for further types of wall creations containing complex geometry into future versions of Adeon. While the user is drawing/designing a brick wall, the system Adeon, recognizes the type of wall configuration and displays relevant design, construction, machine and cost information pertinent to the type of wall drawn onto the editor and in turn, transforms that design into readable machine code for digital fabrication. Figure 14 displays the interaction between the designer and the system.

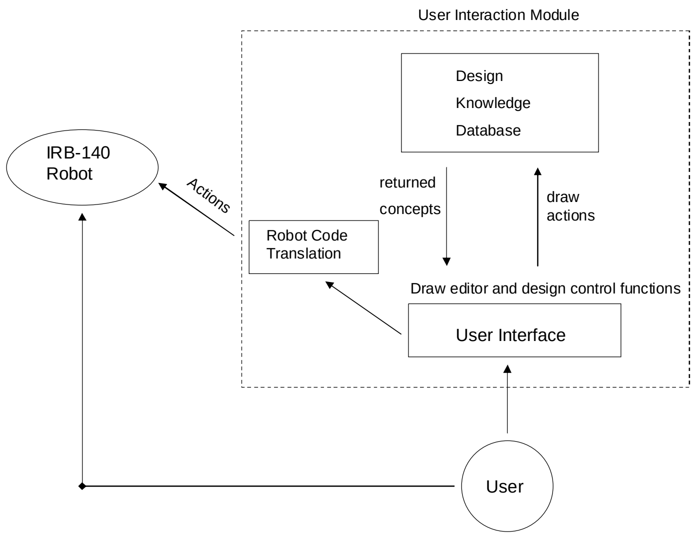


Figure 14. Adeon’s system architecture, an interaction between user, application, and digital fabrication device.

Currently, Adeon interprets user drawing sketches upon the draw window and returns design, construction, and machine data in natural language text.

7.2 Design Scenario I: Vertical Brick Wall

For example, suppose the designer would like to design a vertical brick wall using the IRB-140 (Figure 15).

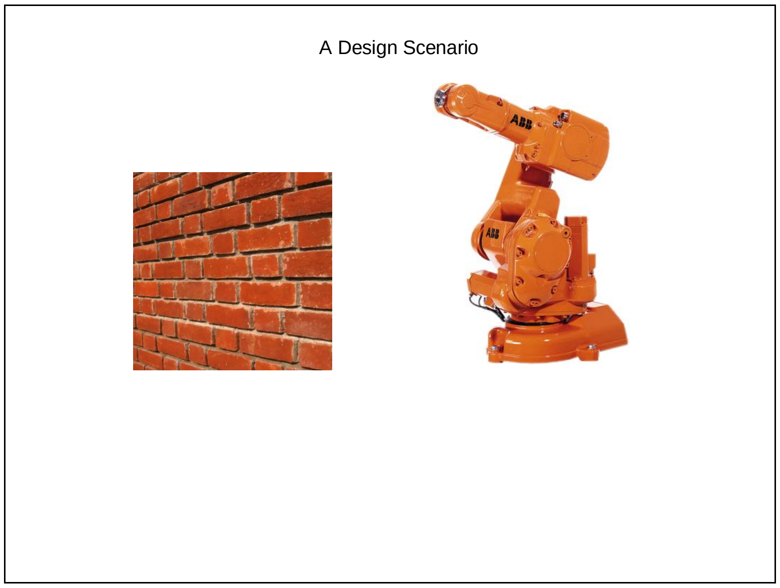
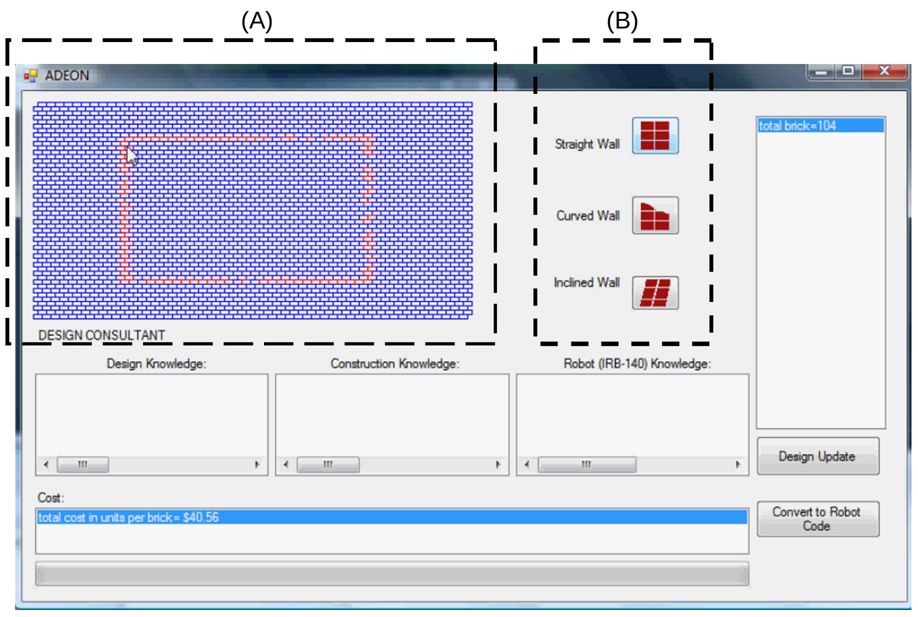


Figure 15. Creating a brick wall with a digital fabrication device.

The user instantiates Adeon and selects the vertical wall design tool available on the Graphical User Interface (Figure 16). As the user is drawing a vertical wall, the system updates itself to reflect the current costs of total units per brick used in the design, total brick count, and lastly design, construction, and robot knowledge.

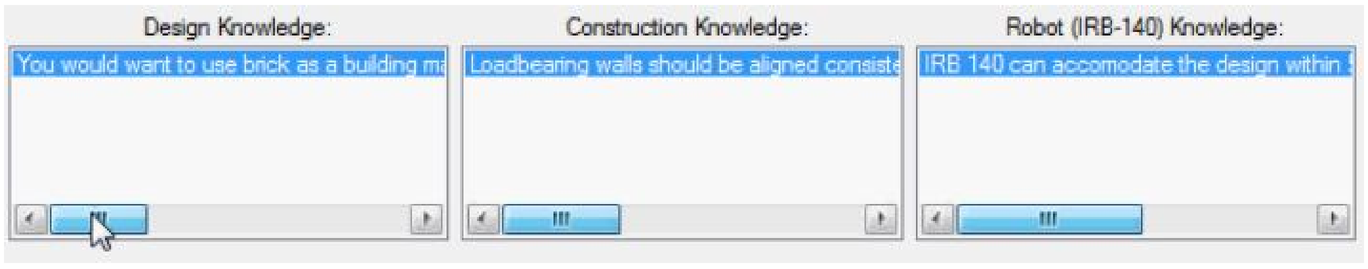


total brick=104

total cost in units per brick= $40.56

Figure 16. Adeon's GUI . A) The user draws a vertical wall in the brick design window by, B) selecting the vertical wall tool.

Upon completion of the vertical brick wall within the editor, Adeon reflects an estimated total cost of units per brick as well as total bricks used in the design. The cost suggests the use of a modular brick unit at the rate of $0.39 per block. Three text windows also display three types of knowledge relevant to the current design (Figure 17).



You would want to use brick as a building ma

Loadbearing walls should be aligned consiste

IRB 140 can accomodate the design within

Figure 17. Display of relevant knowledge modes for current design drawn in editor.

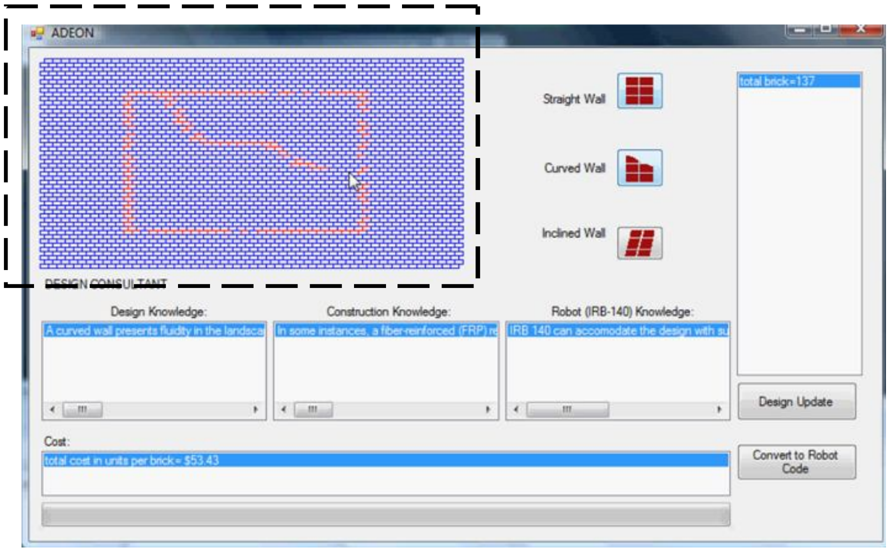
The “Design Knowledge” window informs the user in the vertical wall design example: *You would want to use brick as a building material because you want the aesthetic of having the forces of gravity and weight to be evident in the construction.* The Design Knowledge window outputs a sentence that articulates design at a level of aesthetic reasoning, where “forces of gravity and weight” are meant to imply brick as a visually heavily weighted object. Design knowledge at this level of abstraction describes a relationship between a visual representational figure and a rhetorical one. In T. Knight and G. Stiny’s *Classical and Non-Classical Computation*, verbal representation “is the kind…we use all the time to communicate to each other…and to conduct most of the affairs of the day” (Knight and Stiny 356). Despite the division between the verbal and visual representation, both representations form an interconnected relationship (Knight and Stiny 355) dependent upon one another.

Adjacent to the “Design Knowledge” window on the graphical user interface is the “Construction Knowledge” window. In the vertical wall design example, the output under construction knowledge reads: *Loadbearing walls should be aligned consistently from floor to floor and should be continuous from the roof to the building foundation.* This particular knowledge reflects a design intention for the creation of a loadbearing wall system, and directly indicates valuable information for designing in the situation that the designer may otherwise overlook. If the walls were *not* aligned from floor to floor and from roof to building foundation, then the resulting configuration becomes a partition wall without shear dependency. The functional requirements of a brick wall is given by Adeon’s “knowledge” repository and access to the basic information for understanding building construction, this allows the user to make informed design choices based upon Adeon’s output consultation.

The remaining window, “Robot (IRB-140) Knowledge” also displays pertinent textual information regarding the design of a vertical brick wall relative to the digital fabrication machine itself. The “Robot Knowledge” window box displays, *IRB-140 can accommodate the design within 50 to -230* *degrees range of movement*. That is, Adeon contains knowledge that informs the user whether the current vertical wall design accommodates the machine constraints and requirements for successful product fabrication.

7.3 Design Scenario II: Curved Brick Wall

Should the designer decide to switch from designing a vertical to a curved wall instead (Figure 18), the system recognizes the switch in tasks and is capable of relating pertinent information to the new design scenario.



total brick=137

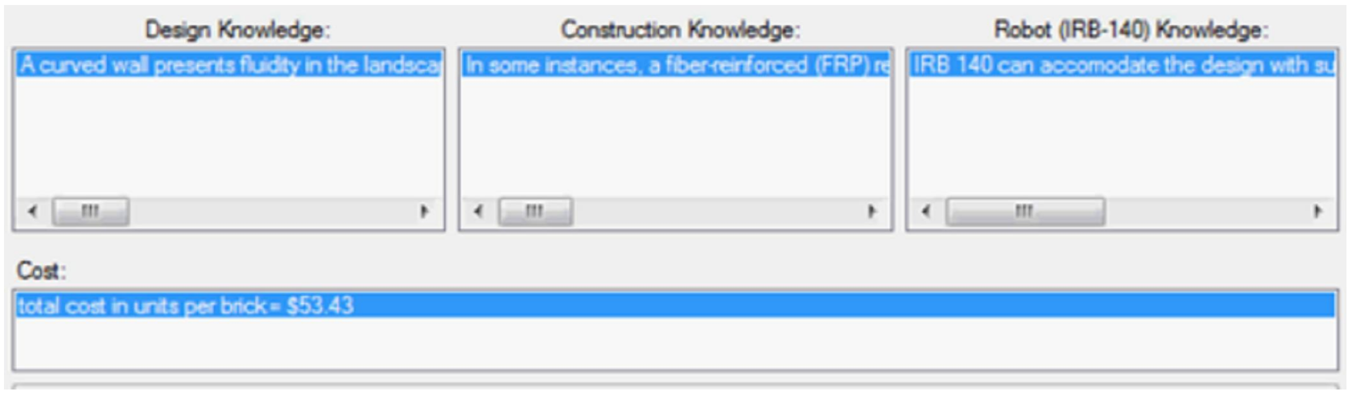
A curved wall presents fluidity in the landsca

In some instances, a fiber-reinforced (FRP) re

IRB 140 can accomodate the design with su

total cost in units per brick= $53.43

Figure 18. Drawing the curved wall upon selection on the Graphical User Interface.



A curved wall presents fluidity in the landsca

In some instances, a fiber-reinforced (FRP) re

IRB 140 can accomodate the design with su

total cost in units per brick= $53.43

Figure 19. Adeon's display of updated knowledge based on new curve wall design.

Upon the new curved wall design (Figure 19), the Design Knowledge label box now reads, *A curved wall presents fluidity in the landscape. An example of this is Atlantida by Eladio Dieste where walls are generated by straight line segments whose ends were translated along a sinusoidal path*. The Construction Knowledge labelbox, also updated, now advises: *In some instances, a fiber-reinforced (FRP) rebar is used for highly specialized brick work because of its high tensile strength and light weight, corrosion resistance and dielelectric (nonconductive) properties*. Lastly, the Robot (IRB-140) Knowledge box reads, *The IRB-140 can accommodate the design with a supplementary load at 32 inches reach of the 5th axis*. Along with this, total costs of bricks per unit and bricks used for the design also display updated information.

7.4 Program Structure

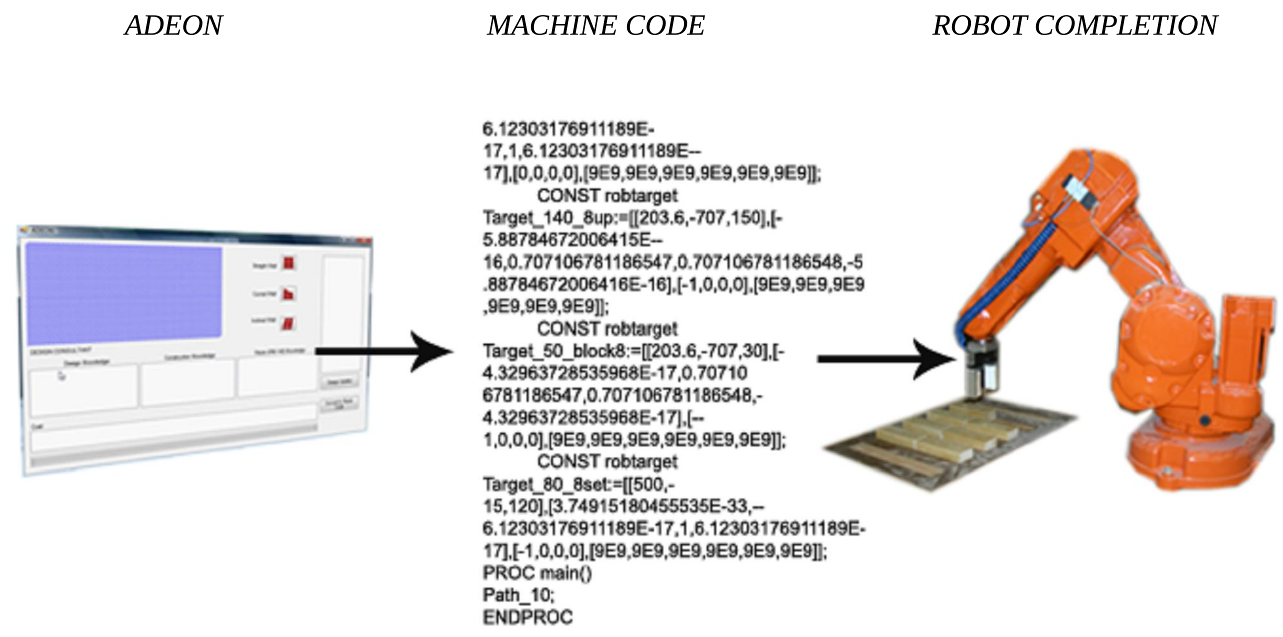
The software framework is divided into Graphic Components, Graphic Data, and Maps, as well as the necessary Form Control to instantiate key behaviors. The Graphic Component contains the graphic styles for the bricks shown as default on Adeon's start up interface, containing an array of Bricks based on location and size. Each set of geometry, or Graphic Data drawn by the user stores itself as data onto a list.

That geometry data, or user input drawn information, is recognized by each brick as a cell containing a point, size, and location index that serve as a component to the array of bricks as a Map (see Appendix G). The term, Map, refers to a given location that acknowledges the user's drawn input into the drawing window and can thereby determine which brick cell the user physically draws on based on 'x' and 'y' Cartesian coordinates, the amount of brick cells the user has drawn across, convert user mouse input to location, as well as to convert a two-dimensional array index to a linear array index for each brick cell.

The Form Control links the user's input on the graphical user interface to specified behaviors or tasks, as well as initialize the Map function. Within the Form Control are the three list boxes containing Design, Construction, and Robot Knowledge. Upon listening, or instantiating either the vertical, curved, or inclined wall option on the graphical user interface, the draw function automatically follows suit.

7.5 ADEON and IRB -140 in Operation

Upon completion of a satisfactory design on the draw editor as well as the reflection of relevant information, the user can now convert the drawn data into machine readable code for the IRB-140 to recognize and perform the design operation as in Figure 20.



```
6.12303176911189E-
17,1,6.12303176911189E--
17],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget
Target_140_8up:=[[203.6,-707,150],[-
5.88784672006415E--
16,0.707106781186547,0.707106781186548,-5
.88784672006416E-16],[-1,0,0,0],[9E9,9E9,9E9
,9E9,9E9,9E9]];
        CONST robtarget
Target_50_block8:=[[203.6,-707,30],[-
4.32963728535968E-17,0.70710
6781186547,0.707106781186548,-
4.32963728535968E-17],[--
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget
Target_80_8set:=[[500,-
15,120],[3.74915180455535E-33,--
6.12303176911189E-17,1,6.12303176911189E-
17],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
PROC main()
Path_10;
ENDPROC
```

Figure 20. Adeon outputs machine readable code for robot operation tasks.

The following images (Figure 21 and Figure 22) display the curved wall design made with ADEON and now under handling operation by the IRB-140.

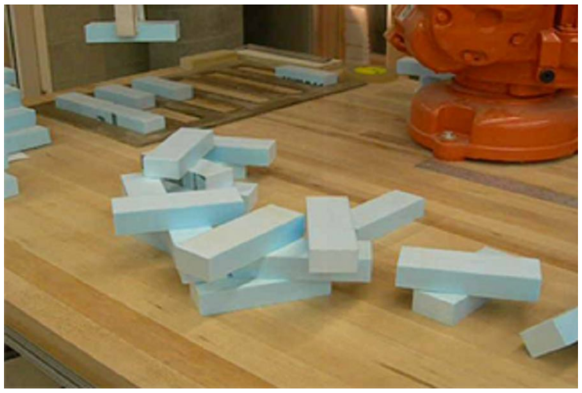


Figure 21. Curved wall, conversion from Adeon to physical creation by the IRB-140.

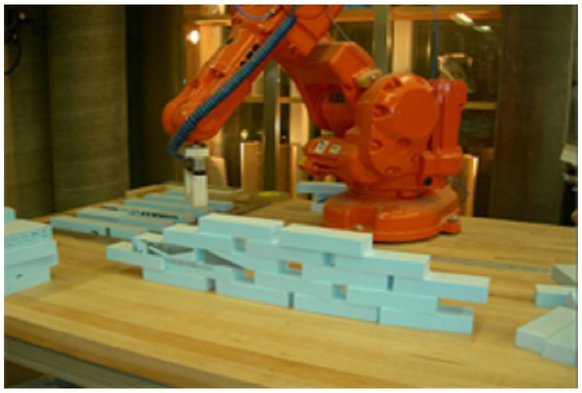


Figure 22. Vertical Wall configuration

Some of the problems encountered while using the IRB-140 is in its inability to detect whether or not a 4”x6” block has been stacked correctly on top of one another. The IRB-140, at its current state, does not recognize the design configuration, or how the weight of material interplays in the successful stacking of blocks. The IRB-140 used in this project performs routine tasks set by machine instructions. The next chapter will reflect on some of the issues and how to resolve some of the issues encountered during the experiment.

<u>Chapter Seven Recap</u>

Chapter Seven presents the thesis deliverable, the Adeon system, and demonstrates its use in a design scenario: the making of a curved and vertical wall using the IRB-140 robot. The chapter details how Adeon’s graphical user interface functions as a tool for bridging the gap between design, construction, and machine knowledge.

PART V A REFLECTION ON STRATEGIES

This final chapter reflects on the research and the strategies implemented into this thesis as well as to suggest implications of how Adeon can impact the way designers consult experts in the field, and how machines will change designers.

8 Design Consulting Redefined, Adeon in the Future

8.1 Adeon Revisited

This thesis began as an exploration in finding ways for which design tools can exhibit common knowledge that only a designer can understand from work and education experience. Adeon, at the present, suggests the possibility for implementing an architectural knowledgebase that would facilitate designers to make design choices based on the alternatives suggested by Adeon as an intelligent system. So far, Adeon can relate three areas of knowledge: Design, Construction, and Machine information. With the use of the Open Mind Common Sense database, users of Adeon can continually update relevant design information to reflect and associate with different design possibilities.

The future development of Adeon, will be that the system becomes the consultant that architects and designers refer and work with in design projects (Figure 23). Placed on a central server, designers can tap into resourceful information associated with the current context of their designs for production. The

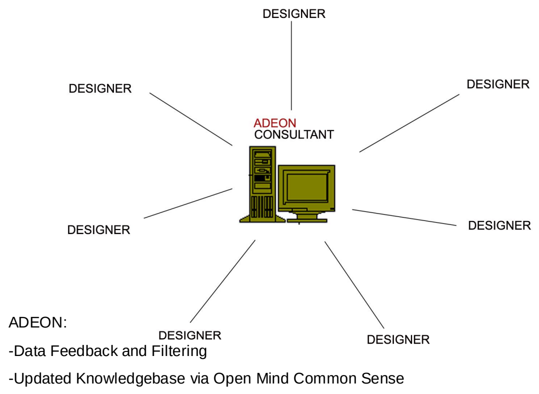
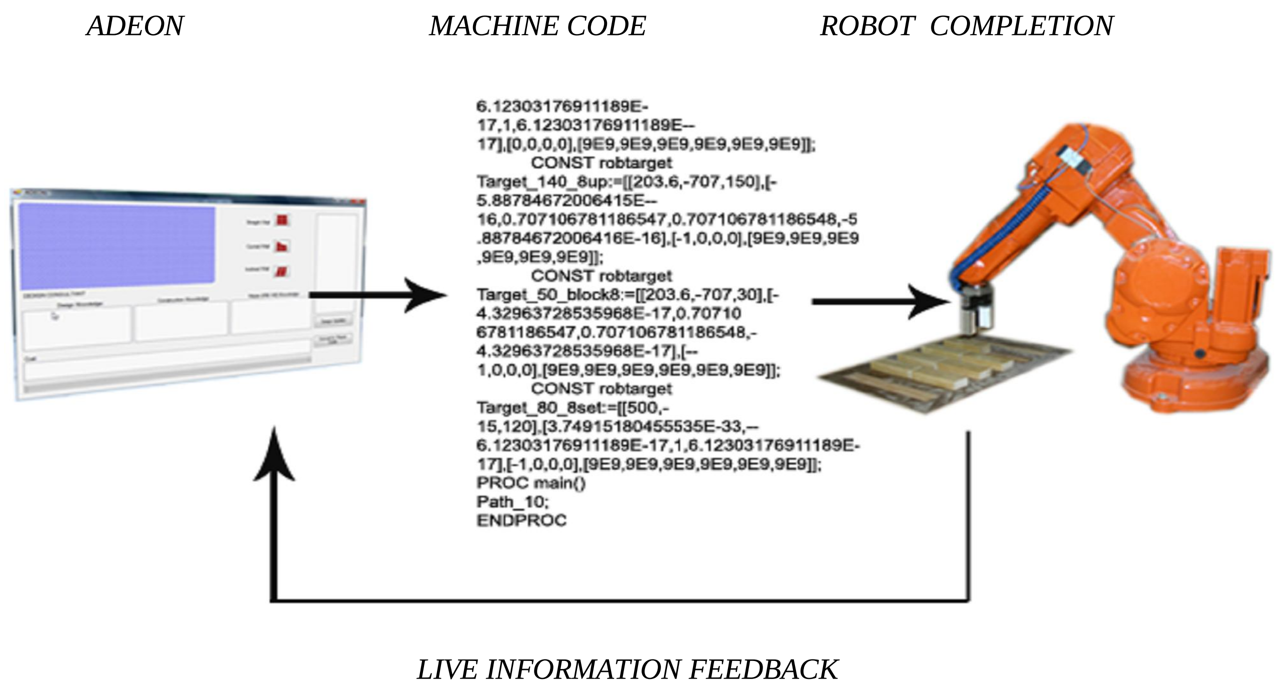

DESIGNER
DESIGNER
DESIGNER
ADEON
CONSULTANT
DESIGNER
DESIGNER
DESIGNER
DESIGNER
ADEON:
-Data Feedback and Filtering
-Updated Knowledgebase via Open Mind Common Sense

Figure 23. Adeon, a design consultant accessed by designers on a central server.

Forthcoming features of Adeon will see the implementation of additional data feedback and filtering of information at the user's discretion. S/he will be able to select the type of architectural knowledge to be shown and can input new knowledge into the Open Mind Common Sense database by directly interacting within Adeon’s environment.

8.2 Future Recommendations

The use of the IRB-140 together with Adeon, currently demonstrates a one-way relationship where the IRB-140 interacts with Adeon *after* a design has been drawn onto the editor for machine readable interpretation. A future implementation is to let Adeon and the IRB-140 to demonstrate a two-way relationship instead, where the IRB-140 interacts with three dimensional spatial design and can reflect/output its understanding of assembly and the particular idiosyncrasies that can occur during production (Figure 24). A two-way relationship between Adeon and the IRB-140 can be brought possible by implementing robot vision-based processing, where the IRB-140 can scan its environment during production assembly and recognize anomalous qualities about the artifact.



```
6.12303176911189E-
17,1,6.12303176911189E--
17],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget
Target_140_8up:=[[203.6,-707,150],[-
5.88784672006415E--
16,0.707106781186547,0.707106781186548,-5
.88784672006416E-16],[-1,0,0,0],[9E9,9E9,9E9
,9E9,9E9,9E9]];
        CONST robtarget
Target_50_block8:=[[203.6,-707,30],[-
4.32963728535968E-17,0.70710
6781186547,0.707106781186548,-
4.32963728535968E-17],[--
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget
Target_80_8set:=[[500,-
15,120],[3.74915180455535E-33,--
6.12303176911189E-17,1,6.12303176911189E-
17],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
PROC main()
Path_10;
ENDPROC
```

Figure 24. With vision-cam, the robot can deliver live information feedback to Adeon.

8.3 The Machines among Architects

The challenges designers face when using machines occur when tasks are not properly understood by the machine, or when a machine misinterprets a designer’s intentions. Currently, the means for remedying this type of encounter demands the user to do any of the following: summon technical support, resort to technical manuals and/or online support forums, or in some cases, hire a consultant. Whether or not this proves as a barrier to the progression and development of a project, machines require the capability to integrate and adapt itself into understanding common design practices; to collaborate as an active agent in various field processes. This thesis looks to make that possible, offering a contribution on how architects can relate differently with machines and extend to the initial formations for a collaborative approach. Our machines, as Rodney Brooks puts it, present themselves as an external reflection of the biological (Brooks 2002). If this is true, then our machines as design tools are an extension and expression of that biological nature in ultimate creativity.

Bibliography

Acocella, Alfonso 2006, *Stone Architecture: Ancient and Modern Construction Skills*, 1st edn, Rizzoli, New York.

Anderson, S (ed.) 2004, *Eladio Dieste: Innovation in Structural Art*. Princeton Architectural Press, New York.

Basic Brick Masonry Skills (videorecording) 1995, Milton Y, Haynes L, and Hull S, MWM Productions.

Brooks, R. 2002, *Flesh and Machines*. New York: Pantheon.

Casselman, Bill. *Latin Dictionary*, University of Washington. http://www.math.ubc.ca/~cass/frivs/latin/latin-dict-full.html. Viewed April 2008.

Computer Science: Reflections on the Field, Reflections from the Field. 2004. National Research Council of the National Academies. National Academies Press: Washington, D.C.

Gramazio, F and Kohler M 2008, Architecture and Digital Fabrication. www.dfab.arch.ethz.ch. Viewed on May 2008.

Crevier, D, 1993, *AI: The Tumultuous History of the Search for Artificial Intelligence,* BasicBooks, New York.

Grouting Masonry (videorecording) 1989, Wright W, Travis, L, Merrigan M, U.S.C. Teleproductions.

Hoang, Han 2005, *Automated Construction Technologies: Analyses and Future Development Strategies,* MIT, Cambridge.

Knight, T. and Stiny, G. 2001, Classical and Non-Classical Computation. *Arq*, 5, 369.

Lawson, Bryan 1980, *How Designers Think,* Architectural Press, Westfield.

Lobell, John 1979, *Between Silence and Light: Spirit in the Architecuture of Louis I. Kahn*. Shambhala, Boulder.

Minsky, Marvin 1988, *The Society of Mind.* Simon and Schuster, New York.

MIT Common Sense Computing Group. 2008. Open Mind Commons. http://commons.media.mit.edu:3000/. Viewed on February 2008.

MIT Common Sense Computing Group. ConceptNet http://web.media.mit.edu/~hugo/conceptnet/. Viewed on February 2008

Lieberman, H Liu, P Singh, B Barry. 2004, *Beating common sense into interactive applications,* AI Magazine 25(4): 63-76. AAAI Press.

Liu, H. and Singh, P. “ConceptNet: A Practical Commonsense Reasoning Toolkit”, *BT Technology Journal* 22(4). 2004. Kluwer Academic Publishers.

Mueller Erik T 2006, *Common Sense Reasoning*. Morgan: New York, 2006.

Poole, D, Boerlaye, B, Crowley, M, “On the Use of Possibilistic Bases for Local Computations in Product-Based Possibilites Networks”, Conference proceedings *Advances in Artificial Intelligence: 20th Century Conference of the Canadian Society for Computational Studies of Intelligence 2007*. Proceedings May 28-30 2007.

Rosenbloom, P, Laird, J, & Newell, A. 1993, *The Soar Papers: Research on Integrated Intelligence.* MIT Press, Cambridge.

Schon, Donald 1987, *Educating the Reflective Practitioner*, Josey-Bass, San Francisco.

Skibniewski, M and Woolridge, 1992, “Robotic Materials handling for Automated Building Construction Technology” in *Automation in Construction*, Elsevier Science Publishers, 1, pp. 251-266.

Simon, H. A. 1965, *The Shape of Automation,* Harper & Row, New York.

Sutherland, I 1975, “Structure in Drawings and the Hidden-Surface Problem”, in *Reflections on Computer Aids to Design and Architecture*, N. Negroponte, (ed.) New York

Schodek, D, Bechthold M, Griggs, K, Kao K, Steinberg M 2005, *Digital Design and Manufacturing: CAD/CAM Applications in Architecture and Design,* Wiley, Hoboken.

Villalon, R. and Lobel, J. 2007, *Materializing Design: Contemporary Issues in the Use of CAD/CAM Technology in the Architectural Design and Fabrication Process.* ASCAAD International Conference on Nov. 28-30. Alexandria, Egypt.

Wiggins, Glen 1997, *Louis I. Kahn: The Library at Phillips Exeter Academy.* Von Nostrand Reinhold, New York.

APPENDIX A: TABLE OF FIGURES

(All images are properties of their rightful owners and subject to copyright)

APPENDIX B

CASE STUDY ANALYSIS NOTES: BASIC BRICK MASONRY SKILLS

NOTES from: Basic Brick Masonry Skills (videorecording) 1995, Milton Y, Haynes L, and Hull S, MWM Productions.

The BIA, Brick Institute of America

Terms, Tools and Techniques of the brick mason

Basic Language of the Brick Mason:

Aggregate: refers to the mineral materials sand and gravel to make mortar and concrete

Bat: brick piece, half a piece or smaller in size

Bed Joint

Bond: mortar and masonry adhesion

Burning the Joint: tooling the joint after it has set

Buttering: put mortar on brick

Closer Brick: Last brick laid on the course

Course: a continuous row of horizontal masonry units

Crowding the line: laying of bricks next to line

Dry Bonding, lining brick without mortar

Fat Mortar: sticky adhesive mortar

Furrowing: shallow indentions cut mortar

Head Joint, vertical mortar joint

Lead, a guide in leaning a mortar

Lean Mortar: not sticky mortar

Line: string stretched from lead to lead on top edge of the brick course

Mud: common name of mortar

APPENDIX B

Ranging the Corner: alighning to corners

Rowlock: msu set on its side

Slack to the line: far from the line

Soldier: face side facing out

Tempering: adding water and mixing mortar to proper consistency

Tooling: act of compressing, shaping, or finishing mortar joitns with special tool

Trig: Device used to support a line at the center of the wall

Wythe: a vertical wall one unit thick

The Brick Mason's Trowel

Types and Uses of Trowels:

The narrow London

The wide London

The Philadelphia

Most commonly used is Narrow London

Wide London and Philadelphia used in concrete block work

Mixing Mortar: mix mortar at the job site, know how to mix by hand and machine. Gas powered or electric mortar mixer.

APPENDIX B

Hand Mixing Mortar:

Assemble mixing tools – masonry cement

-sand

-clean water

-mortar box

mortar hoe

square point shovel

bucket or container

-box 1x1x1 to measure one cubic foot of sand

-Type ‘N’ Mortar – 1 cu. Ft. cement per 3 cu. Ft. sand

-7 shovelsful sand = 1 cu. Ft.

Place bag of mortar in mortar box, divide in half, take other out, spread ½ bag of mortar mix over sand, spread other sand on top of mortar cement, take second half of mortar and spread over layer of sand

With mortar hoe, thoroughly mix dry ingredients

Pull ingredients to center of mortar box and make depression at center of mixture

Pour water into the depression

Using the hoe, mix ingredients

Let mixture stand for 5 minutes. Then mix again with hoe

Clean tools

APPENDIX B

Picking up Mortar

Cupping method, used with mortar board, cutting from main pile, and shaping it to long tapered form approx. length of the trowel blade

Clock method

Stringing Motar: unload trowel evenly. Length of three bricks, 2 feet long. Cut any excess mortar, don’t put back on board

Furrowing brick: pull trowel across mortar

Use excess mortar to fill gaps

Stringing mortar over a line

1. load trowel with mortar
2. use sweeping motion to put mortar length of 3 bricks
3. cut off excess mortar to fill gaps
4. furrow motar and use excess to end of string

practice placing brick to the line

5. leave 1/16” space between the line and the brick
6. cut excess mortar from the wall and make sure aligned to brick below it
7. butter the next brick

APPENDIX B

Grouting Brick Masonry

NOTES from: Grouting Masonry (videorecording) 1989, Wright W, Travis, L, Merrigan M, U.S.C. Teleproductions.

Brick Institute of Masonry
Masonry institute of America
National Concrete Masonry Association

-Masonry, endured by time. Monumental structures used in the past.

-Unreinforced masonry need damping properties, ductility, lateral strength

-Reinforced masonry came because of seismic, wind, and earth pressures and heavy loads

-Includes steel reinforcement and grout

-Reinforced masonry allow for thinner walls to be constructed, longer spans, and taller and stronger walls.

-Grout excellence in resistance to sound transmission

-Grout increases fire resistance rating, a solid grouted wall 6-8 inches thick has a 4 hour fire rating.

-Grout is not mortar or not concrete. It's a cementious masonry unique to reinforce masonry.

-Sand, Portland cement, water, and pea gravel are the basic ingredients of grout

2 types of grout used in masonry construction:

-Fine Grout – 1 part Portland Cement, 2 ¼ to 4 parts sand and water

-Coarse Grout – 1 part Portland cement, 2 ¼ to 3 parts sand and water, 1 to 2 parts pea gravel.

-Fine grout is used in small and congested areas because of steel reinforcement.

-Coarse grout is used to the width of the grout space or where less congested.

-Slump test, 8-10 inches

APPENDIX B

-Masonry may be fully grouted once masonry and steel reinforcement are put in place. Compeletely filled with grout inside.

-Hollow unit masonry walls can be partially grouted

-High Lift grouting, when walls exceed 5 feet. Need clean out spaces to blow excess dry mortar

-Low Lift grouting, without clean out spaces

-Grout bonds creates a homogenous system, consolidated

-Building code, cold weather construction

-Both Fine and Coarse grout must have a minimum of 2000 psi compressive strength to ensure bounding of grout to reinforced steel.

APPENDIX C: CASE STUDY ANALYSIS PHILIPS EXETER LIBRARY, EXTRACTION FOR THE OMCS DATABASE

LIBRARY AT PHILIPS EXETER ACADEMY
ARCHITECT: LOUIS KAHN
TRADITIONAL BRICK ARCHITECTURE

Design Knowledge **IRB-140 Knowledge**	**Construction Knowledge**
Louis Kahn **can** design brick walls. (CapableOf) Brick **is a type of** load-bearing material. (IsA) Philips Exeter Library **is** designed in brick to keep with the traditional New England campus. (HasProperty) Philips Exeter Library **is a** brick building designed to keep with the traditional New England campus. (IsA) The Philips Exeter Library **is the** building designed by architect, Louis Kahn. (DefinedAs) Philips Exeter Library's brick piers **are** wider nearer to the ground where their loads are greater. (HasProperty) The brick wall **is** interspersed with apertures for windows. (HasProperty) **You would put** openings in a brick walls **because you want** a door or window. (MotivatedByGoal)	**An** exterior brick wall **wants to** be designed for stopping water ingress from melted snow, rain, and ice. (Desires) **Something you need to do before** you lay bricks **is** to mix mortar. (HasPrerequisite) Hand mixing mortar **requires** masonry cement, sand, clean water, mortar box, mortar hoe, square point shovel, and a bucket. (HasPrerequisite) Mortar **is created by** mixing masonry cement, sand, and clean water. (CreatedBy) **The first thing you do when you** prepare mortar **is** place the bag of mortar inside a mortar box. (HasFirstSubevent) **One of the things you do when you** have mortar in a mortar box **is** to divide the mix in half. (HasSubevent) **One of the things you do when you** divide mortar mix in half **is** spread sand on top of the mortar cement. (HasSubevent)

Design Knowledge	Construction Knowledge
A window's width and height **is the** maximum distance of brick piers. (DefinedAs) **The effect of** placing a window in a brick wall **is that** the width and height are determined by the maximum distance of the brick piers. (Causes) The brick wall at Philips Exeter Library **is used for** load-bearing. (UsedFor) Brick walls **can be used for** as a veneer instead of as a load-bearing material. (UsedFor) Brick walls **can be used for** as a skeletal structure instead of as a load-bearing material. (UsedFor) **You would** use brick as a veneer **because you want** to create an aesthetic effect. (MotivatedByGoal) **You would want to** use brick as a building material **because you want** the aesthetic of having the forces of gravity and weight to be evident in the construction. (MotivatedByGoal)	**One of the things you do when** you mix dry mortar **is** place a depression at the center. (HasSubevent) **One of the things you do when** you mix dry mortar **is** pour water. (HasSubevent) **One of the things you do when** done mixing mortar **is** let mixture stand for five minutes. (HasSubevent) **The first thing you do when you** lay brick **is** load the trowel with mortar. (HasFirstSubevent) **One of the things you do when you** lay mortar on a brick wall **is** to place mortar three brick lengths long. (HasSubevent) **One of the things you do when you** place mortar three brick lengths long **is** cut off excess mortar to fill brick gaps. (HasSubevent) **One of the things you do when you cut off excess mortar** to fill brick gaps **is** to furrow the mortar and use as excess to end of brick string. (HasSubevent)

Design Knowledge IRB-140 Knowledge	Construction Knowledge
The brick piers of Philips Exeter Library **are used for** accommodating the building's load. (UsedFor) The brick piers of Philips Exeter Library **are** progressively narrower at the base. (DefinedAs) **You would** extend a brick exterior wall of Philips Exeter Library **because you want** to increase its sense of being a screen. (MotivatedByGoal) **You would** design Philips Exeter Library with warm tones and rich texture **because you want** an inviting quality provided by the natural light. (MotivatedByGoal)	**One of the things you do when you** place brick to the line **is** leave 1/16" space between the line and the brick. (HasSubevent) **One of the things you do when you** leave 1/16" space between the line and brick **is** cut excess mortar from the wall and make sure the brick is aligned to the other brick below it. (HasSubevent) **One of the things you do when you** align brick to the wall **is** to butter the next brick. (HasSubevent) **An** exterior brick wall **wants to** be designed for accommodating differential movement. (Desires) Brick walls **are the** modular building blocks bonded together with mortar to form walls that are durable, fire-resistant, and structurally efficient in compression. (DefinedAs)

Construction Knowledge

Masonry **is the** building with units of various natural or manufactured products, such as brick, stone, or concrete block, usually with the use of mortar as a bonding agent. (DefinedAs)

A bat **is the** half piece of brick. (DefinedAs)

A bond **is the** mortar and masonry adhesion (DefinedAs)

Burning the Joint **is the** tooling of the joint
after it has set. (DefinedAs)

A Line **is the** string stretched from lead to lead on top edge of the brick course. (DefinedAs)

Mud **is the** common name for mortar. (DefinedAs)

Ranging the Corner **is the** aligning of brick to corners. (DefinedAs)

Tempering **is the** adding of water and mixing of mortar to proper consistency. (DefinedAs)

Tooling **is the** act of compressing, shaping, or finishing mortar joints with a special tool. (DefinedAs)

A Trig **is the** device used to support a line at the center of a brick wall. (DefinedAs)

A Wythe **is the** vertical wall one unit thick. (DefinedAs)

Buttering **is the** placing of mortar on brick. (DefinedAs)

Closer Brick **is the** last brick laid on the course. (DefinedAs)

A course **is the** continuous row of horizontal masonry units. (DefinedAs)

Crowding the Line **is the** laying of bricks next to a line. (DefinedAs)

Dry Bonding **is the** lining of brick without mortar. (DefinedAs)

Fat Mortar **is the** sticky adhesive mortar. (DefinedAs)

Construction Knowledge

Furrowing **is the** shallow indentations cut into mortar. (DefinedAs)

A Head Joint **is a** vertical mortar joint. (DefinedAs)

A lead **is a** guide in leaning mortar. (DefinedAs)

Lean Mortar **is the** term to describe non-sticky mortar. (DefinedAs)

APPENDIX D: CASE STUDY ANALYSIS NOTES ON ATLANTIDA

Untraditional Curved walls described:

Case Study #1: Eladio Dieste's Atlantida (1958-50)

-Atlantida has sinusoidal walls and roof, before end walls were added.

-Atlantida is a sinusoidal wall.

-Atlantida is made up of cylindrical barrel shells.

-In their curved direction, these act in compression, whereas in their longitudinal axes, they act as beams.

-The shape of Atlantida's walls are generated by straight line segments whose ends were translated along sinusoidal paths.

-"The wall is a seamless whole, not separating the mechanical to the philosophical" (Anderson 2004).

-The vault is uniformly loaded in horizontal projection, therefore the funicular shape is a parabola.

-A funicular is a shape subjected to a given pattern of loads and its inversion assumes the same pattern of loads assumed by a flexible string or cable.

-Atlantida's masonry structure is subject to bending.

-There are non-loadbearing elements added or needed to sculpt the interior space.

-The wall is dancing.

-The wall presents fluidity in the landscape.

-The wall is fluid.

-A dynamic partition

-The surface as barrier.

-A non-static wall

-The bricks are dance in sync with the landscape.

-The bricks flow smoothly as an exterior shell.

APPENDIX D

Consequences to Construction:

-Reinforced brick work

-Reinforced brick work consists of steel-reinforcement .

-Steel reinforcement bars are manufactured by hot-roll process as round rods with lugs, or deformations which inhibit longitudinal movement of the bar in the surrounding "concrete".

-Steel reinforcement are in sizes #3 - #8, the numbers are the number of eights of an inch in the nominal diameter of bars.

-In some instances, a fiber-reinforced plastic (FRP) rebar is used for highly specialized concrete reinforcement because of its high tensile strength and light weight, corrosion resistance, and dielectric (nonconductive) properties.

-FRP rebars are manufactured in the same sizes as steel rebars and also have deformations on the surface.

APPENDIX E: NOTES ON LOUIS KAHN'S PHILLIPS EXETER ACADEMY LIBRARY

Traditional Brick Wall Enclosure

-The brick piers between the windows become wider nearer to the ground where their loads are greater.

-The brick wall is interspersed with apertures for windows.

-The windows width and height are determined by the maximum distance of the brick piers.

-The brick is treated as a load-bearing material not as a veneer or skeletal structure

-The forces of gravity and weight of the masonry is evident in the construction.

-The brick piers are wider at the base than near the roof to accommodate load.

-The brick piers are progressively narrower towards the base.

-The exterior wall extends "to increase its sense of being a screen."

-The walls have warm tones and rich texture which add an inviting quality provided by the natural light.

-The walls do not bring each side to a traditional ninety-degree corner.

-The wall adds a screen-like quality to the façade.

Kahn: "If you talk to a brick and ask it what it likes, it'll say it likes an arch. And you say to it, look arches are expensive and you can always use a concrete lintel to take the place of an arch. And the brick says, I know it's expensive and I'm afraid it probably cannot be built these days, but if you ask me what I *like* its still an arch."

(Kahn, Louis. "An Architect Speaks his Mind." *House and Garden* 142, no. 4 (October 1972).

-Kahn "masonry was 'dancing like an angel' at the top and 'grunting' below".

-Kahn refers to the 'grunting below' quality as the weight of the building at the typical grade-level corner is almost palpable.

APPENDIX E

Consequences to Construction

-The masonry walls consist of modular building blocks bonded together with mortar to form walls that are durable, fire-resistant, and structurally efficient in compression.

-Load-bearing brick construction

Typical Solutions for Constructing with brick:

-Constructing a concrete or steel frame and facing that frame with individual bricks.

-Build large brick panels off site or on the ground and then raise them on the structural frame.

Construction for Masonry Bearing Walls

Foundation – concrete –formwork, type of concrete, concrete needs rebar, type of rebar, set time, slump test, rebar steel ties, set.

APPENDIX F: EIGHTEEN RELATIONS FOR OMCS DATABASE

1. CausesDesire – What does it make you want to do?
____ would make you want to _____.

2. MadeOf – What is it made of?
____ is made of _____.

3. IsA – What kind of thing is it?
___ is a kind of _____.
___ is a type of _____.
___ are a kind of ____.

4. UsedFor – What do you use it for?
___ is used for ____.
___ can be used for ____.
___ is for ___.

5. CapableOf – What can it do?
___ can ____.

6. Part of – What is it part of?
___ is part of ____.

7. DefinedAs – How do you define it?
____ is the _____.

8. CreatedBy – How do you bring it into existence?
____ is created by _____.

9. HasFirstSubevent – What do you do first to accomplish it?
The first thing you do when you ____ is ____.

10. HasLastSubevent – What do you do last to accomplish it?
The last thing you do when you ____ is _____.

11. HasPrerequisite – What do you need to do first?
Something you need to do before you ____ is _____.
____ requires _____.
If you want to _____ then you should _____.

12. AtLocation – Where would you find it?
You are likely to find ____ in ______.
You are likely to find ____ on _____.

You are likely to find ____ near ____.
You are likely to find ____ around ____.

13. MotivatedByGoal – Why would you do it?
You would ____ because you want ____.

14. Desires – What does it want?
_____ wants to ______.
An _____ wants to _____.
A ____ wants to _____.

15. Causes – What does it make happen?
The effect of _____ is ______.
The effect of _____ is that _____.

16. HasSubevent – What do you do to accomplish it?
One of the things you do when you ____ is _____.
Something that might happen when you ____ is ____.
Something that might happen while ____ is ____.

17. HasProperty – What properties does it have?
___ is generally ____.
___ are ___.
___ is ___.

18. ReceivesAction – What can you do to it?
___ can be ___.
___ is ____.

APPENDIX G: RAPID SOURCE CODE

```
MODULE Module1
!150 is the apex
        CONST robtarget Target_80_1up:=[[3.6,-550,150],[-5.88784672006415E-
16,0.707106781186547,0.707106781186548,-5.88784672006416E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
       PERS tooldata RAD_AL2100:=[TRUE,[[0,0,121],[1,0,0,0]],[0.17,[0,0,1],[1,0,0,0],0,0,0]];
!target10 = grab block
!target_set describes location of block, dropped.
        CONST robtarget Target_10_block1:=[[3.6,-550,30],[4.32963728535968E-
17,0.707106781186547,0.707106781186548,4.32963728535968E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_80_apex:=[[500,0,200],[-3.74915180455534E-33,-
6.12303176911189E-17,1,-6.12303176911188E-17],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_80_1set:=[[480,15,30],[3.74915180455535E-33,-
6.12303176911189E-17,1,6.12303176911189E-17],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_80_2up:=[[103.6,-550,150],[-5.88784672006415E-
16,0.707106781186547,0.707106781186548,-5.88784672006416E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_10_block2:=[[103.6,-550,30],[8.71576399210524E-
33,0.707106781186547,0.707106781186548,8.65927457071935E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_90_2set:=[[500,-15,30],[3.74915180455535E-33,-
6.12303176911189E-17,1,6.12303176911189E-17],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_90_3up:=[[203.6,-550,150],[-5.88784672006415E-
16,0.707106781186547,0.707106781186548,-5.88784672006416E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
```

APPENDIX G

```
        CONST robtarget Target_10_block3:=[[203.6,-550,30],[8.71576399210524E-
33,0.707106781186547,0.707106781186548,8.65927457071935E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_80_3set:=[[480,15,60],[3.74915180455535E-33,-
6.12303176911189E-17,1,6.12303176911189E-17],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_100_4up:=[[303.6,-550,150],[-5.88784672006415E-
16,0.707106781186547,0.707106781186548,-5.88784672006416E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_10_block4:=[[303.6,-550,30],[-4.32963728535968E-
17,0.707106781186547,0.707106781186548,-4.32963728535968E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_90_4set:=[[500,-15,60],[3.74915180455535E-33,-
6.12303176911189E-17,1,6.12303176911189E-17],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_110_5up:=[[403.6,-550,150],[-5.88784672006415E-
16,0.707106781186547,0.707106781186548,-5.88784672006416E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_20_block5:=[[403.6,-550,30],[-4.32963728535968E-
17,0.707106781186547,0.707106781186548,-4.32963728535968E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_80_5set:=[[480,15,90],[3.74915180455535E-33,-
6.12303176911189E-17,1,6.12303176911189E-17],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_120_6up:=[[3.6,-707,150],[-5.88784672006415E-
16,0.707106781186547,0.707106781186548,-5.88784672006416E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
        CONST robtarget Target_30_block6:=[[3.6,-707,30],[-4.32963728535968E-
17,0.707106781186547,0.707106781186548,-4.32963728535968E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
```

APPENDIX G

```
	CONST robtarget Target_90_6set:=[[500,-15,90],[3.74915180455535E-33,-
6.12303176911189E-17,1,6.12303176911189E-17],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST robtarget Target_130_7up:=[[103.6,-707,150],[-5.88784672006415E-
16,0.707106781186547,0.707106781186548,-5.88784672006416E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST robtarget Target_40_block7:=[[103.6,-707,30],[-4.32963728535968E-
17,0.707106781186547,0.707106781186548,-4.32963728535968E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST robtarget Target_80_7set:=[[480,15,120],[3.74915180455535E-33,-
6.12303176911189E-17,1,6.12303176911189E-17],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST robtarget Target_140_8up:=[[203.6,-707,150],[-5.88784672006415E-
16,0.707106781186547,0.707106781186548,-5.88784672006416E-16],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST robtarget Target_50_block8:=[[203.6,-707,30],[-4.32963728535968E-
17,0.707106781186547,0.707106781186548,-4.32963728535968E-17],[-
1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
	CONST robtarget Target_80_8set:=[[500,-15,120],[3.74915180455535E-33,-
6.12303176911189E-17,1,6.12303176911189E-17],[-1,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];
PROC main()
Path_10;
ENDPROC


PROC Path_10()
!SetDO, open gripper
	SetDO DO10_1,1;
	MoveJ Target_80_1up,v100,z100,RAD_AL2100\WObj:=wobj0;
```

APPENDIX G

```
	MoveL Target_10_block1,v100,fine,RAD_AL2100\WObj:=wobj0;
!Reset, close gripper
	Reset DO10_1;
	MoveJ Target_80_1up,v50,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_80_1set,v100,z100,RAD_AL2100\WObj:=wobj0;
	WaitTime 2;
	SetDO DO10_1,1;
	MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_80_2up,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_10_block2,v100,z100,RAD_AL2100\WObj:=wobj0;
	!PickBlock;
	WaitTime 2;
	Reset DO10_1;
	MoveJ Target_80_2up,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_90_2set,v100,z100,RAD_AL2100\WObj:=wobj0;
	WaitTime 1;
	SetDO DO10_1,1;
	MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_90_3up,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_10_block3,v100,z100,RAD_AL2100\WObj:=wobj0;
	WaitTime 2;
	Reset DO10_1;
	MoveJ Target_90_3up,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
	MoveJ Target_80_3set,v100,z100,RAD_AL2100\WObj:=wobj0;
```

```
WaitTime 2;
SetDO DO10_1,1;
MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_100_4up,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_10_block4,v100,z100,RAD_AL2100\WObj:=wobj0;
WaitTime 2;
Reset DO10_1;
MoveJ Target_100_4up,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_90_4set,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_110_5up,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_20_block5,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_110_5up,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_80_5set,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_120_6up,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_30_block6,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_120_6up,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_90_6set,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_130_7up,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_40_block7,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_130_7up,v100,z100,RAD_AL2100\WObj:=wobj0;
MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
```

```
        MoveJ Target_80_7set,v100,z100,RAD_AL2100\WObj:=wobj0;
        MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
        MoveJ Target_140_8up,v100,z100,RAD_AL2100\WObj:=wobj0;
        MoveJ Target_50_block8,v100,z100,RAD_AL2100\WObj:=wobj0;
        MoveJ Target_140_8up,v100,z100,RAD_AL2100\WObj:=wobj0;
        MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
        MoveJ Target_80_8set,v100,z100,RAD_AL2100\WObj:=wobj0;
        MoveJ Target_80_apex,v100,z100,RAD_AL2100\WObj:=wobj0;
ENDPROC
PROC Path_20()
ENDPROC

ENDMODULE
```

APPENDIX H: System Source Code

//Map Source Code

```
// © 2008 Rachelle Villalon. All rights reserved.

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;
using Study10.GraphicComponents;

namespace Study10.Maps
{
    public class Map: GraphicComponents.Idraw
    {
        //members
        List<Cell> _cells = new List<Cell>();
        List<bool> _flags = new List<bool>();

        //map info
        //geometry data
        Point _location;
        Size _size;
        Size _cellSize;
        double _offset = 0.5;

        public double OFFSET
        {
            get { return _offset; }
            set { _offset = value; }
        }
        public Size SIZE
        {
            get { return _size; }
            set
            {

                _size = value;

            }
        }
        public Point LOCATION
        {
            get
            {
                return _location;
            }
            set
            {
                _location = value;
            }
        }



        public Size CELL_SIZE
        {
            get { return _cellSize; }
            set
            {
                _cellSize = value;
                foreach (Cell c in _cells)
                {
                    c.SIZE = value;
                }
            }
        }

        public Map()
        {
            LOCATION = new Point();
            SIZE = new Size(10, 10);
            CELL_SIZE = new Size(4, 2);
            init();
        }

        public void init()
        {
            int length = SIZE.Height * SIZE.Width;
            for (int i = 0; i < length; i++)
            {
                _cells.Add(new Cell());
            }
            for (int i = 0; i < SIZE.Width; i++)
            {
                for (int j = 0; j < SIZE.Height; j++)
                {
                    set(new Cell(), new Point(i, j));
                }
            }

        }

        public void set(Cell cell, Point index)
        {
            int lIndex = convertIndex(index);

            // chekc if index out of bound
            if (lIndex >= _cells.Count)return;

            cell.INDEX = index;
            double cellX = (double)(LOCATION.X + (_cellSize.Width * index.X));
            double cellY = (double)( LOCATION.Y + (_cellSize.Height *
index.Y));

            //offset the even rows
            if ((index.Y % 2) == 0)
            {
                Console.WriteLine("cellY=" + cellY);
                cellX += (_cellSize.Width * _offset);
            }

            cell.LOCATION = new Point((int)cellX, (int)cellY);
            cell.SIZE = _cellSize;
            _cells[lIndex] = cell;

        }

        public void add(Cell cell)
        {

            _cells.Add(cell);

        }
        public int getBrickCount()
        {
            int count = 0;
            foreach (Cell c in _cells)
            {
                if (c is Brick) count++;
            }
            return count;
        }

        //convert 2D array index to linear array index
        private int convertIndex(Point p)
        {
            return p.X + (_size.Width * p.Y);
        }

        public void draw(Graphics g)
        {
            foreach (Cell c in _cells)
            {
                c.draw(g);
            }
        }
        public static Point convertMouseInputToLoc(int x, int y, Map map)
        {
            int nx = (x * map.CELL_SIZE.Width) + map.LOCATION.X;
            int ny = (y * map.CELL_SIZE.Height) + map.LOCATION.Y;
            return new Point(nx, ny);
        }
    }
}
```

//FORM SOURCE CODE

```
// © 2008 Rachelle Villalon. All rights reserved.
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using Study10.Maps;
using Study10.GraphicsData;
using Study10.GraphicComponents;
using System.Runtime.InteropServices;
using System.IO;

namespace Study10
{
    public partial class Form1 : Form
    {
        Map _map;

        //added for code to show within label
        Label labl;

        //mouse flags
        bool _LBD = false; //left button down

        public Form1()
        {
            InitializeComponent();
            initializeMap();
            this.DoubleBuffered = true;
        }


        public void initializeMap()
        {
            _map = new Map();
            _map.SIZE = new Size(50, 50);           //was size (50, 10)
            _map.CELL_SIZE = new Size(8, 4);
            _map.LOCATION = new Point(10,10);
            _map.init();
            listBox1.Items.Add("total brick");
            listBox2.Items.Add(".");
            listBox3.Items.Add("Design Knowledge"); //added
            listBox4.Items.Add("test"); //added 5/15
            listBox5.Items.Add(".");
            Data.add(_map);
        }

        public Map getMap()
        {
            return _map;
        }

        protected override void OnPaint(PaintEventArgs e)
        {
            base.OnPaint(e);
            Graphics g = e.Graphics;
            Data.draw(e.Graphics);
            update();

            this.Invalidate();
        }


        public void update()
        {
            decimal cost = 0.39M;
            decimal totalcost;
            listBox1.Items[0]=("total brick=" + _map.getBrickCount());
            listBox1.Invalidate();

            //listbox5 = ROBOT KNOWLEDGE
            //listbox4 = CONSTRUCTION KNOWLEDGE
            //listbox2 = DESIGN KNOWLEDGE
            //listbox3 = COST

            if (_map.getBrickCount() < 1000)
            {
                //COST
                //listBox3.Items[0]=("$0.39 per brick unit");
                totalcost = _map.getBrickCount() * cost;
                listBox3.Items[0]=("total cost in units per brick= $" +
totalcost);
            }
            //CONSTRUCTION KNOWLEDGE
            if (_map.getBrickCount() < 100)
            {
                listBox4.Items[0] = (".");
            }

            else if (_map.getBrickCount() > 130) //FOR CURVED WALL IN
TRANSLATION
```

```
            {
                listBox4.Items[0] = ("In some instances, a fiber-reinforced
(FRP) rebar is used for highly specialized brick work because of its high
tensile strength and light weight, corrosion resistance, and dielectric
(nonconductive) properties");
            }
            else
            {
                listBox4.Items[0] = ("Loadbearing walls should be aligned
consistently from floor to floor and should be continuous from the roof to
the building foundation.");
            }




            //DESIGN KNOWLEDGE
            if (_map.getBrickCount() < 100)
            {
                listBox2.Items[0] = (".");
            }
            else if (_map.getBrickCount() > 130) //Curved wall translated
            {
                listBox2.Items[0] = ("A curved wall presents fluidity in the
landscape. An example of this is Atlantida by Eladio Dieste, where walls are
generated by straight line segments whose ends were translated along
sinusoidal paths.");
            }
            else
            {
                listBox2.Items[0] = ("You would want to use brick as a
building material because you want the aesthetic of having the forces of
gravity and weight to be evident in the construction");
            }

            //ROBOT KNOWLEDGE
            if (_map.getBrickCount() < 100)
            {
                listBox5.Items[0] = (".");
            }
            else if (_map.getBrickCount() > 130)
            {
                listBox5.Items[0] = ("IRB 140 can accomodate the design with
supplementary load at 32 inches reach of the 5th axis." );
            }

            else
            {
                listBox5.Items[0] = ("IRB 140 can accomodate the design
within 50 to -230 degrees range of movement.");
```

```
            }

            return;



        }

        private void Form1_MouseDown(object sender, MouseEventArgs e)
        {
            _LBD = true;

            //Console.Out.WriteLine(" button pressed ");
        }

        private void Form1_MouseUp(object sender, MouseEventArgs e)
        {
            _LBD = false;
        }

        private void Form1_MouseMove(object sender, MouseEventArgs e)
        {
            if (_LBD)
            {

                int x = (int)((double)(e.X - _map.LOCATION.X) /
_map.CELL_SIZE.Width);
                int y = (int)((double)(e.Y - _map.LOCATION.Y) /
_map.CELL_SIZE.Height)-1;
                Point loc=new Point(x,y);
                Console.Out.WriteLine( loc.ToString());
                _map.set(new Brick(), loc);
                Console.Out.WriteLine(" dragging ");
            }
            //Console.Out.WriteLine(" moving");
        }


        private void Form1_KeyDown(object sender, KeyEventArgs e)
        {

        }

        private void listBox2_SelectedIndexChanged(object sender, EventArgs e)
        {

        }

        private void label2_Click(object sender, EventArgs e)
        {
        }


        private void RobConvertCode_Click(object sender, EventArgs e)
        {


            for (int i = progressBar1.Minimum; i <= progressBar1.Maximum; i++)
            {
                progressBar1.PerformStep();

            }

            OpenFileDialog fdlg = new OpenFileDialog();
            TextBox textBox1 = new TextBox();

            fdlg.Title = "IRB-140 RAPID CODE";
            fdlg.InitialDirectory = @"c:\Robot\";
            fdlg.Filter = "All files(*.*) |*.*|All files (*.*)|*.*";
            fdlg.FilterIndex = 2;
            fdlg.RestoreDirectory = true;
            if (fdlg.ShowDialog() == DialogResult.OK)
            {

                textBox1.Text = fdlg.FileName;
            }
```

APPENDIX H

//DATA SOURCE CODE

```
// © 2008 Rachelle Villalon. All rights reserved.

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using Study10.GraphicComponents;
using System.Drawing;

namespace Study10.GraphicsData
{
    public class Data
    {
        public static List<Idraw> _drawables = new List<Idraw>();

        public static void add(Idraw d)
        {
            _drawables.Add(d);
        }

        public static void remove(Idraw d)
        {
            _drawables.Remove(d);
        }

        public static void draw(Graphics g)
        {
            foreach(Idraw d in _drawables)
            {
                d.draw(g);
            }
        }
    }
}
```

APPENDIX H

//CELL SOURCE CODE

```
// © 2008 Rachelle Villalon. All rights reserved.
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;
using Study10.GraphicComponents;

namespace Study10.Maps
{
    public class Cell : Idraw
    {
        //geometry data
        Point _location;
        Size _size;
        Point _index;

        public Point INDEX
        {
            get
            {
                return _index;
            }
            set
            {
                _index = value;
            }
        }
        public Size SIZE
        {
            get { return _size; }
            set { _size = value; }
        }
        public Point LOCATION
        {
            get
            {
                return _location;
            }
            set
            {
                _location = value;
            }
        }

        public virtual void draw(Graphics g)
        {
            g.DrawRectangle(Pens.Blue, new Rectangle(LOCATION, SIZE));
        }


      }
}
```

//DRAW SOURCE CODE

```
// © 2008 Rachelle Villalon. All rights reserved.

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;

namespace Study10.GraphicComponents
{
    public interface Idraw
    {
        void draw(Graphics g);

    }
}
```

APPENDIX H

//BRICK SOURCE CODE

```
// © 2008 Rachelle Villalon. All rights reserved.

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Drawing;
using Study10.Maps;

namespace Study10.GraphicComponents
{
    public class Brick: Cell
    {


        //graphics styles
        Pen _pen = Pens.Red;

        public Brick()
        {
            LOCATION = new Point();
            SIZE = new Size(2, 1);
        }
        public Brick(Point loc, Size size)
        {
            LOCATION = loc;
            SIZE = size;
        }

       public override void  draw(Graphics g)
       {
             base.draw(g);
           g.DrawRectangle(_pen, new Rectangle(LOCATION, SIZE));

       }


    }
}
```

APPENDIX H

//MAIN( ) PROGRAM SOURCE CODE

```
// © 2008 Rachelle Villalon. All rights reserved.

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace Study10
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

APPENDIX H

//RESOURCES CODE

```
namespace Study10.Properties {
    using System;


    /// <summary>
    ///   A strongly-typed resource class, for looking up localized strings,
etc.
    /// </summary>


[global::System.CodeDom.Compiler.GeneratedCodeAttribute("System.Resources.Too
ls.StronglyTypedResourceBuilder", "2.0.0.0")]
    [global::System.Diagnostics.DebuggerNonUserCodeAttribute()]
    [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]
    internal class Resources {

        private static global::System.Resources.ResourceManager resourceMan;

        private static global::System.Globalization.CultureInfo
resourceCulture;


[global::System.Diagnostics.CodeAnalysis.SuppressMessageAttribute("Microsoft.
Performance", "CA1811:AvoidUncalledPrivateCode")]
        internal Resources() {
        }

        /// <summary>
        ///   Returns the cached ResourceManager instance used by this class.
        /// </summary>

[global::System.ComponentModel.EditorBrowsableAttribute(global::System.Compon
entModel.EditorBrowsableState.Advanced)]
        internal static global::System.Resources.ResourceManager
ResourceManager {
            get {
                if (object.ReferenceEquals(resourceMan, null)) {
                    global::System.Resources.ResourceManager temp = new
global::System.Resources.ResourceManager("Study10.Properties.Resources",
typeof(Resources).Assembly);
                    resourceMan = temp;
                }
                return resourceMan;





        }
```

```
  }

        /// <summary>


///   Overrides the current thread's CurrentUICulture property for


        ///   resource lookups using this strongly typed resource class.
        /// </summary>

[global::System.ComponentModel.EditorBrowsableAttribute(global::System.Compon
entModel.EditorBrowsableState.Advanced)]
        internal static global::System.Globalization.CultureInfo Culture {
            get {
                return resourceCulture;
            }
            set {
                resourceCulture = value;
            }
        }

        internal static System.Drawing.Bitmap curvewall {
            get {
                object obj = ResourceManager.GetObject("curvewall",
resourceCulture);
                return ((System.Drawing.Bitmap)(obj));
            }
        }

        internal static System.Drawing.Bitmap slantedwall {
            get {
                object obj = ResourceManager.GetObject("slantedwall",
resourceCulture);
                return ((System.Drawing.Bitmap)(obj));
            }
        }

        internal static System.Drawing.Bitmap straightwall {
            get {
                object obj = ResourceManager.GetObject("straightwall",
resourceCulture);
                return ((System.Drawing.Bitmap)(obj));
            }
        }
    }
}
```

APPENDIX H

//GENERATED CONTROLS, FORM CODE

```
// © 2008 Rachelle Villalon. All rights reserved.

namespace Study10
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should be
disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code


        private void InitializeComponent()
        {
            this.listBox1 = new System.Windows.Forms.ListBox();
            this.listBox3 = new System.Windows.Forms.ListBox();
            this.button4 = new System.Windows.Forms.Button();
            this.label1 = new System.Windows.Forms.Label();
            this.dsgnconsultlabel = new System.Windows.Forms.Label();
            this.RobConvertCode = new System.Windows.Forms.Button();
            this.listBox4 = new System.Windows.Forms.ListBox();
            this.listBox5 = new System.Windows.Forms.ListBox();
            this.designknowlabel = new System.Windows.Forms.Label();
            this.constknowlabel = new System.Windows.Forms.Label();
            this.robotknowlabel = new System.Windows.Forms.Label();
            this.listBox2 = new System.Windows.Forms.ListBox();
            this.label2 = new System.Windows.Forms.Label();
            this.button3 = new System.Windows.Forms.Button();
            this.button2 = new System.Windows.Forms.Button();
            this.button1 = new System.Windows.Forms.Button();
            this.label3 = new System.Windows.Forms.Label();
            this.label4 = new System.Windows.Forms.Label();
            this.progressBar1 = new System.Windows.Forms.ProgressBar();
            this.SuspendLayout();

            //
            // listBox1
            //
            this.listBox1.FormattingEnabled = true;
            this.listBox1.Location = new System.Drawing.Point(676, 23);
            this.listBox1.Name = "listBox1";
            this.listBox1.Size = new System.Drawing.Size(120, 290);
            this.listBox1.TabIndex = 0;
            //
            // listBox3
            //
            this.listBox3.FormattingEnabled = true;
            this.listBox3.HorizontalScrollbar = true;
            this.listBox3.Location = new System.Drawing.Point(12, 385);
            this.listBox3.Name = "listBox3";
            this.listBox3.Size = new System.Drawing.Size(658, 43);
            this.listBox3.TabIndex = 2;
            //
            // button4
            //
            this.button4.Location = new System.Drawing.Point(676, 319);
            this.button4.Name = "button4";
            this.button4.Size = new System.Drawing.Size(115, 37);
            this.button4.TabIndex = 6;
            this.button4.Text = "Design Update";
            this.button4.UseVisualStyleBackColor = true;
            //
            // label1
            //
            this.label1.AutoSize = true;
            this.label1.Location = new System.Drawing.Point(12, 369);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(31, 13);
            this.label1.TabIndex = 7;
            this.label1.Text = "Cost:";
            //
            // dsgnconsultlabel
            //
            this.dsgnconsultlabel.AutoSize = true;
            this.dsgnconsultlabel.Location = new System.Drawing.Point(12,
219);
            this.dsgnconsultlabel.Name = "dsgnconsultlabel";
            this.dsgnconsultlabel.Size = new System.Drawing.Size(127, 13);
            this.dsgnconsultlabel.TabIndex = 8;
            this.dsgnconsultlabel.Text = "DESIGN CONSULTANT ";
            // RobConvertCode
            //
            this.RobConvertCode.Location = new System.Drawing.Point(676, 378);
            this.RobConvertCode.Name = "RobConvertCode";
            this.RobConvertCode.Size = new System.Drawing.Size(115, 35);
this.RobConvertCode.TabIndex = 9;
            this.RobConvertCode.Text = "Convert to Robot Code";
            this.RobConvertCode.UseVisualStyleBackColor = true;
            this.RobConvertCode.Click += new
System.EventHandler(this.RobConvertCode_Click);
            //
            // listBox4
            //
            this.listBox4.FormattingEnabled = true;
            this.listBox4.HorizontalScrollbar = true;
            this.listBox4.Location = new System.Drawing.Point(233, 261);
            this.listBox4.Name = "listBox4";
            this.listBox4.Size = new System.Drawing.Size(216, 95);
            this.listBox4.TabIndex = 10;
            //
            // listBox5
            //
            this.listBox5.FormattingEnabled = true;
            this.listBox5.HorizontalScrollbar = true;
            this.listBox5.Location = new System.Drawing.Point(455, 261);
            this.listBox5.Name = "listBox5";
            this.listBox5.Size = new System.Drawing.Size(215, 95);
            this.listBox5.TabIndex = 11;
            //
            // designknowlabel
            //
            this.designknowlabel.AutoSize = true;
            this.designknowlabel.Location = new System.Drawing.Point(75, 245);
            this.designknowlabel.Name = "designknowlabel";
            this.designknowlabel.Size = new System.Drawing.Size(99, 13);
            this.designknowlabel.TabIndex = 12;
            this.designknowlabel.Text = "Design Knowledge:";
            //
            // constknowlabel
            //
            this.constknowlabel.AutoSize = true;
            this.constknowlabel.Location = new System.Drawing.Point(282, 245);
            this.constknowlabel.Name = "constknowlabel";
            this.constknowlabel.Size = new System.Drawing.Size(125, 13);
            this.constknowlabel.TabIndex = 13;
            this.constknowlabel.Text = "Construction Knowledge:";
            //
            // robotknowlabel
            //
            this.robotknowlabel.AutoSize = true;
            this.robotknowlabel.Location = new System.Drawing.Point(496, 245);
            this.robotknowlabel.Name = "robotknowlabel";
            this.robotknowlabel.Size = new System.Drawing.Size(143, 13);
            this.robotknowlabel.TabIndex = 14;
            this.robotknowlabel.Text = "Robot (IRB-140) Knowledge:\r\n";
            //
            // listBox2

            this.listBox2.FormattingEnabled = true;
            this.listBox2.HorizontalScrollbar = true;
            this.listBox2.Location = new System.Drawing.Point(12, 261);
            this.listBox2.Name = "listBox2";
            this.listBox2.Size = new System.Drawing.Size(215, 95);
            this.listBox2.TabIndex = 15;
            this.listBox2.SelectedIndexChanged += new
System.EventHandler(this.listBox2_SelectedIndexChanged);
            //
            // label2
            //
            this.label2.AutoSize = true;
            this.label2.Location = new System.Drawing.Point(488, 42);
            this.label2.Name = "label2";
            this.label2.Size = new System.Drawing.Size(67, 13);
            this.label2.TabIndex = 16;
            this.label2.Text = "Straight Wall";
            this.label2.Click += new System.EventHandler(this.label2_Click);
            //
            // button3
            //
            this.button3.Image =
global::Study10.Properties.Resources.slantedwall;
            this.button3.Location = new System.Drawing.Point(561, 170);
            this.button3.Name = "button3";
            this.button3.Size = new System.Drawing.Size(45, 33);
            this.button3.TabIndex = 5;
            this.button3.UseVisualStyleBackColor = true;
            //
            // button2
            //
            this.button2.Image =
global::Study10.Properties.Resources.curvewall;
            this.button2.Location = new System.Drawing.Point(561, 96);
            this.button2.Name = "button2";
            this.button2.Size = new System.Drawing.Size(45, 37);
            this.button2.TabIndex = 4;
            this.button2.UseVisualStyleBackColor = true;
            //
            // button1
            //
            this.button1.Image =
global::Study10.Properties.Resources.straightwall;
```

```
            this.button1.Location = new System.Drawing.Point(561, 23);
            this.button1.Name = "button1";
            this.button1.Size = new System.Drawing.Size(45, 36);
            this.button1.TabIndex = 3;
            this.button1.TextAlign =
System.Drawing.ContentAlignment.BottomCenter;
            this.button1.UseVisualStyleBackColor = true;


            //
            // label3
            //
            this.label3.AutoSize = true;
            this.label3.Location = new System.Drawing.Point(490, 108);
            this.label3.Name = "label3";
            this.label3.Size = new System.Drawing.Size(65, 13);
            this.label3.TabIndex = 17;
            this.label3.Text = "Curved Wall";
            //
            // label4
            //
            this.label4.AutoSize = true;
            this.label4.Location = new System.Drawing.Point(487, 170);
            this.label4.Name = "label4";
            this.label4.Size = new System.Drawing.Size(68, 13);
            this.label4.TabIndex = 18;
            this.label4.Text = "Inclined Wall";
            //
            // progressBar1
            //
            this.progressBar1.Location = new System.Drawing.Point(12, 434);
            this.progressBar1.Name = "progressBar1";
            this.progressBar1.Size = new System.Drawing.Size(658, 23);
            this.progressBar1.TabIndex = 19;
            //
            // Form1
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(815, 471);
            this.Controls.Add(this.progressBar1);
            this.Controls.Add(this.label4);
            this.Controls.Add(this.label3);
            this.Controls.Add(this.label2);
            this.Controls.Add(this.listBox2);
            this.Controls.Add(this.robotknowlabel);
            this.Controls.Add(this.constknowlabel);
            this.Controls.Add(this.designknowlabel);

       this.Controls.Add(this.listBox5);


            this.Controls.Add(this.listBox4);
            this.Controls.Add(this.RobConvertCode);
            this.Controls.Add(this.dsgnconsultlabel);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.button4);
            this.Controls.Add(this.button3);
            this.Controls.Add(this.button2);
            this.Controls.Add(this.button1);
            this.Controls.Add(this.listBox3);
            this.Controls.Add(this.listBox1);
            this.Name = "Form1";
            this.Text = "ADEON";
            this.MouseUp += new
System.Windows.Forms.MouseEventHandler(this.Form1_MouseUp);
            this.MouseDown += new
System.Windows.Forms.MouseEventHandler(this.Form1_MouseDown);
            this.MouseMove += new
System.Windows.Forms.MouseEventHandler(this.Form1_MouseMove);
            this.KeyDown += new
System.Windows.Forms.KeyEventHandler(this.Form1_KeyDown);
            this.ResumeLayout(false);
            this.PerformLayout();

        }
        #endregion

        private System.Windows.Forms.ListBox listBox1;
        private System.Windows.Forms.ListBox listBox3;
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.Button button3;
        private System.Windows.Forms.Button button4;
        private System.Windows.Forms.Label label1;
        private System.Windows.Forms.Label dsgnconsultlabel;
        private System.Windows.Forms.Button RobConvertCode;
        private System.Windows.Forms.ListBox listBox4;
        private System.Windows.Forms.ListBox listBox5;
        private System.Windows.Forms.Label designknowlabel;
        private System.Windows.Forms.Label constknowlabel;
        private System.Windows.Forms.Label robotknowlabel;
        private System.Windows.Forms.ListBox listBox2;
        private System.Windows.Forms.Label label2;
        private System.Windows.Forms.Label label3;
        private System.Windows.Forms.Label label4;
        private System.Windows.Forms.ProgressBar progressBar1;
    }
}
```