The Robotic Façade:
A Design Solution for Energy Conservation in the CityHome of the Future

by

Ronan Lonergan

BE (Hons) Mechanical Engineering
University College Dublin, 2009

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of
Master of Science in Mechanical Engineering
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
September 2011

Author...
Department of Mechanical Engineering
July 27, 2011

Certified by...
Kent Larson, Department of Architecture
Thesis Supervisor

Certified by...
Leon Glicksman, Department of Mechanical Engineering
Thesis Reader

Accepted by...
David E. Hardt, Professor of Mechanical Engineering
Chairman, Department Committee on Graduate Theses

The Robotic Façade:
A Design Solution for Energy Conservation in the CityHome of the Future

by

Ronan Lonergan

Submitted to the Department of Mechanical Engineering
on July 27, 2011, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

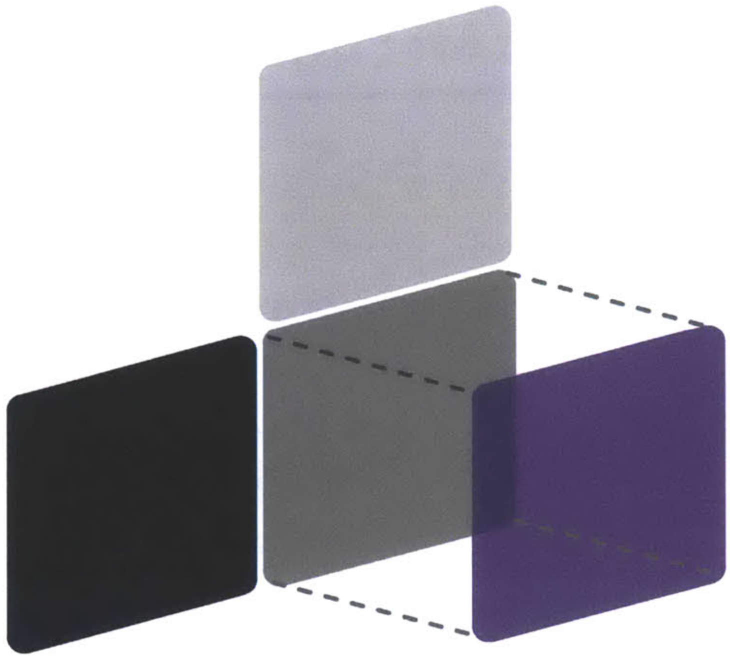
Abstract

This project outlines a design for a façade product that can potentially be used to simplify both the construction and operation of an apartment in an urban setting. Additionally, this façade module has been conceptualized as a way to improve energy performance, making it the energy hub of the home, and housing the majority of the heating, ventilation, and air-conditioning equipment. The design makes use of activity recognition to aid with this energy performance improvement, and adapts the conditions in the living spaces to real-time activities being carried out in the home.

Energy analyses, in conjunction with real-life modeling and deployment tests, were used to verify the concept. In addition, a graphical user interface was built, allowing the home occupant to adjust the system settings of the automated energy-management technologies, as well as enabling the system to give feedback to the user on how certain decisions affect the performance of the home.

Thesis Supervisor: Kent Larson

The Robotic Façade



A Design Solution for Energy Conservation in the CityHome of the Future

by

Ronan Lonergan

MITMECHE

Contents

List of Figures

1. A Vision of how Architecture Should Adapt to Global Urbanization

1.1 An Introduction to the CityHome

Urbanization, the growth of cities, is closely linked to world population growth, and is one the biggest challenges facing governments, and their city architects, in the 21st century. As the United Nations (UN-Habitat) described in their third *World Urban Forum* [1], the rapid expansion of urban areas in the emerging markets in Asia is having a dramatic impact on how people live and work, as well as an associated environmental impact. For instance, as of 2007, more than half the people around the world are "cityzens", or people who live in cities. The emergence of developing countries as urbanized nations is reflected in statistics that describe how, in 1975, the urban population was 813 million in developing countries and 704 million in developed, but, by 2005, the increase in birth rates and migration to cities had increased the urban population to 2266 million in developing countries and 966 million in developed. In fact, by 2030, almost 4 billion people, or 80% of all people who live in urban areas, will live in cities in the developing world. This represents an average urbanization growth rate of 1.78%, or almost twice that of population growth rate [1].

This urbanization is having a dramatic impact on the demand for living space, and the traditional response to this increased demand was to build upwards, with architects prescribing high-rise apartment buildings to solve the problem of a lack of space in urban areas. Furthermore, as cities grow larger still, the average size of an apartment decreases.

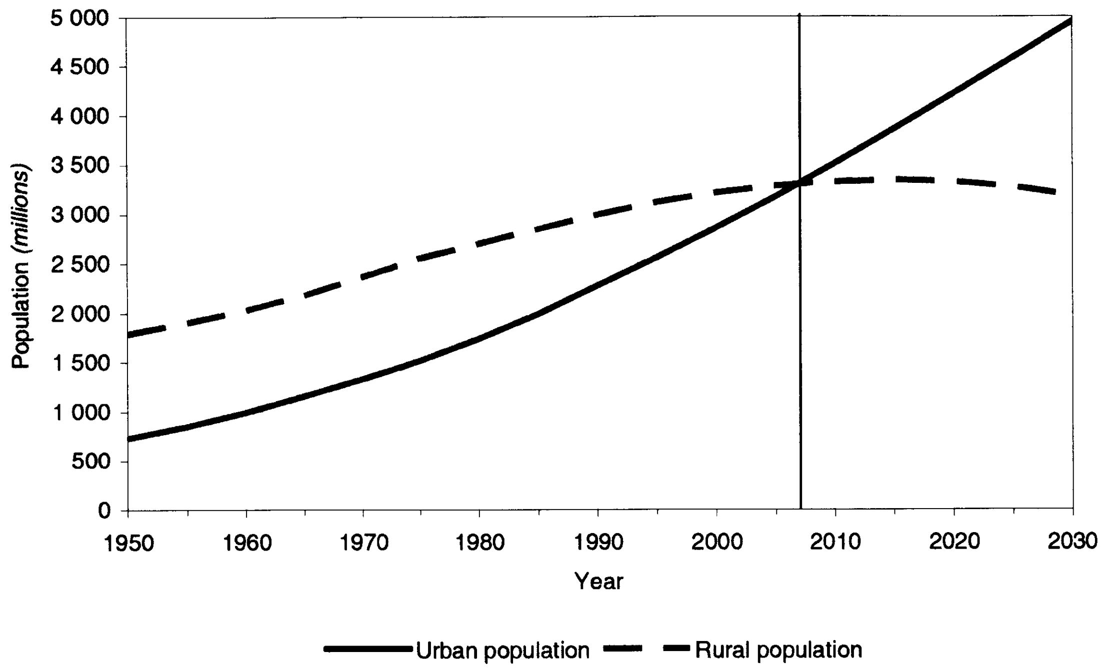


Figure 1: Urban and rural population of the world: 1950-2030 [2]

In addition to the demand for living space, the needs and desires of the people themselves have shifted over the last quarter-century. For the first time in human history, the elderly population in developed nations has surpassed the number of children aged 14 or younger, and by 2050, nearly one third of people in the developed world will be elderly [1]. In this century, people expect ubiquitous mobility-on-demand more than any generation before, be it in the availability of domestic and international air routes, or the ability to travel anywhere in an urban setting via a convenient, and relatively direct, public or private mode of transport, and all this despite the increased energy costs we experience in the 21st century. The CityHome project, part of the Changing Places research group, is a direct response to this state of flux that both developed and emerging nations are experiencing in their cities.

Directly addressing the lack of space in the urban environment, the CityHome is a concept that aims to give more functionality and flexibility to the homeowner through the use of transformable structures to furnish the home. Based around a standard developer-built apartment size (a 2-bay, 750 sq. ft. 1-bedroom, or 3-bay, 1100 sq. ft. 2-bedroom), and using a prefabricated modular chassis (building frame), and mass-customizable interior infill modules (cabinetry), the apartment can not only be customized prior to construction, but the transformable furniture allows the living space to adapt to changing daily needs, so that an extra bedroom, or a bigger entertaining space perhaps, can be created on demand.

Figure 2 gives four renderings of a 2-bay CityHome in different configurations, adapting to the need for larger entertainment and dining space in image A, to the combination of dining and sleeping spaces given in image B, the extended sleeping areas in image C, and the combined gym and lounge spaces in D. These transformations are primarily facilitated by the use of a large, translating cabinet module that can divide the spaces in a multitude of ways.

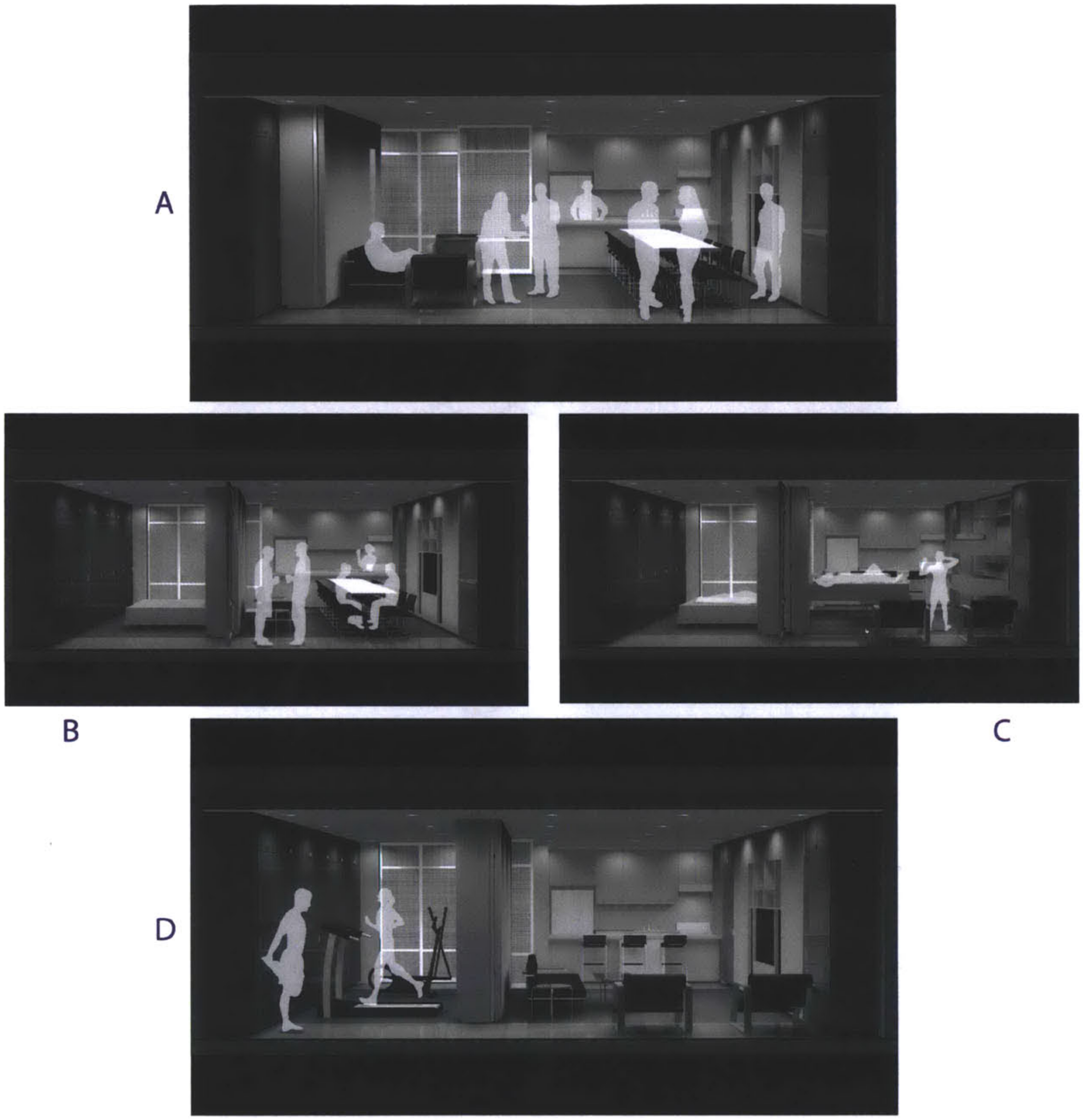


Figure 2: Examples of CityHome apartment transformation

In addition to the standardized, modular chassis that makes up the frame of the building, and the interior modules that allow the apartment to transform, the CityHome has been envisaged as a design that simplifies construction, bringing many of the complicated systems associated with the heating, ventilation and air conditioning equipment to the perimeter of the living space, and placing it in a façade module that constitutes the "energy hub" of the apartment. It is within this framework of customization of the CityHome that the apartment solution design tree was developed, breaking down the design into those three primary components—the chassis modules, the interior modules that allow the apartment to transform, and the façade modules, as described in Figure 3.

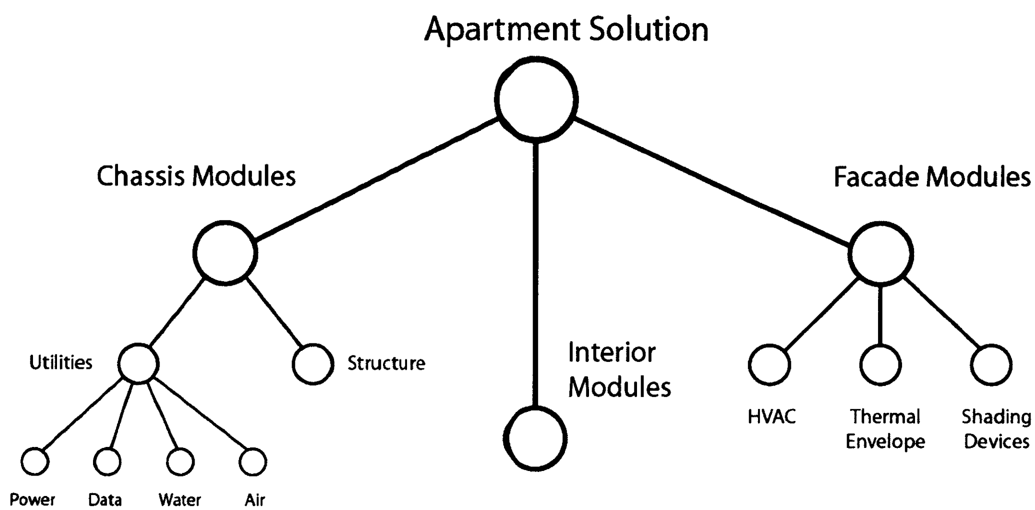


Figure 3: Apartment design tree

In addition to allowing a smaller space to behave like a much larger living area, the CityHome aims to integrate various technologies to enhance the standard of living in the modern urban environment. An extensive sensor array in the apartment allows activity recognition to give pertinent information to the homeowner, such as information about the energy use in the home based on user-habits; but activity recognition can also assist in the health monitoring of elderly residents, enabling the ageing population, as described before, to maintain their independence in their own home for longer. The *PlaceLab* was an extensive experiment undertaken by the Changing Places group to explore such uses for activity recognition, and described in the appendices on page 122.

Related to the CityHome project is the CityCar, a concept that addresses the issue of mobility in an urban context, again where space, for driving and parking in this case, is limited. The CityCar has been designed to occupy one third the amount of space of a regular vehicle when parked, and due to the fact that it is electrically powered, it can potentially be interfaced with the home in novel ways, such as brought in a specialized elevator directly up to the user's home.

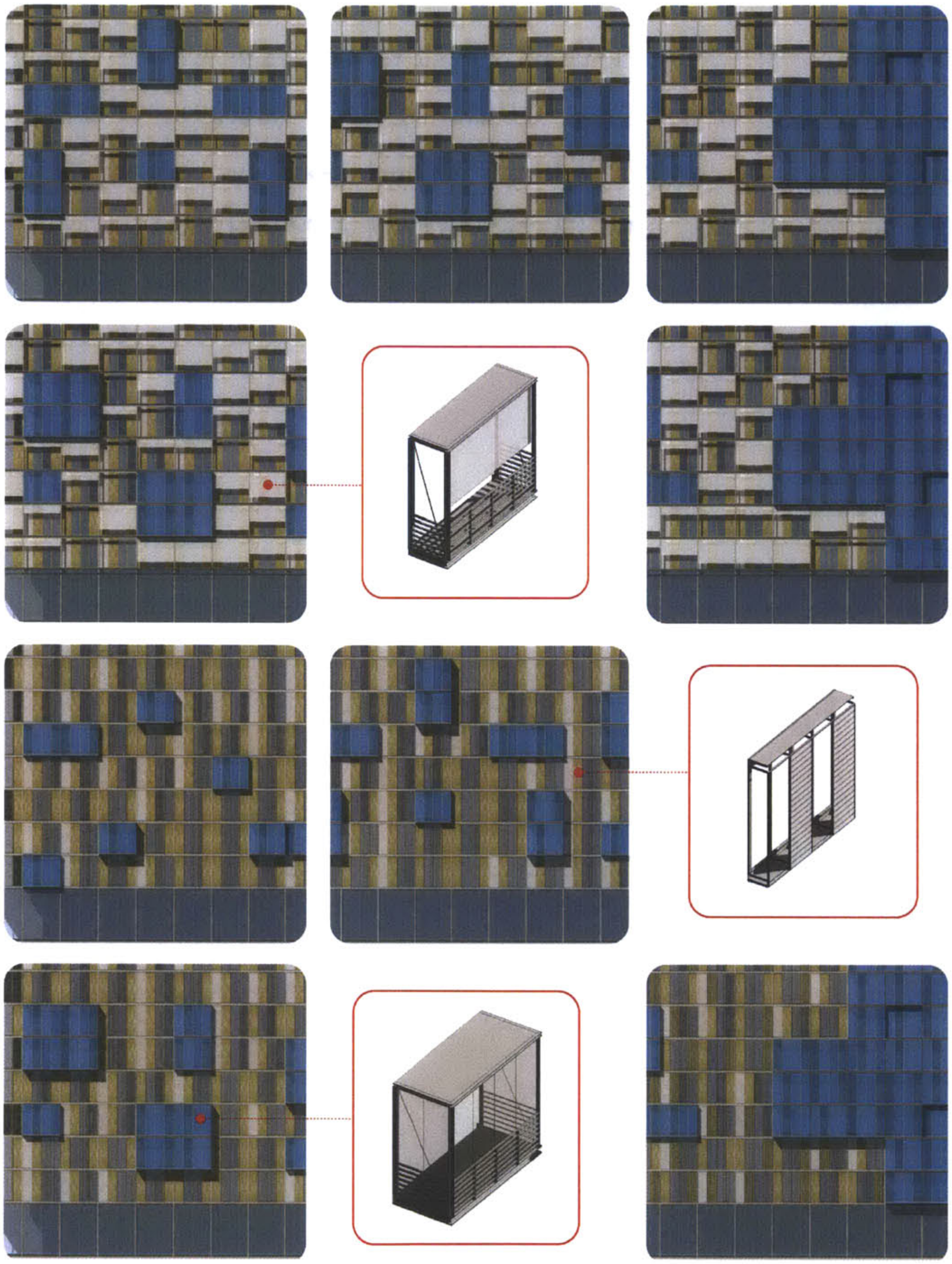


Figure 4: Rendering of CityHome with different façade configurations

Perhaps the greatest impact from urbanization and population growth, beyond simply the increased demand for usable space, is the impact on the energy we are using, still primarily produced from finite fossil-fuel resources (86% [3]), on a daily basis. While six countries;

Bangladesh, China, India, Indonesia, Nigeria, and Pakistan, represent the greatest contributors to population growth (over half the people born on the planet each year [1]), and therefore also some the greatest potential for urbanization and the associated environmental impact from increased energy use, it is the United States that is the world's biggest consumer of energy. While China produces most carbon dioxide emissions of any country [4], the necessity to change energy-usage habits in the U.S. is nonetheless significant. Twenty percent of the world's energy consumption is attributable to the U.S., and buildings represent forty percent of the total (see Figure 5). This project further concentrates on residential buildings, which consume twenty-two percent of all energy in the United States alone. When one considers that, according to Energy Star, 30% of energy in buildings is used inefficiently or unnecessarily [3], the potential to effect change through the design of a more intelligent home is quite significant. In turn, one of the biggest influences on energy use in residential buildings is the interface between the interior and exterior environments, the building skin, or façade. Figure 4 shows several renderings of the exterior of the CityHome, exploring the idea of using modular balcony structures, with different configurations, on the façade of the building.

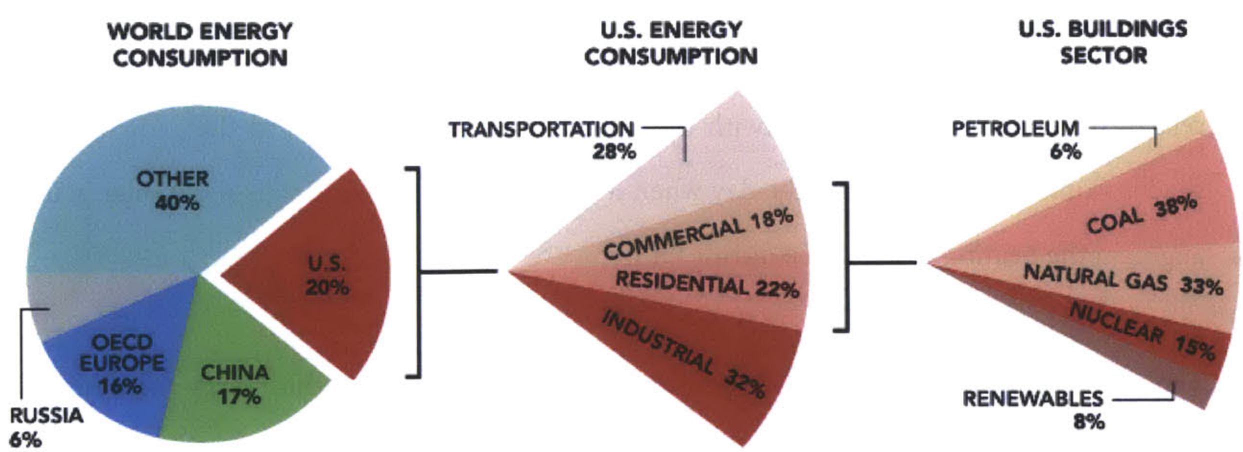


Figure 5: World Energy Consumption 2010 [5]

1.2 An Introduction to the Robotic Façade Project

As outlined in Section 1.1, the CityHome aims to define a new standard for urban living, using space and energy more efficiently, as well as improving the quality of life of the homeowner by incorporating activity recognition technology that can be used to aid the resident. The façade is seen as a modular piece of the apartment solution (as described in Figure 3), housing most of the energy-handling HVAC equipment, thereby reducing the complexity of installation of this equipment in the main apartment chassis, and defining the façade module as the "energy-hub" of the apartment. In addition to controlling the exchange of heat energy between the inside and the exterior environment, the façade is also responsible for protection from glare, inclement weather, and pollution, to name but a few, so the role of the module can be further generalized to "comfort control".

Essentially, this project identifies several shortcomings to how this comfort control is typically achieved in a present-day, developer-built apartment. Related to HVAC systems, it is noted that many apartments are outfitted with oversized equipment, since proper optimization would require prohibitively expensive engineering, that the equipment is complicated and labor intensive to install, and that the equipment is then operated inefficiently by the homeowner. Similarly, for the technologies that influence the other comfort criteria, such as the shading, some of the outstanding problems identified are those related to the lack of integration with the HVAC, and inefficient use, so that, for instance, shades would be closed during the day when it would be most beneficial to allow solar gain to augment the heating of the space during winter.

The overarching goals of the façade project design, as they relate to the energy handling and general comfort-control systems, are to be able to optimize the equipment, without the use of expensive engineers, to simplify the installation, hence reduce the associated expense of installation, and to achieve intelligent and efficient use of the equipment.

It is suggested that a customizer program, similar to how cars are configured today using online configurators, could be used to achieve an efficient choice of equipment sizes and types, and based on a limited number of user inputs, such as the location of the apartment building, the size of the building, and aspect of the façades etc. Furthermore, addressing the lack of standardized interfaces in the architectural and construction industries, such as piping interfaces for water-handling equipment, could streamline the construction process considerably. The façade design, being modular in nature, will specify a standardized interface to the chassis and interior modules, so that functionality similar to that seen with the Universal Serial Bus (USB) in computer hardware is available to the construction process, and so different façade modules can be seamlessly interchanged with the in-place building chassis.

The term "robotic" is used to describe this façade module design project since the design will leverage a sensor network and activity recognition algorithms to achieve efficient operation by using automation whenever possible. One of the bigger design challenges is deciding how much user-input to allow, since the balance between automated and manual control is a delicate one, and automated systems do not always do exactly what the user wants, which can lead to the user becoming frustrated, and abandoning automated control completely. This issue is explored in greater depth further on in the thesis, but it is suggested that through this use of automation, an appealing combination of energy savings and an enhanced living environment can be achieved.

This Robotic Façade can be described as a comprehensive design project, with emphasis placed on developing a product with real marketable potential, rather than an academic exercise. With this in mind, the design decisions made during the project reflect how practical the potential designs were, and rather than re-inventing the comfort control systems, the robotic façade module can be described as a hub that integrates existing technologies in an innovative and more appealing way, merging technologies that are traditionally fragmented, so that users are likely to be compliant with best energy-saving practices, since the system is virtually effortless to use. An example of such fragmented technology may be an automated shading and lighting system, not a novel technology, but seldom used in residential homes. Such systems are generally complicated to install, control, and maintain, use proprietary software and hardware, and often do not interact well with other products, as well as give more options to the user than needed.

The Robotic Façade is a product that could be built and configured off-site, where costly labor associated with the HVAC system installation can be employed more effectively, and then installed on-site in a near-complete condition with only simple interconnections between the HVAC equipment housed on the façade module, and the delivery systems in the rest of the home.

The Robotic Façade aims to find a balance between architectural aesthetics and robust design, with the façade of the Institut du Monde Arabe in Paris, pictured in Figure 6, held as an example of exceptional architectural beauty, but excessive complexity, since the mechanized irises that make up the façade are no longer functional due to the sheer number of moving parts. Additionally, this project is approached as a design solution to urbanization throughout the world; therefore, the façade module's standard design will consider multiple regions across the globe, not just the United States.

A summary of some of the basic design intentions of the project is given in Figure 7.

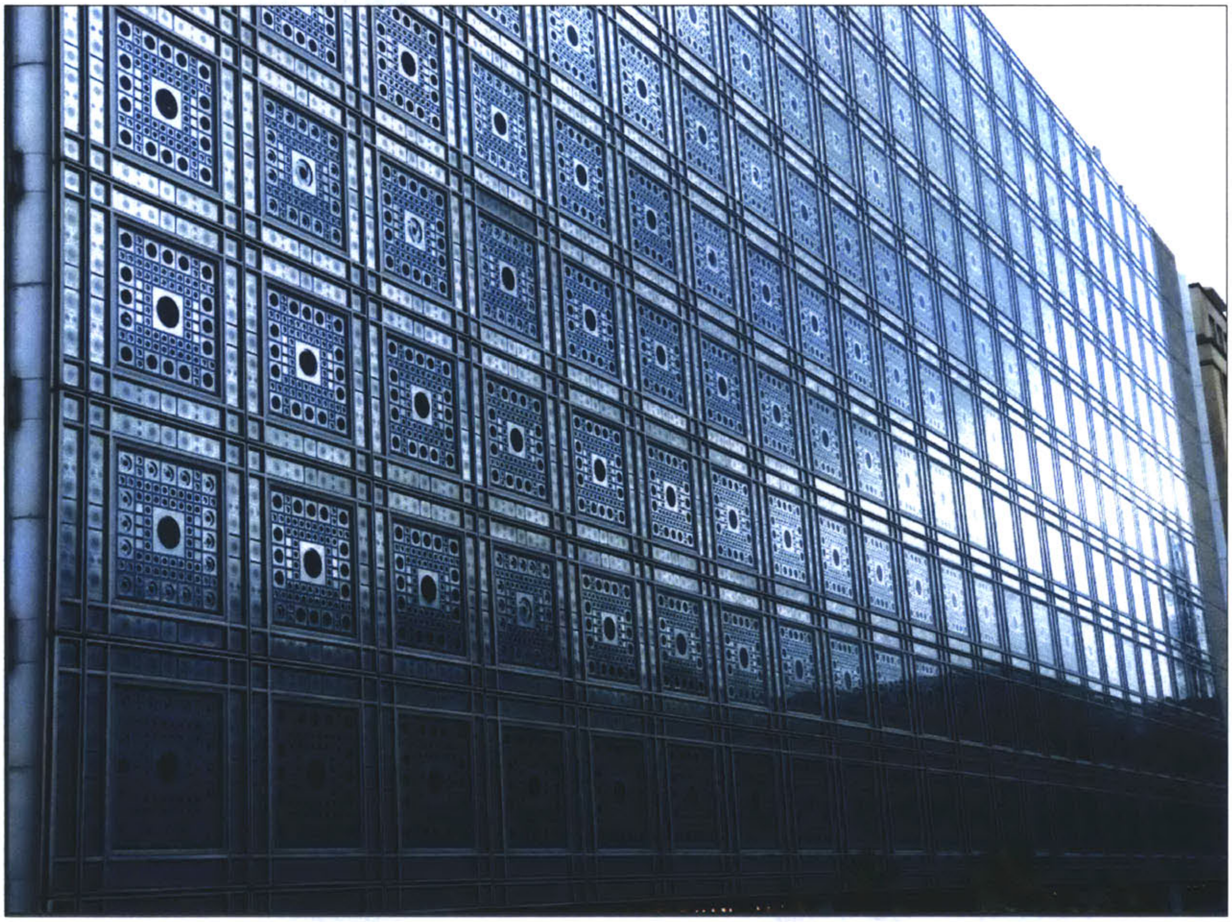


Figure 6: Institut du Monde Arabe [6]

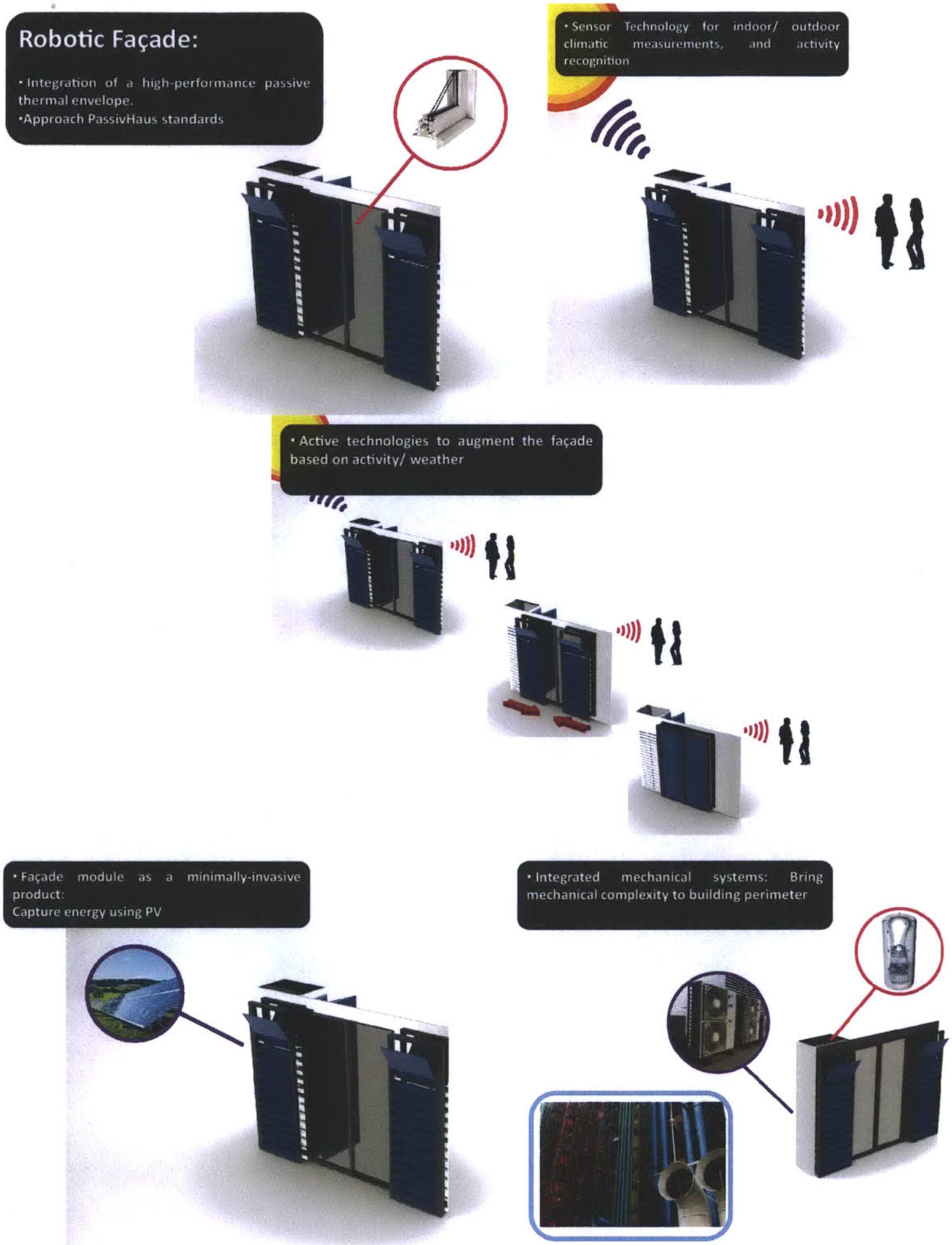


Figure 7: Summary of initial design intent

2. The Design Process

2.1 The Façade as a Tool for Energy Conservation and an Enhanced Living Environment

As outlined in Section 1. , the Robotic Façade will serve as the primary tool for reducing energy use, as well as providing technologies to enhance the quality of life of the CityHome resident, beyond that of a normal apartment.

The following points were those identified during the design iterations which are necessary to reduce the overall energy consumption of the home. They constitute guidelines for the development of a commercially-viable façade product:

i. Reduce heating loads in winter. This refers to the inclusion of high performance insulation and glazing in the façade design so that these passive technologies can reduce the loads placed of an active heating system.
ii. Capture solar energy in winter. This point expands upon the first, specifying that sunlight incident on the glazing during the winter should be allowed to penetrate into the living spaces to augment the heating system.
iii. Reduce cooling loads in summer. This point, like the first, can be achieved by using high performance insulation to keep cooled air within the apartment during warm summer weather.
iv. Natural ventilation in summer. In many cases, the outside air temperature is low enough during summer months to keep the interior of the home within comfort conditions. This point requires that natural ventilation be used whenever possible, since it eliminates the use air conditioning.
v. Reduce electrical loads. Phantom energy loss, or electrical energy usage by devices in standby mode, represents as much as 5% of residential electrical

energy use in the U.S. [7]. The use of sensors attached to the most energy-intensive electrical appliances in the home can potentially reduce these numbers significantly.

vi. Capture and store energy. This point refers to harnessing renewable energy sources in the vicinity of the home. This may include the use of photovoltaic panels, or solar thermal collectors, depending on the region and orientation of the façade.

vii. Develop an industrial design which is mass-customizable. This point refers to the energy savings potential of a mass-produced home. While pre-fabricated homes have enjoyed limited success over the past number of decades, it is thought that by combining the mass-production economies of scale, with the choice offered in a one-off, architect-designed home, the energy saving potential during construction would be significant. The combination of mass-production with wide customization choice is known as mass-customization.

viii. Encourage responsible energy usage. Giving the homeowner information about their energy usage habits in the home, at a time when they are likely to pay attention to the information (just-in-time information), as well as advice as to how they can improve their behavior for the better. This can be achieved through the use of activity recognition, to relay what the resident it doing at a given time, in combination with a system that can predict what impact a given activity will have. An example might be a system that advises a resident, who is using a personal computer in the living room at the time, that shading the windows in the unoccupied bedroom would reduce the air-conditioning loads in the space by 10% for the day.

The proceeding sub-sections expand upon this list of requirements, outlining the functionality required for the Robotic Façade to act, firstly as a HVAC hub, but also as an intelligent interface with the external environment.

2.1.1 The Façade as a HVAC Hub: Required Functionality

Figure 10 expands upon Figure 3, outlining the various HVAC technologies that should potentially be accommodated in the façade module, with the exact configuration varying depending on climatic demands and the geographic location of the apartment. The design will again be modular, allowing the various sub-systems to be swapped in and out of a standard configuration common to all climates and façades.

In order to accommodate the majority of the heating, ventilation and air-conditioning systems at the perimeter of the building, a standardized utility system (air and water handling) is prescribed for the CityHome (see Figure 8). This system includes three primary parts:

i. An air-exhaust network of pipes that run from the back of the apartment space. This network, taking air from the back of the apartment, will create a region of lower pressure than that at the façade. This slightly lower pressure can, in turn, be used to draw air deep into the apartment from the façade during times when natural ventilation is possible.

   The exhaust fans are located in both the bathroom and kitchen, and the connecting ductwork runs along a single over-head path back to the façade module. After discussions with the manufacturer of a high-velocity duct system for conditioned air [8], it was decided that the same technology could be incorporated into the CityHome for air exhaust. This micro-duct system, using a 2-inch diameter duct, is rated to handle over 200 cubic feet per minute, per duct. The small bore of ductwork, has been designed to require one-third the amount of space of a convention system, so it is intended that installation would be possible within the framework of the CityHome chassis, between the exposed ceiling, and the load bearing members.

This active, fan-assisted system, when used in a 3-bay, 1100 sq. ft. CityHome, would be capable of circa one air-change per hour, per duct. It is intended that three micro-ducts will be installed for the space, one for each of the blue inlet boxes shown in Figure 8, giving the potential to draw in over three air-changes per hour in the space.

ii. A domestic hot water (DHW) network that runs from the façade to the back of the apartment, where the kitchen and bathroom will be located. Similar to the air extraction network, the DHW network runs from the façade module, under-floor, back to the kitchen and bathroom.

iii. A radiant under-floor network throughout the apartment to deliver heated/chilled water to the living space. While under-floor heating is relatively common, the use of a chilled floor during times when cooling is required is less prevalent, and Section 2.1.1.1 describes work related to the use of the technology.

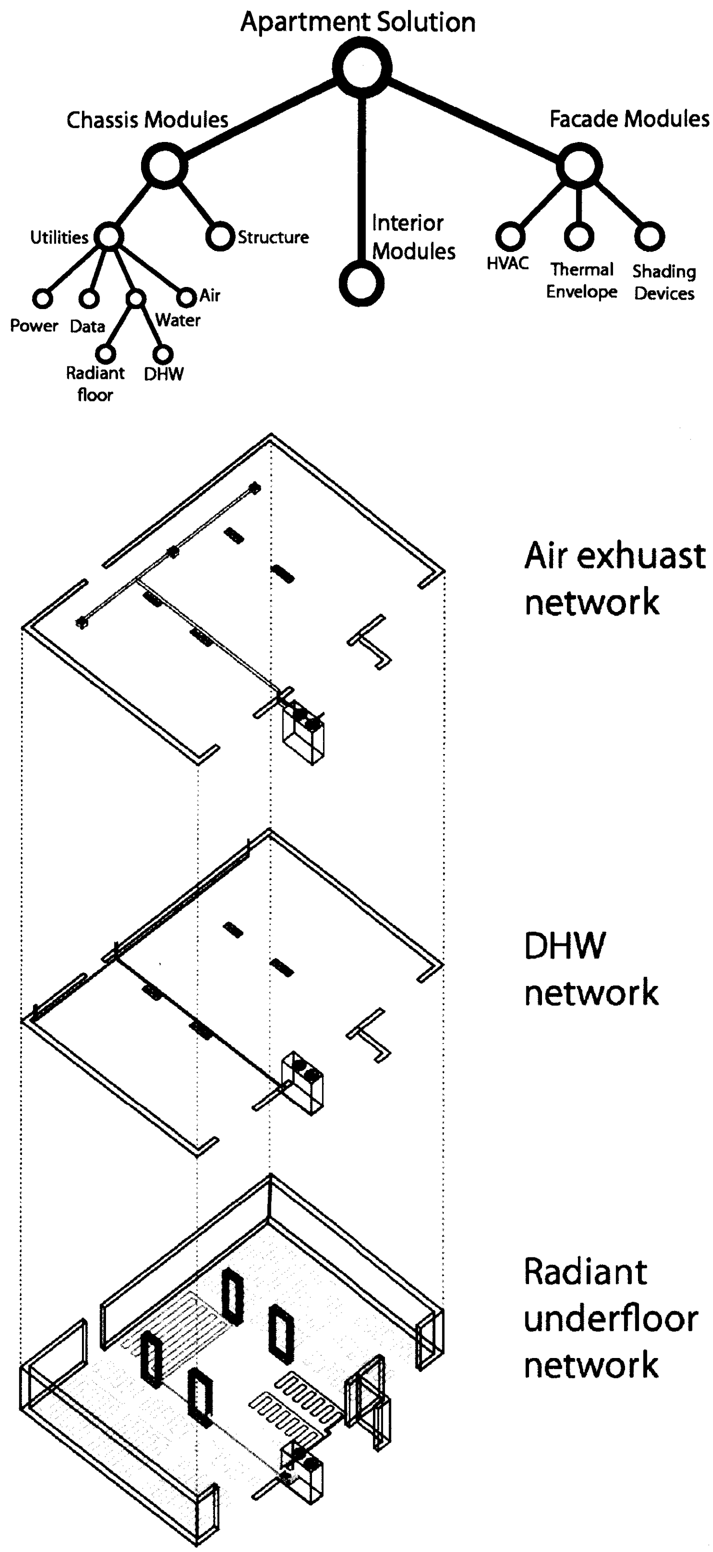


Figure 8: Utilities Solution

2.1.1.1 Radiant Heating and Cooling: State of the Art

It is intended that the modular HVAC of the Robotic Façade system will incorporate radiant heating and cooling, and the following section outlines some of the work that has been carried out by the academic community using related technologies.

While under-floor pipe networks and hydronic heating systems have been used in housing for centuries, and introduced as a modern technology to the United States in the 1930s by Frank Lloyd Wright [9], few attempts to use an under-floor network to cool a space have been made. ASHRAE, the American Society of Heating, Refrigerating, and Air-Conditioning Engineers, clarifies that a radiant system is simply one that transfers more than 50% of its heat by radiation, and outline the drawbacks of such a system in their HVAC handbook [10];

i. Among the most significant drawbacks to the use of radiant floor cooling is the possibility of condensation developing on the floor surface, due to the fact that radiant systems inherently deal with sensible cooling alone. To address this issue in regions where humidity levels may cause condensation on a radiant floor, a separate air treatment system must be incorporated into the equipment.
ii. Radiant floors, used for both heating and cooling, may also suffer from slow response times if not controlled correctly.
iii. In certain instances it has been found that non-uniform surface temperature is caused by incorrect sizing of the under-floor pipe network.

Despite these drawbacks associated with radiant heating and cooling, its popularity is growing in many countries, with 30-50% of new residential buildings in Germany, Austria and Denmark, and 90% in Korea, incorporating radiant under-floor heating [9]. ASHRAE looks favorably upon radiant floors as an energy efficient technology, and again highlights the associated advantages in their handbook [10];

i. Radiant systems allow for control of both the indoor air temperature, as well as the mean radiant temperature of a given spaces, so that overall comfort is better controlled.
ii. Control of mean radiant temperature can be the primary means of meeting acceptable conditions, and can allow the dry-bulb temperature to be lower when heating, and higher when cooling, thereby reducing sensible heating and cooling loads, and saving energy.
iii. Air motion in the space can be minimized since only air necessary to fulfill ventilation requirements is needed. This can reduce drafts.
iv. Radiant floors can take better advantage of high efficiency energy sources such as heat pumps.
v. Radiant systems allow for simultaneous heating and cooling in adjacent zones.
vi. Hydronic radiant systems, since water is such a good thermal storage medium, allow the HVAC equipment to be placed further away from the point of deliver of the heating/ cooling energy. This can be achieved without the associated fan noise of an air distribution system.
vii. Peak loads can be reduced through use of thermal storage in both the water itself, as well as thermal mass.

Certain studies have looked at the energy savings that may be possible when using radiant systems in place of conventional all-air systems, and Stetiu [11] investigates the potential savings in commercial buildings using radiant cooling. In this case the system is a cooled ceiling, which is known to be the most efficient type of radiant cooling configuration [10], but the study gives an indication of the magnitude of energy savings that are possible when using radiant cooling in general. The results find that while radiant cooling is most applicable for hot, dry climates, it can be used in any US location with low risk of condensation, if humidity levels are addressed using an outdoor air treatment system. The results show that energy consumption savings range from 17 to 42%, and average at 30%, for the locations simulated in the study. The use of radiant cooling also reduced the peak load by an average of 27%.

Note that while radiant ceiling panels are highlighted as most efficient [10] due to the beneficial buoyancy effects that can be achieved when cooling air above a space and allowing it to sink into the occupied volume, several studies have looked at the potential for using radiant floor cooling. This offers the advantage of using the same pipe network as for radiant heating in heating dominated climates, when one may not be able to justify the installation of a separate cooling pipe network, and is something shown to be possible by Dieckmann *et al.*[12]. Lim *et al.* [13] discuss how best to control such a radiant floor cooling system, and conclude that control of the water temperature in the pipe network is more effective than controlling the flowrate, and that a cooled floor can maintain acceptable comfort conditions to within very acceptable limits.

Bjarne Olesen, head of the International Centre for Indoor Environment and Energy at the Technical University of Denmark, has been one of the most prolific authors of scientific publications related to radiant floor cooling [14][9]. Olesen [9] gives an overview of radiant floor heating in his review article, indicating that a maximum floor temperature of 29C should be used for heating the occupied zone, but that the temperature may be up to 35C for the perimeter zone (1m) around windows and outer walls. It is stated that the maximum heating capacity in the occupied zone is thus around 100 W/m^2, with the heat exchange coefficient being around 11 W/m^2-K and the radiation contribution to this coefficient being around 5.5 W/m^2-K. It is suggested that a heating system should be designed so that the temperature drop of the water be no more than 10K through the pipe circuit, and that the pipes be placed with 150mm spacing.

In a recent publication, Olesen [14] concentrates on radiant floor cooling, and states that while cooling from floor level results in lower heat exchange coefficient (to around 7 W/m^2-K total, made up of 5.5 W/m^2-K radiation), it can still be used effectively. One of the reasons why cooling from the floor is so effective is due to the high angle factor between the radiant heat source, the floor, and the occupants. Jin *et al.*[15] explain this further, stating that the angle factor, which is directly proportional to radiative heat flux, is about 2.5 times higher for floor cooling than for a ceiling cooling system, meaning that a 1K change in the floor temperature will have 2.5 times the effect on mean radiant temperature than a 1K change in ceiling temperature.

Olesen [14] also notes that a radiant floor cooling system results in acceptable vertical air temperature distribution (it is recommended that the air temperature difference between ankles (0.1m) and head (1.1m) for a sedentary person be limited to 3K), and it was found in experimental studies that floor cooling can limit this range to 0.4-0.5K.

In relation to radiant floor cooling performance and design, Olesen [14] notes that cooling heat flux is limited to around 50W/m^2 due to the prescribed minimum floor temperature of 20C for sedentary people in a given space. This heat flux can, however, be increased (floor temperature lowered) if direct sunlight is incident on the floor, and can reach values in excess of 100W/m^2. It is further stated that floor radiant cooling should be designed with 150mm spacing between pipes, and to allow for only a 3 to 5C increase in water temperature through the pipe circuit.

Dieckmann *et al.* [12] expand upon Olesen's description of radiant floor cooling, indicating that a ceramic floor is more appropriate than a wooden radiant floor, due to the increased thermal insulation offered by a wooden floor, hence the larger temperature differences required to give a certain heat flux, as well as the decreased potential to achieve an even surface temperature across the floor. Note also that while Dieckmann *et al.* [12] concentrate on radiant floors in comparison with radiant ceilings, the possibility to have radiant wall systems is also mentioned, as it is by ASHRAE also [10].

2.1.2 The Façade as an Interface: Required Functionality

As outlined previously, the façade, being the interface between the interior living space and the outside environment, has significant impact on energy management in the home. The required functionality encompasses this energy management role, but extends beyond it, as outlined in the list below (Section 2.1.2.2). The design of the system is such that, as described in Section 1.2, the façade will employ a significant amount of automated control, but also allow for complete manual control, whether or not such manual intervention has a detrimental impact on energy efficiency. In order to bridge the gap between more efficient automated control, and manual control, the system considers the use of just-in-time feedback to the user. This persuasive information regarding the impact upon energy efficiency of any given manual over-write is essentially extending the general requirement to encourage responsible energy usage, as outlined in point vii. of Section 2.1. As such, each of the design requirements in Section 2.1.2.2 is described with each of **manual** and **automated** control, as well as **persuasive information** feedback in mind.

The CityHome will incorporate an extensive sensor array to aid in the activity recognition and automated control of its systems. Through the knowledge built-up during the PlaceLab (see page 122 of appendices) experiments, it is known that such a sensor deployment can be discreet, non-intrusive during operation, and cheap, with multiple sensor types included on a single integrated circuit, and communicate wirelessly with a base station. The following is a brief description of the sensors that can be made available to the final design.

2.1.2.1 Sensor Types

i. Temperature: Multiple temperature sensors within the interior of the apartment can give fine-grained information on thermal conditions, and allow for a targeted response to uncomfortable levels in a smaller thermal "sub-zone" that may be the kitchen area etc.

   Both air and floor temperatures will be monitored so that the space heating can be adapted as required.

ii. Humidity: The relative humidity level is an important metric that relates to the comfort of occupants, and humidity will be of particular importance when using under floor cooling to prevent condensation build-up.

iii. Air quality: It is suggested that the comfort system should monitor the levels of potentially-harmful gases in the home. The types of gases monitored might change depending on geographical region; for instance, radon gas leaking naturally from the earth is an issue is some parts of the world, but others may be important everywhere, including carbon monoxide, carbon dioxide and NOx.

   Rather than attempting to monitor particulate matter in the interior air, it is suggested that a filter system be used to remove such contaminants as the air enters the building. This might be of particular importance if the building is in what is known to be a heavily-polluted city, or very close to a busy road. This sensor type will be more coarse-grained, perhaps only a single sensor will be used for the apartment as a whole.

iv. Air flow rate: This metric has a subtle influence on the comfort of the interior occupants. It is known that the presence of an appreciable air flow within an interior space will generally allow the interior temperature to be higher than would normally be acceptable to maintain comfort conditions with still air. It is expected that the measurement will be inferred from a simple air-speed sensor placed close to the main opening into the interior. This sensor may be a small anemometer, or the information may be received from the fan speed of an energy recovery ventilator aspirating the living space.

v. Light: Lighting-level sensors will make up a fine-grained sensing array and will monitor for unacceptably high or low levels of sunlight and artificial light, based on current activities in the home.

vi. Glare: This is closely related to lighting-levels, and will use the same sensors to predict when glare might be a problem for people in the living space, again based on their current activities.

vii. Sound: This type of sensor may be of importance if natural ventilation is used in a built-up area, where noise pollution might affect the home occupants. These sensors can also be used in combination with passive infrared and object motion sensors (see below) to infer activity based on certain algorithms.

viii. Passive Infrared: This sensor will monitor for motion within the home. It will take the form of a fine-grained infrared sensor array that can be used to predict activity in the home.

ix. Object Motion: Object motion sensors will be attached to particular items that are used within the home. Using a piezoelectric accelerometer, they will be able to detect when the particular item is used or moved.

2.1.2.2 Façade Interface Requirements

i. **Temperature control:** Under-floor heating and cooling will be employed as the primary means to deliver heated and chilled water to the apartment mechanically. **Automation** can be employed in a conventional manner, using a thermostat, and additionally incorporate night set-back. In certain instances the sensing technology may recognize extraordinary circumstances within the home, such as a large number of guests present for a dinner party. This would allow the system to reduce the heating requirement, or might necessitate increased cooling. Incorporating smaller heating zones into the piping layout, in conjunction with the sensor array, allows for fine-grained control of the temperature, based on activity recognition within the home. For instance, if it is known that a person is cooking in the kitchen rather than eating, then the temperature could potentially be reduced in this area to accommodate the decreased heating requirements of a person who is physically active.

If a user feels uncomfortable within the home then a **manual** override of the automated system should be easy to perform from a central control panel. Manual interventions should be temporary, but if the system recognizes that a person is consistently uncomfortable with the temperature set-points they should be offered the option of changing them.

The **persuasive information** aspect of the temperature control is employed when a user attempts to override the automated system set-points for temperature control. In this instance they should be presented with an argument for leaving them at the more energy-efficient level, stating that they would save a specific amount of money by being more energy efficient. It will be important to maintain a balance that allows for a persuasive argument with the user, but does not become a nuisance when trying to change the system settings.

ii. **Natural Lighting Control: Automation** will allow the system to minimize the use of artificial lighting, potentially dimming or switching off lighting when no activity is detected in a particular part of the home at night. During the day this automation will maximize natural light through the shading devices if it is most energy efficient to do so, such as when solar heat gain would decrease the load placed on the heating system.

Manual control will allow the home occupant to open and close the shading at will, despite the fact that it may not be most energy efficient to do so.

Again, **persuasive information** at an appropriate moment should inform an occupant about the impact of allowing, or not allowing, natural sunlight into the living space.

iii. **Ventilation and Exhaust:** Using pertinent sensor information, the amount of fresh air drawn into the system can be **automatically** controlled based on occupancy and activities being carried out in the home. In this case the air may enter through an energy recovery ventilator (ERV) system, where it is filtered and heated/ cooled by the exhaust air, or can come straight through open windows. The flow rate to the interior serves minimum outside air requirements, but can be automated to generate an appreciable breeze in the occupied spaces of the home. If it happens that the space requires mechanical cooling at the time, or is at the cross-over point where natural cooling is no longer acceptable and mechanical cooling would be required, then this air flow might increase the highest temperature that the occupants find acceptable, thereby reducing, or perhaps removing, the mechanical cooling load.

Manual control may be necessary to increase the air flow to the kitchen, for instance, during times when cooking smells need to be removed from the living spaces, but cannot be automatically detected.

Persuasive information may be employed at a time when the system is presented with the scenario whereby a person wants to open all the windows on a very cold winter's day. The response might be to warn that the energy lost by bypassing the ERV system would have a significant energy efficiency impact, and the user may be provided with a dollar amount that could potentially be saved by taking alternative action.

iv. **Humidity Control:** Based on the information from both the internet (prevailing outdoor humidity levels in surrounding area), and interior humidity sensors, the system can **automatically** determine if humidity adjustment (potentially dehumidifying in summer, humidifying in winter) is necessary. When humidity adjustment is not necessary, then natural ventilation through the windows will be possible.

While ASHRAE does not specify a lower limit on relative humidity levels, there may be some people who are more sensitive to dry air than others. For instance, some people experience nose-bleeds when humidity is too low. These conditions will not be anticipated by the automated system, so **manual** control will be necessary. Again, if consistent disapproval with the humidity levels is recorded in the pattern recognition software, then the humidity set-points should potentially be adjusted.

Persuasive feedback related to humidity levels may be of particular importance with the use of under floor cooling. If the user attempts to open the window then he or she should be advised that the humidity levels will increase, and cannot be controlled properly. This would potentially lead to condensation on the chilled floor, and hence should be avoided.

v. **Insulation against Heat and Cold:** Perhaps the most important function of the façade, this will essentially be a passive technology, a high-performance insulation layer that is gasketed to give a thermal barrier approaching that of a Passiv House [16]. **Automation** may allow a secondary layer of exterior shades to cover the glazing, increasing the insulation values somewhat by decreasing the air flowrate over the exterior of the façade.

Manual control of the insulation will primarily be in the form of manual control of the actuated windows.

Similar to the design requirement for ventilation and exhaust, **persuasive information** feedback can be used to inform the occupants that opening the windows may have a significant negative impact on the energy use of the home.

vi. **Wind Protection:** While not designed to withstand a storm, if high winds are predicted and relayed to the system by an **automated** weather update from the internet, then the system may respond by closing external shades across the glazing of the façade.

Manual control of the system may be necessary if the automated weather prediction in inaccurate.

vii. **Sun Protection:** This requirement is one of the more basic expected of the façade; to protect the building occupants, and interior furnishings and infill components, from the harmful effects of direct sunlight. It is a combination of technologies, with multiple functions, that will achieve this sun protection, such as the insulation, glazing, external and interior shading layers, and is automated and manually controlled indirectly through these technologies.

viii. **Glare Protection:** Sensor detection of activities, in conjunction with knowledge of where the Sun is in the sky based on time of year, may indicate that glare will be a problem for occupants. This might be the case if a person is watching TV in a sunlit room. Actuators may respond to this by **autonomously** adjusting the shading system to cover the TV screen, while leaving the majority of the room in sunlight.

Prediction of where glare might be an issue is an inherently complicated procedure to automate, since it will be based on factors that include human opinion. **Manual** override of an automated response will be necessary if the system judges inaccurately that glare is bothersome.

ix. **Visual Protection:** This requirement refers to the need to protect the privacy of the home occupants from the outside. It is most easily **automated** at night, when the system knows where the occupants are in the space, and that lighting levels are such that blinds are required to obscure the view into the home.

Manual control is necessary at unpredictable times when the occupants decide that extra privacy is necessary.

Persuasive information may be required to suggest what type of visual protection is most appropriate to use. In certain instances the user may decide to use shades external to the glazing to obscure the view to the interior, but prevailing climatic conditions may be such that solar gain is beneficial, and the system could advise that internal blinds should be used instead, allowing most of the solar radiation through the glazing while protecting occupant privacy.

x. **Visual Contact and Transparency:** Converse to the façade protecting the privacy of the occupants, it must also allow for a view of the outside environment. The area of glazing on the façade is the metric in this case, and potentially has a significant impact on the energy efficiency of the home.

xi. **Security:** The façade should be designed in such a way that it provides a reasonable level of passive security against burglary of the home. In relation to **automated** control, the object motion and passive infrared sensor network can easily be used to detect when there is an unexpected presence at the façade, and act as a burglar alarm.

xii. **Protection from Mechanical Damage:** Similar to sun protection, this is one of the basic requirements of the façade. Protection of the occupants and interior space from debris of all kinds from the outside is achieved passively by the primary glazing and insulation layers. **Automated** and active protection may be provided by an external layer of shades during periods of high winds or other exceptional circumstances.

xiii. **Noise Protection:** In addition to insulating the home from unfavorable temperature gradients, the façade will provide sound insulation from the external environment. **Automation** may be used to detect sound levels above an acceptable threshold, and close the actuated layers of the façade, which may include windows and external shades, to attenuate the problem.

Manual control will be necessary at times when the occupant's preferences do not align with the system settings for acceptable noise levels.

Persuasive information may be employed to inform the user that while closing the windows will help reduce the noise levels for a given scenario, it will also necessitate the use of mechanical cooling, and thereby increase energy consumption in the home.

xiv. **Fire Protection:** This requirements outlines that the façade should not only offer an acceptable level of protection from fire in adjacent apartments, but also provide a point of egress from the home in an emergency.

xv. **Energy Capture and Storage:** Outlined as one of the main points for increased energy efficiency in the home in Section 2.1, this point refers to the potential to capture solar energy at the façade for electricity generation, or hot water heating. An **automated** system would be employed to regulate the energy storage, as well as adjust the energy capture device, which may be an array of photovoltaic panels positioned on adjustable external, to optimize performance.

2.2 Energy Simulation

2.2.1 An Introduction

The following energy analysis study was carried out to explore the viability of producing a single façade module chassis that would form the basis for all customized modules equipped for different locations around the world. Without considering the specifications of final designs, the analysis attempts to give an indication of the energy saving potential of a basic façade that is highly insulated close to Passiv House [16] standards.

For this study an energy analysis tool named Design Builder [17] was used. Designed as a front-end graphical user interface to the industry-standard U.S. Department of Energy's EnergyPlus [18] simulation program, Design Builder offers an extensive number of configuration options to analyze a given design.

2.2.2 Cities Analyzed

In this study, six cities where chosen for analysis using Design Builder. They include:

i. Boston, U.S.A.
ii. Los Angeles, U.S.A.
iii. Beijing, China.
iv. Mumbai, India.
v. Quebec, Canada.
vi. London, England.

The motivation for choosing these six was to give an array of results applicable to different climate types, and in cities in both the developed and emerging economies, where urbanization is a growing phenomenon.

The conventional Koppen-Geiger climate classification system divides the globe into six climate groups:

A: Equatorial

B: Arid

C: Temperate

D: Continental

E: Polar

H: Alpine

These Koppen-Geiger groups are further sub-divided based on annual precipitation and temperature, resulting in a 2-4 letter code representing each region. It is noted that the majority of the world's population is concentrated in regions that fall under the A, C and D groupings; hence this analysis concentrates on these regions.

Note, however, that when considering a high-performance building façade, the minutiae of a given climate classification, such as the precipitation levels, do not significantly affect the operation of the HVAC equipment. For the purposes of this simplified study, the climates are classified as those that;

i. Require active heating for an extended period during the year.
ii. Allow natural ventilation to condition the internal space.
iii. Require active cooling for an extended period during the year.
iv. Require active cooling and dehumidification for an extended period during the year.

For the purposes of this study, these four simplified climate descriptions are dubbed *HVAC influence types*, and numbered 1 to 4 as above;

i. Influence type 1 (cold)
ii. Influence type 2 (temperate)
iii. Influence type 3 (hot and dry)
iv. Influence type 4 (hot and humid)

Note that while there are other factors that influence internal air temperature, mean radiant temperature, and humidity, such as direct sunlight exposure due to orientation, latitude, or region (which would have to be dealt with using appropriate shading), the main influences related to the Koppen-Geiger climate classifications are the external temperature and humidity, and a given climate zone may experience several or all of the "HVAC Influence Types", depending on time of day or year. Therefore, it is more appropriate to think about how many of the four influence types a given HVAC system should be capable of coping with.

Table 1 describes the climates of each of the six cities, including information on how they are defined within the Koppen-Geiger system, the average max. and min. annual temperatures, and how many of the four *HVAC Influence Types* are likely to be experienced throughout the year in the city.

One might find, for example, that a particular region may experience only a short period of cold weather in any given year that would require heating, in addition to domestic hot water heating, to be used, but such a scenario still necessitates some kind of heating solution, as may be the case for Los Angeles in Table 1

			HVAC "Influence Type"			
City	**Koppen-Geiger classification code**	**Average (min./max.) annual temperatures (°C)**	**Type 1 "cold"**	**Type 2 "temperate"**	**Type 3 "hot & dry"**	**Type 4 "hot & humid"**
Boston	Dfa: Snow, fully humid, hot summer	min -7. max 27.	☑	☑	☑	☑
Los Angeles	Csa: Warm Temperate, steppe, hot summer	min 8. max 28.	☑	☑	☑	☑
Beijing	Dwa: Snow, dry winter, hot summer	min -10. max 31.	☑	☑	☑	☑
Mumbai	Aw: Savanna (Equatorial, winter dry)	min 19. max 33.		☑	☑	☑
Quebec	Dfc: Snow, fully humid, cool summer	min -17. max 24.	☑	☑		
London	Cfb: Warm temperate, fully humid, warm summer	min 2. max 22.	☑	☑		

Table 1: City and Climate Classifications [19]

2.2.3 Analysis Settings

The study considers a 3-bay apartment with total floor space of 1130 sq. ft., which is a typical size for a 2-bedroom, developer-built apartment. In order to properly control the analysis, heat transfer between the apartment and the outside climate was only modeled for one wall, that to which the façade module is attached. The other walls, floors and roofs were considered adiabatic.

The study concentrated on 2 metrics in particular, the overall improvement, or decrease, in energy use for the apartment over the course of a year above that of a more poorly performing building, and the decrease in cooling energy use when adaptive shading is employed.

The first metric, overall energy decrease, essentially considers if it is justifiable to equip the standard façade chassis with very high-performance insulation for all global regions where the CityHome may be constructed. The second metric, that looking specifically at cooling energy decrease, gives an understanding of how the adaptive shading, in isolation, improves the annual energy performance over an identical building which does not employ automated shading.

The overall (annual) energy improvement is compared to two different wall construction types, a low thermal resistance (U-value of 1.2 W/m^2K), and a medium thermal resistance (U-value of 0.6 W/m^2K) wall. In comparison, the robotic façade module will be designed to include insulation rated with a U-value of 0.12 W/m^2K (R=8.3 m^2K/W).

For both the low-resistance and medium resistance walls, double-glazing, with a U-value rating of 2.708 W/m^2K is used, whereas the robotic façade will use triple-glazing, with a low-emissivity coating, rated as having a U-value of 1.949 W/m^2K.

In relation to the cooling energy analysis, that looking at the control of the exterior shades in isolation, the algorithm used in EnergyPlus considered the blinds to be closed whenever the solar irradiance on the façade is over a given 70W/m^2 threshold.

Table 2 outlines the results of the analysis in three columns, the first two giving the percentage improvement of the total heating and cooling annual energy usage over the low thermal resistance (Low-R) insulation, the second over the medium resistance insulation, and the third column the improvement that can be expected through the use of automated shading. The percentage improvements in Table 2 are based on the raw data for the total annual energy usage given in Table 6 on page 125.

A more detailed list of analysis settings is given in Table 4 and Table 5 on page 123.

2.2.4 Analysis Results

		Total (heating & cooling) Annual Percentage Improvement Over Low-R	Total (heating & cooling) Annual Percentage Improvement Over Medium-R		Percentage Decrease in Cooling Energy
Boston	North	10.5	5.1		12.3
	South	5.5	2.0		24.9
	East	8.9	4.0		22.3
	West	8.8	4.1		27.2
LA	North	5.1	2.1		19.1
	South	3.3	0.9		38.3
	East	4.2	1.5		29.7
	West	4.6	1.9		38.3
Beijing	North	9.6	4.8		7.3
	South	4.1	1.3		15.4
	East	7.6	3.5		13.8
	West	8.4	4.0		17.5
Mumbai	North	3.5	1.7		7.6
	South	6.7	3.5		15.2
	East	5.8	3.0		12.5
	West	6.1	3.1		13.9
Quebec	North	11.2	5.5		18.8
	South	6.8	2.7		35.2
	East	9.6	4.4		29.5
	West	9.4	4.4		39.1
London	North	10.5	5.0		20.4
	South	7.2	3.0		36.9
	East	9.4	4.3		30.7
	West	9.2	4.3		37.0

Table 2: Energy Analysis Results

2.2.4.1 Analysis Results: Discussion

From the results in Table 2 it is apparent that while overall energy savings accrued through the use of high performance insulation over the low-resistance and medium-resistance options are modest in some instances, each of the cities and climates considered did show a decrease in energy use. It is suggested that while insulation rated to Passiv House specifications was tested, and indeed the final design will consider other Passiv House elements, the apartment is not designed to manage air-flow as strictly as a Passiv House, so the results could not be expected to show equivalent savings. This first set of results does, however, justify the use of high-performance insulation, a feature which will be carried through to the final design.

In relation to the second set of test results, those considering the use of automated shading to reduce cooling loads; one can see that in each case, the shades helped reduce energy consumption, with some cities and orientations experiencing a reductions close to 40%. The potential to save such a significant amount of energy, coupled with the added functionality offered by external automated shading, such as security and wind protection (see Section 2.1.2.2), makes this technology an attractive standard inclusion for the robotic façade chassis.

Note, however, that this study is intended as a first approximation analysis, and does not constitute an in-depth look at the real-world performance of the CityHome.

It should be considered that the size of the glazing has a significant impact on the energy use in a home, which is why high performance homes limit the glazed area. This study was undertaken before the final Robotic Façade design was envisaged, and uses a glazed area of

40%. The final design increases this percentage to over 60%, which would have a detrimental impact on performance. For further details on this design decision refer to Section 4.1.1.

It is noted that of an air change setting used in these studies, that of 0.49 roomfuls per hour, and given in Table 5 on page 124, is indicative of a very tightly-sealed home, which may not be achieved in the CityHome, which does not strictly adhere to building energy practices such as the Passiv House specifications. This setting has a dramatic impact on energy use, and could be revised upwards by a factor of three or four.

Furthermore, this analysis using Design Builder considers the heat transfer through a single side of the CityHome apartment. There are cases when the apartment will be positioned at the corner of the building, giving two walls that are exposed to the outside elements. In addition, the assumption of an adiabatic interface at the other sides of the apartment is an imperfect one, and will lead to another source of error that should be taken into consideration when viewing the results in Table 2.

3. A Discussion of Mass-Customization

The topic of mass-customization was mentioned briefly in Section 1.1 in relation to how the CityHome was conceived as a form of modular construction, based on a standardized chassis, but with customizable, and transformable, interior modules, as shown in Figure 9. The Robotic Façade will also incorporate elements of mass-customization, with the following sections describing the design choices made in relation to what aspects of the façade module should be customizable.

Joseph Pine, author of *Mass Customization: The New Frontier in Business Competition* [20] is often quoted to describe mass-customization, stating that:

> "Customers don't want choices; they just want exactly what they want…"

The point Pine is making here is essentially that fully satisfying customer needs and desires is difficult, since almost everyone has different needs and desires. Since the dawn of the industrial revolution, if a company wanted to make a product at high volume and low cost, then it traditionally had to limit customer choice. This mass-production approach counters that of conventional residential architecture, where a home is a one-off architect design, and fully customized for a new owner. Mass-customization aims to provide the customer with the economies of scale of mass-production, but with a wider range of choice, through the use of agile design and manufacturing.

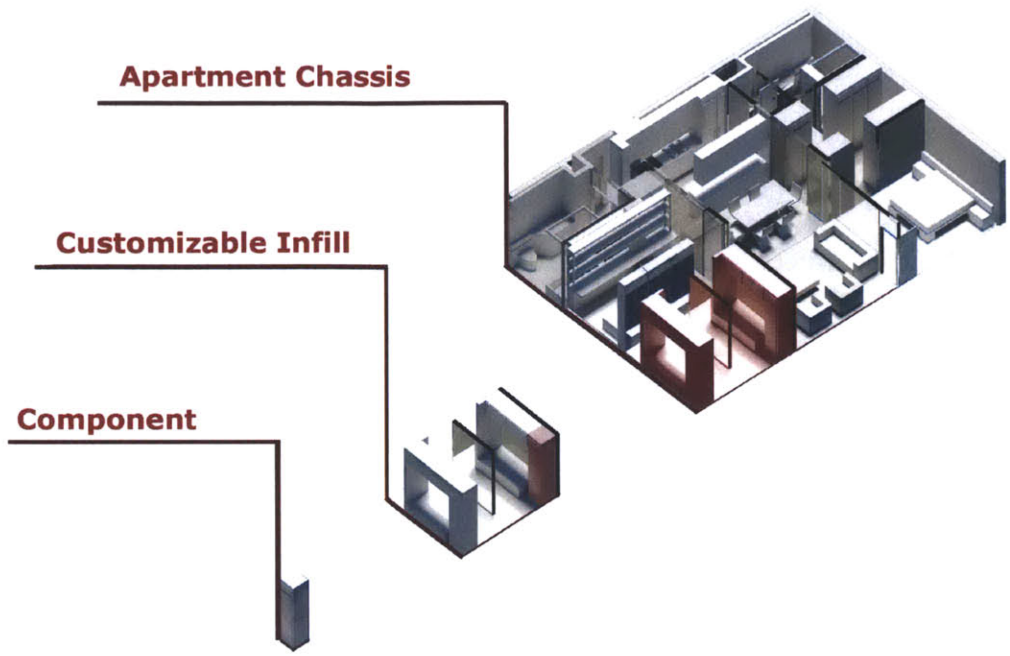


Figure 9: Apartment breakdown into chassis and infill.
Credit: Carla Farina, MIT House_n Research Consortium

During the final Robotic Façade design stages, three primary guidelines, or spectra, were considered in order to define where the final design should offer choice that is most beneficial and viable. These spectra are outlined as;

i. **Service versus Desire:** The product has been envisaged based on certain design requirements (see Section 2.1.2.2), essentially services that the end user needs, and this is balanced against options that the designer or end-user might want, or desires.

ii. **Enabling versus Offering:** How much customization after the initial installation should the Robotic Façade offer. The initial offering may be customizable prior to the modules being delivered, but this spectrum asks if the product should be customizable, or enabling, afterwards, like the transformable furniture of the CityHome interior.

iii. **User-centric versus Design-centric:** User-centricity essentially refers to the degree to which mass-customization is used. While inherently cheaper than traditional one-off customization, there is still a certain price-penalty paid for choice, even with the savings offered by large production batch sizes. A design-centric project refers to the experience being dictated by the designer alone, where little choice is offered, and essentially this spectrum asks where it is most appropriate to offer choice so that it is appreciated, and where should the product be standardized.

3.1 Designer-prescribed Features versus Customization Options

It was decided that the first version of the Robotic Façade design should be built around a non-configurable chassis to house the human-occupiable and HVAC spaces. The position of the HVAC equipment in the façade module, while varying depending on which bay of the apartment the module is attached to, would always result in a positioning that places the equipment as close to the central duct-work lines as possible, as described in Figure 8.

Additionally, the glazing size, glazing type, insulation specification, external shading mechanism, internal blind technology, and window opening mechanism are all pre-defined. The main customization options will be offered in the HVAC equipment choice, external shading design, and overall façade module color scheme. Configuration of the design has been limited in this initial design to improve the likelihood of developing an economically-viable product, with a choice selection that does not overwhelm the consumer. Similar to the CityHome's transformable living space, the Robotic Façade will provide the user with what was described in point ii. above, as greater "enabling" choice, or the ability to define how the space will be used post-installation; allowing transformations from a six-foot extension of the main living space, to a winter garden that can trap warm air for release into the rest of the CityHome.

While the degree to which mass-customizable has been integrated into the Robotic Façade design has been limited in the design, included in the appendix on page 132 is a description of how future versions of design of the Robotic Façade can expand upon the concept in the future.

3.1.1 Customization of the HVAC Systems

While not the primary focus of this project, it is intended that the Robotic Façade will be configured using a HVAC configuration tool, giving an optimized selection of HVAC technologies, based on a limited number of user inputs, and without the need for a consulting engineer. The standardized CityHome and Robotic Façade chassis, in addition to the utilities network, reduce the number of interdependencies significantly for such an optimization procedure, and based on the design tree given in Figure 10, as well as the design calculations from Section 0, the following is a description of the HVAC equipment options.

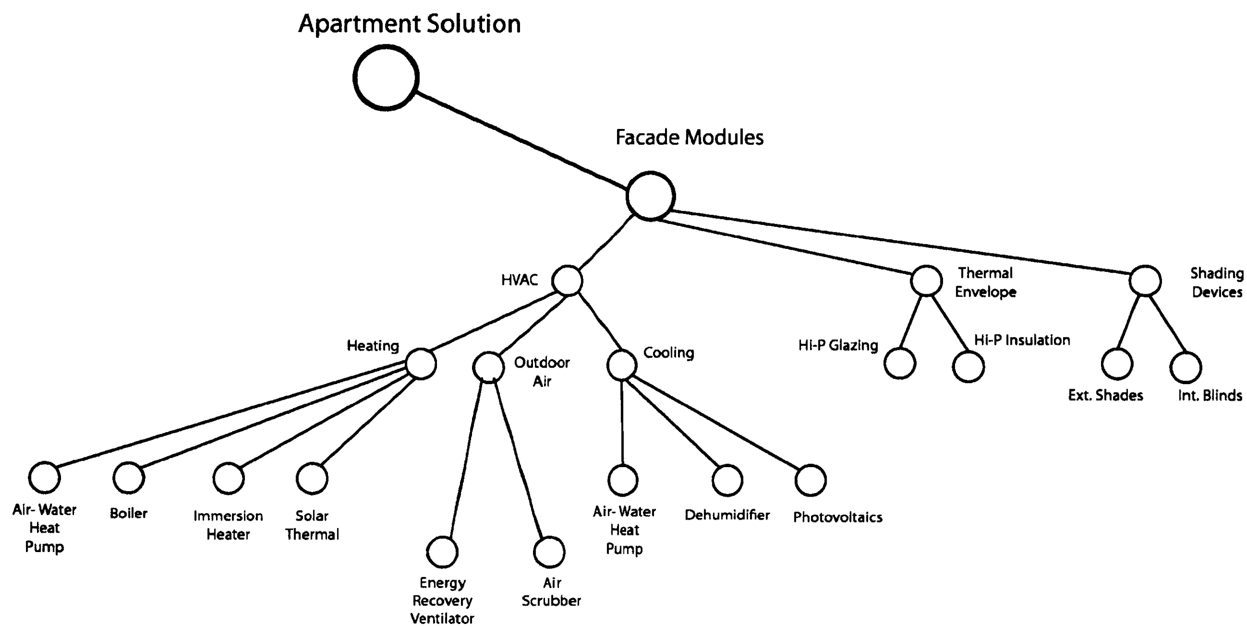


Figure 10: HVAC Design Tree

Once again, the HVAC design is based upon a standard "chassis", or base configuration, that is used with every façade module. Optional components and technologies are those that can supplement the standard solution, as advised by the automated configurator. Table 3 outlines the breakdown of standard and optional technologies. Note that Table 3 again uses the term "HVAC Influence Type", as described in Section 2.2.2.

	HVAC Influence Type		
	Type 1	Type 2	Types 3 & 4
Natural ventilation	**Standard**	**Standard**	**Standard**
Radiant heating & cooling	**Standard**	**Standard**	**Standard**
Air-water heat pump	**Standard**	**Standard**	**Standard**
Boiler	**Optional**		
Immersion heater	**Standard**	**Standard**	**Standard**
Humidity control	**Optional**	**Optional**	**Standard**
Blinds	**Optional**	**Optional**	**Optional**
Shades	**Standard**	**Standard**	**Standard**
High performance glazing	**Standard**	**Standard**	**Standard**
Energy recovery ventilator	**Standard**	**Standard**	**Standard**
Solar thermal heating	**Optional**	**Optional**	**Optional**
Photovoltaic energy capture	**Optional**	**Optional**	**Optional**

Table 3: HVAC configurations

3.1.1.1 HVAC Systems: Further Description

Based on Table 3, the following is a description of the choice of standard and mass-customizable heating, ventilation, and air-conditioning equipment.

Natural Ventilation: Whenever possible throughout the year, and for all climate types, the Robotic Façade will use natural ventilation. This methodology attempts to achieve thermal comfort in the living spaces without the use of mechanical air conditioning. It will take the form of operable windows, as well as an air-inlet through an Energy Recovery Ventilator. If the space is sensed to be too warm, the façade allows outside air in to attempt to cool the interior space.

This approach can achieve large energy savings by displacing the use of active, energy-consuming systems. The control of the natural ventilation system can be fully automated, or users can be instructed about the correct procedure using sensing technology in conjunction with just-in-time information based on current weather conditions. This just-in-time information will be applicable to the operation of the windows, which will not be mechanically actuated in the Robotic Façade (see Section 4.1 for a detailed description of the final design).

There are, however, limitations to this approach, since natural ventilation, just like many "free" energy sources, is not completely reliable. Even in the few regions around the world where natural ventilation may be sufficient to keep the interior space comfortable year-round, there may be exceptional circumstances, such as a larger than normal gathering of people in the home, when mechanical systems are necessary.

Radiant Heating and Cooling: A hydronic system, delivering heated and chilled water to the living space via an under-floor pipe network, was chosen as the primary mechanical HVAC technology. After a review of the potential for the technology to perform as required, given in Section 2.1.1.1, the radiant system has been designed as the primary active control for the interior air temperature and mean radiant temperature.

Note, however, that the hydronic system has been designed to supply, in conjunction with the air-to-water heat pump, either heated or chilled water on any given day. While different zones of the home can be heated or cooled to differing degrees, the system cannot be used to be heat and cool simultaneously. The immersion heater can provide domestic hot water on days when the heat pump is being used to supply chilled water to the floors, but the design is limited to natural ventilation of sub-zones of the CityHome that require conditions that are not being met by the under-floor system. While the optimization of this system is not the focus of this project, it is noted that careful consideration as to when the hydronic system should switch from a heating to cooling mode, or vice versa, is required. It is also noted that to gain the full benefit of the heat-storage potential of a water-based system, this transition should not be made on a daily basis, instead, a long-range forecast of required heating or cooling requirements should dictate when adjustment is appropriate.

Air-to-water Heat Pump: An air-to-water heat pump was chosen as the primary means of supplying heated and chilled water to the hydronic system. A heat pump operates in a similar manner to a conventional air-conditioning unit, using a reversible vapor compression cycle. The pump will have the capability of both heating and cooling water, where the objective is to transfer heat from the outside air into the interior space during heating, and to transfer heat from the interior to the exterior during cooling. In this case the heat is transferred to, or removed from, water, which can be used for domestic hot water, in addition to the radiant heating.

The use of a hydronic system offer the advantages associated with thermal storage in the water, so the device can therefore be sized smaller than would be necessary without heat storage.

While the air-to-water heat pump is a standard technology in the Robotic Façade HVAC chassis, it is limited by the fact that it has diminishing efficiency as the temperature of the outside air drops. There is a point at which it is not economical to install a heat pump over a conventional heating system. An electrical heater may be incorporated into the design to supplement the energy harvested from the outside air in colder conditions, but this may be subject to high electricity prices compared to fossil fuels, depending on the region in the world. In this case, a supplementary boiler may be required. Also, while the heat pump may provide both heating and cooling, this inherently means that the device is running for longer throughout the year, which, consequently, will adversely affect the machine lifetime.

Boiler: It is suggested that a small boiler be offered as an option in climates that experience a particularly cold winter. The boiler itself may be powered by a variety of fuel-types, and will integrate easily into the hydronic system. Thermal storage can potentially offer the same advantages as an air-to-water heat pump, lowering the required boiler capacity in comparison to a furnace using air as the delivery medium.

One of the potential limitations of this technology is the fact that the use of an electrical boiler can be cost-prohibitive in certain cases, and the infrastructure to supply a fossil fuel to the system may be prohibitively complicated.

Immersion Heater: An immersion heater, or electrical element that heats the water in the hot water tank, has been included as a standard piece of equipment in the HVAC array. It can be used in conjunction with a boiler to supplement heating in times when demand exceeds the boiler capacity, and can also be used to supplement a heat pump, or provide domestic hot water at a time when the heat pump is providing chilled water for space cooling.

Note that while an immersion heater offers flexibility for use in many system configurations, and is very simple, reliable, and cheap to install, it is also potentially very expensive to run.

Humidity Control: This control may be for humidification/ de-humidification, and necessary to prevent condensation on the radiant floor at certain times. Humidification may be achieved by injecting water vapor into the incoming air, while de-humidification may use a desiccant or simply pre-cool the incoming air below its current dew point, given the current outside temperature and relative humidity level. The technology will be incorporated into the Energy Recovery Ventilator.

Blinds: Positioned inside the glazing, blinds have a significantly reduced ability to prevent solar gain in the interior space when compared to exterior shades. They primarily serve as a privacy screen, or to provide shading at times when solar gain is desirable.

Shades: Placed outside of the glazing, active shading, as investigated in the energy analysis in Section 0, can be used to lower the peak cooling loads by diminishing solar gain. Additionally, the use of independently-actuated zones of slats on the shades can provide variable shading levels for different areas of the living space.

High-Performance Glazing: Triple glazing, with a low-emissivity coating, and U-value of 1.95 W/m^2K, will be used to complement the high-performance insulation (U-value of 0.12 W/m^2K), as considered in the energy analysis in Section 0. It is acknowledged that this technology is expensive, and the improvement in energy performance will vary, depending on region, but it has been chosen for inclusion in the standard Robotic Façade system due to the dramatic energy performance gain in some regions, and the economies of scale that can be expected from the production of the standard Robotic Façade package in large quantities.

Energy Recovery Ventilator (ERV): The primary means of ventilating the CityHome to provide minimum air requirements, this device recovers both the sensible and latent (moisture content) energy from the exhausted air, and transfers it to the incoming air. Air filtration can also be incorporated. One can expect to recover up to 70% of the sensible energy from the exhaust air.

Solar Thermal Heating: This optional technology harnesses energy from the Sun by heating water for domestic and space heating use. This type of device is relatively simple, cheaper, and viable in more places around the world than a photovoltaic (PV) alternative.

The main limitation to the use of solar thermal technology is the complexity and weight that has to be borne to pump a working fluid to the façade.

Photovoltaic Energy Capture: This technology harnesses energy from the Sun and converts it directly into electrical energy. It can generate up to 190W/m^2 of energy, but while the device (solar paneling) is relatively light, and could easily be installed on the façade structures (louvers etc.), the climates and latitudes in which PV is economically viable are limited, and usually only attractive in conjunction with financial incentives. The technology is also highly dependent on orientation.

4. The Final Design

4.1 Structural Design

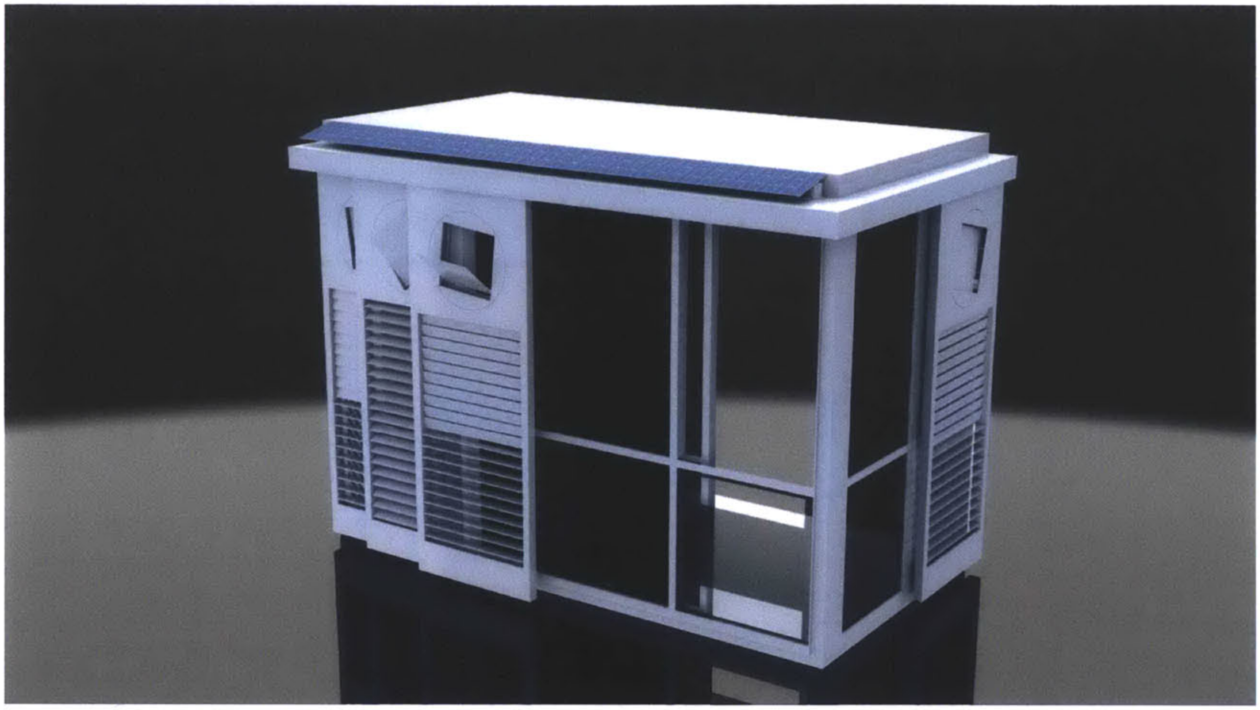


Figure 11: Final Robotic Façade Module

Shown in Figure 11 is the final design for the first generation of Robotic Façade modules. The following is a brief description of the structural features included in the design. Refer to Figure 48, Figure 49, Figure 50, and Figure 51 in the appendices for further technical drawings of the module design.

- The chassis, or structural frame of the module measures 12.5' x 9.5' x 6.5', giving a total volume of approximately 770 cubic feet.
- The HVAC equipment is housed in a dedicated cabinet, vented at the front to allow for air exchange, and fully insulted from the human-occupiable space in the façade module. The cabinet measures 3' in width, 6.5' in depth, and occupies the fully 9.5' height of the space, giving a usable area of 19.5 square feet, and a total volume of 185 cubic feet.

- The remaining section of the façade module is designed as a multi-functional extension of the living space, giving an extra 61 square feet of floor area, and a total occupiable volume of 585 cubic feet.
- The Robotic façade incorporates floor-to-ceiling glazing on two sides, with the main face of the module having a glazing panel measuring 90 square feet. The glazing on the side of the module measures 28.5 square feet.
- The glazing incorporates a transform, or horizontal rail, at a height of 38" above floor level, and has a manually-operable full-length window, measuring 3' x 9.5'. When open, a reinforced-glass balcony railing serves as a safety barrier.
- The module will use insulation rated with a thermal resistance (R-value) of 8.3 m^2K/W, and triple-glazing, with a rated resistance of 1.949 W/m^2K.
- A narrow ledge extends from the top of the module and wraps around both of the glazed-walls. It serves to house the motor and track equipment for the automated shades, as well as provide a platform for accommodate a photovoltaic panel in regions where such a technology is feasible (see page 128 of the appendices).
- The front wall of the façade module incorporates 3 separate shade panels. Each panel slides in front of the HVAC vent panel, exposing the fully-glazed area. Another shade panel is located on the side of the module, covering the smaller glazed area, and sliding back to the interface with the CityHome chassis to expose the glass.
- Each shading panel has two independently-actuated zones of movable slats, allowing fine-grained shading options.
- At the top of each of the shading panels is an actuated light-shelf, with two degrees of freedom. See Section 4.1.2 for further details.

Section 4.1.1 describes the reasoning behind several of the design decisions described here, and as shown in Figure 11.

4.1.1 Design Choices

Relating back to the design requirements outlined in the Section 2.1, the final façade module design considered various options before finalizing the structure shown in Figure 11. This section outlines the reasons why some of the design choices were made.

Relating to the requirement for the Robotic Façade to simplify construction, and house the majority of the heating, ventilation and air-conditioning equipment in such a way that the interface between the façade module and the CityHome chassis is standardized, the design opted to dedicate a cabinet space in the façade chassis to the equipment. Shown in Figure 12, the HVAC cabinet may be positioned to the left or right of the module, depending on which bay of the apartment it is to be attached to, and is positioned so that the equipment is always closest to the central duct line that serves to extract air from, and supply domestic hot water to the back of the CityHome, as described in Figure 8. The HVAC cabinet offers easy access to the equipment via a panel inside the human-occupiable space in the chassis.

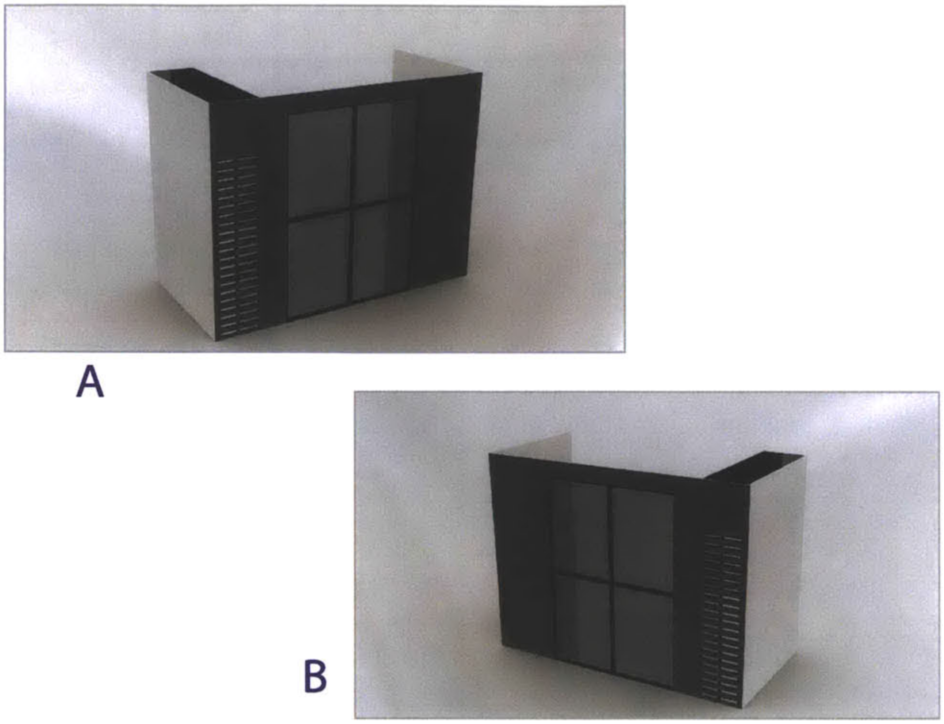


Figure 12: HVAC location

The choice of glazing area for the Robotic Façade is an example of the "Service versus Desire" spectrum discussed in Section 3. While strict adherence to issues of energy performance would limit the allowable glazed area on the façade, it was decided that a larger glazed area would enhance the user-experience to a greater extent. Referring to Figure 13, the design opted for full floor-to-ceiling glazing, equivalent to the six-panel version shown.

The CityHome places emphasis on intelligent use of natural light, since the apartment is relatively deep, and it is felt that this larger glazed area offers the potential to flood more of the living space in sunlight, as well as offering the potential to use the Robotic Façade module as a garden space.

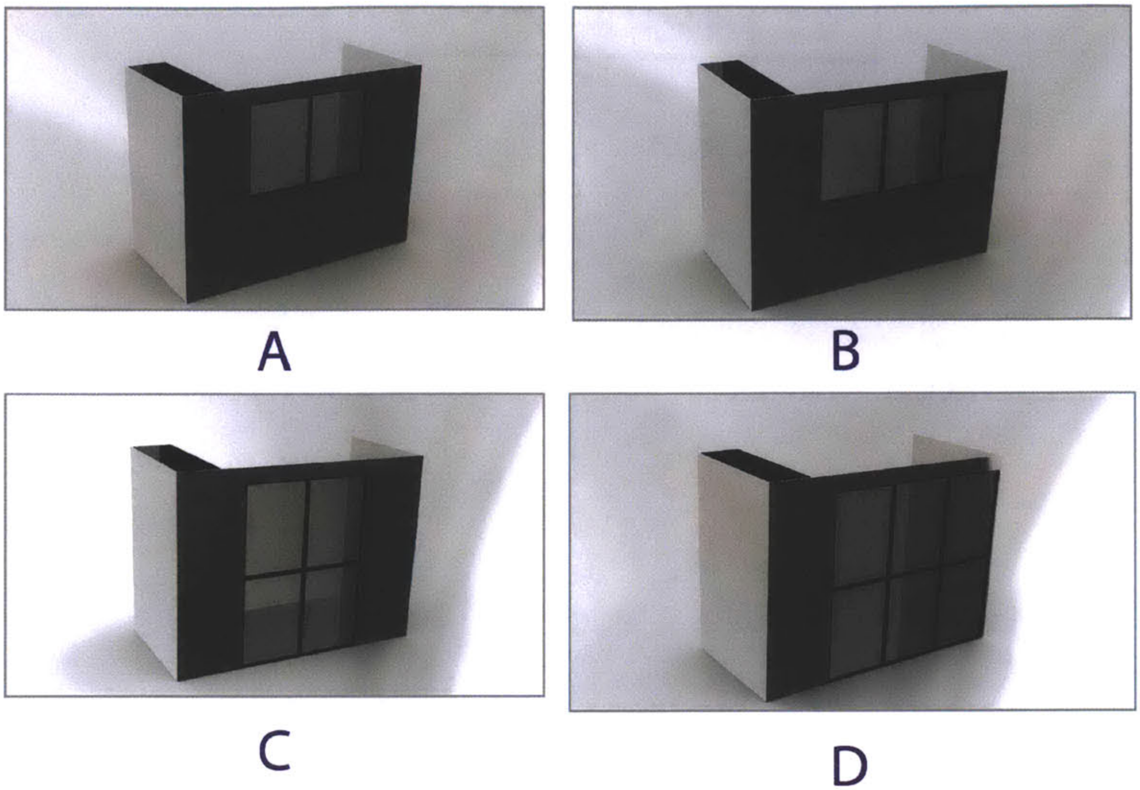


Figure 13: Window Design

The choice of window opening mechanism was critical to several aspects of the design. With reference to Figure 14, and from a security and safety standpoint, it was felt that neither the vertical-slide (images A and B) nor hopper (image C) mechanisms allowed for a satisfactory point of egress in the event of a fire.

In relation to the internal blinds, or external shades and insect screening, it was found that any mechanism that required the window to move outside the plane of the glazing frame, such as the hopper design, would be unacceptable.

The horizontal-slide mechanism (images D and E of Figure 14) was chosen for its simple and effective operation. Moving in the horizontal plane means that counter-weights are not required, which add complexity and potential for failure in a time of emergency for other designs.

A purely manually-operable window mechanism was chosen since it was felt that all natural ventilation could be automated through the air-intake at the energy recovery ventilator, thereby reducing cost and complexity related to the window operation. In addition, a manually-controlled mechanism allows for a tighter seal around the moving frame, increasing the energy performance of the façade.

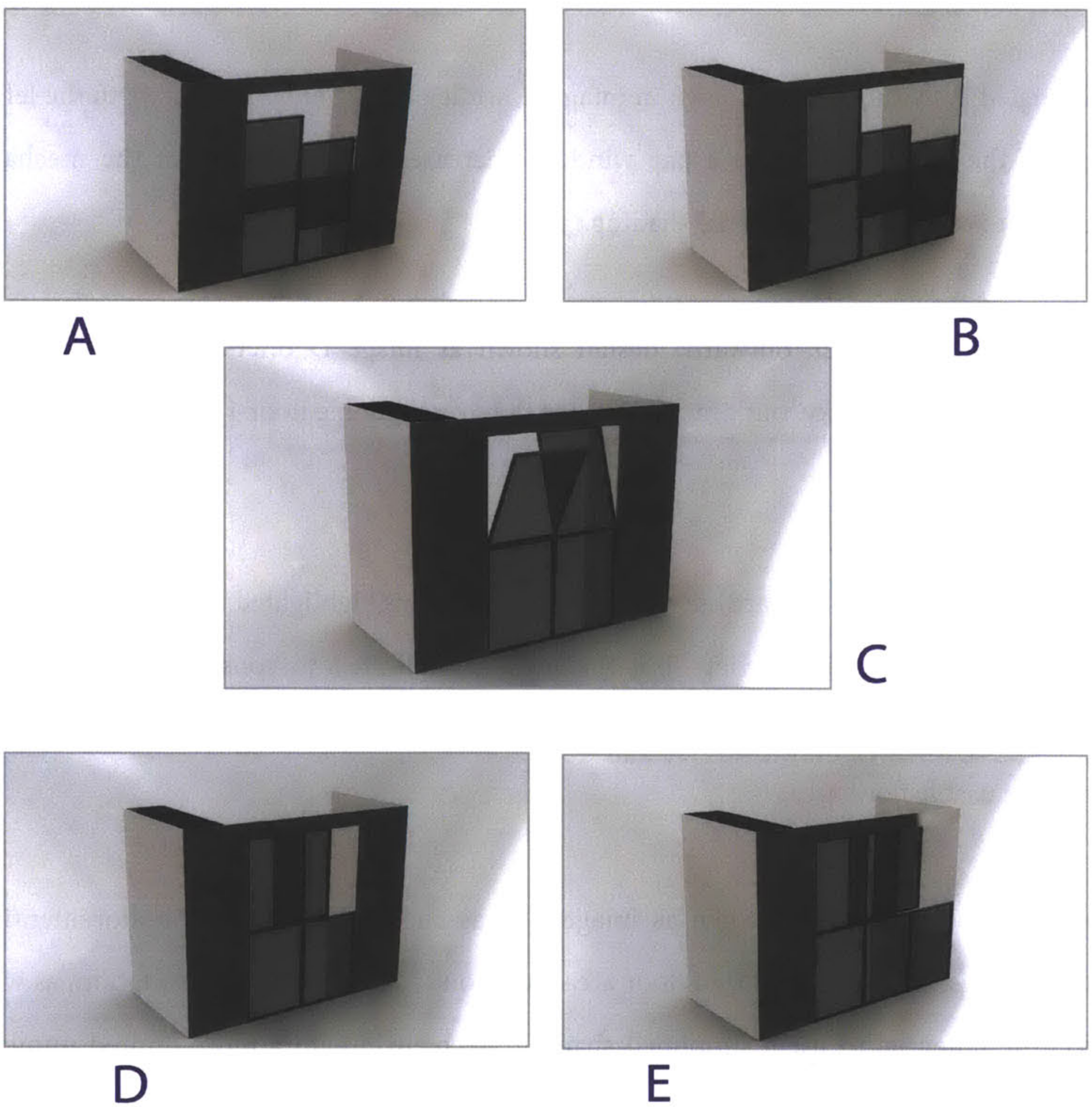


Figure 14: Window Mechanisms

The façade module is limited to a depth of 6.5 feet out from the CityHome chassis at the furthest point. The designs shown in Figure 15 give examples of the basic chassis shapes considered that adhere to this criterion.

Where a downward-facing design, as shown in image B, offers a novel view out of the CityHome, it limits the amount of solar radiation incident on the façade, in addition to limiting the usable floor area in the module.

The angled (vertical) design, shown in image C, while offering enhanced views to the left and right of the façade similar to a bay window, increases the complexity of the mechanism required to automate the external shading.

The angled (horizontal, outward) design shown as image D offers the attraction of an enhanced view of the sky, but again decreases the overall usable floor-space within the façade module.

The angled (horizontal, inward) design given as image E is highlighted as the most interesting from an aesthetic point of view, but the most complicated to construct and outfit with mechanized shades, since it would require two separate drive motors. Again, this design limits the usable floor area.

The simplest design, that shown as image A, was chosen because of the aforementioned shortcomings, since it maximized floor area based on the module depth limitation, as well as offers the potential to use a simplified shading mechanism with a more limited number of drive motors.

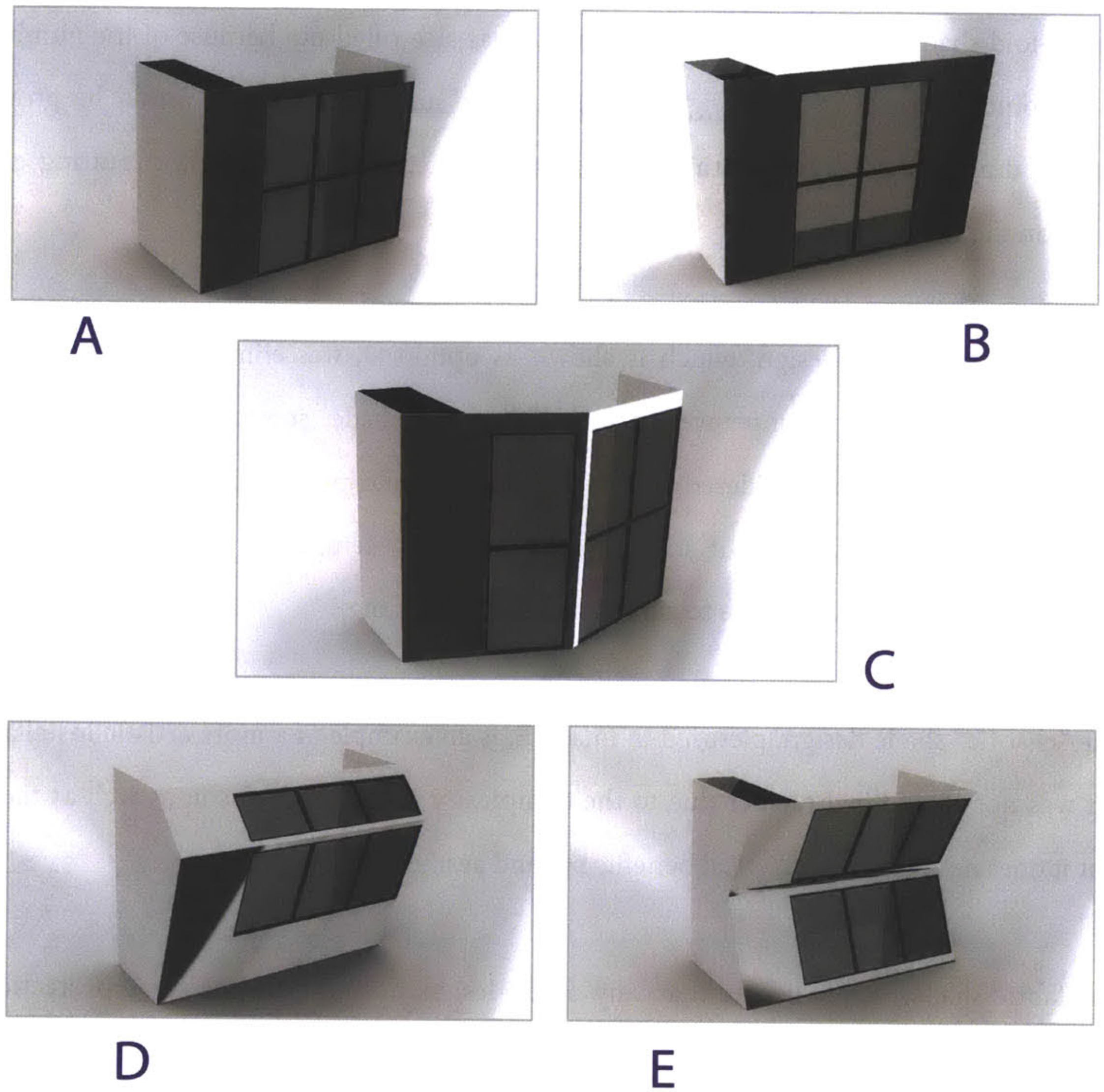


Figure 15: Module Shape

The shading mechanism design considered options which included those outlined in Figure 16, Figure 17, and Figure 18. The sliding mechanism shown as option A, while offering many favorable features, such as design simplicity, was rejected, since it limits the amount of glazing to that which can be accommodated between the shading panels when they are fully retracted, to half the width of the façade module at most.

The bi-fold shading system, shown as option C, was also ruled-out because of the number of pivot points inherent to the design. It was felt that such a mechanism would be prone to damage during icy weather, with the mechanism potentially get stuck and putting excess strain on the motors to operate.

The vertically-folding design, which is shown as option D, was eliminated because of the complexity of using a counter-weight to raise up the shade, something that is prone to breakdown. It was also considered that such a design could potentially be dangerous in the event of an emergency; without access to the control system, it may prove difficult for a firefighter to lift up such a shade to gain access to the building.

The "chaotic" shade design, pictured in image E, is an example of a more artistic installation, but was not considered further due to the complexity of the mechanism, as well as the fact that it does not offer any practical benefits beyond aesthetics.

The fabric shades, while lighter than any other design, do not offer the same protection as rigid designs, or the potential to have zones of independently-actuated slats.

The sliding shades, consisting of three separate panels, and similar to those shown in image B, were chosen for the final design since they offer the required safety, and fine-grained shading functionality, in a simple design, but since all the panels slide onto one another, allow more of the façade to be glazed (3/4 of the façade module width in this case). It is noted, however, that this design will require further refinement to prevent snow and ice from hindering its operation in colder climates during winter months. While this chosen design assumes that the shades are held from the top only, and simply offset from the façade on the bottom by a roller, it will be necessary to fix the shades at both the top and bottom if they are the perform as an extra layer of protection in the event of high winds.

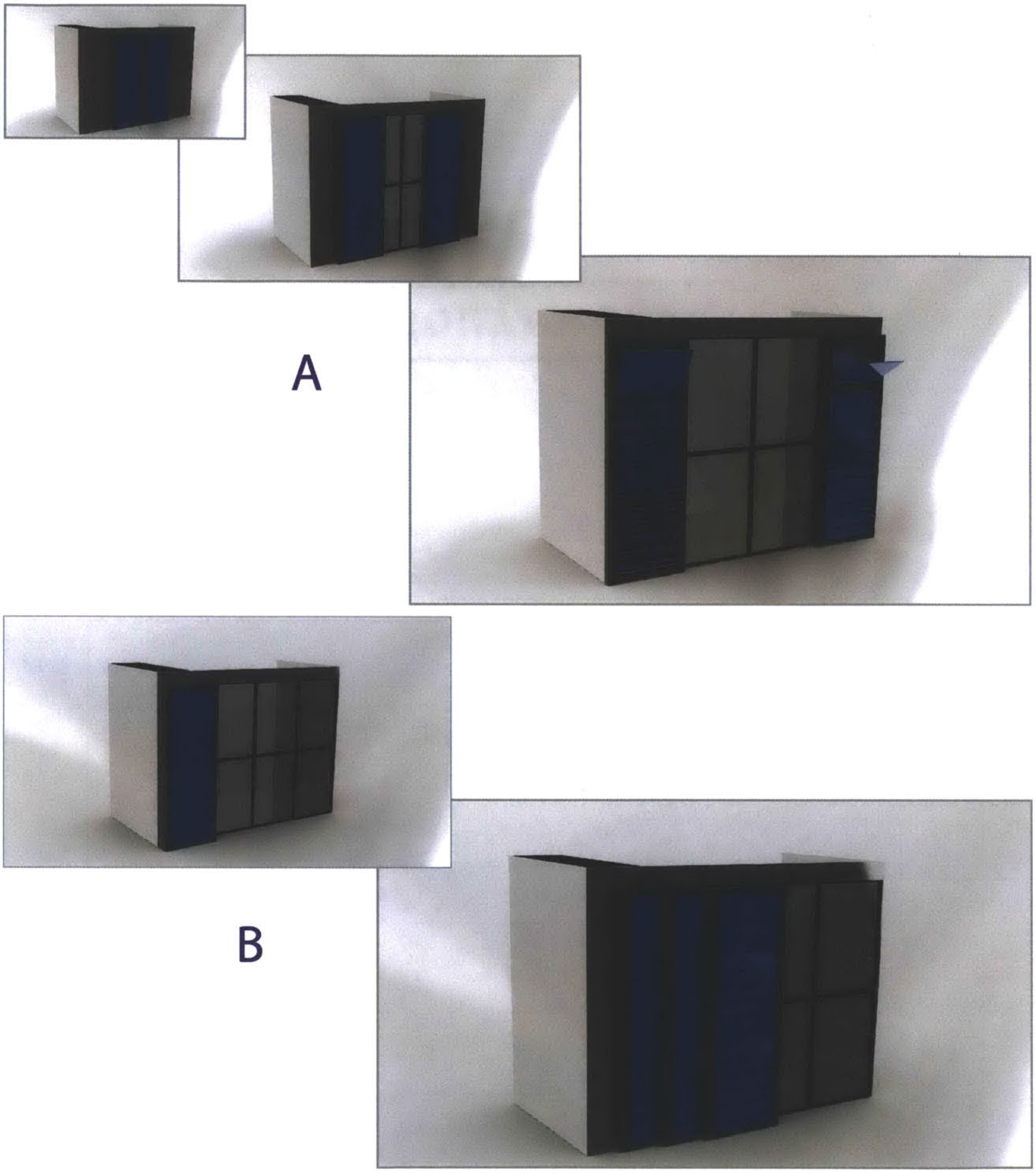
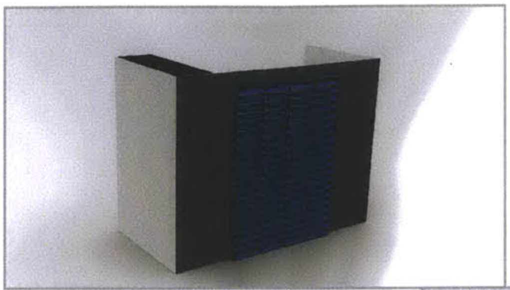
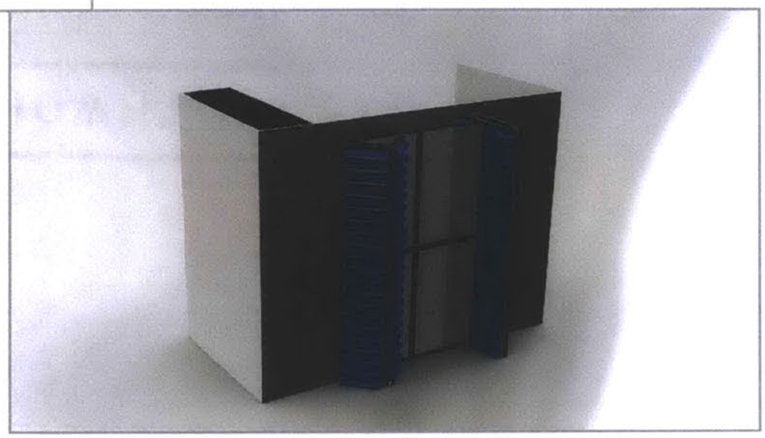
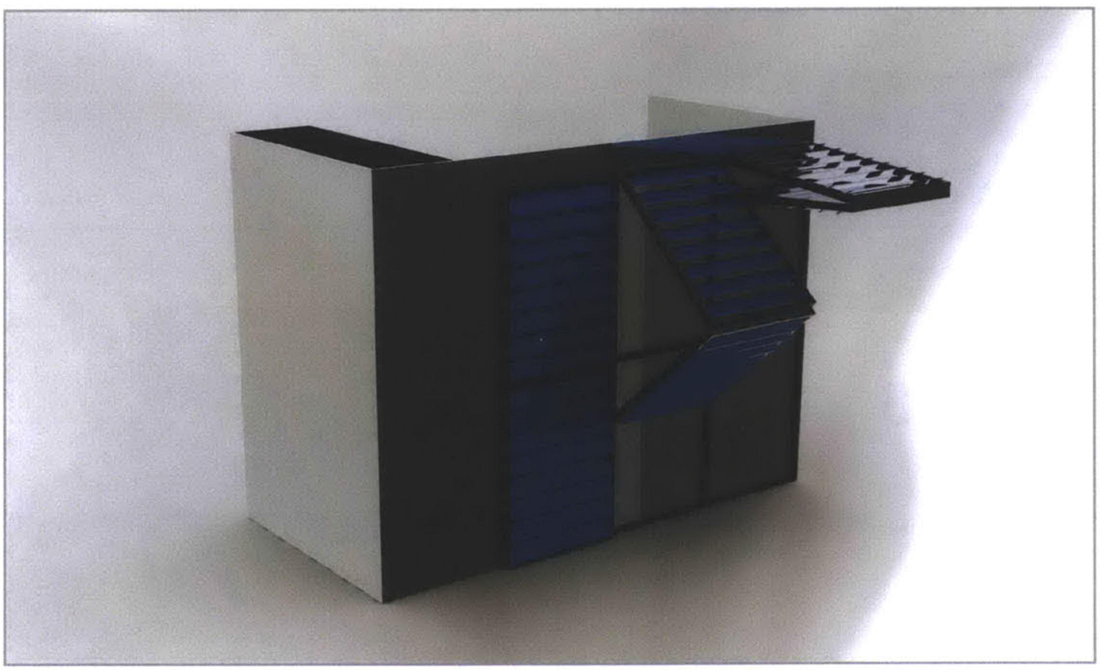


Figure 16: Shading Types (1)



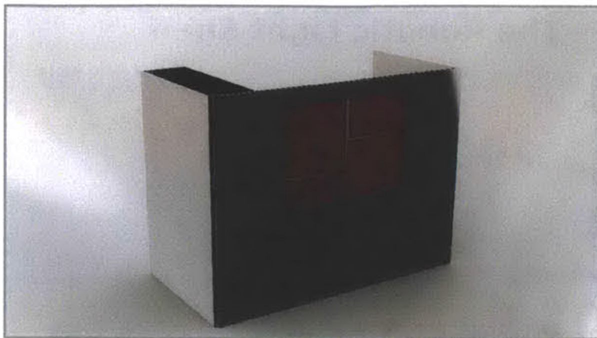
C



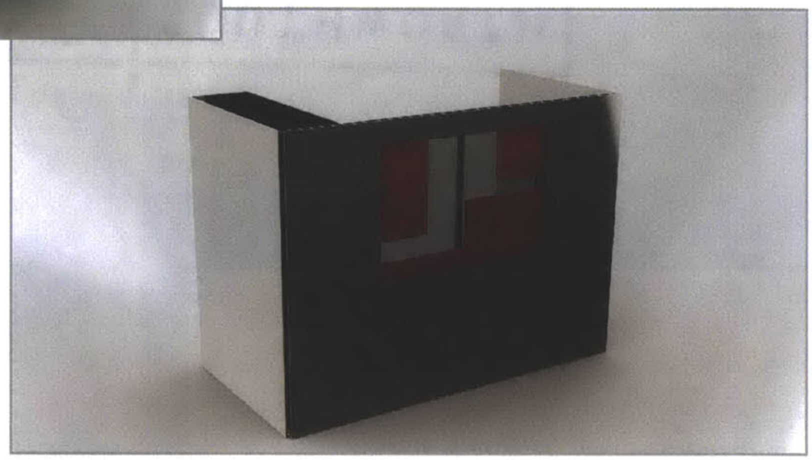


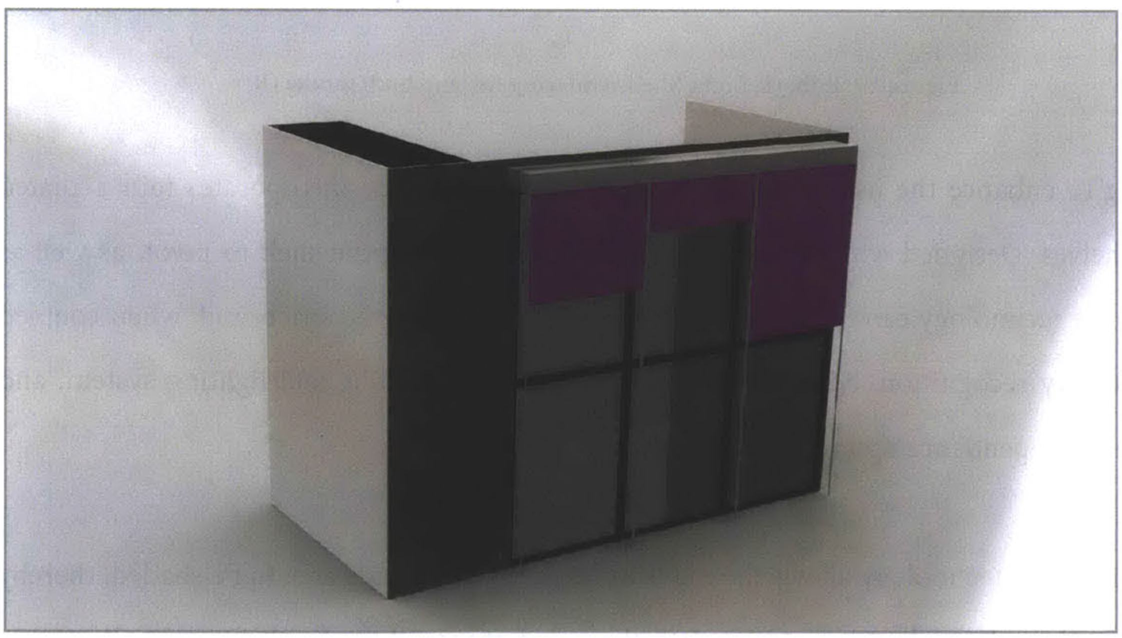
D

Figure 17: Shading Types (2)



E





F

Figure 18: Shading Types (3)

4.1.2 Personalized Sunlight-- The Robotic Light Shelf

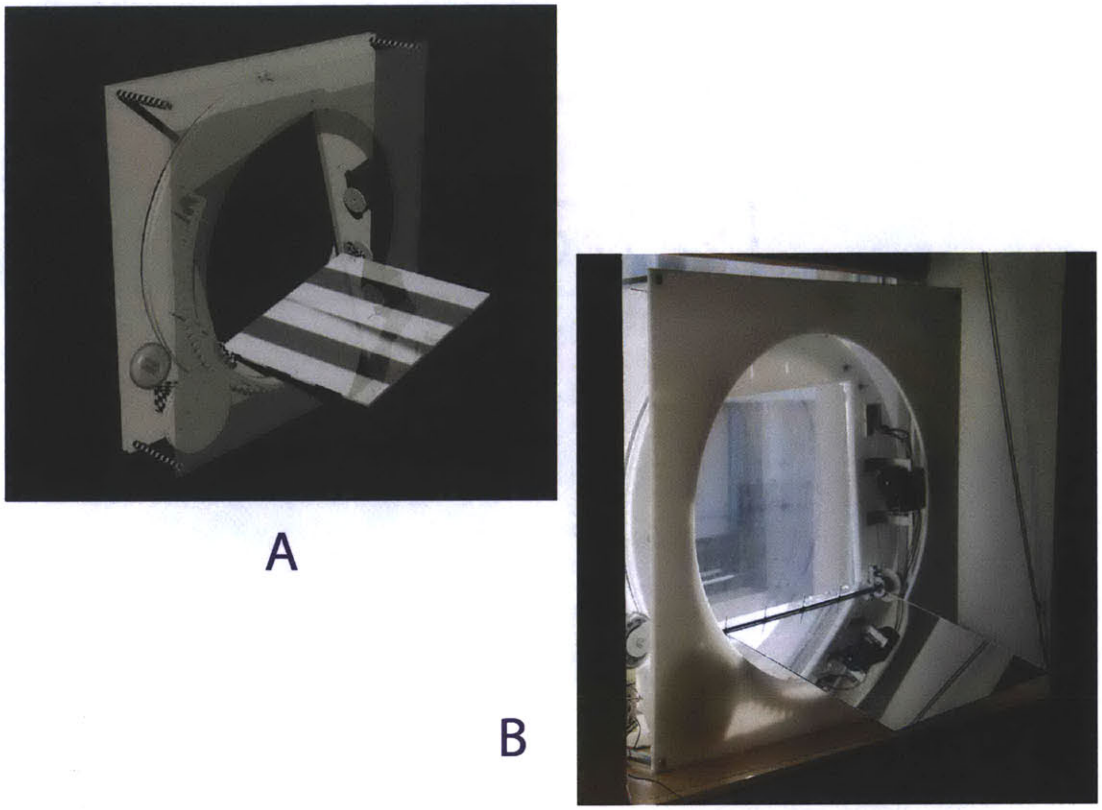


Figure 19: Robotic Light Shelf rendering (A) and built model (B)

Aiming to enhance the use of natural light, the Robotic Façade incorporates four actuated light shelves. Designed with two degrees-of-freedom, allowing the shelf to pivot, as well as rotate, the technology can be used to throw shafts of light into the space, and, when coupled with activity recognition, be used to create an intelligent shading and lighting system, and reduce the dependence upon artificial lighting.

Essentially this technology allows the majority of the glazed façade area to be shaded, thereby reducing the solar radiation entering into the living space, while simultaneously allowing a small unshaded area to reflect sunlight off the mirrored light shelf. The position of the reflected light in living space can then be specified by the user using a control interface to adjust the light shelf angles. The overall energy gain in the space due to sunlight can be reduced, but bright areas of natural light can still be achieved, with activity recognition being used to link certain activities to specific lighting scenarios, achieving a sense of "personalized sunlight".

4.2 Functional Design

4.2.1 Robotic Façade: Interaction with the User

This section outlines how the Robotic Façade will function during everyday life. As outlined in Section 2.1.2.1, the CityHome and Robotic Façade will interact with one another, as well as the home occupants, through an extensive array of sensors. Using activity recognition and sensor pattern information, the Robotic Façade will be able to:

- Regulate the temperature of the living space, allowing it to drift out of comfort conditions, thereby saving energy, when it is detected that nobody is in the home. Such an algorithm would adjust the thermostat set-points, taking into account the likelihood that the homeowner may only be away temporarily, and perhaps using a GPS-equipped system to accurately regulate this dynamic control. A previous project in the Changing Places group looked in greater depth at the potential to use this GPS thermostat technology.
- Regulate the necessary amount of domestic hot water produced by the system. Activity and occupancy recognition will be employed here to detect not only whether people are present in the home, but also the number of people. Based on this information, in addition to approximating functions that take the time of day into account, since demand for DHW may be greater in the morning, when occupants want to shower, the system can predict the demand for domestic hot water, and potentially save energy by not producing water when it will not be used. This may be of particular importance during the summer time, when the hydronic under-floor

system is used for cooling, and the domestic hot water is produced through the use of a relatively inefficient electric immersion heater.

- Predict when it is appropriate to use natural ventilation above the minimum air requirements for the space. An energy recovery ventilator (ERV) will serve as the primary air inlet to the CityHome, allowing energy to the exchanged at the interface. While this energy exchange is beneficial at times when the interior spaces are being heated or cooled through the use of the active hydronic system, the system will bypass the ventilator's energy-recovery mechanism when it is known that the outside temperature is appropriate for natural ventilation. This means that, for instance, if the air in the interior space has been heated to an uncomfortable degree by increased occupancy in the home, cool air from the outside can be drawn into the space, rather than using chilled water. In this case one does not want to recovery the energy from the hot interior air, so the ERV will be by-passed, while still directing the incoming air through a filtration system.

  If human presence is detected in the home, then the Robotic Façade may also inform the occupants that opening the window is also most energy efficient, given the current temperature gradient between interior and exterior.
- Detect when sunlight incident on the façade will have a beneficial, or adverse, impact on the performance of the system. During the winter in regions that experience low temperatures, solar gain, or the energy gained from sunlight passing through the façade, can be used to increase the air temperature in the living spaces. The converse is also true, that such solar gain may be detrimental on days when active cooling of the space is required. A combination of light intensity sensors, coupled with knowledge of the interior and exterior temperatures, will allow the system to adapt the automated external shading to regulate this solar gain for maximum energy performance.

- Detect when glare may be an annoyance for home occupants. Based on knowledge of the angle of the sun, given the time of day, time of year, and region in the world, in addition to information on what a given occupant is doing, and where in the space they are, the Robotic Façade can adapt the shading to minimize glare from the Sun. If, for instance, it is known that a given occupant is watching television, then the system can adjust the façade so that television screen is shaded, but the rest of the space is lit by sunlight. The functionality is facilitated by the two independently-actuated zones of adjustable slats on each shading panel, of which there are four. Each of these eight zones can allow sunlight to pass through to a variable degree, giving fine-grained control of the shading.
- Reflect natural light deep into the living space to where a conventional home would require artificial lighting. Facilitated by the open-plan nature of the CityHome design, in addition to the use of actuated light-shelves on each of the four shading panels, the Robotic Façade can bounce sunlight off of the light-shelf's mirrored surface and into parts of the home farthest from the windows. Coupling this functionality with activity recognition can allow occupants to define their own lighting scenarios so that, for instance, if it is detected that a person is reading a book at the kitchen counter, the actuated light-shelf can be used to position a "pool" of light on the ceiling above them, illuminating the space directly below.
- Predict when increased privacy is required, and adapt accordingly. The façade will incorporate external shades, in addition to internal blinds, which can both be used as privacy screens. In this case, the Robotic Façade may use online weather forecasts to predict the likelihood that sunlight will be shining on the façade in the morning. If it is found that solar gain in the morning would not be beneficial, then the system will use the external shades to protect the privacy of the occupants during the night, and if the solar gain would help the heating system, during winter time perhaps, then the internal blinds would be used.

- Caution the home occupants as to the impact that their actions have on the energy performance of the home. If the users are in the habit of relying on the active cooling and heating systems throughout the year, despite the fact that natural ventilation could be used part of the time, then the user interface that links to the activity recognition system can be used to inform the users of the energy wasted through such actions, with the appeal being made using real dollar amounts of money lost in an attempt to make the information more persuasive. The system can also use activity recognition to predict when a user is most likely to heed information presented to him/her; for instance, an occupant may be more receptive to advice when surfing the internet in the evening after dinner, rather than just as he/ she arrives home from work.

Interaction with the automated system, as mentioned previously, will be through a graphical user interface designed to have minimal user input. This approach contrasts with many home-automation systems, which provide the user with access to an abundance of customization options. It is felt that by limiting the number of choices that have to be made, the user will be more likely to use the system correctly, therefore save energy in the home more effectively. This user interface will allow the user to train the activity recognition software, receive feedback on the performance of the home, and also manage the manual override controls, and may be installed as an application on the Android mobile platform. See Section 5.5 for further details.

4.2.2 Robotic Façade: Interaction with the Outside World

While Section 4.1 described how many of the design decisions for the Robotic Façade were made based on issues of efficiency and cost-effectiveness, the design was still envisaged with architectural aesthetics in mind. No less applicable to the mass-customizable CityHome than a one-off architect-design home, the way the apartment communicates with the outside world is very important if the product is to be appealing to the consumer. Much of the appeal of the Robotic Façade lies in its energy management functionality, but several other aspects were considered during the design stages.

While not the intended goal of the design, it is true, among certain demographics, that being seen to be "energy-conscious" is in vogue. That is to say that many people take pride in the fact that they are aware of their impact on the environment. The Robotic Façade is given a distinct look by incorporating four adjustable light-shelves, not before seen in residential housing. Additionally, but conditional upon world region, the Robotic Façade has a prominent photovoltaic panel built-in, used to off-set energy use by the device, but again serving as a statement to the outside world that the homeowner believes in "green" alternative energy sources.

In addition to the "green energy" statement, the design of the Robotic Façade considered how the homeowners would be able to interact with one another using the façade module. This space projects out 6.5 feet from the side of the CityHome, and it was felt that this presents certain opportunities, as well as potential obstacles, for the success of the design.

A CityHome apartment complex may be made up of multiple floors, so certain design rules were developed so that the Façade Module would not overshadow an apartment below. Essentially this means that vertical shafts of modules, each positioned one on top of the other, are employed. In contrast, the module offers the potential benefit of allowing different homeowners to interact with one another, since the module offers a view to the side, rather than simply away from the building. Social interaction within an apartment complex setting is increasingly limited by the lifestyles of modern professional, so this interaction, albeit limited to visual contact, offers a potentially novel attraction.

5. Prototyping and Testing

In addition to designing the Robotic Façade, and conceptualizing the functionality of that design, this project also saw several aspects of the project fully-realized, from practical component-level design, through to construction and testing. This section outlines the development of three robotic light-shelves, the development of a functional graphical user interface equipped with activity recognition capability, the mockup of an android user interface based on the functional desktop version, and the deployment of sensors and testing of the light shelf and activity technology in a real apartment.

5.1 The Robotic Light Shelf Module Anatomy

Three identical prototype light shelves were built for this project, and detailed plans are given in the appendix in Figure 52 and Figure 53. With reference to Figure 20, the first Robotic Light Shelf prototypes include the following features:

A. The main structure is made from one-quarter inch thick acrylic. The pieces are precision cut from a SolidWorks 3-D model using a laser cutter. The material choice is such that when sunlight hits the white acrylic its slight transparency makes it appear to glow.

B. The mirrored surface is 1/16-inch acrylic, and can be replaced with a diffraction grating or other materials to vary the type of light thrown into the space.

C. The light shelf's mirrored surface is mounted on the primary rotating platform.

D. A large drive gear (50-teeth over 180°) allows the rotating platform to rotate through 140 degrees, limited by safety switches. The geometry for the gears in this design was calculated using a plug-in for the Rhinoceros CAD program, GearGen [21].

E. A servo motor is used rotate the mirrored surface. Although a range of motion of 180 degrees is available to the motor, only a 90 degree range is utilized in this design.

F. A 3-axis accelerometer, with a sensitivity of +/- 1.5g, where g is 9.81 m/s^2, is used to detect the orientation of the rotating platform, and replay this back to a microcontroller.

G. Another drive gear (25-tooth), with retaining plates to keep it aligned with the larger gear, is attached to a drive servo.

H. A continuous-rotation drive servo is used to rotate the rotating platform through 150 degrees. Note: A stepper motor (1200mA, 4V), was originally chosen to act as the drive motor, which would have negated the need for an accelerometer to confirm the position of the shelf, but it was found that the motor was not strong enough to rotate the shelf, despite being one of the largest motors with a form-factor capable of fitting in the design. This continuous-rotation servo, rated at 4.8 kg-cm, was chosen as a replacement.

I. A snap-action switch is used as a safety stop for the rotating platform. Should something in the control code go wrong, the large drive gear is designed with end-stops which will hit this switch, and automatically cut power to the drive motors.

J. Another snap-action switch serves as the upper safety stop for the rotating platform.

K. A hub is attached to an axle to gain the necessary leverage to pivot the light shelf

L. A wire linkage transmits the force from the servo to the rotating axle.

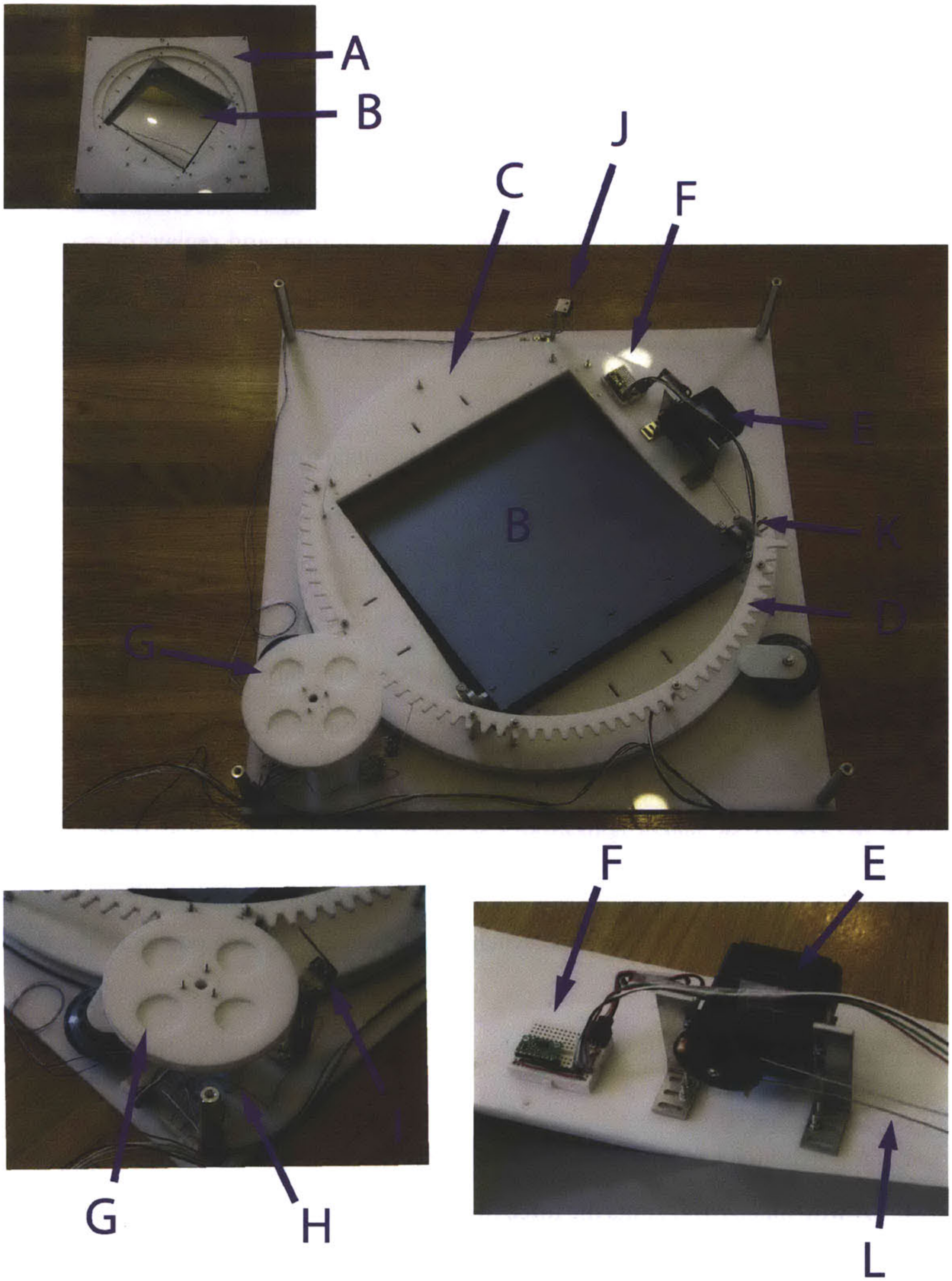


Figure 20: Robotic Light Shelf Anatomy

5.2 Robotic Light Shelf Control

5.2.1 Arduino Control

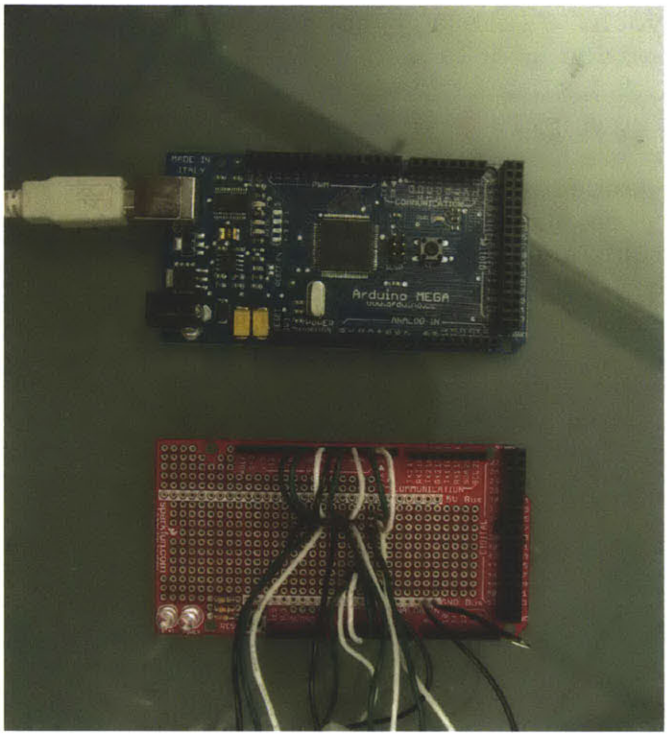


Figure 21: Arduino Mega

An Arduino Mega [22] is used to control all three light shelves simultaneously. Pictured in Figure 21 is the Arduino Mega microcontroller (image A), used in conjunction with a prototyping board (image B) that fits onto the microcontroller, and can be easily replaced to make circuit modifications. While each shelf is designed to run on 5V, the same voltage outputted from the Arduino, a separate power supply was employed to avoid the risk of trying to draw an excessive amount of current from the delicate microcontroller circuit.

The following is pseudo-code description of the programming running on the Arduino Mega, with the full code provided in the appendices on page 136.

```
*****************************************************
#include <Servo.h> //Library to control a generic servo motor
//using pulse-width modulation

// Associate input pins with meaningful variable names

// Set up accelerometer— relate voltage outputs from
accelerometer to real-world angle position of rotating
platform of light shelf

// Note that the Arduino will communicate via a serial (USB)
connection to a larger control user-interface coded using
Java. The Arduino is programmed to accept a single byte value
(0-255) from this serial input, to be interpreted as an
address to the light shelf, and an angle to be adjusted to.

Void setup(){   // run once at start of program
// Begin serial communication

// Attach servos to the Arduino in the code

}

Void loop(){ // Looped through continuously until program
terminated externally

// Allow loop to continue as long as an emergency stop switch
has NOT been triggered

// Get light shelf module ID from serial connection

// If the module ID is 215 then this refers to light shelf 1,
216 to shelf 2, and 217 to shelf 3

// Get the desired angle of rotation from the serial port and
check if it is within an acceptable range

// Rotate the shelf to the desired angle, delay code by 4
seconds to allow the shelf to catch up

// Get light shelf pivot angle from the serial port and check
if it is within an acceptable range. Move shelf to this pivot
angle
}

*****************************************************
```

5.2.2 Java Control

The graphical user interface, activity recognition, and underlying control software running on the desktop software was written in Java. This code communicates with the Arduino microcontroller across a serial port connection, sending single-byte values to convey the adjustments that are required to the position of the light shelves.

The system has been designed in such a way that it should be able to maintain the position of a pool of light in the living space, as cast by the light shelf, taking into account the time of day. This means that a user can define where in the room they want natural light during a given activity, and the code should know how to position the shelf correctly so as to reproduce this lighting condition given a different time of day or year.

The following is a brief summary of the methodology used to adjust the light shelf to a previously-saved lighting position in the living space. Inputs to the code include: α and θ, the pivot angle and rotation angle of the light shelf, respectively, and shown in Figure 22. Also required are the angle offset of the façade from due-south, δ, given in Figure 23, and the distance from the corner of the light shelf, considered the origin at (0,0,0), to the ceiling, labeled as distance L_1 in Figure 24.

- Calculate the current solar azimuth (Az) and elevation angles (h). Refer to Figure 23 for geometry description.
- Calculate the surface normal, $\bar{n}$, of the mirrored surface that, given the current solar position, would result in a reflection to the saved point in the living space. This is the *required* surface normal, and shown in Figure 25.
- Calculate the surface normal of the mirrored surface, given its current angles of α and θ. This is the *current* normal.

- Calculate the angle between the required normal and the current normal.

Iterate pivot angle, α, from 0°to 90° in 0.1° increments.

- Iterate rotation angle, θ, from -70° to +70° in 0.1° increments.
- Return the values of α and θ that result in the lowest angle between required and current normals.
- Adjust light shelf module to the returned values of α and θ.

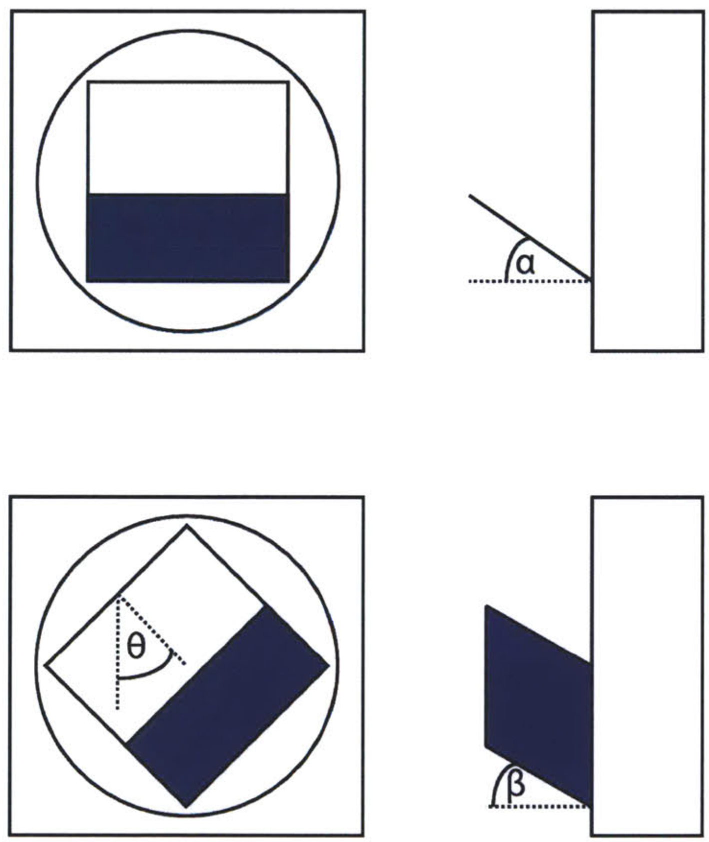


Figure 22: Light Shelf Geometry

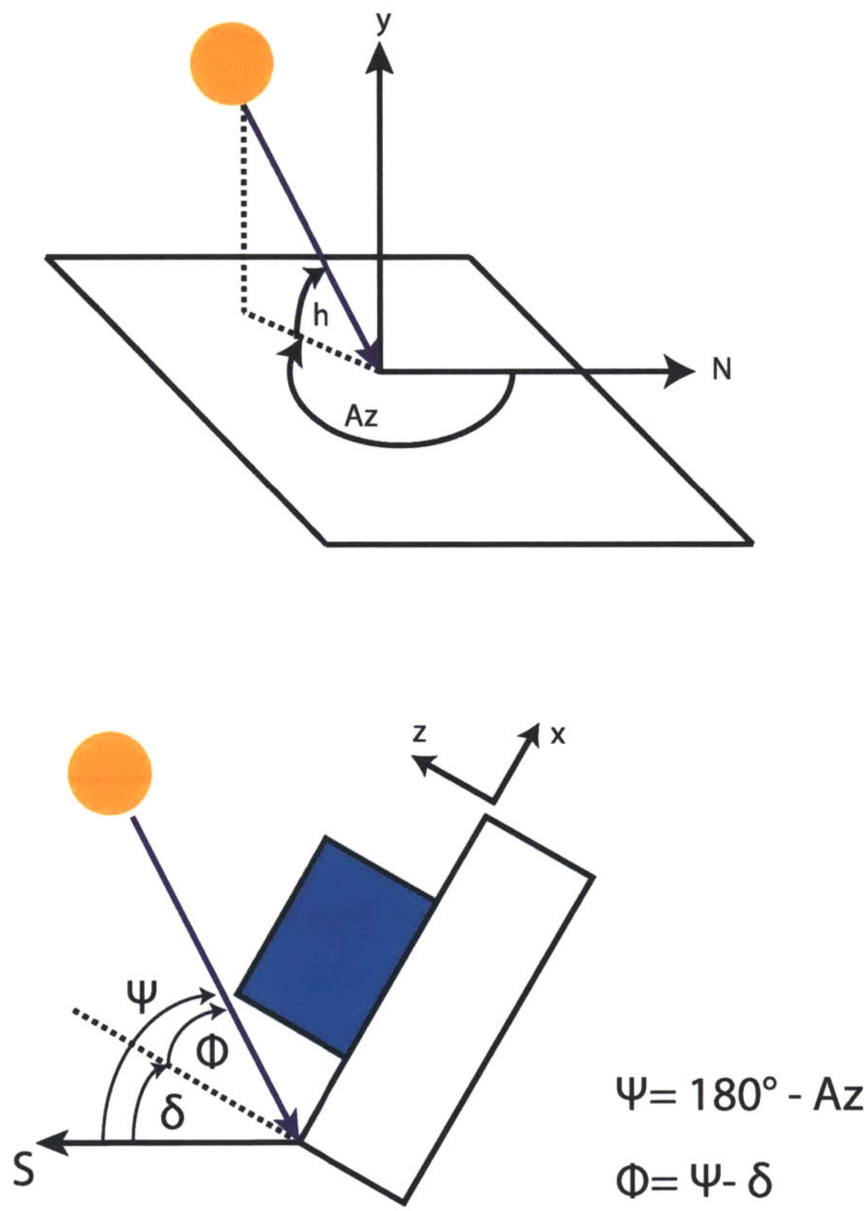


Figure 23: Solar Geometry

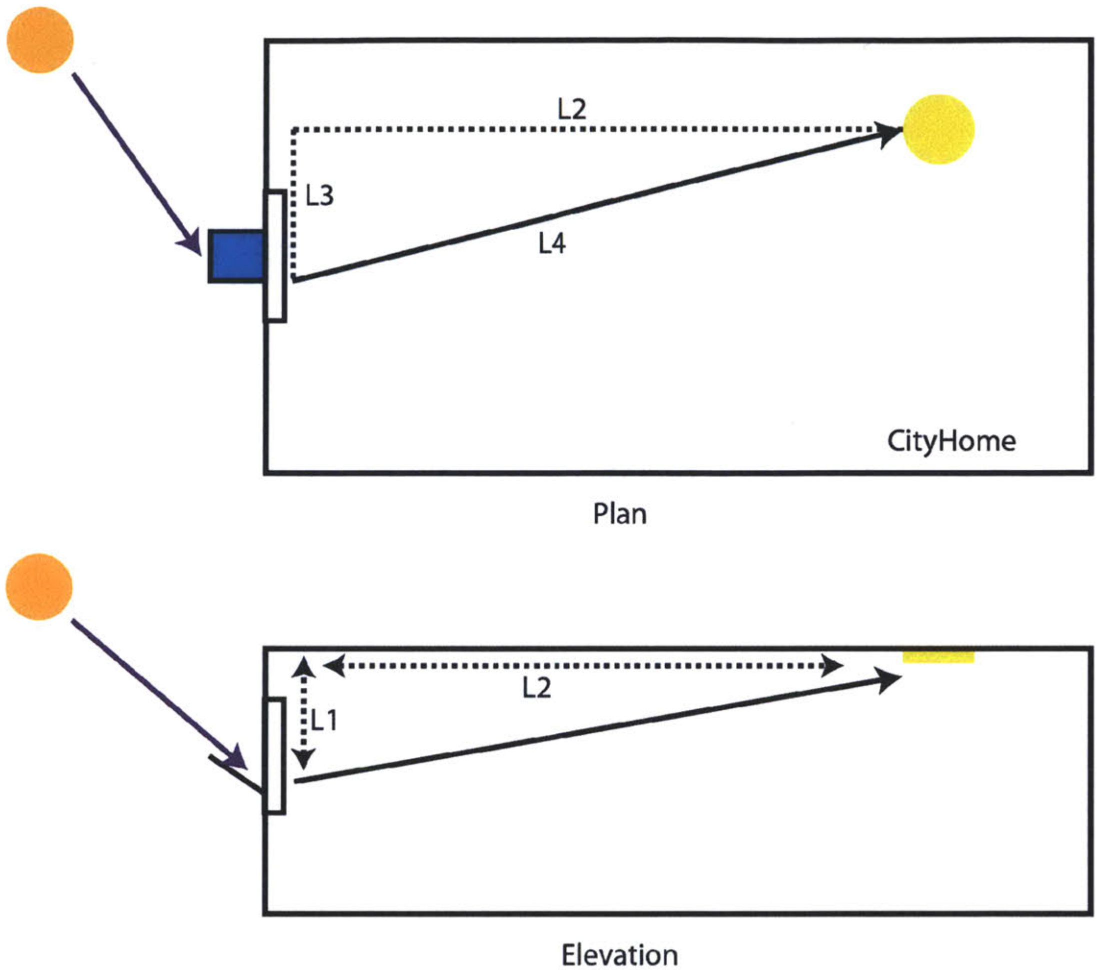


Figure 24: Apartment Geometry

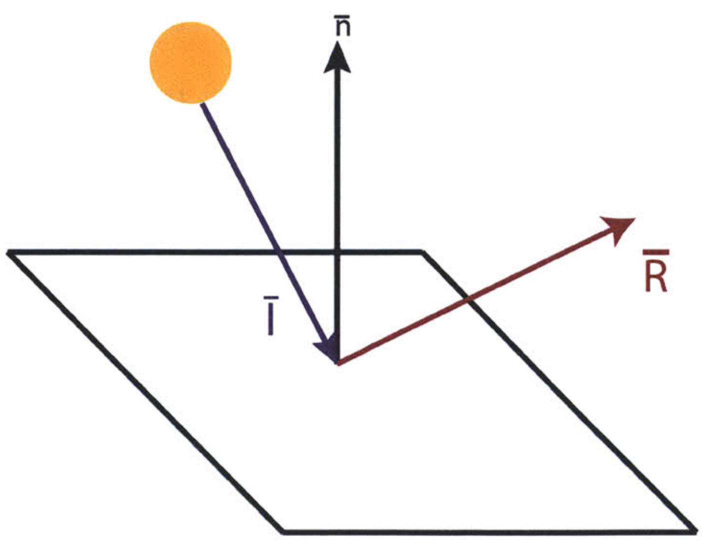


Figure 25: Reflection Geometry

5.2.3 Calculation of Solar Angles

A series of equations are used to calculate the solar azimuth (Az) and elevation (h) angles. The inputs to the code are:

i. The day of the year
ii. The hour of the day
iii. The longitudinal position of the user
iv. The time zone

Refer to page 149 of the appendices for complete description of the calculation.

5.2.4 Calculation of Light Shelf Geometry

Again, refer to Figure 22, Figure 23, Figure 24, and Figure 25 for a description of the geometry in question. The mirrored surface of the light shelf is considered a plane, and can be characterized by its normal, $\hat{n} = (n_1, n_2, n_3)$.

Given the solar azimuth, Az, and the solar elevation angle, h, the incident ray of light on the light shelf can be found. This incident light can be expressed as a unit vector, $\hat{I} = (I_1, I_2, I_3)$. Refer to Figure 25.

The reflected light can then be expressed as a unit vector, $\hat{R} = (R_1, R_2, R_3)$, as shown in Figure 25, and using the reflection vector, the point of intersection of the ceiling and the reflected vector can be found. The code used to find this point of intersection is given in the appendices on page 151.

To establish the light shelf at a position that reflects light to the same point on the ceiling, given a different time of day and year, the system code iterates through all possible combinations of the rotation and pivot angles for the light shelf, and finds the best match to that required to position the light at the saved point on the ceiling. Refer to page 156 of the appendices for further details.

5.3 MIT Environmental Sensors (MITes)

During the initial prototyping, the MIT Environmental Sensors (MITes) platform was used for activity recognition. While the sensor kit, developed within the House_n Research Consortium in MIT's Department of Architecture (Kent Larson, Director, and Stephen Intille, Technology Director), accommodates six different types of sensors, the first Robotic Façade test set-up uses passive infra-red (PIR) sensors, commonly referred to as *occupancy* sensors, as well as on-object accelerometers, or *object-motion* sensors. Image A of Figure 26 shows a number of object-motion sensors, with images B and C showing the internal circuitry, including the on-board battery (image C).

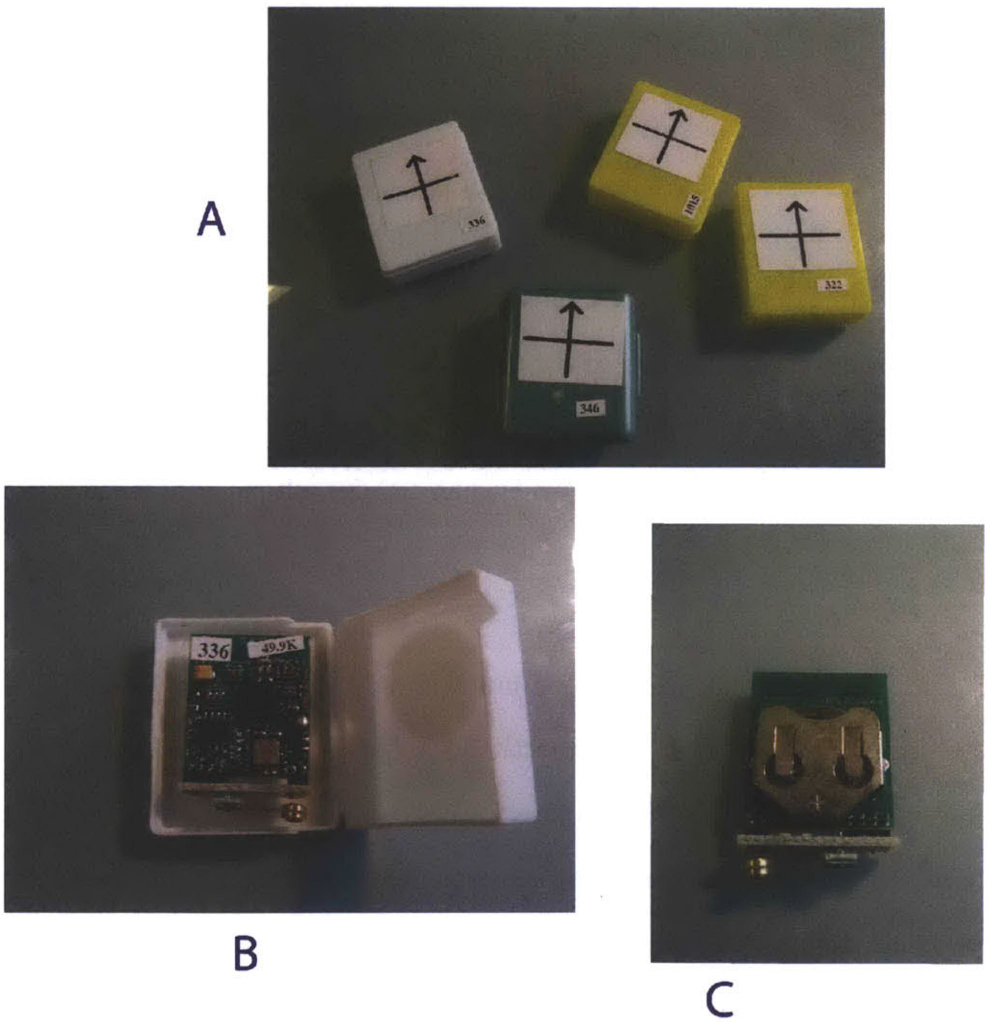


Figure 26: MIT Environmental Sensors (1)

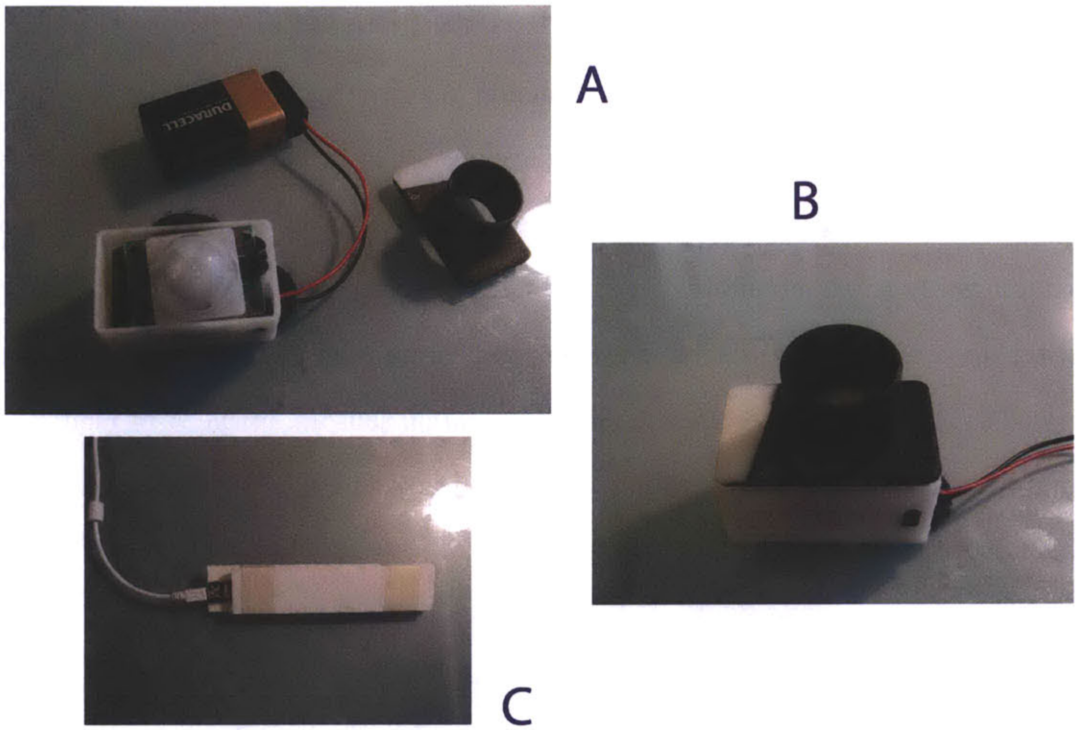


Figure 27: MIT Environmental Sensors (2)

Figure 27 shows an example of a PIR sensor, using a larger 9V battery, housed externally, than the object motion sensors, due to the increased power consumption of the technology. Image B shows the sensor with a shield attached, used to focus the IR beam from the sensor to achieve a more fine-grained sensor grid when using multiple PIR sensors in the same space.

The MITes offer the benefit of wireless communication to a single receiver antenna, pictured as image C in Figure 27, which is connected to a desktop computer running the activity recognition software written for this project. Each sensor is powered by an on-board battery, and the hardware is optimized to minimize energy use. Both sensor types used during testing awaken when activity is detected; the PIR sensors broadcast a unique ID number, and the object motion sensors broadcast their ID number, as well as information on activation intensity.

5.4 Desktop-based Graphical User Interface and Activity Recognition

A graphical user interface was designed and written in Java for this project. Essentially the key elements of the interface include the ability to view a pattern of activity, which, in turn, enables a user to train the software to recognize that same pattern of sensor activations, and associate it with the activity. In addition, the user interface allows each of the light shelves to be controlled manually, as well as associating a given activity with a particular position for the light reflected from the light shelf.

Figure 28, part A, shows one of the main tabs of the user interface, that used to save a given set of sensor activations to an activity type. Five sensors, labeled one through five, are registered with the system in this example, shown along the vertical panel on the left, but the software has been designed to scale automatically to the number of sensors registered with the system. The sensors can also be given more meaningful names when attached to devices around the home.

Shown in Figure 28 part B are examples of sensor activity graphs. Each of the horizontal graphs represents a pattern of activations of a single sensor over time. The graph auto-scrolls to the current time, as indicated by the scroll-bar at the bottom of the screen. Each filled box represents an activation, and each un-filled box a period when the sensor is in stand-by mode.

Figure 29 part C demonstrates the first step for a user to save a sensor activation pattern and associate it with a given activity. By clicking anywhere on the sensor activation graphs, the graph auto-scroll is paused, and a vertical bar is drawn to indicate the first boundary of interest. Clicking a second time at a different location, as shown in Figure 29 part D, draws a second vertical bounding line, and fully defines the time period of interest. Essentially this means that only sensor activations within these two bounding lines are saved, and everything else ignored.

Figure 30 part E indicates how the sensors of interest within the bounded region are chosen using tick-boxes, and highlighted in the figure by red arrows. Two buttons are provided to select all sensors registered with the system, and another to deselect all. This selection process means that only certain sensors will be considered in the detection of any given activity. This is important if, for example, a user knows that there may be many different types of sensors being triggered around the home at a given time, but only two sensors in particular are meaningful for a given activity type. A more definite example is that of a home with more than one occupant. One of the occupants may be in the bedroom, causing sensors to detect human-presence, while the other is saving an activity pattern associated with eating breakfast. All sensor activations will appear on the GUI as per Figure 28 part B, but only the three sensors attached to the cupboard containing the breakfast cereal, the kitchen table, and a particular chair, will be of interest in the recognition of the breakfast activity.

Figure 30 part F indicates how a simple text field is used to name the activity pattern, and subsequently to save the pattern for detection in the future. Note that the auto-scrolling graph screen un-pauses when clicked for a third time, and the two bounding lines are erased. The sensor activations are saved in this graph for as long as the program is running, so the process is easily repeated if a mistake is made. Another fool-proofing feature is that none of the saving operations can be completed out of order, so that the save button cannot be pressed until both bounding lines have been placed, sensors of interest have been chosen, and the activity has been named.

Figure 31 part A is an example of the Automated Control tab of the GUI. This interface allows;

- Current sensor activations to be viewed as a color-changing vertical list along the left hand side of the screen.
- A list of saved activity patterns that can be recognized by the system to be viewed in the center of the screen on the white background.
- Manual control of each of the three light shelves rotation and pivot angles, as well as control over whether the shading system is open or closed. These controls are given in the vertical panel along the right hand side of the interface.
- The user to associate a given light shelf angle setting with a given activity type so that the light shelf will re-position itself to throw light to an unchanging position in the living space if a linked activity is detected.

Figure 31, part B, shows how a sensor activation, indicated by a color-change of the sensor name from pink to green, and highlighted by a red arrow here, is displayed on the Automated Control tab. Here, a simple activity recognition pattern, one linked to that single sensor4, is shown to be flagged by the system. The second red arrow highlights how the system prints out the name of the activity recognized.

The activity recognition algorithm used in this project is very simple, but the back-end system-code has been designed to incorporate a more advanced set of activity recognition rules in the future. In this iteration, the algorithm does not acknowledge a sensor activation until it has been repeatedly activated a given number of times, thereby avoiding the possibility that a brief sensor activation was a glitch in the signal transmission. Once a sensor activation is acknowledged, a boolean variable associated with that particular sensor is set to true. This variable has a time-out count, so that if the sensor is not activated again, the boolean value will change to false after a set time-delay. The system generates activity codes for each of the activity files generated from the process described in Figure 28, Figure 29, and Figure 30, and as shown as a list of files in the center of the GUI in Figure 31. In these activity codes is information on which sensors are of interest, and when a given pattern of the boolean variables, both true and false values depending on the sensor activations, matches any given activity code, the system prints the recognized activity name to the screen.

Figure 32 part C indicates how a given set of light-shelf parameters, which include the light-shelf ID number, and the rotation and pivot angles, can be sent directly to a given module from the user interface to test the lighting position, or associated with a given activity by highlighting the activity file from the white list, changing the angle values along the right-hand side of the screen, naming the association, as highlighted in the red box, and pressing the save button. Again, this user-interface has been designed with several fool-proofing features, including the inability to send information to the Arduino microcontroller without specifying the module of interest first, and the inability to try to position the light shelf at an angle that it outside of its physical design constraints.

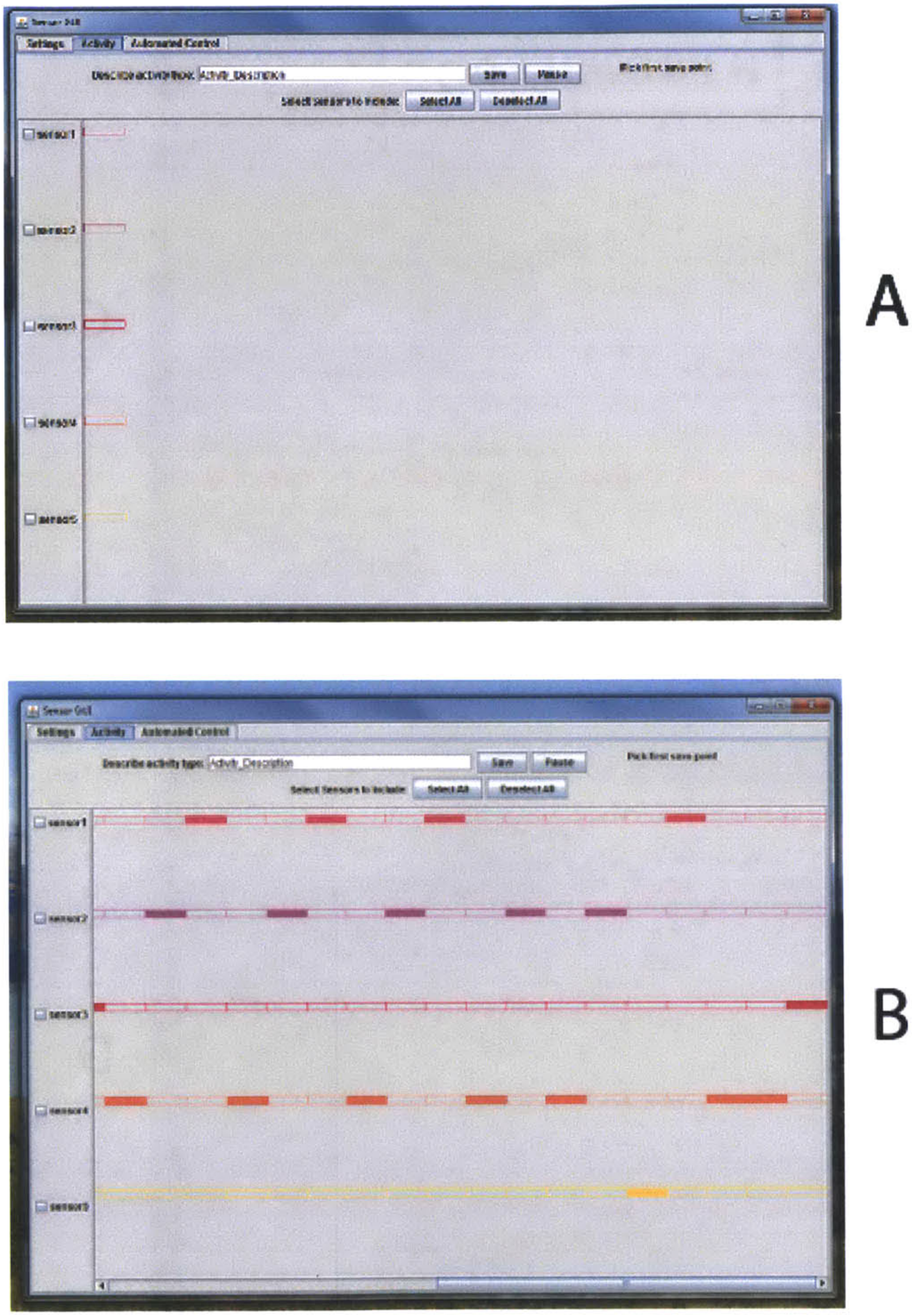
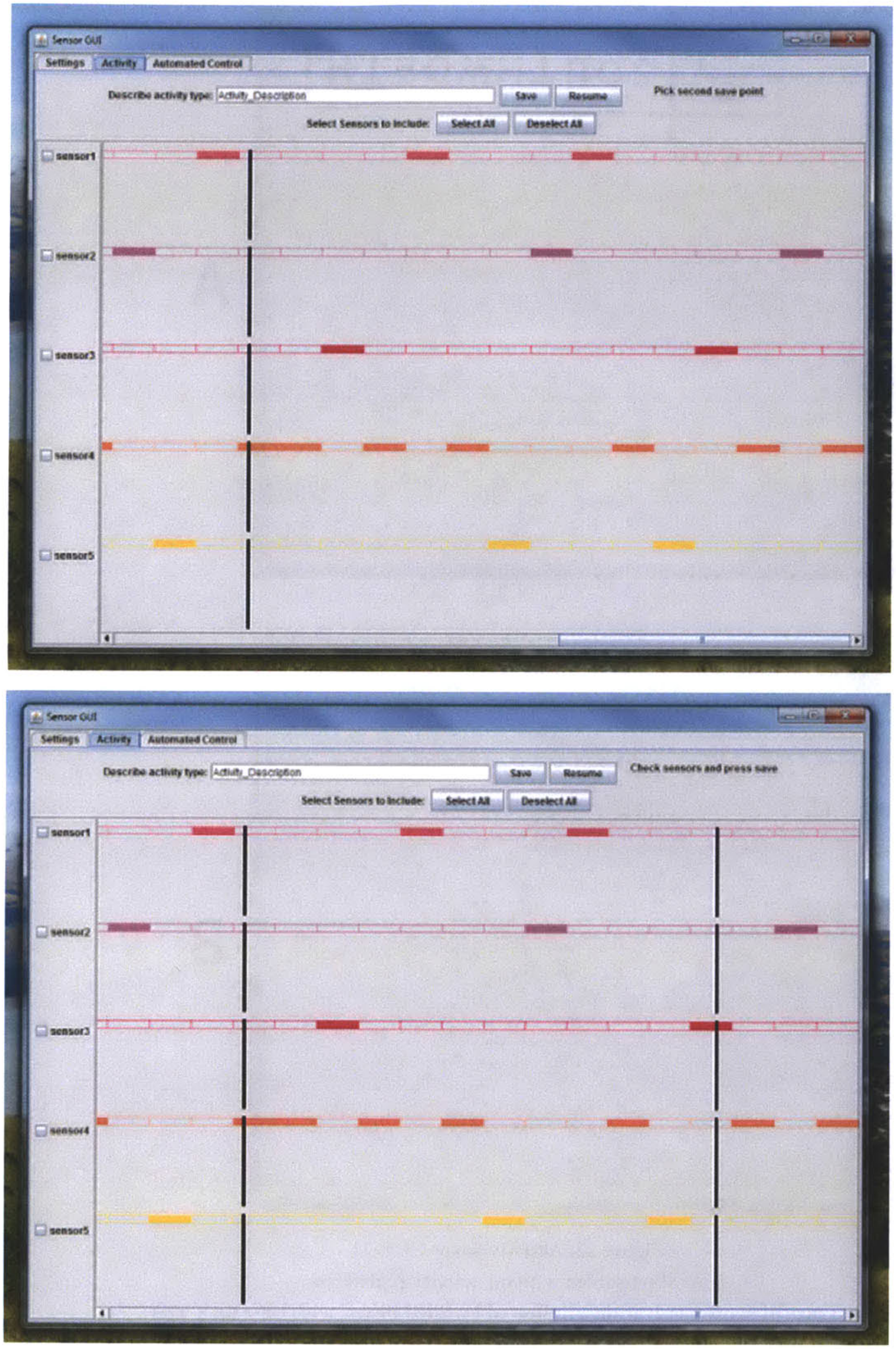


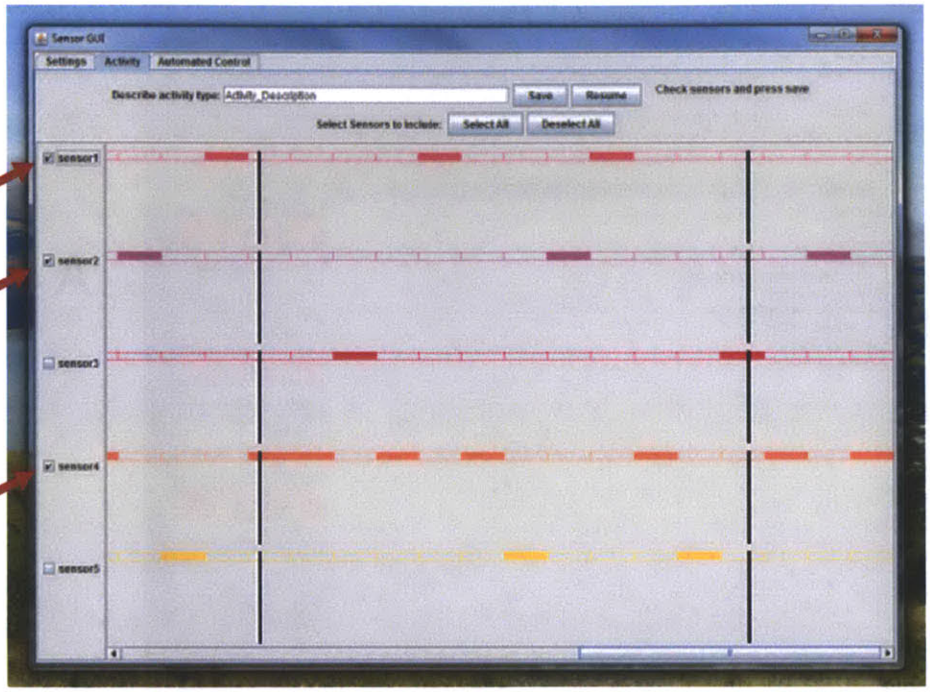
Figure 28: Activity Saver GUI (1)
A: Start-screen with no activity registered
B: Trace of sensor activations, indicated by filled boxes, with time on x-axis



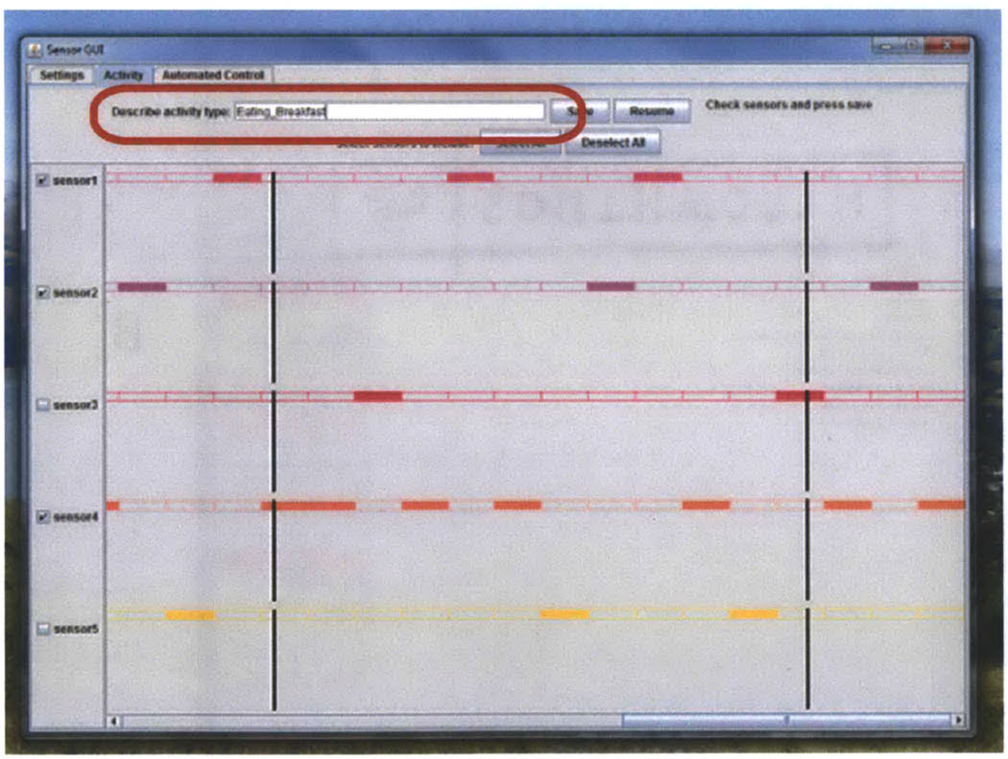
C

D

Figure 29: Activity Saver GUI (2)
C: Start point of save-interval selected, and highlighted by vertical line
D: End point of save-interval selected, and indicated by second vertical line

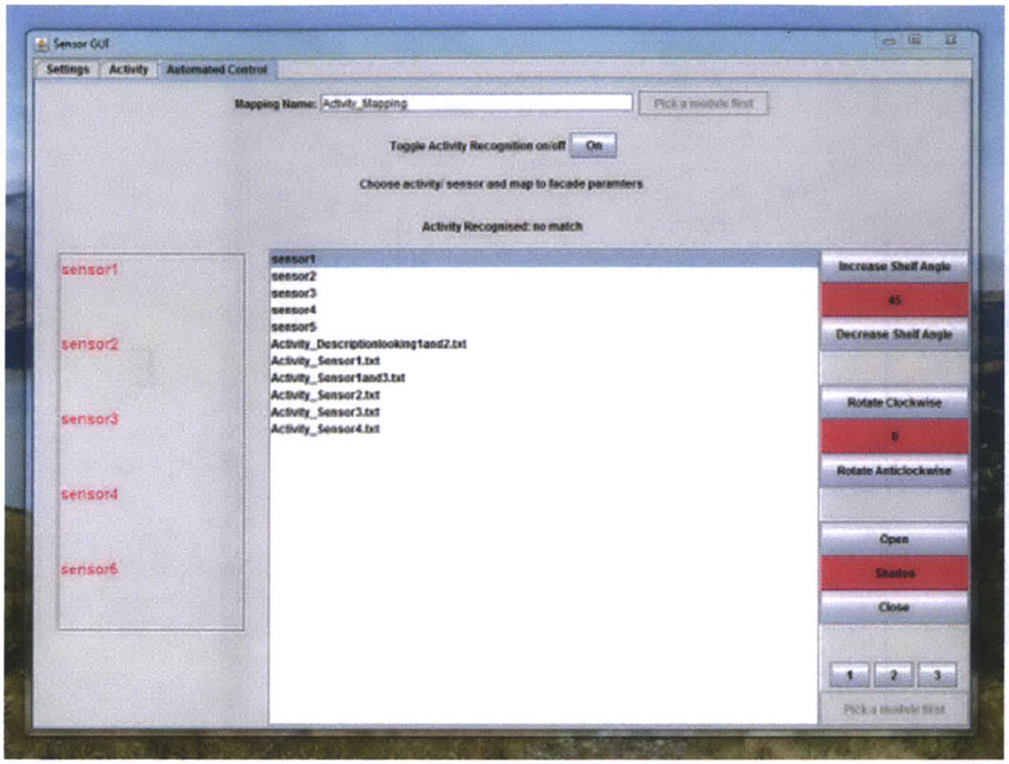


E

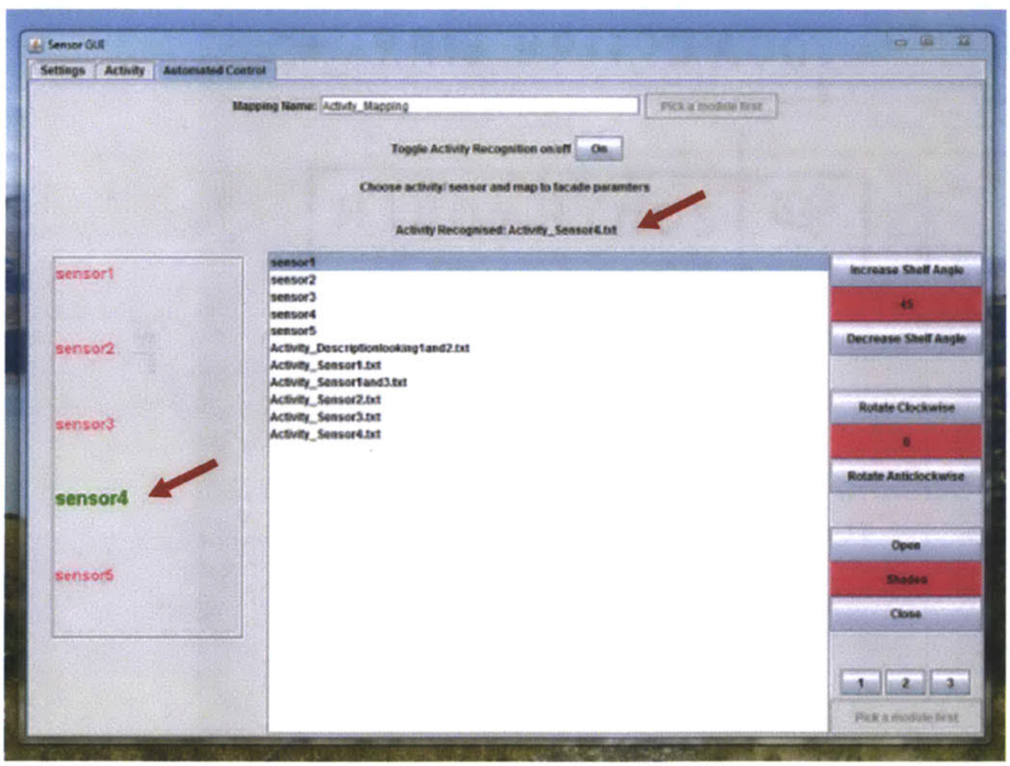


F

Figure 30: Activity Saver GUI (3)
E: Sensors of interest selected, highlighted by red arrows
F: Activity named, highlighted in red, and saved

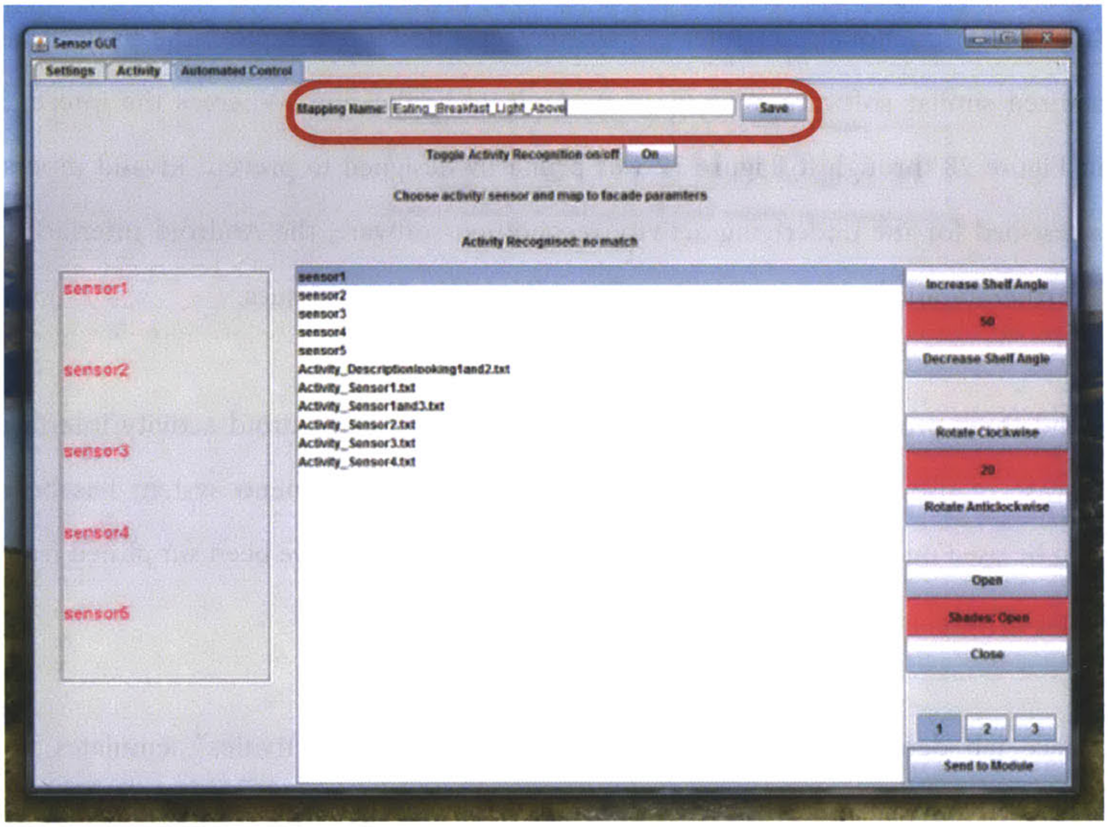


A



B

Figure 31: Activity Recognition GUI (1)
A: Activity Recognition start-screen
B: Real-time sensor activity indicated by sensor name changing color to green. Activity pattern recognized, and activity file name printed to screen



C

Figure 32: Activity Recognition GUI (2)
C: Mapping of recognized activity to light position by linking activity file name to light shelf angles, and saving the mapping in the highlighted text field

5.5 Android Graphical User Interface

In addition to the functional user-interface written for a desktop or laptop, the project has conceptualized similar software, written as an Android application. Whereas the interface shown in Figure 28 through to Figure 32 was primarily designed to prevent invalid inputs, and as a test-bed for the underlying activity recognition software, the Android interface is seen as a further iteration, this time concentrating on human factors issues.

Built around three primary tabs, as well as a settings menu, the Android activity interface considers ease of use for an inexperienced home occupant. The menu system has been designed to be used only with one's finger, and the actual controls have been simplified from the desktop software.

The interface tab depicted in Figure 33, which is labeled as "Activities", emulates the functionality of Figure 28. Screenshots A, B and C of Figure 33 describe how, once again, sensor activations are displayed as auto-scrolling graphs, with unfilled boxes indicating that particular sensor is in standby, and filled boxes indicating an activation. Again, the bounds of the sensor activation pattern of interest are chosen by touching the active graph area at the desired points, which pauses the graphs and places a visual marker in the form of purple vertical bounding lines.

Figure 34, part D, gives an example of how the on-screen Android keyboard will automatically pop-up when the text-field for the activity name is highlighted. Part E shows an example of a confirmation screen for saves, and part F shows how the settings menu can be accessed from the dedicated menu button that appears as a hardware button on all Android devices.

While the desktop system's user-interface has settings hard-coded into it, such as location and orientation of the façade, Figure 35 shows the Android mock-up interface that automatically finds the location, hence the longitude and latitude, via the device's built-in GPS antenna, as well as the prevailing outdoor environmental conditions via a connection to the internet. The indoor air temperature is found using a set of wireless sensors, and the only user-input required is that of the orientation of the façade.

The tab shown in Figure 36, "Automated Control", emulates that shown in Figure 31, allowing the user to associate a given activity with a particular lighting position within the space. The primary improvement made to the Android interface is the inclusion of a simplified directional pad, or D-pad, for moving the light shelf to a desired position. This means that the user need not have any understanding of what adjustments are necessary to the rotation and pivot angles of the light-shelf to achieve the desired lighting result.

Figure 37 indicates that the manual control will have its own dedicated tab in the Android interface. It allows the light-shelves to be operated without a link to activity recognition, and this is also seen as one of the best places to give persuasive feedback on the energy-impact of decisions made using manual control, with a small window informing the user of how much energy may be wasted as a result of a given choice.

Note that while this Android mock-up concentrates on the Robotic Light-Shelf control, the same activity recognition tabs and basic interface design can be used to control other aspects of the Robotic Façade, such as the complete HVAC system.

Figure 33: Android Activity Tab

A: Auto-scrolling graph of sensor activations, indicated by filled colored boxes, with time along the x-axis

B: Start of save interval selected by touching screen, which pauses the auto-scroll, and places a vertical line at that point

C: End of save interval selected by touching screen a second time, placing a second vertical line on sensor activity graph

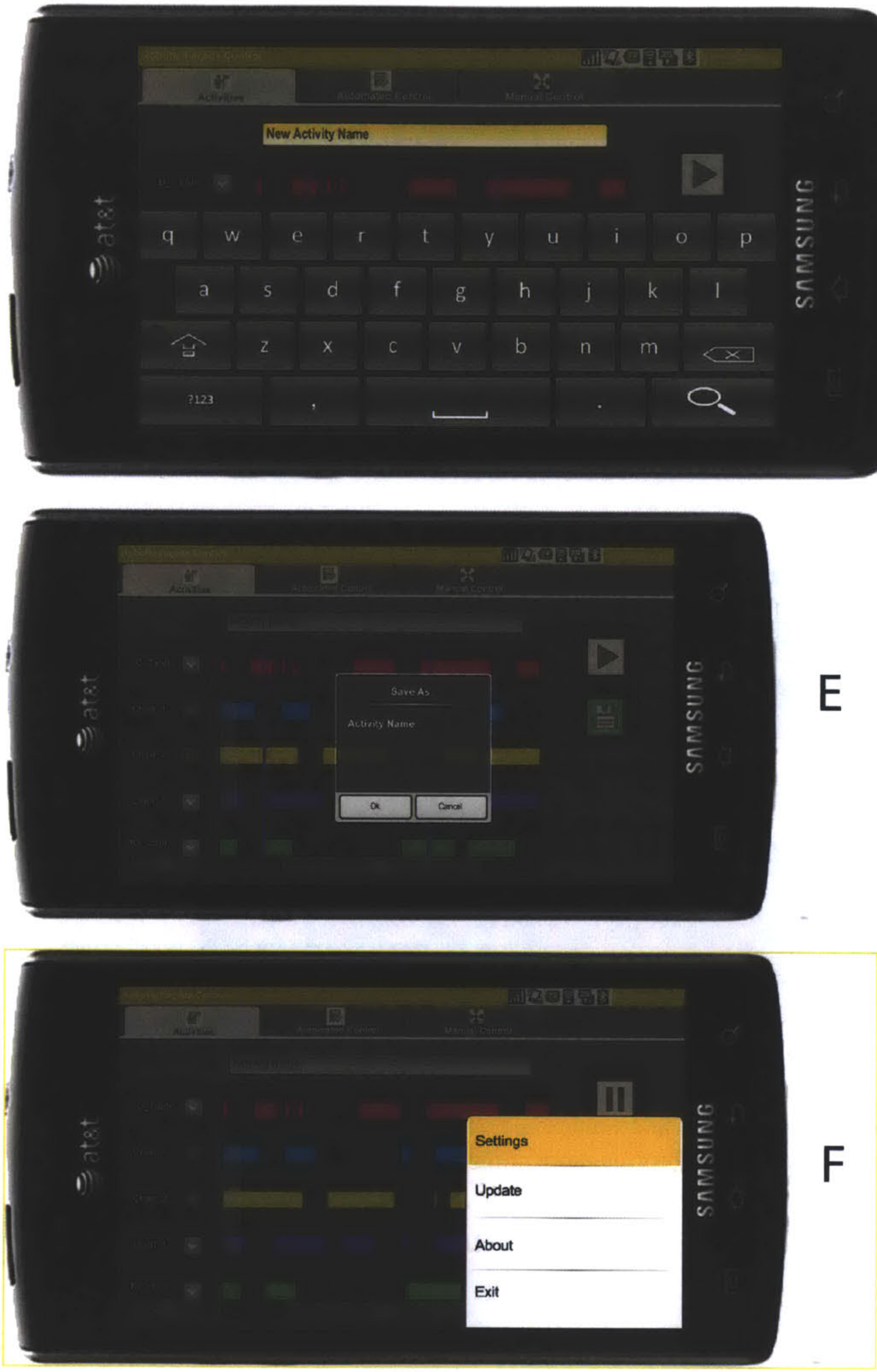
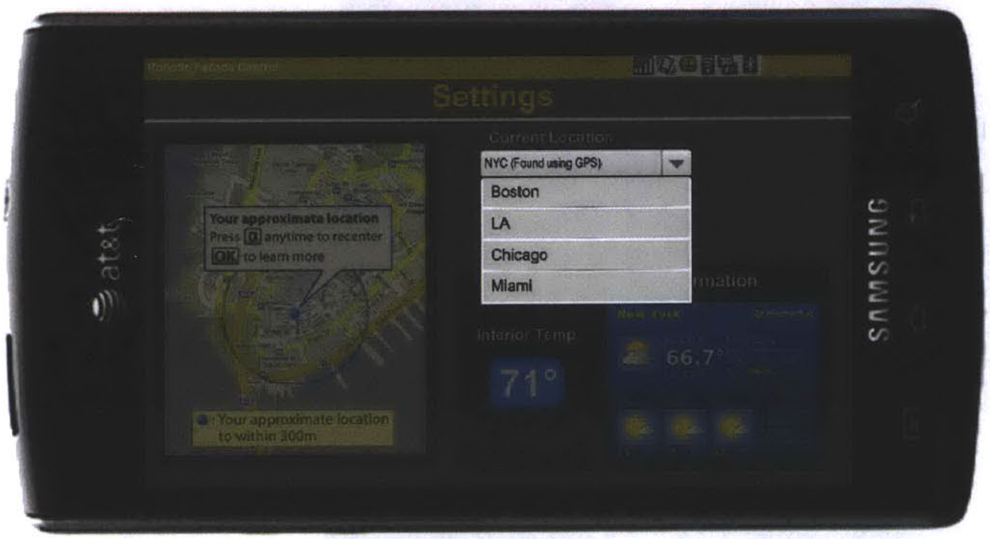


Figure 34: Android Input Examples
D: Example of on-screen keyboard used to input activity name
E: Save confirmation screen
F: Additional options menu

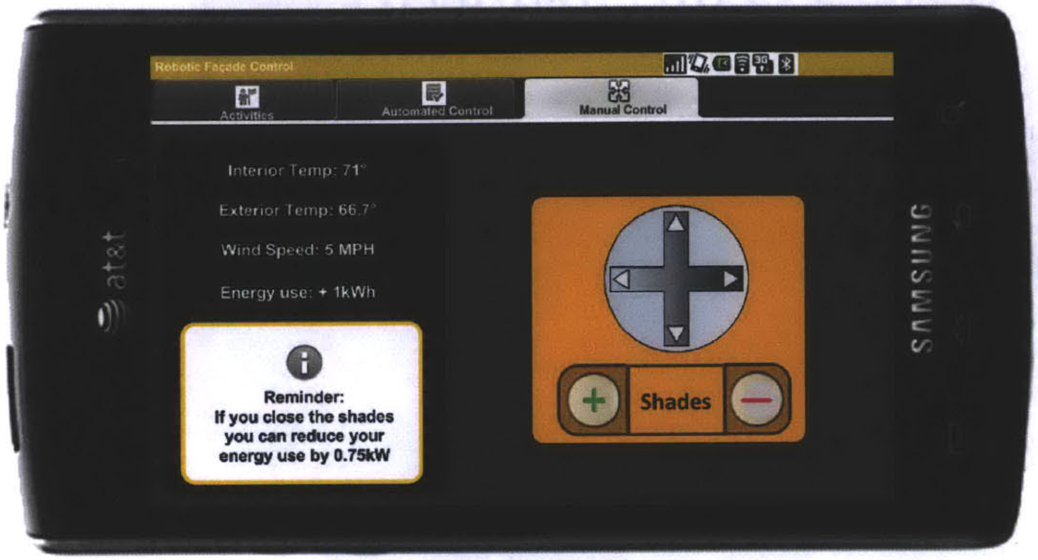
G



H

Figure 35: Android Settings Menu
G: Settings menu showing location found using internal GPS antenna, outside weather conditions found using internet connection, and location and orientation settings inputted by user
H: Example of location drop-down menu if GPS location unsuccessful

Figure 36: Android Automated Control Tab
I: Interface allowing a selected activity to be linked to a light shelf position
J: Example of on-screen keyboard used to name and save mapping



K

Figure 37: Android Manual Control Tab

K: Left: Information window showing energy savings. Right: Directional control pad used to manually move the area of reflected light to desired position in living space.

5.6 Real-world Deployment and Testing

Part of the project involved testing the Robotic Light Shelf, sensors, and the activity recognition graphical user interface in a real home setting. One of the light shelf modules was installed in an open-plan apartment of similar size to the CityHome, as shown in Figure 38. This apartment has hosted several studies within the Changing Places research group so is already equipped with MITes object motion sensors (some of the technical details previously described in Figure 26 and Figure 27) on every movable object. Figure 41 gives several examples of where the sensors are deployed.

Figure 38 shows how the Robotic Light Shelf module was installed in the apartment, and, while not positioned external to the glazing, was relatively unobtrusive to the interior furnishings and décor, as well as being capable of catching the sunlight at an adequate intensity, exemplified in image B.

From this study it was found that the shelf is capable of illuminating areas that were previously inaccessible to direct natural sunlight, as shown in the contrasting images of Figure 39. By directing light onto a lightly-colored ceiling, a diffuse "pooling" effect is created overhead, illuminating much of the space below. Additionally, and as shown in the contrasting images of Figure 40, the system has proven to be effective at augmenting the natural lighting levels when used to track the activity of a home occupant.

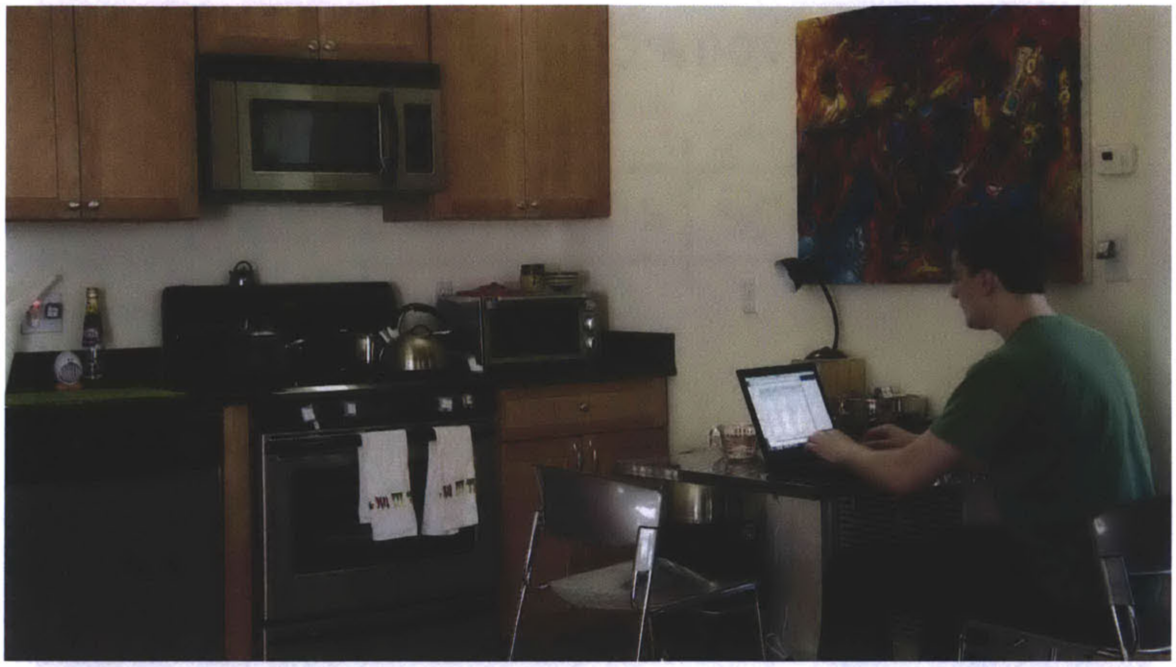
Additionally, the Robotic Light Shelf was tested as a platform for controlling the type of natural light that enters into the space, and achieving certain design aesthetics. The mirrored surface was replaced by a large diffraction grating, and the results are shown in Figure 42.

A

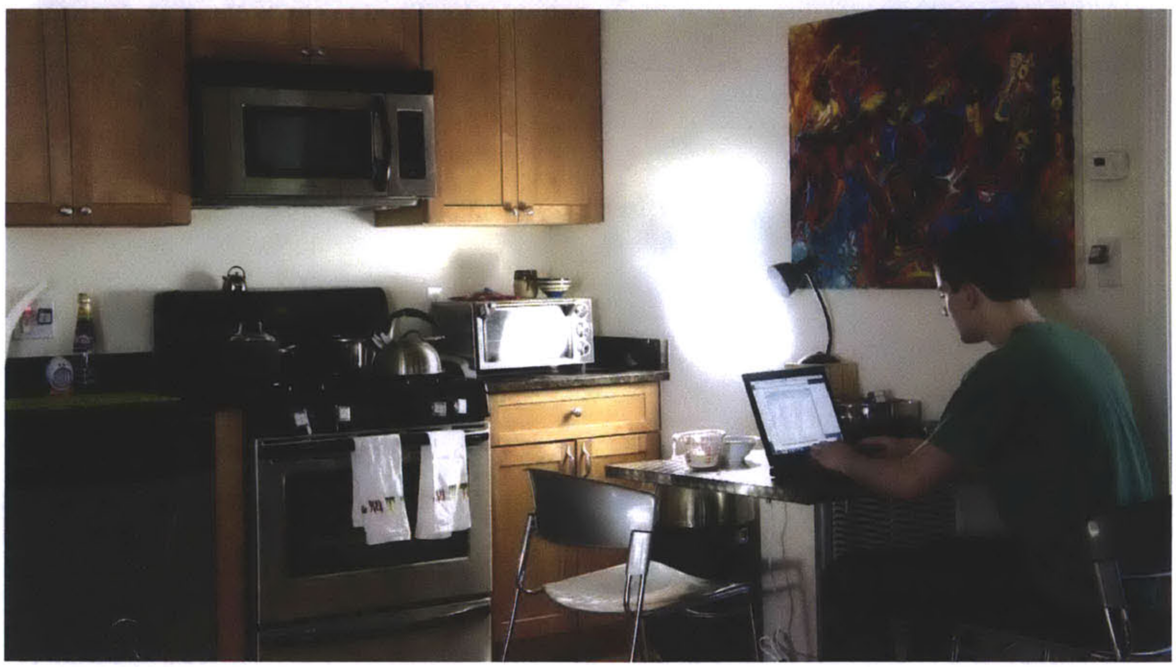
B

Figure 38: In-home testing (1)

Figure 39: In-home testing (2)



A



B

Figure 40: In-home testing (3)

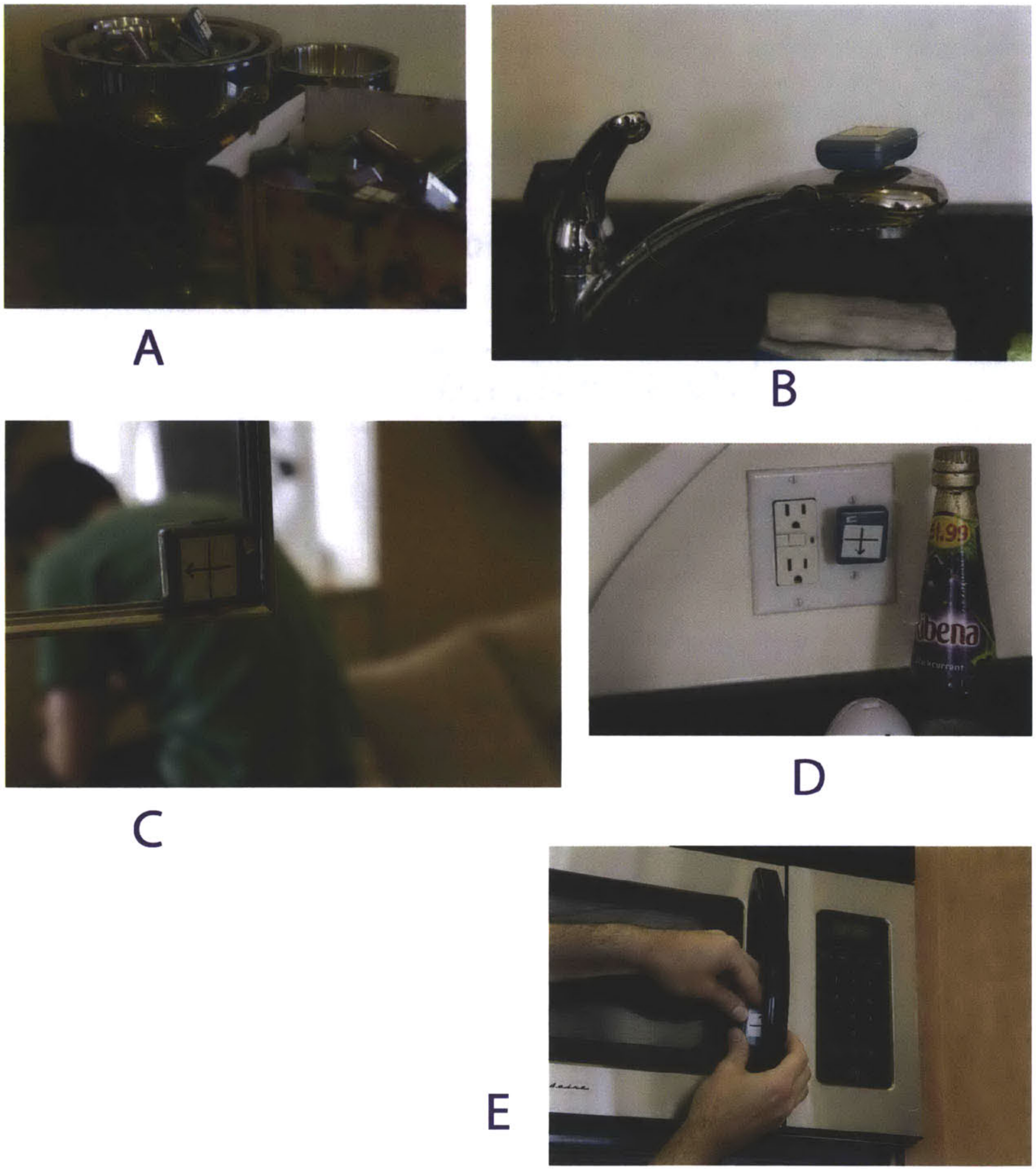
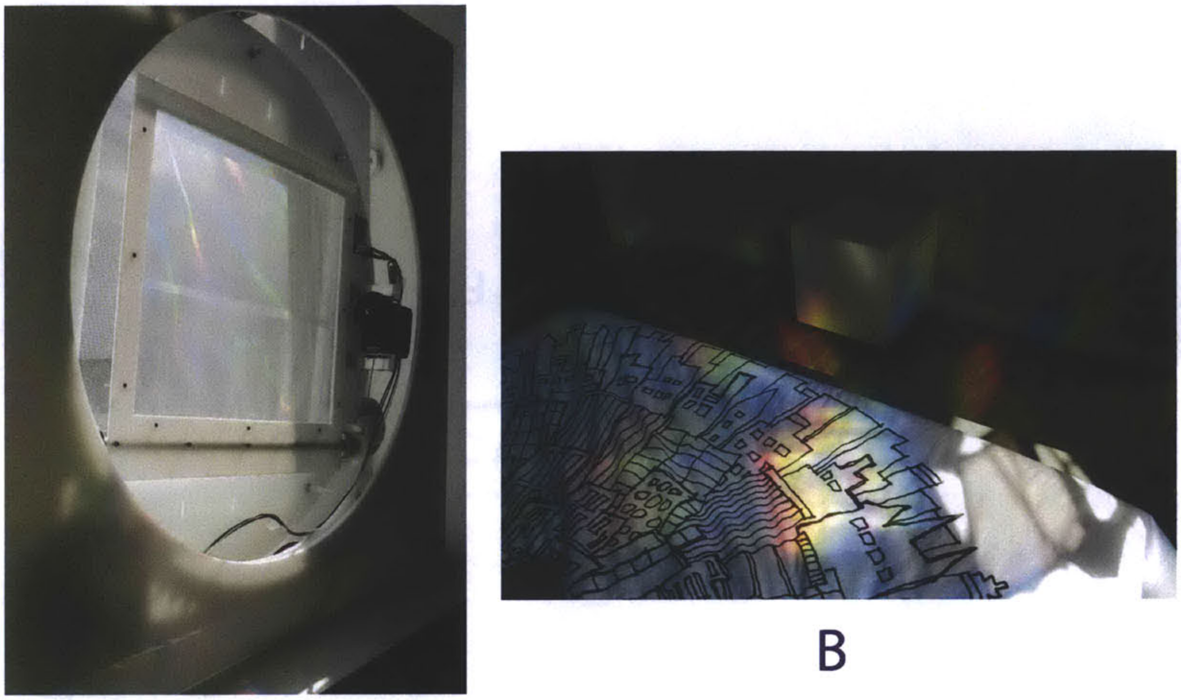


Figure 41: In-home testing (4)



B

A

Figure 42: In-home testing (5)

6. Conclusions and Future Work

This project has outlined a vision for how a façade can be described as a standalone product, in the form of a balcony module, capable of integrating into an apartment design in an urban setting, while housing the majority of the heating, ventilation, and air-conditioning equipment, and thereby simplifying apartment construction and operation. This concept has been developed into a design that addresses the complete energy-management issues in the home, and uses novel technologies, such as activity sensing and automated light shelves, to improve upon energy-usage patterns by the home owner, as well as improve his/ her quality of life in the home.

A construction and testing phase to the project, while not producing a full mockup of the Robotic Façade module, was able to determine that the design includes several novel technologies, and, when linked to the GUI designed and produced for the project, amount to what is seen as an appealing end-product that integrates many beneficial technologies, but in such a way that makes them more user-friendly.

Going beyond this initial phase, it is suggested that future work on the Robotic Façade should include the construction of a module mock-up that incorporates all the features of the final design. The user-interface, while functional on a desktop or laptop computer, should be further refined to approach that simplified Android concept described in Section 5.5, and integrate all of the HVAC technologies into the code. Further to this, this project has not attempted to produce a HVAC online customizer, as mentioned in Section 1.2, which is something that can be developed further during future project revisions.

Appendix A PlaceLab

The PlaceLab [23] was one of Changing Places' (formally House_n) earlier studies of technologies that were designed to improve the energy performance of the home, as well as the quality of life of the home occupants. A real life one-bedroom condominium was constructed to the specifications of the research group, with almost every aspect of life in the home monitored through a large array of sensors. Non-intrusive studies were carried out, whereby volunteers lived in the PlaceLab for varying amounts of time, but with no contact with the researchers. In addition to information about how people respond to stimuli to reduce energy usage in the home, these studies also produced rich datasets on activity patterns, and have colored the development of more small-scale and portable sensor kits that can be used to perform similar activity recognition and intervention studies in any home.

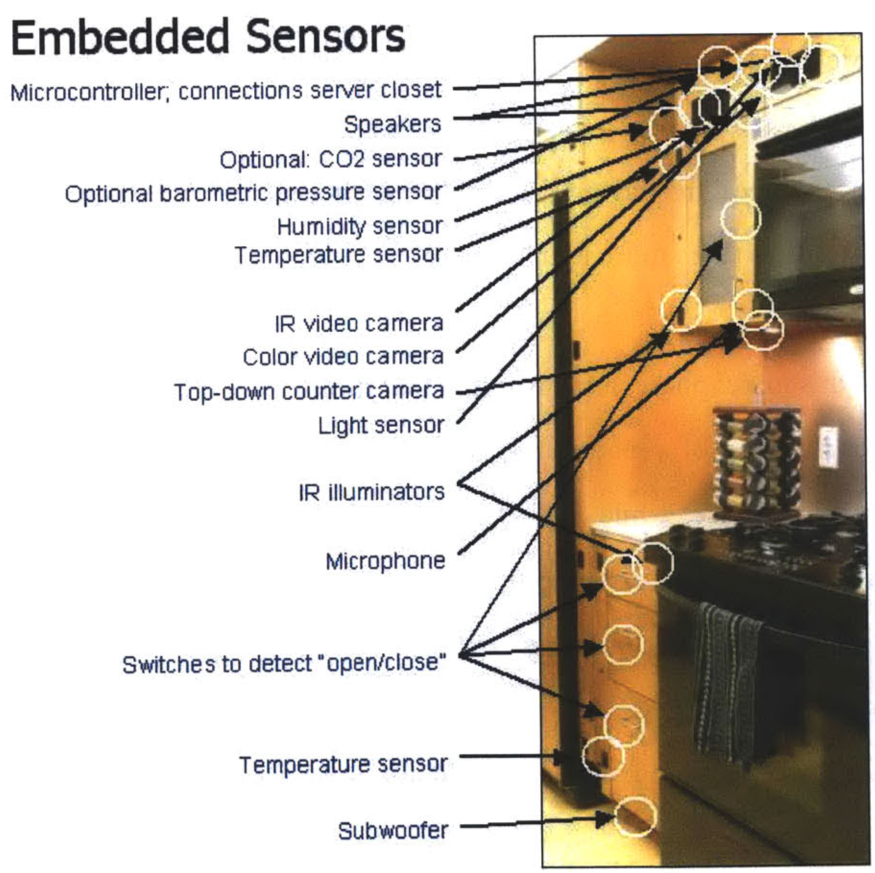


Figure 43: PlaceLab Kitchen Sensors [23]

Appendix B Energy Analysis Settings

Standard Apartment Design
Three-bay apartment. Each bay 4.11m (13'6") wide. Total apartment width 12.33m (40'6").
Apartment depth 8.53m (28')
Floor-to-ceiling height 2.74m (9').
Total apartment floor surface area 105m^2 (1134 sq. ft.)
Façade exterior walls R-value 8.3 m^2K/W (U-value of 0.12W/m^2K).
Windows triple-glazed, R-value 0.513 m^2K/W (U-value 1.949 W/m^2K).
Glazing clear-colored.
50% of façade covered by glazing.
Shading is installed external to the glazing, and is automatically used when the solar radiation incident on the window exceeds the 70W/m^2 setpoint.
Low thermal mass within the building.

Table 4: Energy Analysis Apartment Assumptions

HVAC Assumptions
People density of 0.02 people/m^2
Occupancy schedule assumed to be 5pm-8am
Heating setpoint 18C
Heating setback 12C
Cooling setpoint 25C
Cooling setback 28C
Acceptable temperature for natural ventilation 22C
Maximum relative humidity of 60%
Lighting density 5W/m^2
100 liters of domestic hot water used per day, and heated from 10C to 65C
Fresh air rate is 10 liters/sec-person.
Air change rate is 0.49 roomfuls per hour.
Heating Coefficient of Performance: 0.65
Cooling Coefficient of Performance: 1.67

Table 5: Energy Analysis HVAC Assumptions

Appendix C Further Energy Analysis Results

		Total Annual Energy Use (kWh)		
		Proposed Robotic Façade	Low-R	Medium-R
Boston	North	15592	17430	16431
	South	13039	13805	13302
	East	14789	16225	15412
	West	14898	16339	15527
LA	North	5879	6195	6003
	South	5648	5840	5698
	East	5897	6156	5988
	West	6358	6664	6480
Beijing	North	16919	18720	17774
	South	13743	14336	13922
	East	15912	17228	16491
	West	16326	17816	17002
Mumbai	North	12522	12979	12738
	South	13167	14118	13638
	East	13144	13958	13547
	West	13107	13956	13520
Quebec	North	23248	26189	24595
	South	19763	21199	20303
	East	21920	24248	22935
	West	21967	24252	22970
London	North	14129	15792	14878
	South	12503	13475	12887
	East	13597	15006	14212
	West	13575	14950	14178

Table 6: Analysis Results: Annual Energy Use

Appendix D Robotic Façade HVAC Cabinet

Figure 44 gives a rendered view of several components inside the robotic façade's HVAC cabinet. The space, measuring 185 cubic feet, is intended to accommodate the majority of the heating, ventilation and air-conditioning equipment, so that the Robotic Façade module can be fully equipped off-site, using expensive labor necessary to commission this equipment in a more cost-effective manner in factory conditions, and then shipped to the construction site to be as a pre-assembled product to be connected to the primary CityHome chassis.

It is intended that the final façade product will have custom-sized equipment produced for it, but to give an initial indication of the space available in the HVAC cabinet, commercially-available products are shown in the rendered images. These products have been drawn to exact scale, and with reference to the numbers shown in Figure 44, the following components are included:

1. The HVAC cabinet, measuring 3' x 6.5', with a height of 9.5'.
2. Reneaire Breeze BR 70 Energy Recovery Ventilator.
3. Lochinvar Knight Wall Mount gas-fired boiler.
4. 85 liter hot water tank.
5. 43 liter chilled water tank.
6. Daikin Altherma Air-Water heat pump system.

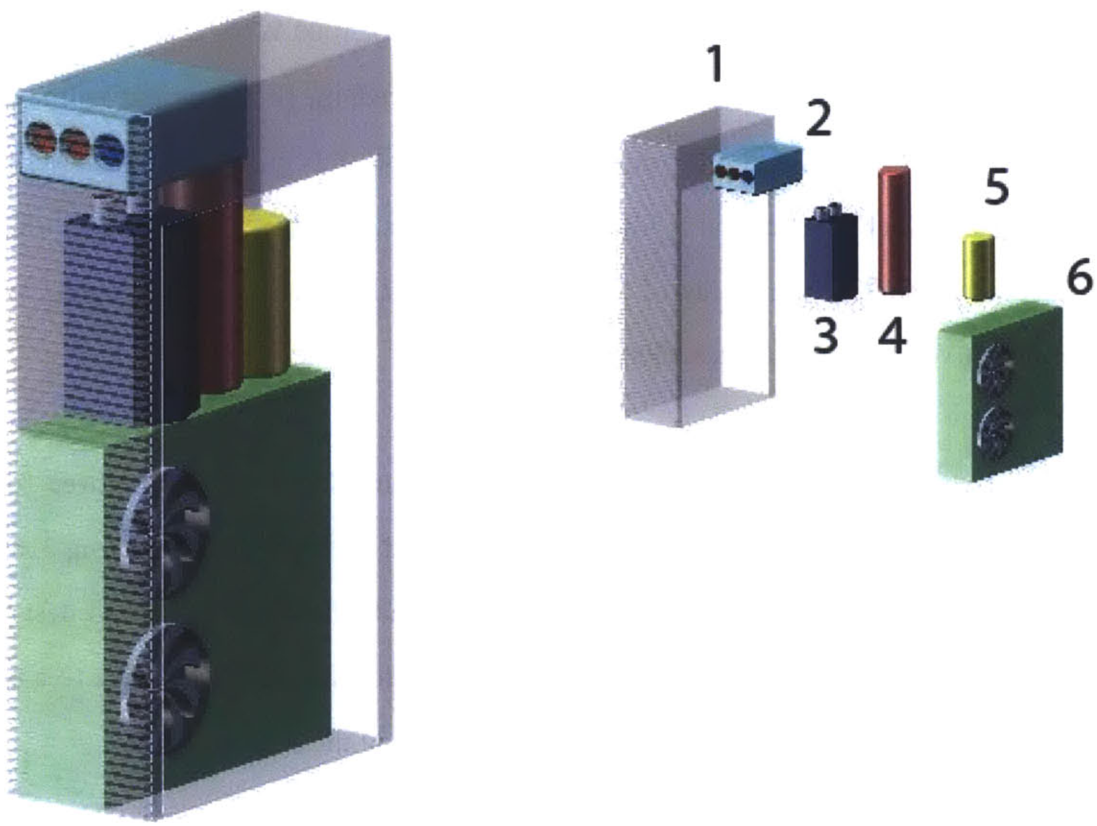


Figure 44: HVAC Cabinet

Appendix E Potential Use of Photovoltaics with The Robotic Façade

As part of the design process, the potential to incorporate photovoltaic panels on the façade module to off-set the energy consumed by the automated shading systems, was considered.

The potential design is built around a 4" x 4" solar cell, manufactured by Solar Winds USA [24], delivering 2W when under standard test conditions of 1000 W/m^2 incident light, and priced at $2.10 per cell. Two configurations were considered, the first, pictured as image A in Figure 45, simulates the use of cells on each of the movable slats of the shades, as well as the back of the light shelves. Each slat is fitted with seven cells, and there are twenty-four slats in total. Each light shelf has an array of thirty-six cells, totaling 240 for the complete system, delivering 480W, and costing circa $500.

The second configuration, which is shown as image B in Figure 45, places three rows of thirty-six cells on a fixed plate above the shades, totaling at 108 cells, and delivering 216W for circa $227. This configuration was chosen for further study due to its simplified design; the cells positioned on the slats of the shades have to be able to transmit power back to a battery bank, but still translate back and forth with the movement of the shades.

In addition to the cells, the following equipment is also specified for a PV system:

- Charge controller. Estimated price circa $100.
- Deep draw battery. It is suggested that a battery similar to that used in a golf cart is used. A 6V, 1.2kW-hr version is estimated at $125.

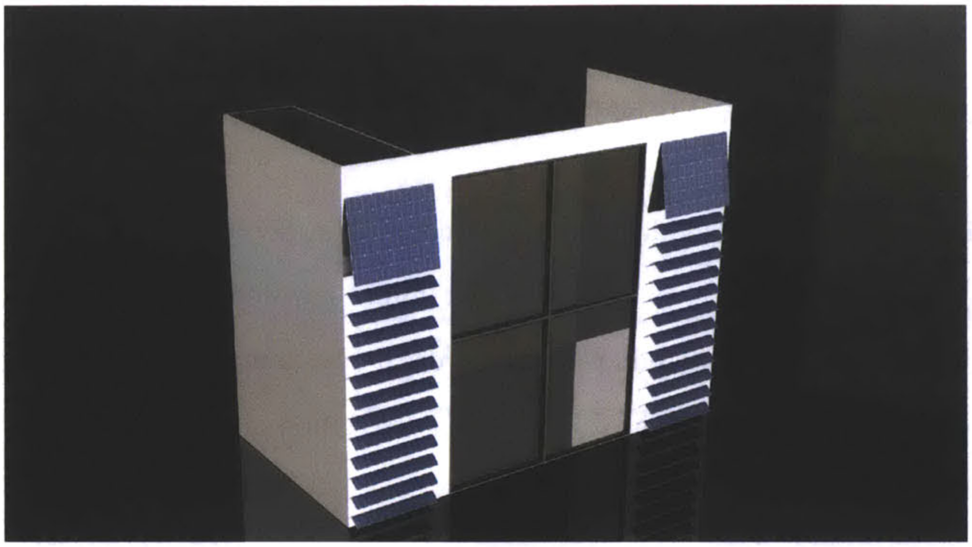
- Motor to power actuated shades. A conservative power rating estimate of 100W means that a 1.2kW-hr battery with 80% deep draw ability has 0.96kW-hr available, and would give 9.6 hours of motor power. It is further estimated that an opening/ closing operation of the shades would last 3 minutes, so a single charge of the battery would power circa 190 actuations, or over 6 actuations per day.

In order to give a first approximation as to where PV panels may be viable to power the shade actuation, a tool named PVWatts [25], developed by the National Renewable Energy Lab, was employed. Given the area and efficiency of the solar cells proposed in configuration B of Figure 45, PVWatts was used to simulate the energy that could potentially be captured for four orientations in each of the cities tested in Section 0. The results of these simulations are given in Table 7, and indicate that, despite the unfavorable latitudes and orientations in certain cases, the fact that such little energy is used by the proposed actuation system means that all locations are capable of capturing enough solar energy to power the motorized shades.

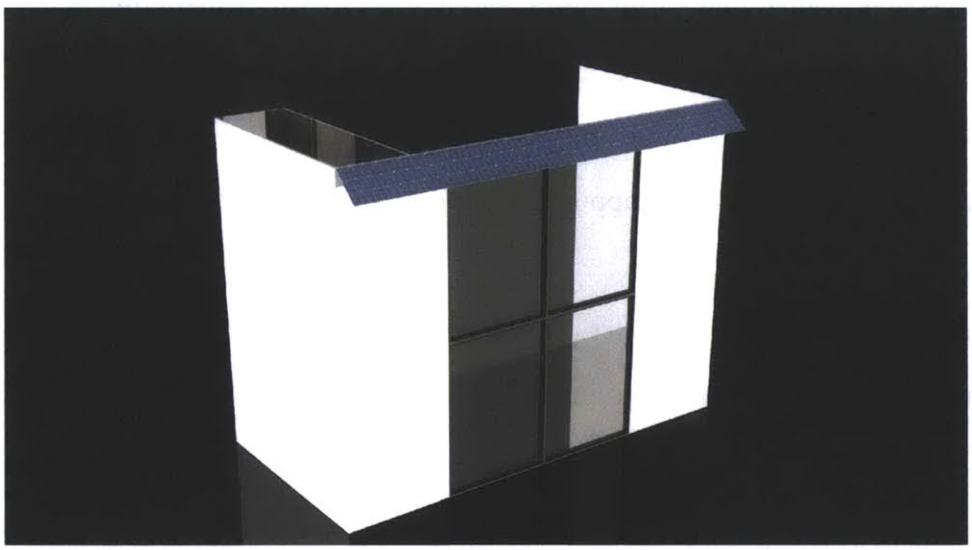
Note that Table 7 only includes yearly totals for the number of kW-hr of energy that can be captured, but from the more extensive set of results that was analyzed, it was apparent that each month in each location was delivering over the 1 kW-hr of energy, which surpasses the energy required in the design.

Boston							
North		South		East		West	
	kWh		kWh		kWh		kWh
Year	**138.024**	**Year**	**336.096**	**Year**	**250.560**	**Year**	**243.864**
USD	16.286	USD	39.660	USD	29.566	USD	28.776
LA							
North		**South**		**East**		**West**	
	kWh		kWh		kWh		kWh
Year	**213.408**	**Year**	**396.576**	**Year**	**301.752**	**Year**	**330.048**
USD	26.676	USD	49.572	USD	37.718	USD	41.256
Beijing							
North		**South**		**East**		**West**	
	kWh		kWh		kWh		kWh
Year	**151.632**	**Year**	**348.408**	**Year**	**255.096**	**Year**	**253.152**
Mumbai							
North		**South**		**East**		**West**	
	kWh		kWh		kWh		kWh
Year	**289.656**	**Year**	**351.648**	**Year**	**318.816**	**Year**	**322.056**
USD	12.043	USD	14.621	USD	13.255	USD	13.390
Quebec							
North		**South**		**East**		**West**	
	kWh		kWh		kWh		kWh
Year	**111.672**	**Year**	**315.144**	**Year**	**230.040**	**Year**	**232.416**
USD	9.618	USD	27.139	USD	19.809	USD	20.014
London							
North		**South**		**East**		**West**	
	kWh		kWh		kWh		kWh
Year	**76.680**	**Year**	**215.352**	**Year**	**146.016**	**Year**	**154.440**
USD	3.569	USD	10.021	USD	6.795	USD	7.187

Table 7: PVWatts Results



A



B

Figure 45: PV configuration options

Appendix F Robotic Façade: Future Customization Options

While the initial Robotic Façade module design, as outlined in Section 4.1, incorporates minimal customization options, it is intended that future revisions will leverage the standardized chassis frame, in conjunction with standardized interconnections, to allow for a greater degree of choice for the user. The following is a description of a proposed modular façade approach.

Essentially the façade is broken down into a grid, based on a one-foot square unit, and each sub-module that makes up the complete façade solution is based upon this standard unit. The sub-modules, be they dedicated shading units, or a glazing panels, are broken down into three module types, and are affixed to a support framework. The module types are defined as:

i. Self-contained modules: This description refers to those modules that can fully function independently of adjacent modules

ii. Layer-dependent modules: These modules rely on other interior/ exterior module layers to operate effectively; e.g. the use of an un-sealed external shading layer that requires the underlying glazing to be air-tight.

iii. Laterally-dependent modules: These modules require supporting modules to one or both sides for effective operation. E.g. a sliding shade that moves fully off to one side.

The hierarchy shown in Figure 46 provides a description of the sub-module types, while the rendering shown in Figure 47 illustrates how the interchangeable modules might fit together.

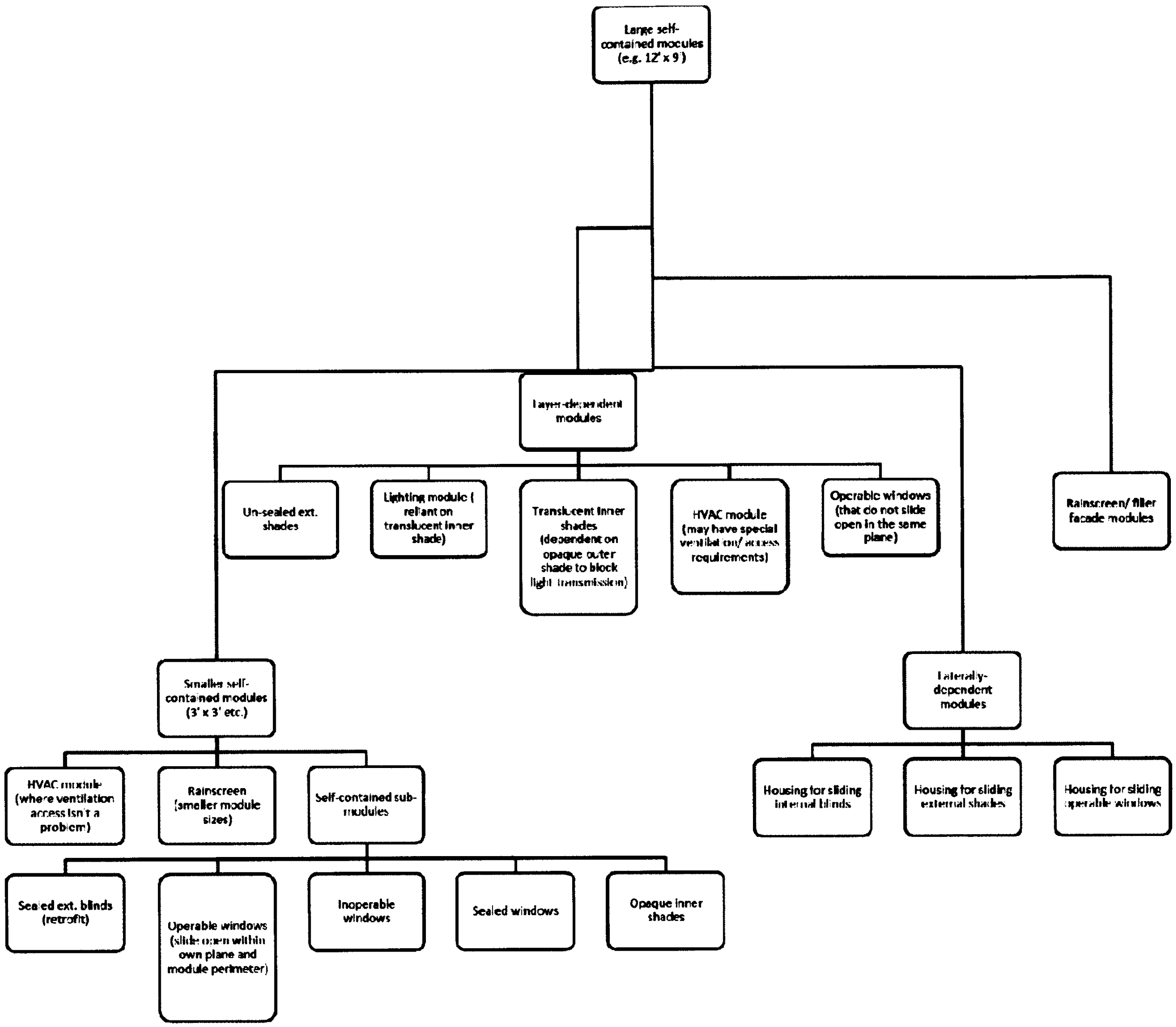


Figure 46: Robotic Facade modular hierarchy

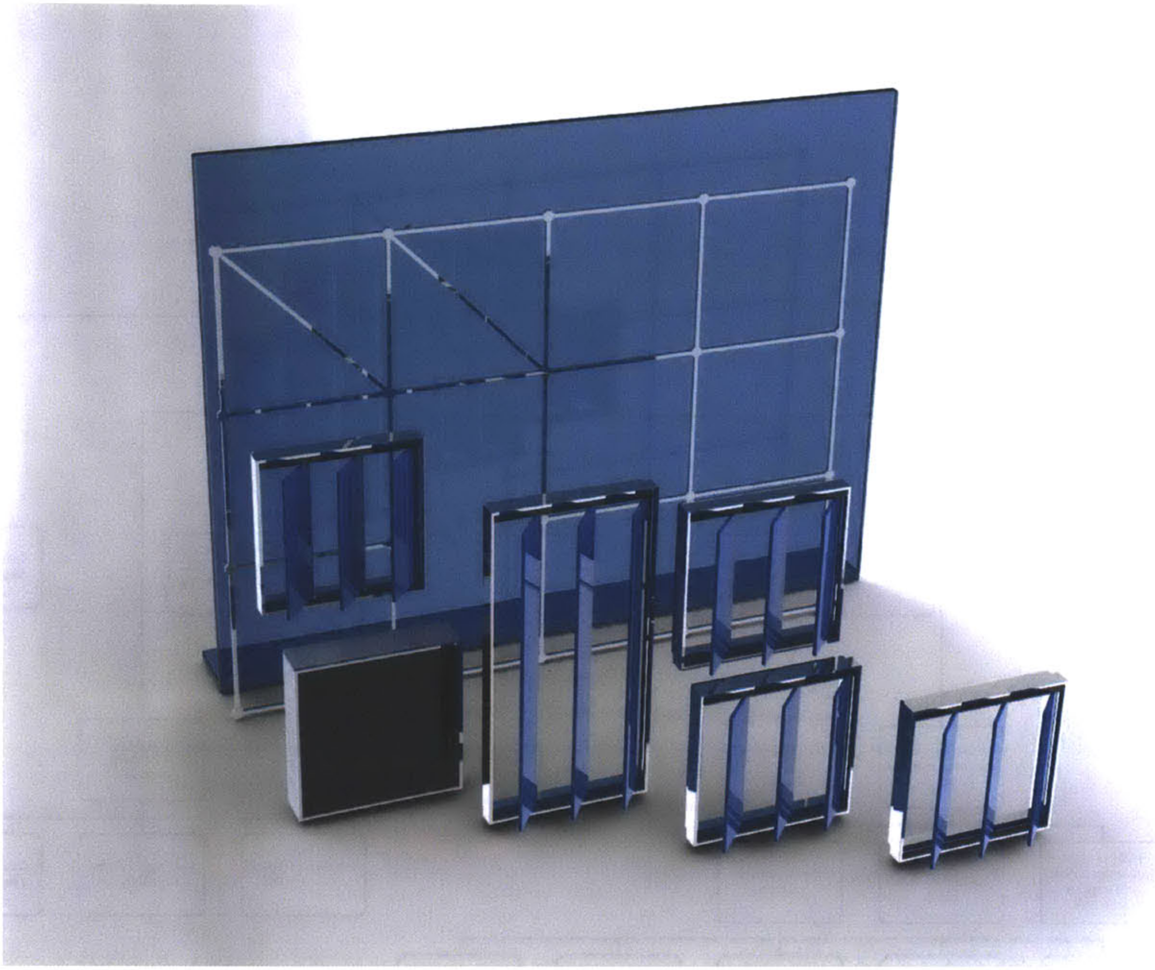


Figure 47: Example of interchangeable facade modules

Appendix G Robotic Façade Final Design

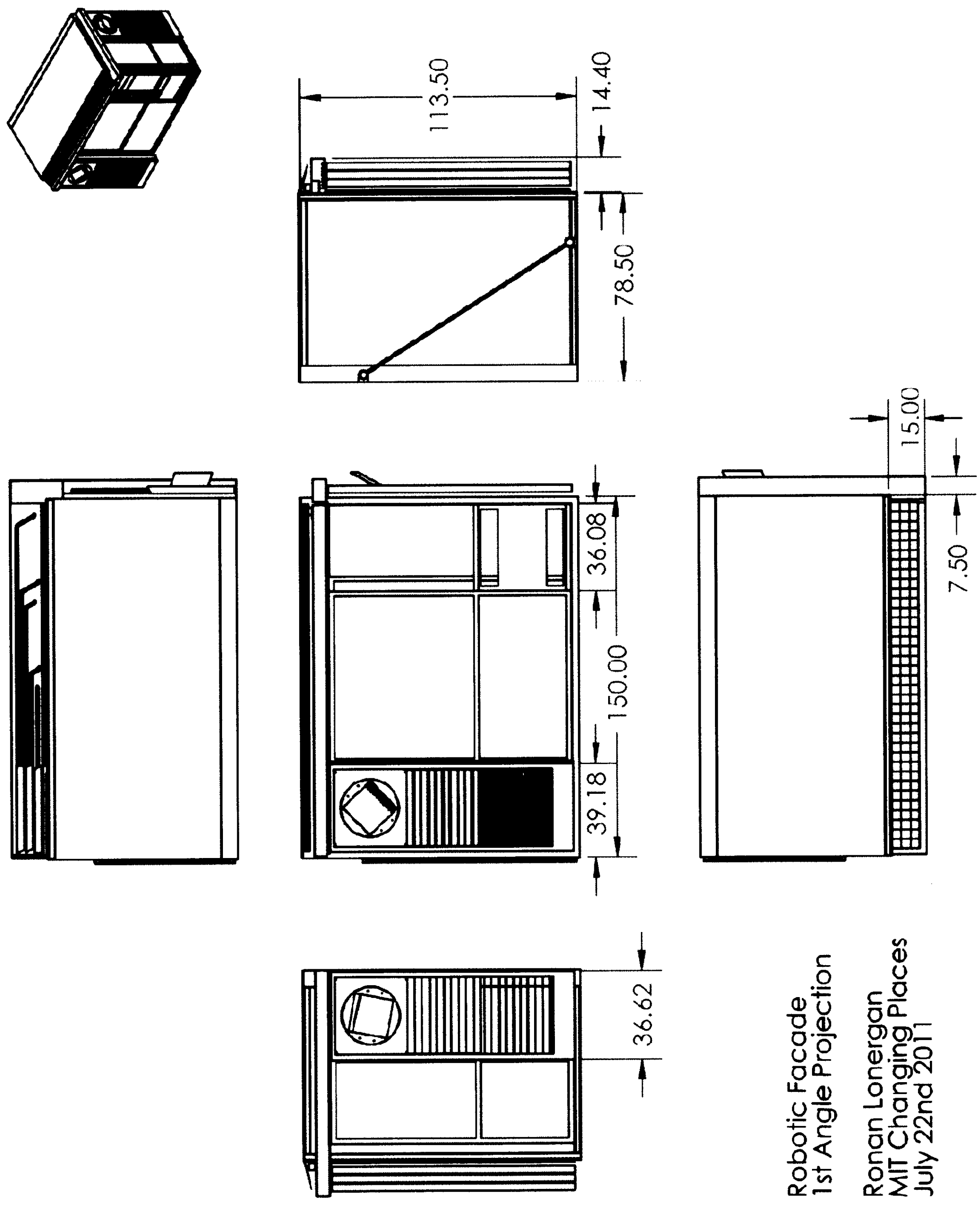


Figure 48: Robotic Facade Module orthographic projection

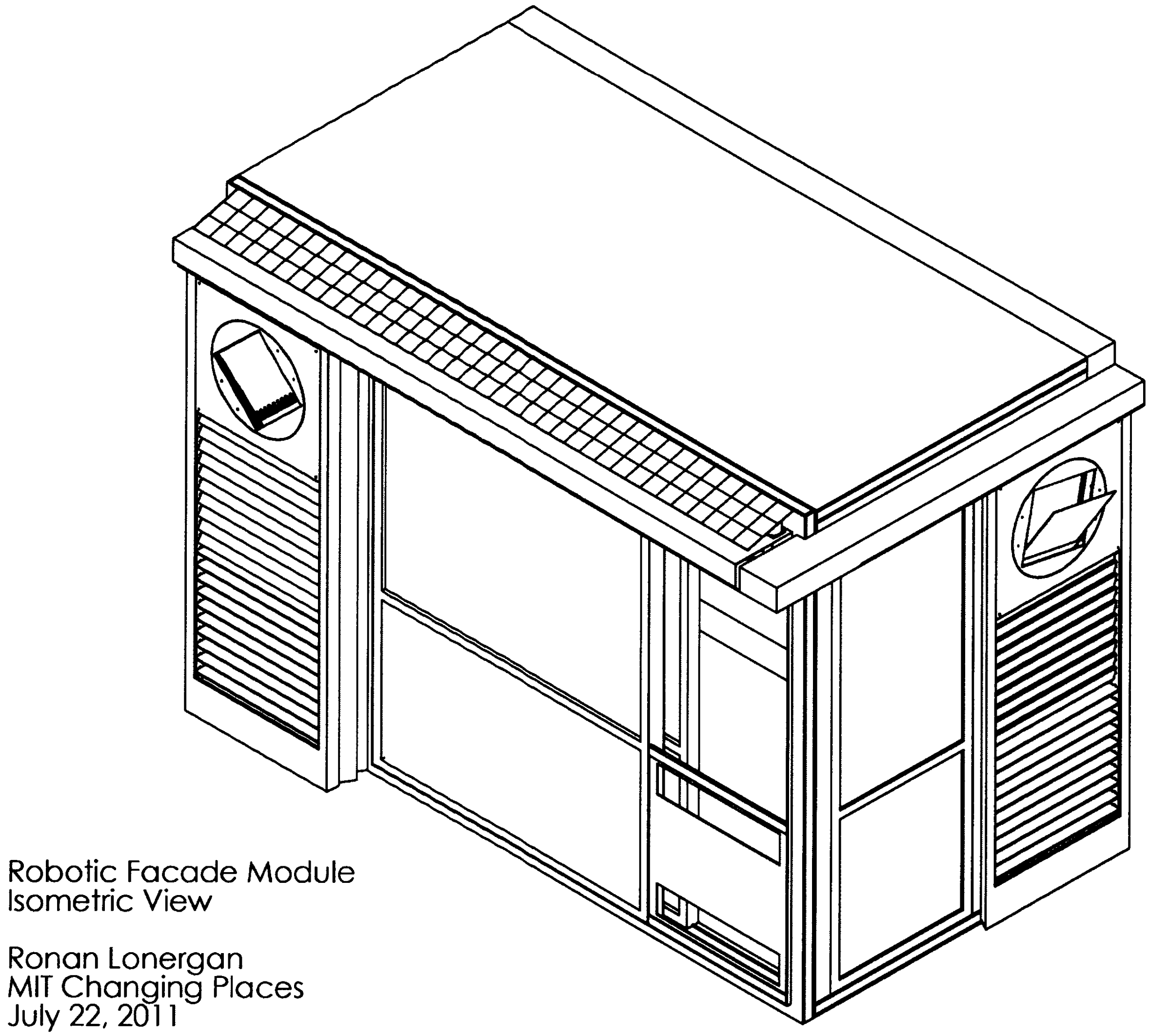


Figure 49: Robotic Façade Module isometric (front)

Robotic Facade-- Rear

Ronan Lonergan
MIT Changing Places
July 22, 2011

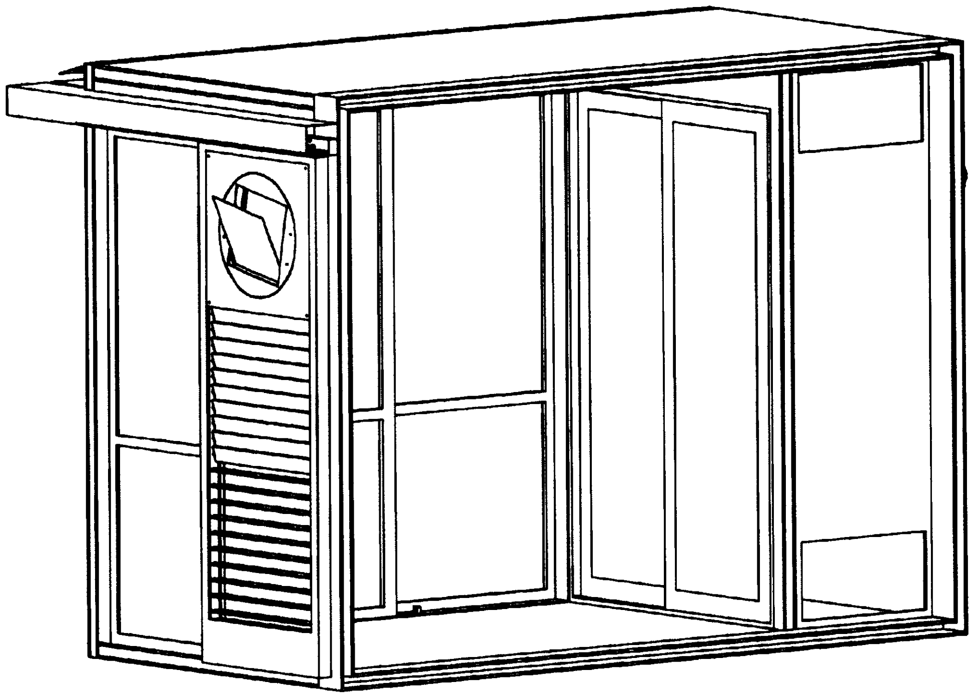


Figure 50: Robotic Façade isometric (rear)

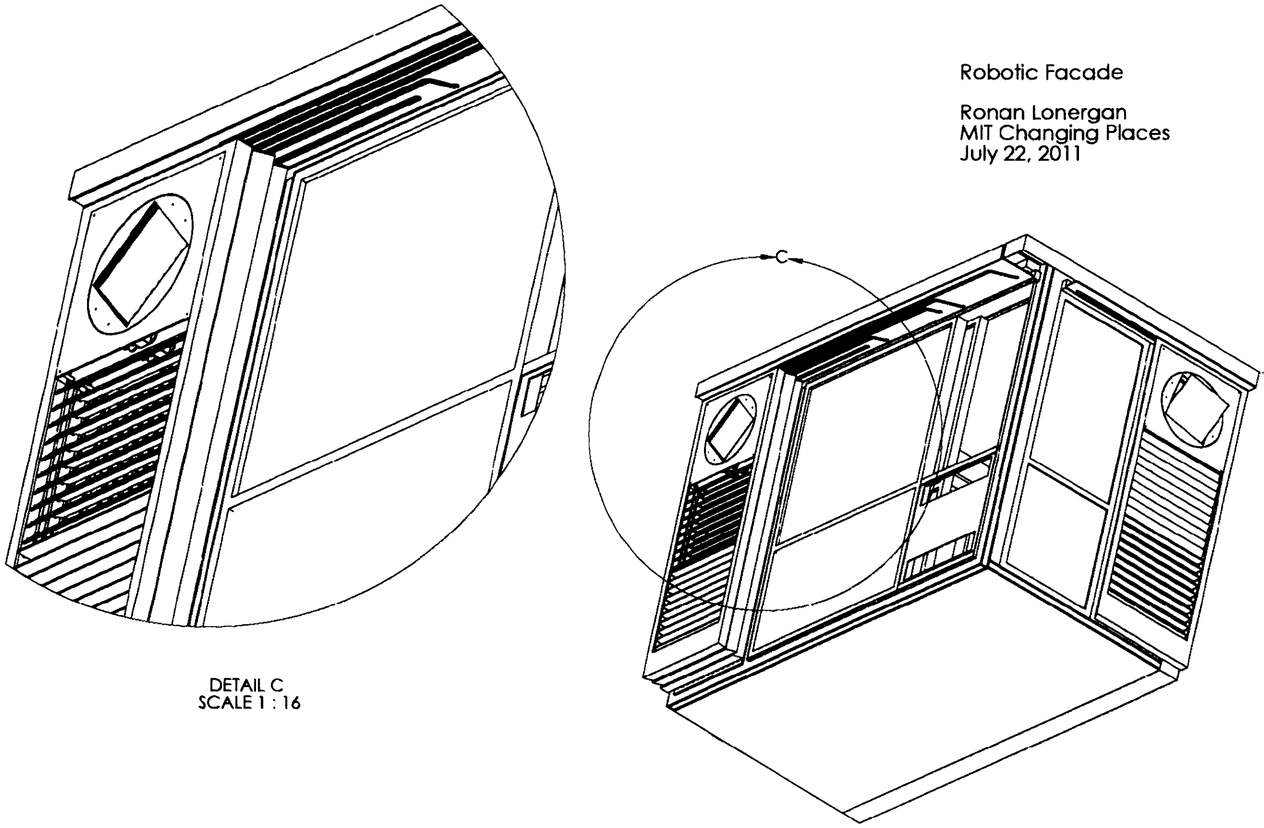


Figure 51: Robotic Façade Module (shading mechanism)

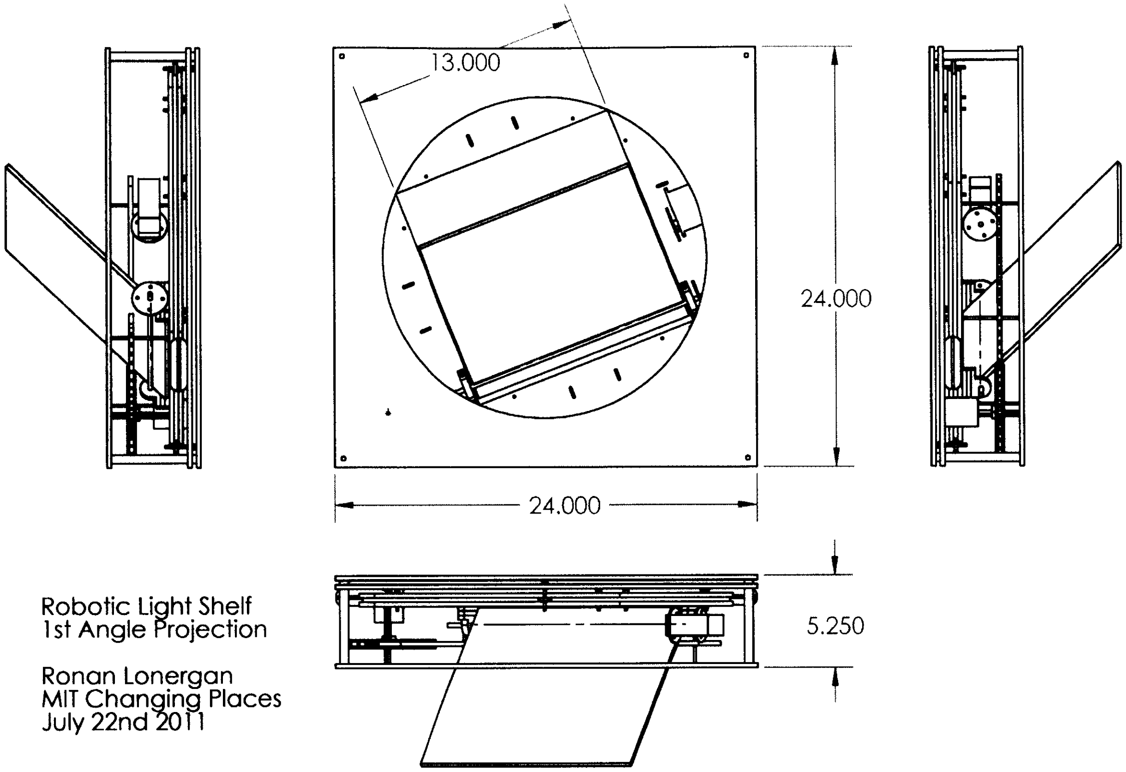


Figure 52: Robotic Light Shelf orthographic projection

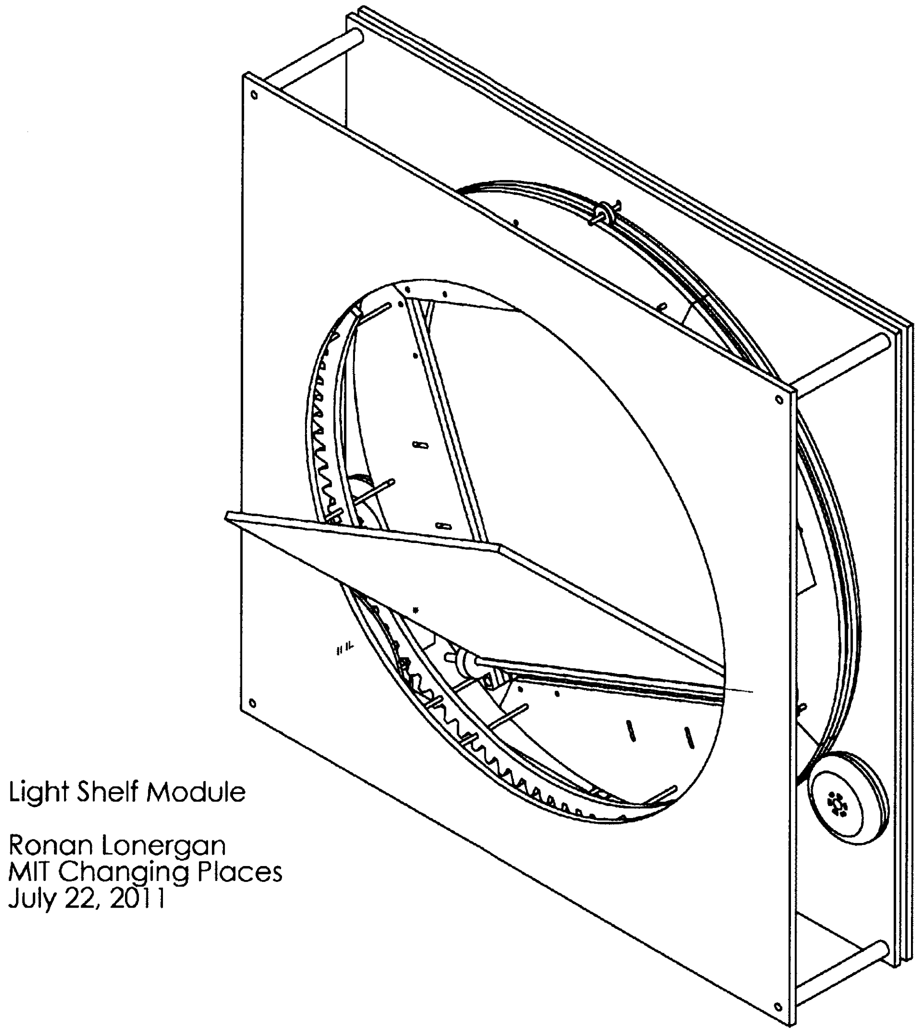


Figure 53: Robotic Light Shelf isometric view

Appendix H Arduino Code

The following is a copy of the code used to control one of the light shelf modules using an Arduino Mega. The parts that refer to the other two modules have been commented out, but can still be used in the future. This material can be uploaded to an Arduino microcontroller from the Arduino IDE.

```
//************************************************************
****
//************************************************************
****
#include <Servo.h>


   // Servo names—3 light shelves with 2 servo motors each
Servo continuous_servo1;  // create servo object to control a
servo
Servo lightShelf1;          // big light shelf servo 1

//************************************************************
****
// Accelerometer pins— Will interpret the voltage output from
// 3 accelerometers


int accelPinY1=2;    // accelerometer from LS 1 (Y)

//************************************************************
****

int intFailedTries=0;

byte emergency_stop =0;


//************************************************************
*

// Accelerometer 1 setup

// N.B. instead of considering the range of motion of the
shelf
// to be -70 deg to +70 deg, the new range will be +1 to +140

const int lightShelfNegAngle1= 1; // degrees from neutral
position
```

```
const int lightShelfPosAngle1= 141;   // degrees from neutral
position

const int lightShelfNegReading1= 195;  // from accelerometer 1
(Update as required)
const int lightShelfPosReading1= 508;  // from accelerometer 1

const int neutralAngle1= 70; // position shelf should return
to when reset

int tempValue1=0;
int tempValue2=0;

int lightShelfModuleID=1;    //which of the shelves are we
trying to control

int rotateAngle1=0;


int shelfAngle1=0;



//***********************************************************
*
// NB this code will receive a single byte value (0-255) from
an external program
// and must be adapted to receive integers outside of this
range.
// The serial monitor within the Arduino IDE will not work as
expected because
// each digit that is entered is essentially considered a
char, not an int. The
// code has been optimized to interface with a Java program
through a serial connection
void setup()
{

  Serial.begin(9600); //Prepare serial port for use

  //**************************************************
  // SERVO PINS
  continuous_servo1.attach(4);
  lightShelf1.attach(5);

  attachInterrupt(0, emStop, LOW);     // end-stop switch 1 on
pins 2 and 3
  attachInterrupt(1, emStop, LOW);    // end-stop switch 2

  Serial.println("Select which lightshelf you would like to
control:");
  Serial.println("Enter 215 for shelf 1");
```

```
  Serial.println("Enter 216 for shelf 2");
  Serial.println("Enter 217 for shelf 3");

}

void loop()
{


  while(emergency_stop==0)
  {
    // Establish which module we are talking about
    // 215=  lightshelf module 1
    // 216=  lightshelf module 2
    // 217=  lightshelf module 3

    lightShelfModuleID=saveSerialValue();
     Serial.println("Light shelf module found:");

      Serial.print("module ID: ");
      Serial.println(lightShelfModuleID, DEC);

    //*******************************************************
    //*******************************************************

    if(lightShelfModuleID==215){
      Serial.println("Light shelf 1");
      rotateAngle1=saveSerialValue();

      Serial.print("rotateAngle1: ");
      Serial.println(rotateAngle1, DEC);
      moveAll(lightShelf1,  continuous_servo1,rotateAngle1,  0,
accelPinY1);
      delay (4000);

      if(rotateAngle1<1 ||rotateAngle1>141){
        Serial.println("Inputted angle value out of bounds!");
// have to make sure that an out-of-bounds angle isn't given
from within Java code
        rotateAngle1=70;

      }

      shelfAngle1=saveSerialValue();
      Serial.print("shelfAngle1: ");
      Serial.println(shelfAngle1, DEC);
      moveAll(lightShelf1,     continuous_servo1,rotateAngle1,
shelfAngle1, accelPinY1);
      delay (4000);

      if(shelfAngle1<0  || shelfAngle1>90){
```

```
        Serial.println("Inputted  shelf  tilt  angle  out  of
bounds!");
        shelfAngle1=0;
      }

  //*******************************************************
    }else if(lightShelfModuleID==216){
      Serial.println("Light shelf 2");

      // rotateAngle2=saveSerialValue();

//         Serial.print("rotateAngle2: ");
//         Serial.println(rotateAngle2, DEC);
//
//
//        if(rotateAngle2<1 ||rotateAngle2>141){
//                Serial.println("Inputted  angle  value  out  of
bounds!");    // have to make sure that an out-of-bounds angle
isn't given from within Java code
//          rotateAngle2=70;
//        }

     // shelfAngle2=saveSerialValue();
//        Serial.print("shelfAngle2: ");
//        Serial.println(shelfAngle2, DEC);
//
//        if(shelfAngle2<0  || shelfAngle2>90){
//             Serial.println("Inputted shelf tilt angle out of
bounds!");
//          shelfAngle2=0;
//        }

    //*******************************************************
    }else if(lightShelfModuleID==217){
      Serial.println("Light shelf 3");

       //rotateAngle3=saveSerialValue();
//         Serial.print("rotateAngle3: ");
//         Serial.println(rotateAngle3, DEC);
//
//        if(rotateAngle3<1 ||rotateAngle3>141){
//                Serial.println("Inputted  angle  value  out  of
bounds!");    // have to make sure that an out-of-bounds angle
isn't given from within Java code
//          rotateAngle3=70;
//        }

    //  shelfAngle3=saveSerialValue();
//        Serial.print("shelfAngle3: ");
//        Serial.println(shelfAngle3, DEC);
//
//        if(shelfAngle3<0  || shelfAngle3>90){
```

```
//           Serial.println("Inputted shelf tilt angle out of
bounds!");
//        shelfAngle3=0;
//      }
//
    }else{
     Serial.println("Error with light shelf ID!!");
    }

    //**********************************************************
    //**********************************************************


  }

  Serial.println("Stopped for emergency");

}
//  END OF MAIN LOOP

//****************************************************

void emStop(){
  //This is an interrupt service routine
  emergency_stop=1;
  continuous_servo1.write(90);
 // resetShelf(lightShelf1, continuous_servo1, accelPinY1);

//  continuous_servo2.write(90);
//  resetShelf(lightShelf2, continuous_servo2, accelPinY2);
//
//  continuous_servo3.write(90);
//  resetShelf(lightShelf3, continuous_servo3, accelPinY3);
}
//
////****************************************************
//
void continuousSpeed(Servo servoname, byte direct){

  // direct picks the rotation direction, 0 or 1

  if(emergency_stop==0){     //check in case there has been an
emergency stop situation

    if (direct==0){
      servoname.write(110);
      delay(250);
      servoname.write(90);   //explicitly tell servo to stop
turning after a given delay to present runaway
    }
    else if (direct==1){
      servoname.write(70);
```

```
      delay(250);
      servoname.write(90);   //explicitly tell servo to stop
turning after a given delay to present runaway

    }
  }
  else{
    servoname.write(90);
  }
}
//
////*********************************************************
*
// Function to return an int value corresponding to angular
position. Pass the accelerometer name
int getAngleVoltage(int anglePinName){

  int angleVolts=  analogRead(anglePinName);
  return (angleVolts);

}
//
////*********************************************************
// FOR MODULE 1
// input an angle in degrees, return angle corresponding to
servo

int mapAngle1(int angDegrees){
  int   angleVolts=   map(angDegrees,   lightShelfNegAngle1,
lightShelfPosAngle1,                    lightShelfNegReading1,
lightShelfPosReading1);
  return angleVolts;

}

////********************************************************

void  driveContinuous(Servo  servoname,  int  angle1,  Servo
shelfServoName, int accelPin){
  // pass the name of the continuous servo you want to drive,
the angle (in degrees to drive it to), the name of the servo
that drives
  // the associated shelf, and pin that the accelerometer is
on
  // move light shelf to neutral unright
  moveShelf( shelfServoName, 0);
  delay(1500);
  int target=0;
  if(accelPin==2){
  target= mapAngle1(angle1);
  Serial.println("Mapped angle 1");
  }else if(accelPin==3){
```

```
    //target= mapAngle2(angle1);
    Serial.println("Mapped angle 2");
  }else if(accelPin==4){
     // target= mapAngle3(angle1);
      Serial.println("Mapped angle 3");
    }


  Serial.print("Target: ");
  Serial.println(target,DEC);

  int currentPosition= getAngleVoltage(accelPin);

  while((currentPosition  >  (target+10))||(currentPosition  <
(target-10))){

    if( currentPosition > (target+10)){
      continuousSpeed(servoname,1);
      delay(1500);
      currentPosition= getAngleVoltage(accelPin);
      // Serial.print("Hi: ");
      // Serial.println(currentPosition,DEC);
      Serial.print("Target: ");
      Serial.println(target,DEC);
      Serial.print("Current: ");
      Serial.println(currentPosition,DEC);
    }
    else if(currentPosition < (target-10)){
      continuousSpeed(servoname,0);
      delay(1500);
      currentPosition= getAngleVoltage(accelPin);
      // Serial.print("Lo: ");
      //  Serial.println(currentPosition,DEC);
      Serial.print("Target: ");
      Serial.println(target,DEC);
      Serial.print("Current: ");
      Serial.println(currentPosition,DEC);
    }
    else{
      break;
    }
    //int currentPosition= getAngleVoltage(accelPinY1);
  }

  Serial.println("There!");
}

////**********************************************************
**
//// code to move the light shelf to a given angle

void moveShelf(Servo servoname, int angle){
```

```
  if(0<= angle <= 90){
    servoname.write(angle);
  }
  else{
    Serial.println("Angle outside acceptable range!");
  }
}

////*****************************************************

void   resetShelf(Servo   shelfName,   Servo   motorName,   int
accelPin){
  emergency_stop=0;
  moveShelf( shelfName, 0);
  driveContinuous(motorName, 0, shelfName, accelPin);
}

////****************************************************
//// combined rotation and shelf opening

void   moveAll(Servo   shelfName,   Servo   motorName,   int
rotationAngle, int shelfAngle, int accelPin){
  driveContinuous(motorName,     rotationAngle,     shelfName,
accelPin);
  moveShelf(shelfName, shelfAngle);
}

//******************************************************
//******************************************************

  int saveSerialValue(){
     Serial.flush();            //Include both of these lines
to reset the serial reader
     tempValue1, tempValue2=0;

     while (Serial.available()==0)
    {;}

    if(Serial.available()>0){
      //Serial.print("Peek:");
      //Serial.println(intAsciiToBinary(Serial.peek()));
      return Serial.read();
    }

    }

//******************************************************
//******************************************************
```

Appendix I Further System Calculations

I.1 : Solar Angle Calculations

The following steps were implemented in the system code to calculate the current solar azimuth (Az), and elevation (h) [26]:

i. Find Y, the year minus 1900.

ii. Find Z(J) from the following table:

January	J=1	Z(J)=0.5	July	J=7	Z(J)=180.5
February	2	30.5	August	8	211.5
March	3	58.5	September	9	242.5
April	4	89.5	October	10	272.5
May	5	119.5	November	11	303.5
June	6	150.5	December	12	333.5

iii. Find D, the number of days:

$$D = \text{integer}(365.25 \times Y) + Z(J) + K + UT/24$$

where K is the day of the month and UT is the universal time

iv. Find T, the fraction of a Julian century:

$$T = D/36525$$

v. Find L, the mean longitude of the sun:

$$L = 279.697 + 36000.769 \times T$$

vi. Find M, the mean anomaly of the sun:

$$M = 358.476 + 35999.050 \times T$$

vii. Find epsilon, the obliquity:

$$\text{epsilon} = 23.452 - 0.013 \times T$$

viii. Find lambda, the ecliptic longitude of the sun:

$$\text{lambda} = L + (1.919 - 0.005 \times T) \times \sin(M) + 0.020 \times \sin(2M)$$

ix. Find alpha, the right ascension of the sun:

alpha = arctan (tan(lambda) x cos(epsilon))

x. Find delta, the declination of the sun:

delta = arcsin (sin(lambda) x sin(epsilon))

xi. Find LONG, the east-longitude of your location.

xii. Find HA, the hour angle of the sun:

HA = L - alpha + 180 + 15 x UT + LONG

xiii. Find the elevation angle of the center of the sun, h:

h [degrees] =

ARCSIN [SIN(LAT) x SIN(DEC) + COS(LAT) x COS(DEC) x COS(HA)]

Find the azimuth of the sun, Az:

Az [degrees] = ARCTAN [SIN(HA) / (COS(HA) x SIN(LAT) - TAN(DEC) x COS(LAT)]

I.2 : Robotic Light Shelf Reflection Position Calculation

The following Java code is used to calculate the position of the reflected light in the living space using the solar position, and the current light-shelf rotation and pivot angles.

```
//****************************************************************
// Author: Ronan Lonergan
// Date: 15th May 2011
public class reflectionCalculator {    //calculates the position of
the point on the ceiling


	private int sunElevationAngle; // denoted as h in labbook
(deg)
	private int sunAzimuth;    //denoted as Az in labbook (deg)
	private int rotationAngle; // denoted as theta (deg)
	private int shelfSlopeAngle;    // denoted as alpha (deg)
	private int cmOriginToCeiling;   // number of centimeters from
(0,0,0) to ceiling ==> l1
	private int southAngleOffset;    // angle between due-south
and angle of the light shelf (delta) (deg)

	private double xSunPoint, ySunPoint, zSunPoint;
	private double xShelfPoint1, yShelfPoint1, zShelfPoint1;
	private double xShelfPoint2, yShelfPoint2, zShelfPoint2;
	private double lightShelfWidth = 12*2.54; // width of light
shelf in cm

	private double xIntersectPoint, yIntersectPoint,
zIntersectPoint; // point of intersection of reflected ray and
ceiling

	public reflectionCalculator(int sunElevationAngle, int
sunAzimuth, int rotationAngle, int shelfSlopeAngle, int
cmOriginToCeiling, int southAngleOffset){
		this.sunElevationAngle = sunElevationAngle;
		this.sunAzimuth= sunAzimuth;
		this.rotationAngle= rotationAngle;
		this.shelfSlopeAngle= shelfSlopeAngle;
		this.cmOriginToCeiling= cmOriginToCeiling;
		this.southAngleOffset= southAngleOffset;
```

```
	}

	public void reflectionCalc1(){
		//Sun vector
		int psi= sunAzimuth-180;
		int phi= psi-southAngleOffset;
		double gamma= rotationAngle*shelfSlopeAngle/90.0;
		double beta= 1.0*shelfSlopeAngle*(1-
(Math.abs(rotationAngle)/90.0));

		xSunPoint= Math.sin(1.0*phi*Math.PI/180.0);
		ySunPoint= Math.sin(1.0*sunElevationAngle*Math.PI/180.0);
		zSunPoint= Math.cos(1.0*phi*Math.PI/180.0);

		xShelfPoint1= -1.0*Math.sin(1.0*gamma*Math.PI/180.0);
		yShelfPoint1= 1.0*Math.sin(1.0*beta*Math.PI/180.0);
		zShelfPoint1= 1.0*Math.cos(1.0*gamma*Math.PI/180.0);

		xShelfPoint2=1.0*Math.cos(rotationAngle*Math.PI/180.0);
		yShelfPoint2=1.0*Math.sin(rotationAngle*Math.PI/180.0);
		zShelfPoint2=0.0;

		//Equation of plane in space:
		// ax+by+cz+d=0

		double aPlane= (yShelfPoint1*zShelfPoint2)-
(yShelfPoint2*zShelfPoint1);
		double bPlane= (-1.0*xShelfPoint1*zShelfPoint2)+
(xShelfPoint2*zShelfPoint1);
		double cPlane= (xShelfPoint1*yShelfPoint2)-
(xShelfPoint2*yShelfPoint1);
		double dPlane= 0;

		// aPlane, bPlane are the coeffs for the equation of the
plane of the lightShelf
		// normal to lightShelf is aPlane i + bPlane j + cPlane k



		// xSunPoint, ySunPoint, zSunPoint make up a vector from
(0,0,0) to the SunPoint
		// have a normal vector to light shelf plane
```

```
            // eqn of plane made up of incident (sun to light shelf)
vector, and light shelf normal
            // is Ax+By+Cz+D=0 ==> call this the "reflectionPlane"

            double AreflectPlane= ySunPoint*cPlane-
(bPlane*zSunPoint);
            double BreflectPlane= aPlane*zSunPoint-
(xSunPoint*cPlane);
            double CreflectPlane= xSunPoint*bPlane-
(aPlane*ySunPoint);
            double DreflectPlane= 0.0; // since the plane passes
through (0,0,0)


            // Equation of the ceiling plane.
            // takes the same form as before; ax+by+cz+d=0
            // but it is parallel to x and z axes
            // Equation of ceiling plane is therefore; y=
cmOriginToCeiliing+(lightShelfWidth/2)*Sin(theta)


            double xCeilingPlane= 0.0;
            double yCeilingPlane= 1.0;
            double zCeilingPlane= 0.0;
            double dCeilingPlane= -1.0*(cmOriginToCeiling +
((lightShelfWidth/2)*Math.sin(rotationAngle*Math.PI/180)));

            // cross product of ceiling plane and reflection plane
            // essentially the cross product of the two normals of
these planes
            // from this one finds a vector that is parallel to the
line of intersection

            double xIntersection= 1.0*yCeilingPlane*CreflectPlane;
            double yIntersection= 0.0;
            double zIntersection= -1.0*yCeilingPlane*AreflectPlane;

            // need a point on the line of intersection:
            // know yvalue is dCeilingPlane
            // know there will always be a point at z=0;
            // point is therefore (-
BreflectPlane/AreflectPlane*dCeilingPlane, dCeilingPlane, 0)

            double xPointOnIntersect= -
1.0*BreflectPlane/AreflectPlane*dCeilingPlane;
```

```
            double yPointOnIntersect= dCeilingPlane;
            double zPointOnIntersect= 0.0;

            // line of intersection is then: PointOnIntersect
+t(Intersection), where t can take any value


      //*************************************************************
************

            // Reflection vector= incident vector + 2*(dot prod of -
1*incident vector and surface normal)muliplied by surface normal
            // incident vector is from the Sun to (0,0,0)

            double incidentX= -1.0*xSunPoint;
            double incidentY= -1.0*ySunPoint;
            double incidentZ= -1.0*zSunPoint;

            // surface normal dot -1*incident vector == surface
normal dot SunPoint

            // length of SunPoint vector
            double lengthSunPoint= Math.sqrt(Math.pow(xSunPoint, 2)+
Math.pow(ySunPoint, 2)+ Math.pow(zSunPoint,2));

            // length of vector normal to light shelf surface
            double lengthSurfaceNormal= Math.sqrt(Math.pow(aPlane,
2)+ Math.pow(bPlane, 2)+ Math.pow(cPlane,2));

            //Cos of the angle between normal and incident light ray:

            double cosAngle= ((xSunPoint*aPlane) + (ySunPoint*bPlane)
+ (zSunPoint*cPlane))/(lengthSunPoint*lengthSurfaceNormal);

            // Reflection vector:

            double xReflect= incidentX + (2.0*cosAngle*aPlane);
            double yReflect= incidentY + (2.0*cosAngle*bPlane);
            double zReflect= incidentZ + (2.0*cosAngle*cPlane);

            // Reflection line (passes through (0,0,0))
            // reflection line= t(xReflect)i + t(yReflect)j+
t(zReflect)k where t is any number
```

```
            // point of intersection of ceiling plane and reflection
line:

            double t= -1.0*dCeilingPlane/yReflect;

            // ** The required point of intersection: **
            xIntersectPoint= t*xReflect;
            yIntersectPoint= t*yReflect;
            zIntersectPoint= t*zReflect;

      }

      public double getXintersect(){
            return xIntersectPoint;
      }

      public double getYintersect(){
            return yIntersectPoint;
      }

      public double getZintersect(){
            return zIntersectPoint;
      }
}

//*******************************************************************
**
//*******************************************************************
**
```

I.3 Robotic Light-Shelf Maintain-Position Calculator

The following Java code returns rotation and pivot angle values that will position the reflected light at a saved point in the living space.

```
//*****************************************************************
// Author: Ronan Lonergan
// Date: May 15th, 2011

public class normalCalculator {

	// Given the sun position and the point where we want the sun
on the ceiling, this code calculates where the shelf should position
itself

	private int sunAzimuth;    // in degrees
	private int sunElevationAngle;   // in degrees
	private double intersectionPointX;    //The location of the
spot where the ray of light is required to hit the ceiling (in
centimeters)
	private double intersectionPointY;    // in cm
	private double intersectionPointZ;    // in cm
	private int southAngleOffset; // in degrees
	private double xSunPoint, ySunPoint, zSunPoint;
	private double n1, n2, n3; // calculated normal coeffs

	private double bestGuessTheta, bestGuessAlpha; // rotation
angle and slope angle that give the required reflection
	private double smallestAngle=180.0;
	//private double rotationAngle, shelfSlopeAngle;

	public normalCalculator(int sunAzimuth, int sunElevationAngle,
double intersectionPointX, double intersectionPointY, double
intersectionPointZ, int southAngleOffset){


		this.sunAzimuth= sunAzimuth;
		this.sunElevationAngle= sunElevationAngle;
		this.intersectionPointX=intersectionPointX;
		this.intersectionPointY=intersectionPointY;
		this.intersectionPointZ= intersectionPointZ;
		this.southAngleOffset= southAngleOffset;
```

```
	}

	public void calculateNormalCoeffs(){

		//Sun vector
		int psi= sunAzimuth-180;
		int phi= psi-southAngleOffset;


		xSunPoint= Math.sin(1.0*phi*Math.PI/180.0);
		ySunPoint= Math.sin(1.0*sunElevationAngle*Math.PI/180.0);
		zSunPoint= Math.cos(1.0*phi*Math.PI/180.0);

		// make SunPoint a unit vector;

		double lengthOfSunPoint= Math.sqrt(Math.pow(xSunPoint,
2)+Math.pow(ySunPoint, 2)+ Math.pow(zSunPoint, 2));

		xSunPoint= xSunPoint/lengthOfSunPoint;
		ySunPoint= ySunPoint/lengthOfSunPoint;
		zSunPoint= zSunPoint/lengthOfSunPoint;
	// incident vector is from the Sun to (0,0,0)

		double incidentX= -1.0*xSunPoint;
		double incidentY= -1.0*ySunPoint;
		double incidentZ= -1.0*zSunPoint;

	// reflected vector
	// intersectionPointX i + intersectionPointY j +
intersectionPointZ k

		double lengthOfReflectionVector=
Math.sqrt(Math.pow(intersectionPointX,
2)+Math.pow(intersectionPointY, 2)+ Math.pow(intersectionPointZ,
2));


	intersectionPointX=1.0*intersectionPointX/lengthOfReflectionVe
ctor;

	intersectionPointY=1.0*intersectionPointY/lengthOfReflectionVe
ctor;
```

```
	intersectionPointZ=1.0*intersectionPointZ/lengthOfReflectionVe
ctor;

	// reflected - incident /2 = Ai + Bj + Ck

		double A= (intersectionPointX-incidentX)/2.0;
		double B= (intersectionPointY-incidentY)/2.0;
		double C= (intersectionPointZ-incidentZ)/2.0;

		double denominator1= Math.sqrt((-1.0*A*incidentX)-
(1.0*B*incidentY)-(1.0*C*incidentZ));

		n1=1.0*A/denominator1;
		n2=1.0*B/denominator1;
		n3=1.0*C/denominator1;


		angleGuesser();



	}

	public void angleGuesser(){

		for(int i=0; i<900; i++){
			for(int j=-700; j< 700; j++){
				double angleFound= calculateAngle(i/10, j/10);
				if(angleFound<smallestAngle){
					bestGuessAlpha=i/10;  //slope angle
					bestGuessTheta=j/10;  //rotation angle
					smallestAngle=angleFound;
				}
			}
		}
		System.out.println("Smallest angle between required and
guessed normals: " +  smallestAngle + " deg");
		System.out.println("At shelf angles: theta: " +
bestGuessTheta + " deg. Alpha: " + bestGuessAlpha + " deg.");

	}

	public double calculateAngle( double shelfSlopeAngle, double
rotationAngle ){
```

```
            double gamma= rotationAngle*shelfSlopeAngle/90.0;
            double beta= 1.0*shelfSlopeAngle*(1-
(Math.abs(rotationAngle)/90.0));

            double xShelfPoint1= -
1.0*Math.sin(1.0*gamma*Math.PI/180.0);
            double yShelfPoint1=
1.0*Math.sin(1.0*beta*Math.PI/180.0);
            double zShelfPoint1=
1.0*Math.cos(1.0*gamma*Math.PI/180.0);

            double
xShelfPoint2=1.0*Math.cos(rotationAngle*Math.PI/180.0);
            double
yShelfPoint2=1.0*Math.sin(rotationAngle*Math.PI/180.0);
            double zShelfPoint2=0.0;

            //Equation of plane in space:
            // ax+by+cz+d=0

            double aPlane= (yShelfPoint1*zShelfPoint2)-
(yShelfPoint2*zShelfPoint1);
            double bPlane= (-1.0*xShelfPoint1*zShelfPoint2)+
(xShelfPoint2*zShelfPoint1);
            double cPlane= (xShelfPoint1*yShelfPoint2)-
(xShelfPoint2*yShelfPoint1);
            double dPlane= 0;

            // Unit normal to plane:

            double planeNormalLength= Math.sqrt(Math.pow(aPlane,
2)+Math.pow(bPlane, 2)+ Math.pow(cPlane, 2));

            //make unit normal (this is the guessed normal)
            double planeNormX= aPlane/ planeNormalLength;
            double planeNormY= bPlane/ planeNormalLength;
            double planeNormZ= cPlane/ planeNormalLength;

            // make required normal unity length (should be already)

            double requiredNormalLength= Math.sqrt(Math.pow(n1,
2)+Math.pow(n2, 2)+ Math.pow(n3, 2));

            double requiredNormX= n1/requiredNormalLength;
            double requiredNormY= n2/requiredNormalLength;
```

```
            double requiredNormZ= n3/requiredNormalLength;

            // dot the two normals with each other

            double dotProd= (planeNormX*requiredNormX) +
(planeNormY*requiredNormY) + (planeNormZ*requiredNormZ);

            double angleBetweenNormals=
Math.acos(dotProd)*180.0/Math.PI; // in degrees

            return angleBetweenNormals;



      }

      public double getBestGuessTheta(){
            return bestGuessTheta;      //returns the rotation angle
      }

      public double getBestGuessAlpha(){
            return bestGuessAlpha;
      }


}
//****************************************************************
//****************************************************************
```

Bibliography

[1] UN-HABITAT, "World Urban Forum III: An International UN-HABITAT Event on Urban Sustainability," 19-Jun-2006.

[2] United Nations, *World Urbanization Prospects: The 2003 Revision*. 2004.

[3] Energy Star, *Energy Star Facts on Energy Use*. U.S. Department of Energy, U.S. Environmental Protection Agency, 2010.

[4] National Energy Technology Laboratory, *2008 Buildings Energy Data Book*. U.S. Department of Energy, 2009.

[5] National Energy Technology Laboratory, *2010 Buildings Energy Data Book*. U.S. Department of Energy, 2011.

[6] "L'Institut du monde arabe à Paris." [Online]. Available: http://www.imarabe.org/. [Accessed: 24-Jul-2011].

[7] J. Deyette and B. Freese, *Burning Coal, Burning Cash*. Union of Concerned Scientists, 2010.

[8] "Unico System Inc. - Small Duct High Velocity HVAC The Unico System." [Online]. Available: http://www.unicosystem.com/. [Accessed: 24-Jul-2011].

[9] B. Olesen, "Radiant floor heating in theory and practice," *ASHRAE JOURNAL*, vol. 44, no. 7, p. 19-+, Jul. 2002.

[10] "Chapter 6: Panel Heating and Cooling," in *2008 ASHRAE Handbook- Heating, Ventilating, and Air-Conditioning Systems and Equipment*, American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2008.

[11] C. Stetiu, "Energy and peak power savings potential of radiant cooling systems in US commercial buildings," *Energy and Buildings*, vol. 30, no. 2, pp. 127-138, Jun. 1999.

[12] J. Dieckmann, K. McKenney, and J. Brodrick, "Radiant Floor Cooling in Practice," *ASHRAE JOURNAL*, vol. 51, no. 11, p. 70-+, Nov. 2009.

[13] J. Lim, J. Jo, Y. Kim, M. Yeo, and K. Kim, "Application of the control methods for radiant floor cooling system in residential buildings," *BUILDING AND ENVIRONMENT*, vol. 41, no. 1, pp. 60-73, Jan. 2006.

[14] O. Bjarne, "Radiant Floor Cooling Systems," *ASHRAE Journal*, vol. 50, pp. 16-20, Sep. 2008.

[15] X. Jin, X. Zhang, Y. Luo, and R. Cao, "Numerical simulation of radiant floor cooling system: The effects of thermal resistance of pipe and water velocity on the performance," *BUILDING AND ENVIRONMENT*, vol. 45, no. 11, pp. 2545-2552, Nov. 2010.

[16] Ö. I. für et al., *Passivhaus-Bauteilkatalog | Details for Passive Houses: Ã–kologisch bewertete Konstruktionen | A Catalogue of Ecologically Rated Constructions*, 3rd ed. Springer Vienna Architecture, 2009.

[17] "DesignBuilder - Building design, simulation and visualisation - Building Simulation... Made Easy." [Online]. Available: http://www.designbuilder.co.uk/. [Accessed: 24-Jul-2011].

[18] "Building Technologies Program: EnergyPlus Energy Simulation Software." [Online]. Available: http://apps1.eere.energy.gov/buildings/energyplus/. [Accessed: 24-Jul-2011].

[19] "BBC Weather." [Online]. Available: http://news.bbc.co.uk/weather/. [Accessed: 12-Jul-2011].
[20] J. Pine, *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press, 1999.
[21] "Rhinoceros GearGen." [Online]. Available: http://www.rayflectar.com/Rhino/RhinoScripts-Gallery.htm. [Accessed: 25-Jul-2011].
[22] "Arduino - HomePage." [Online]. Available: http://www.arduino.cc/. [Accessed: 24-Jul-2011].
[23] K. Larson, "House_n The PlaceLab." [Online]. Available: http://architecture.mit.edu/house_n/placelab.html. [Accessed: 24-Jul-2011].
[24] "Solar Winds USA." [Online]. Available: http://www.solarwindsusa.com/magento/solar-cell-polycrystalline-4-x-4-100mm-x-100mm-no-tabs.html. [Accessed: 25-Jul-2011].
[25] National Renewable Energy Lab, "PVWATTS v. 1." [Online]. Available: http://rredc.nrel.gov/solar/calculators/PVWATTS/version1/. [Accessed: 25-Jul-2011].
[26] Y. Ma, G. Li, and R. Tang, "Optical performance of vertical axis three azimuth angles tracked solar panels," *Applied Energy*, vol. 88, no. 5, pp. 1784-1791, May. 2011.