The University of Nottingham

Doctoral Thesis

Multi-Agent Stochastic Simulation of Occupants in Buildings

Author:
Jacob Chapman

Supervisors:
Prof. Darren Robinson
Dr Peer-Olaf Siebers

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

The Faculty of Engineering
Architecture and Built Environment

January 2017

THE UNIVERSITY OF NOTTINGHAM

Abstract

Faculty of Engineering
Architecture and Built Environment

Doctor of Philosophy

Multi-Agent Stochastic Simulation of Occupants in Buildings

by Jacob Chapman

One of the principle causes for deviations between predicted and simulated performance of buildings relates to the stochastic nature of their occupants: their presence, activities whilst present, activity dependent behaviours and the consequent implications for their perceived comfort. A growing research community is active in the development and validation of stochastic models addressing these issues; and considerable progress has been made. Specifically models in the areas of presence, activities while present, shading devices, window openings and lighting usage.

One key outstanding challenge relates to the integration of these prototype models with building simulation in a coherent and generalizable way; meaning that emerging models can be integrated with a range of building simulation software. This thesis describes our proof of concept platform that integrates stochastic occupancy models within a multi agent simulation platform, which communicates directly with building simulation software. The tool is called Nottingham Multi-Agent Stochastic Simulation (No-MASS).

No-MASS is tested with a building performance simulation solver to demonstrate the effectiveness of the integrated stochastic models on a residential building and a non-residential building. To account for diversity between occupants No-MASS makes use of archetypical behaviours within the stochastic models of windows, shades and activities. Thus providing designers with means to evaluate the performance of their designs in response to the range of expected behaviours and to evaluate the robustness of their design solutions; which is not possible using current simplistic deterministic representations.

A methodology for including rule based models is built into No-MASS, this allows for testing what-if scenarios with building performance simulation and provides a pragmatic basis for the modelling of the behaviours for which there is insufficient data to develop stochastic models. A Belief-Desire-Intention model is used to develop a set of goals and plans that an

agent must follow to influence the environment based on their beliefs about current environmental conditions. Recommendations for the future development of stochastic models are presented based on the sensitivity analysis of the plans.

A social interactions framework is developed within No-MASS to resolve conflicts between competing agents. This framework resolves situations where each agent may have different desires, for example one may wish to have a window open and another closed based on the outputs of the stochastic models. A votes casting system determines the agent choice, the most votes becomes the action acted on.

No-MASS employs agent machine learning techniques that allow them to learn how to respond to the processes taking place within a building and agents can choose a strategy without the need for context specific rules.

Employing these complementary techniques to support the comprehensive simulation of occupants presence and behaviour, integrated within a single platform that can readily interface with a range of building (and urban) energy simulation programs is the key contribution to knowledge from this thesis. Nevertheless, there is significant scope to extend this work to further reduce the performance gap between simulated and real world buildings.

Acknowledgements

I wish to express my gratitude to my supervisors; Professor Darren Robinson and Assistant Professor Peer-Olaf Siebers for their tireless work discussing new ideas and providing the main motivation for the completion of this work. It has been a great honour working with them both and their passion, knowledge and drive moved this thesis from a conceptual idea to a finished product that can be released into the world. Darren's knowledge of building simulation and occupant modelling and Peer-Olaf's skills in agent simulation and the social sciences were key to the development and usefulness of Nottingham Multi Agent Stochastic Simulation (No-MASS) tool.

I have been lucky to work with some great and very knowledgeable colleagues in the Laboratory for Urban Complexity And Sustainability at The University of Nottingham. A special thanks to Said Jaboob for helping me understand his work in stochastic modelling and Ana Sancho Tomás for helping develop my ideas on building performance simulation/ agent simulation and the difficulties in linking the two.

I am thankful to The Horizon Digital Economy Research Centre for Doctoral Training for funding my work and supporting me in my application to push No-MASS. Giving me time to develop, test and extend my ideas on the areas of knowledge I wished to focus my expertise on is greatly appreciated. I also wish to mention the help and support I received from DesignBuilder for their help in integrating No-MASS within their building simulation tool. With Andy Tindale and his team this work is now accessible to building designers, vastly improving the impact No-MASS will have on the world.

I would like to thank my family for their help and support, especially Chrisminder Hare, your motivation and hard work has helped significantly. This thesis is dedicated to the memory of my beloved grandfather, Norman Barr.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The resources needed to sustain the world's ever expanding population are reaching new levels. In 2015, representatives from 195 countries signed an agreement with the aims of:

> "Holding the increase in the global average temperature to well below 2 °C above pre-industrial levels and pursuing efforts to limit the temperature increase to 1.5 °C above pre-industrial levels, recognizing that this would significantly reduce the risks and impacts of climate change;
>
> Increasing the ability to adapt to the adverse impacts of climate change and foster climate resilience and low greenhouse gas emissions development, in a manner that does not threaten food production; and
>
> Making finance flows consistent with a pathway towards low greenhouse gas emissions and climate-resilient development."
>
> (United Nations, 2015)

It is therefore important to reduce global energy use arising from the contribution of fossil fuels and the associated emission of greenhouse gasses.

In Europe the energy used within residential sector and the services sector accounted for 41% of the total energy used in 2013 and represented an increase of 5% relative to 1990 levels (Figure 1.1). Building performance simulation is used to model the flows of energy in buildings and their systems at the design/ retrofit stage, to ensure they not only meet the demands of the occupants but also allow designers to test strategies for improving building performance.

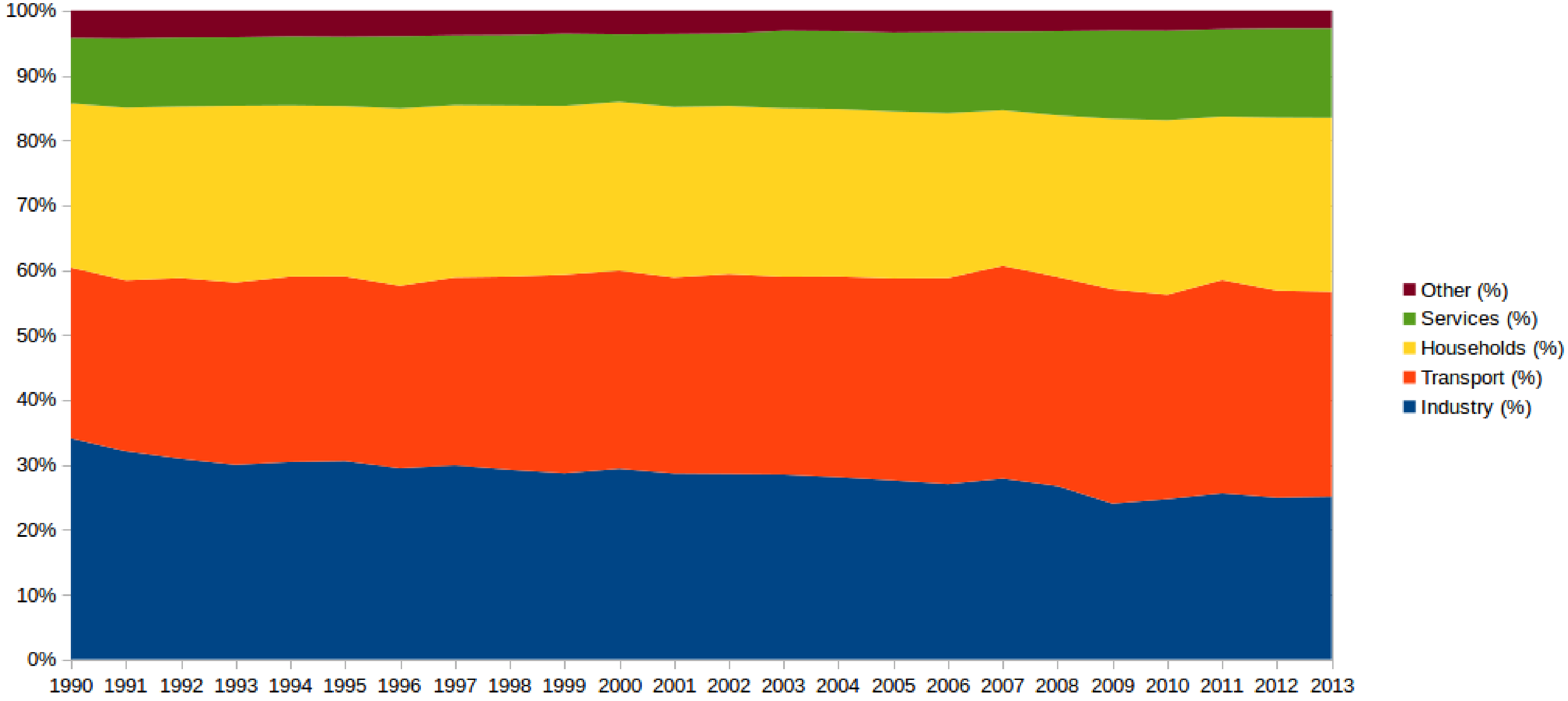


FIGURE 1.1: Energy Used By Sector (European Energy Agency, 2015)

Although a powerful building (re-)design decision support tool, building performance simulations can be subject to limitations. Studies have found that buildings may use twice as much energy as predicted at design stage. Bordass et al. (2001) studied 16 buildings which were predicted to have low energy use, however real world tests showed they were not low energy, but used as much as an average building. Under the stringent design and construction standards imposed by Passivhaus, Blight and Coley (2013) observed that on average a difference of 21% between simulated and actual energy use. It has been reported that for high energy buildings such as labs there is on average a factor of 2.5 difference between predicted and actual energy use (Turner and Frankel, 2008). Baker and Steemers (2003) found that building design parameters such as the plan, orientation and facade have been found to cause fluctuations in energy from 2.5 times. System parameters such boiler efficiency bring fluctuations to 5 times, leaving occupants to account for the other half of the 10 times fluctuation in energy.

Predicted building performance continues to deviate – sometimes considerably – from that which is observed post-build. The reasons are many and complex. These can be categorised as (Chapman et al., 2016): (type I) errors in modelling deterministic phenomena or indeed the neglect of these phenomena, (type II) errors in the inputs to these deterministic models, (type III) errors in modelling stochastic phenomena or indeed the neglect of these phenomena, (type IV) errors in the inputs to these stochastic models. Type I errors might include ignoring thermal storage in the modelling of heat diffusion, assuming thermophysical properties to be constant in the dynamic modelling of heat diffusion, or assuming that heat diffuses exclusively in one direction. Type II errors might relate to the characterisation of the bulk thermophysical properties of building materials, or assuming that multilayer constructions are perfectly homogenous and known; where as in reality workmanship is imperfect and unknown. Type III errors can be sub-categorised according to whether stochastic perturbations to heat flows in buildings are a) climatic, or b) human-behavioural in nature. Type IIIa errors might relate to wind pressures across the envelope and the corresponding impacts on convective heat transfers and infiltration, or the effects of cloud cover on transmitted shortwave irradiation. Whereas, type IIIb might relate to occupants' presence and associated metabolic heat gains, interactions with the envelope (e.g. windows and blinds), lights, appliances and systems. Finally, type IV errors relate to the empirical coefficients that are estimated for the models that are structured to address type III errors and their suitability to the particular context under consideration. Thus, and as with deterministic phenomena, with respect to stochastic phenomena a distinction is made between the model structure and its ability to capture the underlying stochastic phenomena in principle, and the calibration of this model to a particular circumstance.

Using suitable calibration techniques simulations can more accurately represent the aggregate behaviour of a real building, improving predictions to within 6% (Norford et al., 1994). Using 5 different bespoke models of occupancy, Menezes et al. (2012) managed to improve their predictive accuracy from a 70% under prediction to a 3% under prediction. The initial model used basic deterministic schedules with the final model using monitored data for lighting, appliances and catering equipment. There is clearly a performance gap that has not been covered by deterministic techniques, this gap can be reduced with accurate representation of real world occupant interactions. But whilst calibration from a real world building can capture the occupant interactions taking place, it is costly and not possible at the design stage.

This thesis focuses on the simulation of people within buildings and the effects of their energy related behaviours on building performance. Building performance simulation software such as EnergyPlus (Crawley et al., 2000) and ESP-r (Clarke, 2001) tend to have deterministic rules governing occupant behaviour. The deterministic rules and schedules that typically represent occupant interactions within building performance simulation software do not necessarily emulate the mean and certainly not the variance observed in real world behaviours and the performance impacts of them. There have been attempts at addressing this (Bourgeois et al., 2006) but not in a generic and portable way.

With the objective of addressing these limitations, and following the suggestions of Robinson et al. (2011), our approach is to use multi-agent stochastic simulation for modelling occupant behaviour; to combine stochastic models into a single package that can be used to support building and urban performance simulation using a range of software. The term agent has many meanings and has evolved over time. Agents are often described as objects within programs that control their actions based on their perceptions of the environment (Huhns and Singh, 1998). Multi-agent simulation is a tool that has been developed primarily in the social sciences to effectively model human interaction (Bonabeau, 2002, Zhang et al., 2011). Its use in the social sciences has typically been to study behaviours that emerge from bottom up interactions, allowing the creator to make judgements as to what has caused these emergent behaviours and whether they correspond with expectation from social theory. An agent should have the following properties; they should be autonomous, have social ability, perceive and react to the environment and be proactive with their choices (Wooldridge and Jennings, 1995). Each agent has rules and behaviours, making them excellent in principle at modelling group and individual interactions (Axtell, 2000). This work uses multiple agents to model occupants within buildings, using a combination of statistical techniques, interaction rules and machine learning.

1.1 Hypothesis

> An effective modelling strategy for simulating occupants in building performance simulation is multi-agent simulation. Simulating an occupants' energy related behaviours through agents that make use of stochastic models, theory driven rules, social interactions and machine learning techniques.

To develop and test this hypothesis the following aims are defined:

1. Represent the stochastic nature of occupants energy related behaviours, couple stochastic models of occupant interactions with building performance simulation.
2. Ensure diversity between occupants is accounted for.
3. Develop theory driven rules using belief-desire-intention techniques for the phenomena for which data are absent or insufficient.
4. Enable agents to communicate through a social interaction framework.
5. Integrate machine learning techniques to enable agents to learn how to react to the environment to improve their comfort.

This latter is particularly interesting for it may open up the possibility of representing stochastic behaviours of occupants in building simulation without the need complex data hungry models.

1.2 Methodological Approach

A multi-agent simulation framework has been developed, which has been named Nottingham Multi-Agent Stochastic Simulation (No-MASS). The No-MASS framework integrates existing stochastic models of occupant interaction into a tool that can be coupled with building or urban energy performance simulation tools. The coupling allows for simulated occupants to make changes within the simulated building environment and receive responses arising from the effects of the interactions which may stimulate future interactions. Sensitivity analysis is used to scrutinise the inputs of each stochastic model ensuring that the models have a significant impact on predicting the building performance results. Parameters that do not can be removed and the models simplified. Current stochastic models do not cover all of the energy related behaviours of occupants, therefore a belief-desire-intention (BDI) rule system is used to model other interactions, supplementing the data-driven stochastic models. For example switching off the light during sleep. Agents can have unique desires causing changes within the environment that may be in conflict with the desires of other agents. To solve this problem an agent social interaction model is developed to govern the interactions between agents. For more complex interactions where BDI rules would be difficult to design, agent machine learning techniques are used, allowing the agent to learn how to respond to different stimuli.

1.3 Research Structure

The research context, the hypothesis and structure of the work have been set out in Chapter 1. Chapter 2 begins with a review of occupant representations in building performance simulation and highlights the effects occupant interactions can have on the energy performance of the building. Also explored in Chapter 2, is how occupants are modelled in the domains of social sciences, economics and computer science, to observe how methods from these domains can be applied to building performance simulation. Chapter 3 discusses the proof of principle framework that was developed in response to needs derived from the literature and describes how the framework interfaces with building performance simulation. Stochastic models of interaction within the framework are then tested against a case study using a deterministic representation of occupants for comparison in Chapter 4. As the availability of stochastic models of interactions do not cover every aspect of an occupant's interaction with a building, a new approach is integrated into No-MASS for creating rule based interaction ideas based on an agent belief-desire-intention method is described and tested in Chapter 5. But often building occupants do not make changes to the environment without considering the effects on other people, so presented and tested in Chapter 6 is No-MASS's approach for handing negotiated social interactions. In Chapter 7 machine learning techniques within agents are demonstrated allowing agents to learn how to respond to the processes taking place within a building; thus agents can choose a strategy without the need for context specific rules. Finally, in 8 this thesis closes by identifying the principle contributions to knowledge arising from this work and a discussion for the scope of augmenting these contributions in the future. The thesis structure is summarised in Figure 1.2 below.

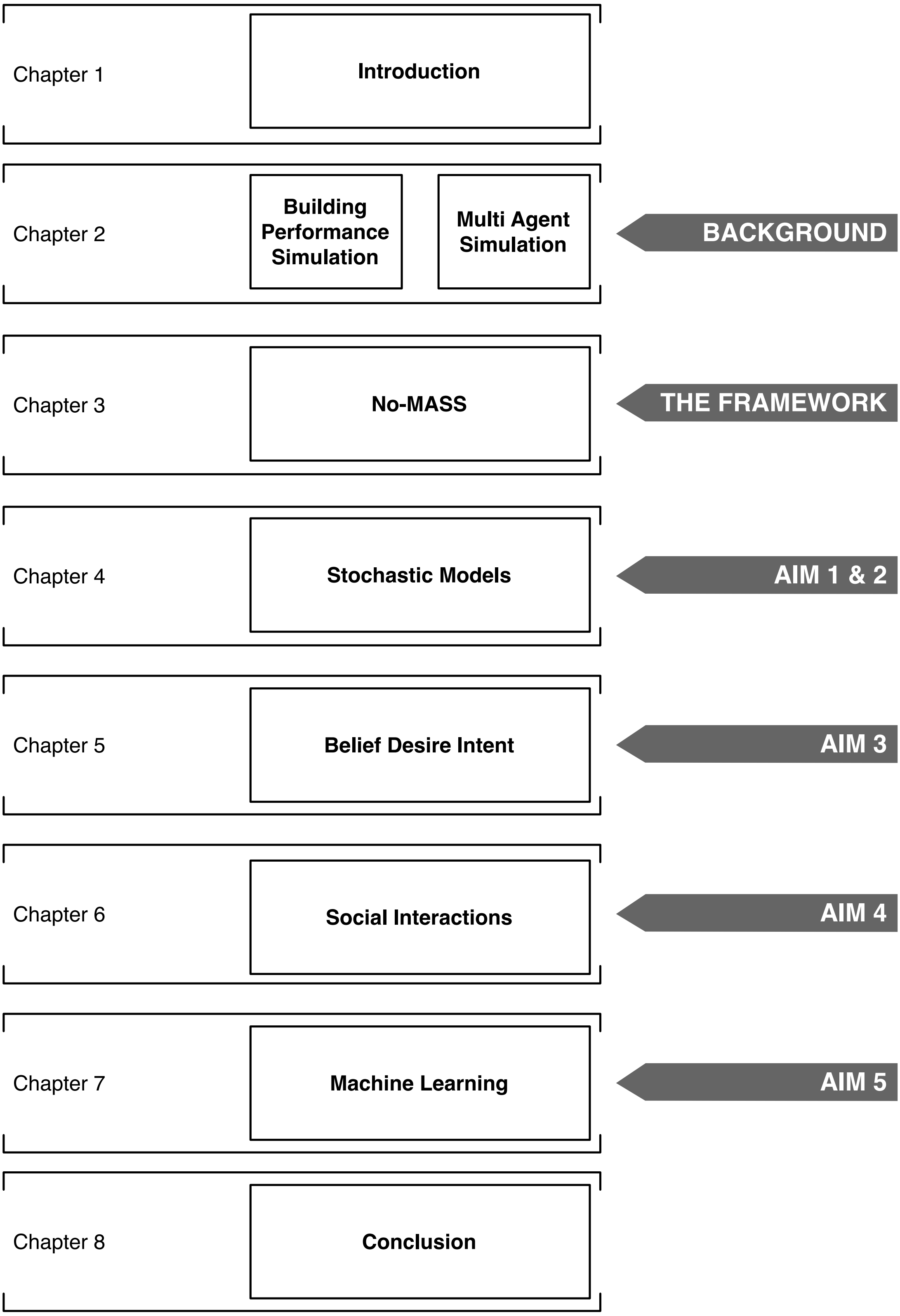


FIGURE 1.2: Thesis Structure

Chapter 2

Occupants in Building Performance Simulation

This thesis focuses on the development and application of a new simulation platform, conceived to systematically improve upon the representation of type IIIb and associated type IV errors in building simulation the were identified in Chapter 1. In this endeavour, work begins with a quick review of building simulation tools. Then the Empirical evidence maintaining that human behaviour impacts building's performance is presented. In conjunction with this evidence the corresponding progress made in the development of stochastic models of human behaviour to address these impacts is presented. Finally, the use of agents to model people in other fields is discussed.

2.1 Building Performance Simulation

There are many building performance simulation tools often with competing algorithms and methodologies, but they all aim to support the building designer in the testing of strategies to improve the buildings performance; to reduce energy use whilst maintaining comfort and ensuring that the design is robust.

In the United States there were initially two government funded building performance simulation tools that had very similar capabilities (Crawley et al., 2001). DOE-2 and BLAST both simulated the performance of buildings but had different features that made one better in some cases and the other better in others, for example BLAST was better at systems simulation and DOE-2 better at building simulation. They were developed over many years, making them costly to maintain and so the respective developer teams pooled their resources, taking the best parts of both tools and developed a single application named EnergyPlus. EnergyPlus can simulate surface/ air heat transfers and included a building systems manager to manage heating ventilations and air conditioning (HVAC), electrical equipment, air loops and photovoltaics (PV). Now a more comprehensive open source building systems simulation platform, it has been extensively tested, and is well documented. Made accessible through a range of both free and commercial graphical user interfaces (GUIs), most notably the DesginBuilder GUI, EnergyPlus is the most widely used of all available building performance simulation tools.

The ESP-r tool is another building performance simulation tool that simulates heat, moisture and air flow, electrical power flow, HVAC systems and lighting and renewable energy systems. It can also calculate performance appraisal measures such as life cycle analysis, indoor air quality, thermal comfort and environmental impact (Clarke, 2001). It has also been extended to simulate occupant interactions using sub hourly occupancy controls (Bourgeois et al., 2006) and computation fluid dynamics has been integrated. Although powerful, it is also complicated and unproductive to use by non-experts and, although open source, is relatively poorly documented, limiting its uptake.

TRNSYS is a modular transient simulation program initially developed for the study of solar energy systems. The main library consists of 150 models that support weather data processing, building performance simulation, the simulation of solar thermal processes and HVAC systems more generally. The library system makes it expandable, however unlike ESP-r and EnergyPlus it is neither free to use nor open source.

There a many other building simulation tools and because they often perform well in some areas and poorly in others comparisons between their ability to simulate the performance of buildings are difficult. Crawley et al. (2008) highlight that the terminology used throughout the different simulation tools make it difficult to compare their abilities, however they make a good attempt at doing so. These tools although useful, our work focuses on reducing the extent of the deviations between the simulated results and the real world performance of the building.

2.2 Building Performance Deviations

Seligman et al. (1978) concluded from a study of 28 identical houses there were deviations of 2 to 1 in energy demands which is thought to be primarily caused by behavioural diversity between occupants; as the houses were themselves identical. In a study of four identically constructed houses, Bahaj et al. (2007) found that in certain periods of the year energy demands varied by a factor of six due to occupancy. In their study of 22 identical residential houses in Germany, Maier et al. (2009) have identified a factor of 2 variation in heating demand. Meanwhile, post occupancy evaluations of UK EcoHomes have found that occupants' planned behaviour accounts for a variation of 51% in heating demand between dwellings (Gill et al., 2010). The variations due to occupancy in identical buildings highlight the difficulty in performing accurate simulations of energy performance. Design stage estimations can make predictions underperform by more than half. However, calibrating models so the simulations accurately represent the real building can improved predictions to within 6% (Norford et al., 1994). Altering the heating controls using an occupant comfort measure (predicted mean vote), Andersen et al. (2007) found that a simulated building could vary in energy use by 324% from a low consumption scenario to a high consumption scenario. Using 5 different bespoke models of occupancy, Menezes et al. (2012) managed to go from a 70% under prediction of actual energy to a 3% under prediction. The initial model used basic deterministic schedules with the final model using monitored occupant data for lighting, number and use of appliances and catering equipment.

The endeavour to reduce the performance gap by accounting for variability between occupants has led to models addressing the stochastic nature of occupants' behaviours, integrated within building performance simulation software. These range from hard coded integration in the case of lighting behaviour models in Reinhart's (2004) Lightswitch2002 algorithm to Haldi and Robinson's (2011) integration of occupant presence, window and blind models into CitySim. More recently, Vorger (2014) hard coded models of presence, activities, approximate use of heating systems, window and blind interaction within the building simulation software Pleiades+COMFIE. In line with empirical findings, Haldi and Robinson (2011) found with CitySim that variations in stochastic behaviour accounted for a factor of two variation in heating demand. Likewise, when comparing an ideal and worst case occupant scenario to demonstrate the range of influence occupants have, Roetzel et al. (2011) found that there was a factor of 2 difference in the simulated annual energy use for both heating and cooling. Bonte et al. (2014) found a similar factor of two variation arising from the integration of models of blinds, lights, windows, temperature setpoints and clothing into a building simulation tool.

Although these efforts have usefully demonstrated the potential impact of stochastic behaviours (and models of them) on building performance, and that these closely concur with empirical observations, they are software specific and lack generality. This criticism was partially addressed by Bourgeois et al. (2006) who developed a general solution for the integration of lighting, windows and blind models with ESP-r, called Sub-Hourly Occupant Control (SHOCC). But this approach is also software specific and does not support more complex features.

In thesis postulates that a comprehensive behavioural modelling framework should support:

> "The definition of archetypes and archetypal behaviours to account for diversity between occupants, social interactions between members of a population and corresponding implications for their behaviours, and behaviours that are conditional on others having already been exercised or indeed on proximity to the building envelope or system, with corresponding implications for interaction probability."
>
> (Chapman et al., 2016)

2.3 Stochastic Interactions of Occupants

As noted above the endeavour to reduce the performance gap by accounting for variability between occupants has led to the development of models addressing the stochastic nature of occupants' behaviours. Presence within a zone is a requirement for most occupant interactions within a building. Reinhart (2004) developed a simple stochastic occupancy model in the Lightswitch2002 algorithm, randomising arrival and departure times by 15 minutes from deterministic schedules. This moved the schedules away from repeated daily profiles but this is not based on empirical data. Page et al. (2008) used longitudinal occupant data to build an inhomogeneous Markov chain model to predict the likelihood of an occupant being present within a zone at a given time of day; although it does not predict movement between zones and long term absences are poorly represented (eg due to illness, business trips or vactions). More recently, Wang et al. (2011) built a Markov chain model for predicting time-dependent transitions between rooms. Chang and Hong (2013) developed curves of occupant presence based on automated lighting sensors. Finally, Feng et al. (2015) used these 3 different approaches in an algorithm where occupancy schedules are derived from Chang and Hong, the number of occupants in a room calculated using Page et al.'s model and transitions between rooms using Wang et al.'s model. As Page et al.'s model is the only model to be rigorously validated it remains to be seen how accurate the other models are. Vorger (2014) developed probability distributions from French survey data for predicting long term absences due to holidays and sickness. Although the data is heavily influenced by the French school holidays, the same methodology could be applied to other locations.

Once occupancy is known presence dependent behaviors can be predicted. The Lightswitch2002 algorithm developed by Reinhart (2004) predicts the probabilistic use of lighting at arrival, during presence and at departure, as a function of minimum indoor illuminance. The algorithm also predicts blind usage, but this is based on a deterministic solar threshold irradiance of $50W/m^2$. Haldi and Robinson developed two occupant interaction models, one for windows (Haldi and Robinson, 2009) and one for external shading devices (Haldi and Robinson, 2010). The shading model uses a hybrid of Markov chains to predict if they are lowered or raised and a Weibull distribution to predict the opening fraction at arrival or departure. The window opening model, also uses a hybrid approach. Transitions in opening status are modelled as a discrete time Markov process, whereas the duration that a window will remain open is modelled as a continuous time random process using a Weibull distribution. Both models performed well under validation.

It is also possible to predict the activities that an occupant maybe performing at a given time. Widén et al. (2009) used time use survey data describing occupants' activities to occupants activity dependent use of electrical appliances and their corresponding energy demand. Widén et al. (2012) expanded on this by building a Markov model to predict activities from the time use data; the outputs of which he uses in a separate model that predicts electrical demand from the activity performed. However, this model only predicts if an activity takes place not when it will start and appliances' dynamic usage behaviour are accounted for. Wilke (2013) created a time dependent Markov chain model that predicts the transition from one activity to another at a given time, coupled with a model to predict the duration for which this activity survives, from which the load profile is predicted based on that activity. Appliances are assigned based on household socio-demographic characteristics using a logistical regression based on an appliance use survey. This model performs well under validation, but it is complex and computationally expensive. In the quest for a parsimonious model (one which yields acceptable accuracy for the least computational cost and complexity), Jaboob (2015) tested a range of modelling strategies, again using time use survey data, selecting time dependent Markov model of activities.

These models cover a rather comprehensive range of energy related behaviours but there remain gaps to be filled. Energy related behaviours relating to curtains interactions, window opening with regard to air quality and noise pollution, etc. To capitalise on the value of these models and to facilitate the straight forward integration of future models they should be integrated with a robust multi-agent simulation framework which is itself integrated with building performance simulation software using an open co-simulation framework. Achieving this in a coherent and generalised way is important if these models are to gain widespread use in building performance simulation, which continues to use outdated deterministic rules and schedules to support building design.

2.4 Multi-Agent Simulation

Multi-agent simulation has been used to represent people in a variety of areas. Epstein and Axtell's (1996) SugarScape simulates a simple society where inhabitants need to eat resources in an artificial world to survive. Each agent moves around a plane looking for and consuming sugar based on predetermined rules. These models demonstrated how societies can develop over time, congregating around areas of resource. The model was expanded to include sugar and spice as resources that could be traded, from which the authors showed that the prices of both converged to an equilibrium, as economic theory would predict. Meanwhile Axelrod (1987) tested agents, employed in combination with a genetic algorithm, against a prisoner's dilemma scenario (should you defect against or cooperate with your accomplice who has the same options). Later, Axelrod and Hamilton (1984) found that 95% of all populations evolved towards the optimal *tit-for-tat* strategy, demonstrating the effectiveness of agents in exploring alternative decisions. Traffic flow within cities has also been modelled with agents, allowing traffic planners to make informed decisions to improve congestion (Balmer et al., 2006, Nagel et al., 1999). Each agent in these scenarios occupies a vehicle, with their own goals and decisions to make. Another example is city wide disaster scenarios, evaluating traffic flow during an evacuation and the effects of different city road layouts (Ring, Grid, etc.) on evacuation time (Chen and Zhan, 2006). Finally, Siebers and Aickelin (2011) models the effects of changes in employee empowerment on customer satisfaction in shops.

Agent cognition is often based on the belief-desire-intention (BDI) system formulated by Rao and Georgeff (1995). In this system an agent has beliefs about the current state of the environment and related desires about what it wants to achieve, they commit to an intent which is the desire they want to achieve. A plan, made up of a set of actions, is chosen to realise there intent. This methodology has been used in the context of building performance simulation, where agents obtain a belief about the state of the current environment from a building performance simulation tool. In this vein, Andrews et al. (2011) combine the Radiance ray tracing tool with agents, to simulate interactions with lighting and shading. Agents build up their understanding of the environment with data from Radiance (room illuminance level) then, based on their assigned personal characteristics, develop plans of actions and act on the plan that maximizes their utility (satisfaction). To simulate diversity each agent was given an archetype of either green activist, good citizen, healthy consumer or traditional consumer. These archetypes were developed from questionnaires administered to building occupants, to better understand their preferred lighting levels and the energy use they are comfortable with. The agents' assigned archetype would effect their desire thus altering their intended method of interaction. Kashif (2014) uses a similar approach to predict the use of fridge-freezers, where an occupant would first perceive their hunger, second conceive a desire based on social normals, household rules and culture. They then finally perform an action, remove food from the freezer, increasing the electrical load on the fridge and then cook. These approaches attempt to encapsulate the human decision making processes involved in each activity. But they can lead to very complex and unwieldy models that have a weak empirical basis. Paradoxically this is also their strength, that with relatively little data reasonably reliable aggregate behaviours can be simulated.

Come what may it is important that empirical studies results have a high degree of certainty and that agents' rules and behaviours are grounded with data based on reality (Gimblett, 2002). In recent years there has been a move from models based on social theoretical rules and behaviours, to those derived from observation (Janssen and Ostrom, 2006). By using previously developed stochastic models of occupant behaviour it is possible to predict agent behaviours based on solid empirical evidence. As an alternative to the BDI approach, Liao et al. (2012) use room occupancy data to inform their agents' behaviours for the prediction of presence across multiple rooms and occupants. However, Liao et al. note that for larger numbers of agents, it is often difficult to obtain high quality data from which to infer reliable rules. The model was also developed for a very specific use case of university buildings where students and professors have very different schedules. It has yet to be seen if this method can be applied to other building uses. Furthermore, these agents do not have the properties (social ability, reactivity and pro-activness) required by Wooldridge and Jennings (1995) to be formally designated as intelligent agents. More recently, Langevin et al. (2014) use data taken from a one year study of an air conditioned office to develop rules that allow an agent to proactively restore thermal comfort based on thermal sensation. These rules allow an agent to make changes to clothing, to operate windows/fans/heaters and to change set point temperatures. These values are then parsed to a dynamic building simulation program using the Building Control Virtual Test Bed (BCVTB). The Langevin agents are only as good as the realism given to the agent attributes specified for the model (such as clothing levels), the corresponding comfort model and the limited thermal inhomogeneity in the simulated indoor environment. With thermal discomfort as the trigger for a specified behaviour (based on the stationary ISO 7730 model), an error in the (dis)comfort prediction will inevitably undermine the faithfulness of the predicted interactions and their consequences. An improvement would be to more explicitly represent the dynamic relationship between environmental stimuli and interactions for prediction.

In contrast with previous strategies to model occupants' behaviour that have tended to be based either on data-driven stochastic models or BDI rules, integrated with a specific dynamic building simulation program, this work proposes a more general framework. The virtues of both modelling approaches can be combined (data-driven where data is abundant and BDI where it is not) and interfaced with a range of building simulation programs; whether at the building or the urban scale. To this end and in the first instance, existing models of occupants' activities, metabolic heat gains, use of windows, lights and shading devices are integrated with a bespoke platform called Nottingham Multi-Agent Stochastic Simulation (No-MASS). These agents' interactions are parsed to a building simulation program, which in turn parses environmental parameters to No-MASS, to impact on future behaviours. This provides a generic way to integrate existing and future stochastic models, speeding up time from model development to integration and thus availability of missing models for use by the broader simulation community, increasing their usefulness. The remainder of this thesis is dedicated to describing this new framework and evolutions of it, from population generation, through parameter assignment to simulation (pre and runtime), and to demonstrating its utility.

Chapter 3

No-MASS

There remain many gaps in our ability to model occupants' stochastic behaviours (such as their interactions with heating and cooling systems, hot water devices, curtains, their use of windows to evacuate pollutants, etc). However there is the availability of a sufficient core set of rigorously formulated and validated models with which to evaluate the proof of principle of No-MASS as a platform for addressing type IIIb and IV errors and thus of evaluating the robustness of buildings' performance. This chapter introduces No-MASS, then discuses how it is used in the following chapters, the models included and how it is implemented. Also described is how it can be connected to other simulation tools through a generic interface, such as EnergyPlus, and finally explain its inclusion in a leading commercial building simulation tool, called DesignBuilder.

3.1 Concept

The No-MASS simulation tool was developed in two phases. The first phase (alpha) is deployed in Chapter 4 to test the effects of stochastic models on buildings, while the second phase (beta) is used in Chapters 5 to 7.

Alpha Version

The initial family of models integrated with No-MASS includes models of occupants' activities (Jaboob, 2015), presence and corresponding metabolic heat gains (Page et al., 2008), window interactions (Haldi and Robinson, 2009), shading interactions (Haldi and Robinson, 2010) and lighting interactions (Reinhart, 2004). These models were chosen as they have been empirically verified and were readily available. Simulations with No-MASS follow the process outlined in the conceptual flow-diagram in Figure 3.1. A number of pre-processes are first performed. Initially an agent population is created, with each agent assigned a profile that influences their subsequent behaviours. For example, socio-demographic characteristics influence the applicable probabilities with which time dependent activities will be predicted; likewise, the probability of being present at a given time step and the corresponding location. This maybe the agents' office when occupying (and sleeping in) a non-residential building or their bedroom for sleeping when occupying a residential building. These characteristics and the subsequent modelling of activities is constrained in the present prototype version of No-MASS to cases of adults that act independently but whose activity choices may be influenced by the composition of the household. For example, couples may have different activity profiles than single adults living alone and elderly agents may perform different actions at different times compared with younger adults: there is a greater chance that an elderly retired occupant will be present and cook during the day, whereas a younger occupant will more likely be out at work. The profiles are defined through a simulation input file, that contains the characteristics of each agent.

Once the agents are defined a pre-process of those models that do not utilise transient environmental inputs, such as models that depend on time only, is run. A distinction between residential and non-residential buildings is made. For residential buildings the activity that an agent will perform and the corresponding location at each time step is processed; whereas in non-residential buildings, a separate presence model is used to calculate whether an agent is present in a given zone at each time step. Once these pre-processes are complete EnergyPlus is called to simulate the building's energy flows for the first time step. At the end of each subsequent time step, No-MASS is called by EnergyPlus. Environmental conditions are parsed from EnergyPlus to No-MASS, which then uses these to predict our agents' behaviours. Each agent is called independently and at random. For residential buildings the pre-processed activity and location for the present timestep is retrieved and used to calculate the metabolic gains for that agent and location. In non-residential buildings only the pre-processed presence is retrieved to calculate metabolic gains. Next agents' interactions with shading devices, windows and lighting are predicted. The outputs from all models are then parsed back to EnergyPlus, which resolves the energy consequences of these interactions when simulating the building's energy flows during the next time step, so that there is no within timestep iteration. This process continues until the end of the simulation period.

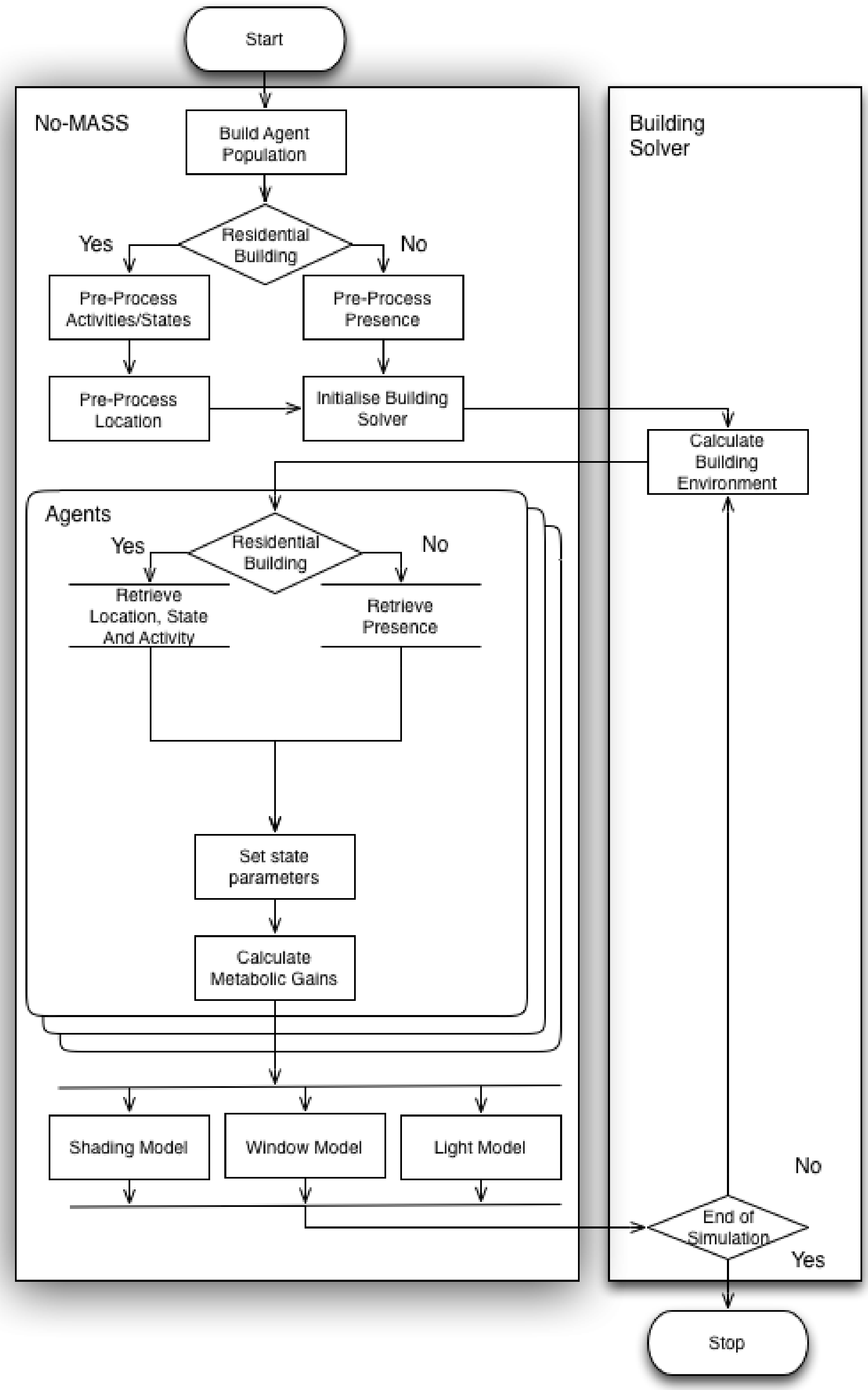


FIGURE 3.1: No-MASS Flow Diagram of alpha version

Beta Version

These core models integrated with the core version provide us with a rigorous and useful starting point, regarding the modelling of occupants' stochastic behaviours that impact on the buildings performance. However not all such stochastic phenomena are covered. In response Figure 3.2 gives an overview of an extension of the alpha version: a beta version of No-MASS with two new models included, the first a social interaction framework and the second a heating setpoint model.

As noted earlier, No-MASS builds an agent population, assigning a profile to each member dependent on the input parameters supplied in the No-MASS configuration file. These profiles are made up of the stochastic model coefficients for windows and shading, or in the case of the presence and activity model the probabilities of being in a state at a given point in time. However now an improved activity model is implemented, this allows occupant profiles to be defined through social demographical characteristics, these alter when and where the agent will be present in a household (Jaboob, 2015). These demographical characteristics consist of the agents age range, marital status, whether they are employed, their family configuration, education level and gender. A retired person will spend more time at home during the day then a professional below the age of 60, a student will sleep and wake later.

The use of shades, windows, lights and heating system setpoints are modelled via a social interaction framework; which allows communication with other occupants, creating a conflict resolution mechanism and co-operative strategy. A voting system is used to allow agents to voice how they wish to interact with the environment, if the other agents in the zone agree an interaction takes place. Unlike the alpha version the social interaction model allows the stochastic models for window, shading and lighting have been moved inside the agents themselves. A benefit of this is that each agent can be given a different set of coefficients for each model meaning more diversity between agents. Belief-Desire-Intention (BDI) rules are applied, allowing rules to be integrated where data-driven stochastic models are not available. Outputs from the building solver and other models such as activities are used as inputs to deterministic rules, thus allowing No-MASS to be extended beyond the stochastic models. Finally, an agent reinforcement learning model has been introduced that can model either heating system setpoints or window interactions. Agents learn how to interact in a way that keeps discomfort to a minimum based on the temporal spacial restraints of the environment.

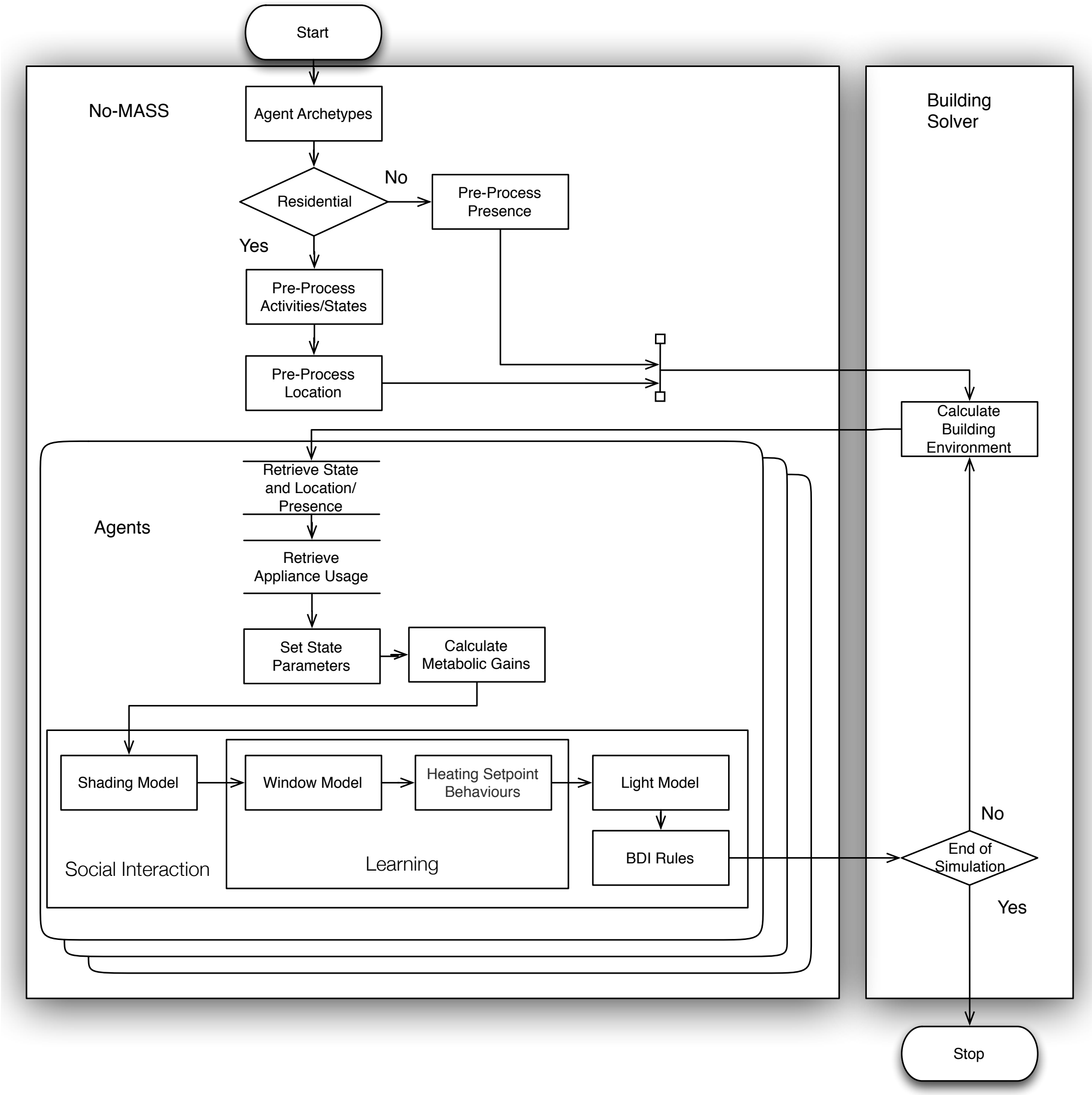


FIGURE 3.2: No-MASS Beta Version Flow Diagram

3.2 Implementation

No-MASS was built from the ground up, having tested a number of current multi-agent simulation development platforms, including Repast Simphony and Anylogic. Although these platforms are very useful they come with significant overheads, typically requiring a large library of graphical user interface routines to be imported, making it more difficult to package a No-MASS equivalent with simulation tools such as EnergyPlus and the DesignBuilder interface to it. C++ was chosen as the development language as it is simple to integrate with EnergyPlus (Crawley et al., 2001), our chosen building simulation tool is also developed in C++. Using the same language allows for easy communication between the two tools. EnergyPlus developed by the US Department of Energy, is well tested, well documented and open source; allowing us to readily understand how to connect to it. There are also two interfaces that allow other tools to interact with it, without altering the EnergyPlus source code. The first is through the building controls virtual test bed (BCVTB) and the second is through the Functional Mockup Interface (FMI) (Nouidui et al., 2013). The No-MASS platform connects to EnergyPlus using FMI, which is an open standard so that No-MASS could in principle be integrated with any other FMI compliant simulation tool. This is chosen over the BCVTB as it allows direct communication through C++ double precision arrays using predefined calling points, whereas BCVTB requires calls over sockets adding complexity and slowing the processing time. The calling points are well documented, with No-MASS only using the *initialise* function, the *receive an array of doubles* function for the environmental variables and the *send an array of doubles* function for the occupant interactions. The array of values that No-MASS receives at each time step is defined in the XML file ModelDescription.xml. At the beginning of the time step the following environmental variables are received: horizontal sky illuminance, rain status, outdoor air dry-bulb temperature, zone air temperature, zone humidity, indoor radiant temperature and indoor illuminance. Returned to EnergyPlus are the number of occupants in a zone, their metabolic gains, appliance gains, the window status, the blind shading fraction, the lighting status and the heating setpoint. Due to the window, shading and location/presence models used within No-MASS a sub-hourly timestep is recommended (ie. 5 minutes), as longer timesteps may overestimate the implication of the occupant interactions. For example the response time to an agent opening a window may be short with the room cooling in just a few minutes. An agent can only respond at the next time step, if the timesteps are not sufficiently short in length, the open window may over cool the room. Figure 3.3 shows the effects of different simulation timesteps on the performance of the simulation on a non-residential monozone building. The timestep choice will have a differing impact depending on the complexity of the building and the occupants simulated.

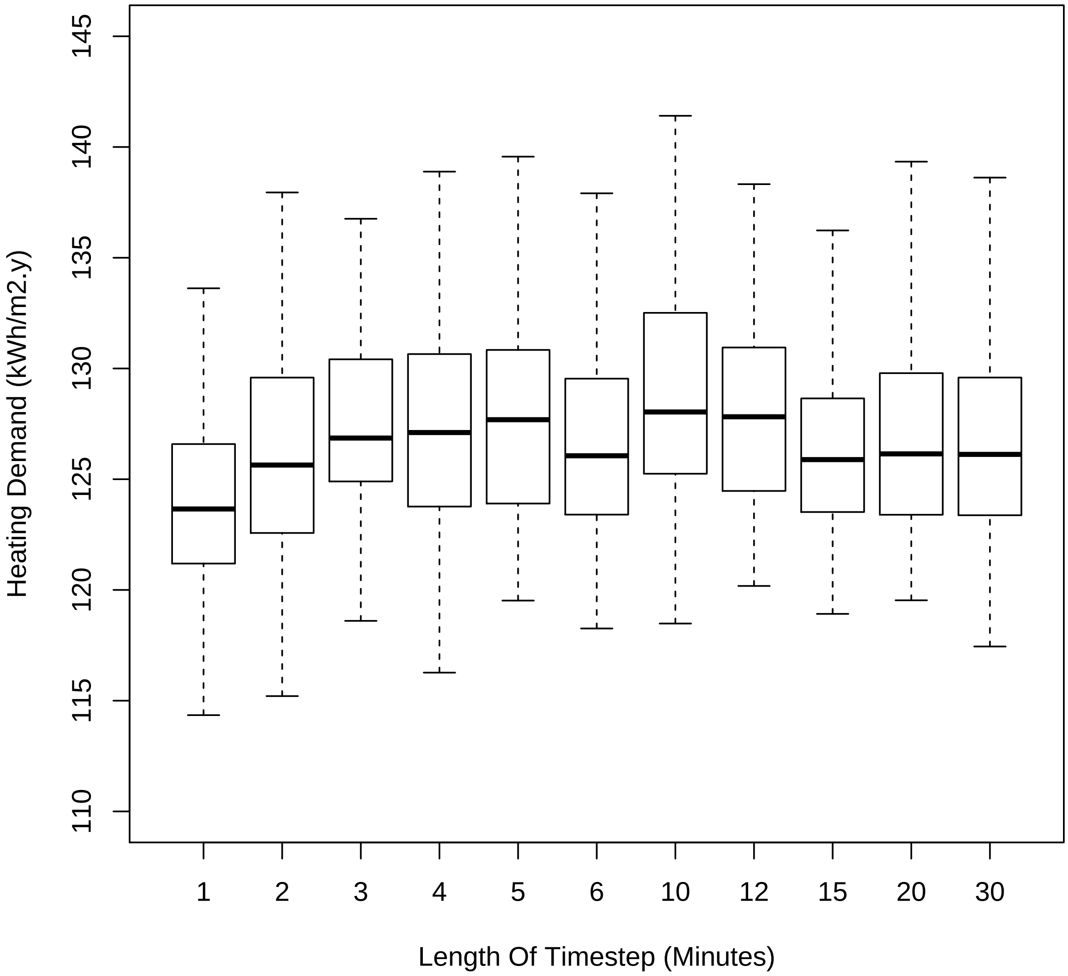


FIGURE 3.3: Time step sensitivity analysis, using a single zone non-residential building, highlighting the effects of different time steps when using No-MASS, 100 replicates each

No-MASS has been developed as both a Linux shared object and as a Microsoft Windows dynamic link library. Figure 3.4 shows how the system connects to EnergyPlus. In the same way that EnergyPlus reads in the building configuration and weather data from the IDF file and the EPW file, No-MASS reads in data from an XML file called NoMass-Config.xml. This file contains information about the occupants that is used to build the agent population, and the subsequent processing of an agent activity profile (a series of parameters defining the the socio-demographic characteristics of the agent, ie. gender, age, income level, etc) that is used to calculate the probability of an activity taking place at each timestep, as well as the bedroom or office that this agent will assigned to during the corresponding activity (eg. the bedroom while sleeping). It also defines the window and shading model coefficients for each model, allowing for diversity between occupants and models to be represented as needed.

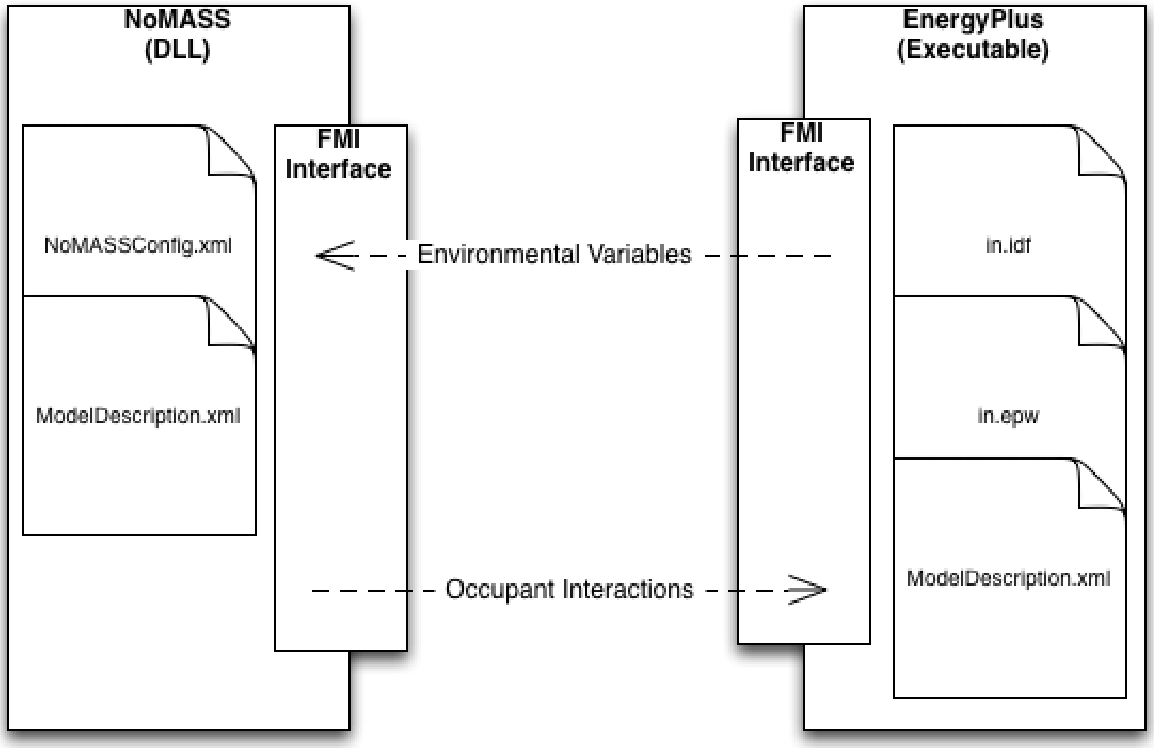


FIGURE 3.4: EnergyPlus, FMU Data Flow Diagram

3.3 Integration

3.3.1 EnergyPlus

EnergyPlus is typical in its use of deterministic rules that represent occupant interactions within buildings. The types of interactions that can be set are limited to single event driven procedures, for example if x happens at time t do y. Heating setpoints in Energyplus are determined by a combination of temporal (the time and date) and comfort related eg. air temperature, operative temperature, thermal comfort (Fanger's predicted mean vote) and humidity parameters. An example of a schedule defining the heating setpoints would be:

$$
\begin{aligned}
&Schedule: Compact,\\
&Kitchen\ Heating\ SP\ Sch,\\
&Temperature,\\
&Through:\ 31\ Dec,\\
&For: Weekdays,\\
&Until:\ 05:00,\ 12,\\
&Until:\ 10:00,\ 18,\\
&Until:\ 17:00,\ 12,\\
&Until:\ 23:00,\ 18,\\
&Until:\ 24:00,\ 12;
\end{aligned}
$$

This describes the heating setpoints for the kitchen zone for weekdays throughout the year heating the zone to $18°C$ between 5am to 10am and 5pm to 11pm, and to $12°C$ otherwise. For window openings the interactions are set in a similar schedule format, albeit with more parameters to choose from (including air temperature, enthalpy, constant, thermal comfort (ASHRAE 55 adaptive and CEN 15251 adaptive), adjacent temperature and adjacent enthalpy). Shading interactions are enabled through a schedule then set through a single setpoint value, either a temperature for heating or cooling, or solar irradiance incident either at the window or on the horizontal plane. Lights are set though a fraction schedule where the values are a fraction of the design level to allow for dimming. Presence within a zone is also set through schedule of the fraction of the total number of occupants that can occupy a zone.

Functional Mockup Interface (FMI)

No-MASS couples with EnergyPlus through the open FMI standard, a generic programming interface that allows other programs to interface to EnergyPlus to extend its functionality. It provides a description of how information can be passed from EnergyPlus at runtime to a functional mockup unit (FMU) which can then make calculations, the results of which are parsed back to EnergyPlus. These results are then used by EnergyPlus for future calculations.

An FMU consists of:

- The FMU Library (Windows Dynamic Link Library, Linux Shared Object)
- A Model Description File

- Any Other Configuration Files

The library file contains the programmed equations, in a programming language that supports C interfaces. When the No-MASS source code is compiled it is in the form of the library file. No-MASS works as both a Windows dynamic link library or a Linux Shared object. It can also be compiled on Macintosh computers; however EnergyPlus does not support FMI on Macintosh computers so no testing has been conducted on the MacOS.

FMI works using a simple data exchange methodology (Figure 3.5). At each timestep EnergyPlus sends a set of predefined variables ($\mathbf{x}$) to the FMU (e.g. Zone Mean Air Temperature and/or Site Rain status). The FMU performs a calculation on the values and returns a set of results ($\mathbf{y}$) to EnergyPlus (e.g. occupant location and/or shade status). These return values can be made to overwrite EnergyPlus values for the next timestep.

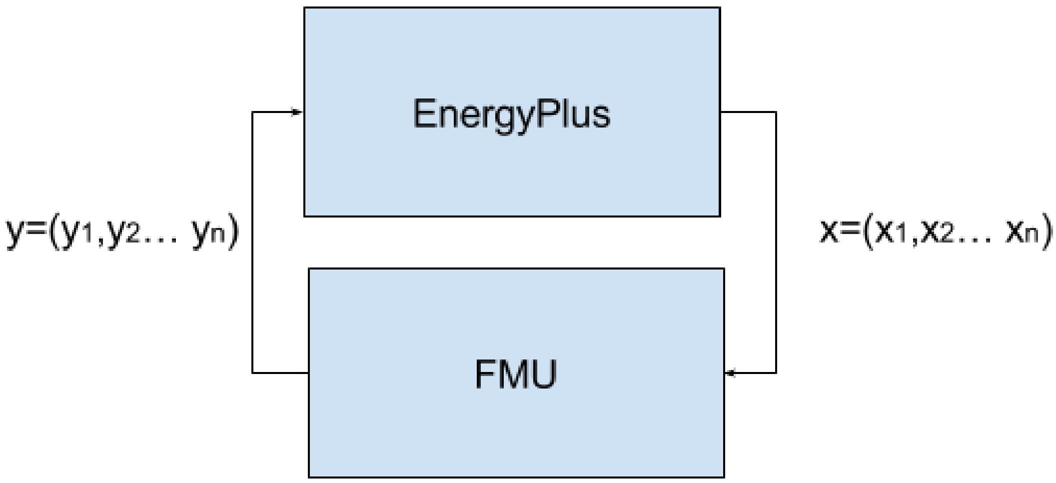


FIGURE 3.5: No-MASS Data Flow Diagram

Within EnergyPlus there are three types of external interface that can be written to:

$$ExternalInterface : FunctionalMockupUnitImport : To : Variable$$
$$ExternalInterface : FunctionalMockupUnitImport : To : Schedule$$
$$ExternalInterface : FunctionalMockupUnitImport : To : Actuator$$

No-MASS uses the schedules and actuators to write values to EneryPlus, overwriting the predefined deterministic rules, at each timestep. The flow of processing from EnergyPlus to the FMU is given in Figure 3.6.

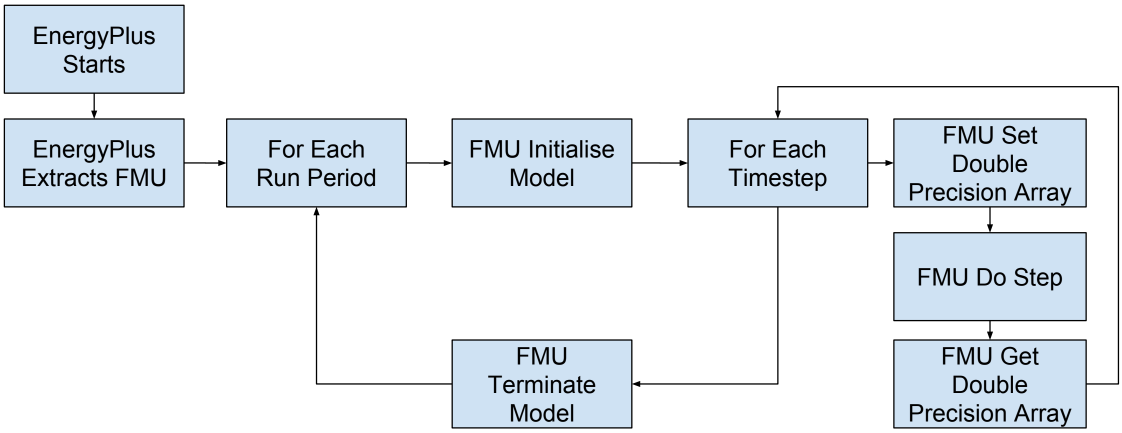


FIGURE 3.6: FMI Interface Flow Diagram

EnergyPlus starts, reads in its configuration file, and determines that it needs to run a simulation for use with an FMU. EnergyPlus then searches for the FMU, in this case No-MASS, and extracts the information it needs to communicate with the FMU. For each run period the FMU is initialised by calling the initialise method of the FMU. Then at each timestep EnergyPlus calls the FMU, setting the variables in the FMU by passing an array of double precision values defined in the model description XML file. Then EnergyPlus calls the FMU do step method where the FMU performs any calculations (see Figure 3.1 for the processes inside No-MASS). To receive the return values EnergyPlus calls the get double precision array method in the FMU and receives an array of double precision values back, as specified in the model description XML file. Once all timesteps are completed the FMU is terminated and the simulation is ended.

DesignBuilder

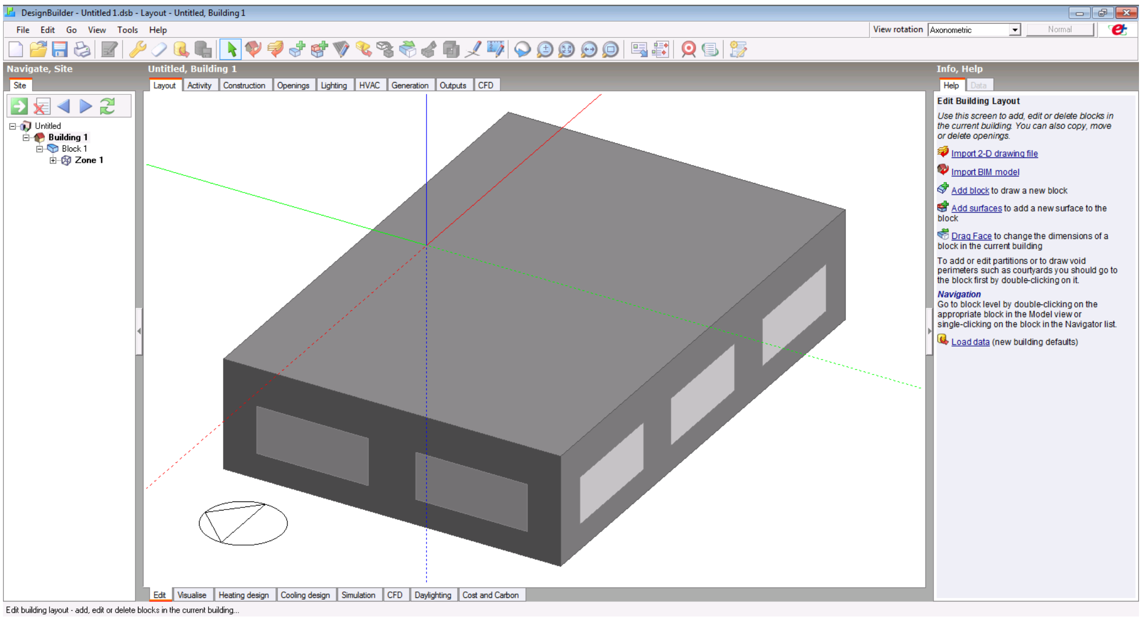


FIGURE 3.7: Main DesignBuilder interface

.

DesignBuilder provides access to EnerygPlus through an easy-to-use interface that serves more than 3500 customers in 80 different countries. Their main market sectors are services engineers, building simulation experts, architects (technical), energy assessors (UK), LEED, BREEAM, Green Star assessors, university R&D and teaching. To enable researchers and early adopters to use and test No-MASS, GUI components controlling No-MASS were integrated with the DesignBuilder GUI (See Appendix A for screen shots of the tool in use). In this way No-MASS can now in principle be used by any building designer without the need to know the complexities of EnergyPlus, FMI, and No-MASS.

The steps in developing and simulating a No-MASS compliant model are as follows: Using the DesignBuilder GUI a building is designed. A user can then enable the No-MASS occupants in the simulation configuration dialogue. Once enabled on the main occupant activity screen there are a choice of occupant templates. There are predefined sets of occupants for offices or residential buildings (eg. family with children, working professional, students, retired, etc.) that can be chosen. Each set of occupants can be further edited allowing different combinations of presence, activities, shades and windows profiles to be defined. The current limitations of the GUI are that it does not allow for agent learning and as of yet BDI rules have not been added.

This tool allows for quick and easy setup of complex building types with differing stochastic occupants. A designer can now test the building against a range of different use cases, with occupants simulated as agents and interactions based on real world empirical, validated models and data.

Chapter 4

Data Driven Stochastic Models, Coupled with Building Performance Simulation

No-MASS couples stochastic models of occupant interaction with building simulation, the process of which was described in the previous chapter. This chapter describes how these models work and the effects they have on building performance. The models are also analysed in terms of their inputs to ensure that each has a significant effect on building performance.

4.1 Stochastic Models

Activity model

The activity model (Jaboob, 2015) predicts the time-dependent probability that one of a set of ten activities will be performed in the home. These activities include sleeping, passive, audio/visual, IT, cooking, cleaning, metabolic, washing appliance use, personal washing and absence from the building. The activities were derived from the UK Time Use Survey (TUS) data 2000-2001 (Gershuny and Fisher, 2013). Jaboob's work was to better predict activity-dependent use of energy in homes. The data was filtered into 10 activities, those within the home aggregated into a set of nine meta activities, with activities taking place outside of the home aggregated into a single activity, out of the home. This formulation of 10 energy-related activities reduced the risk of over-fitting in their model. Most of these activity aggregates are self-explanatory, however some are less so. The activity passive refers to occasions when an occupant is awake, but not physically active; whereas metabolic refers to when an occupant is awake and physically active. After implementing and testing a number of different prediction methodologies Jaboob (2015) found that modelling the activities as a time-dependent Bernoulli process using multinomial logistic regression was the most parsimonious (good predictions and least computationally expensive) approach.

$$P(x,t) = \frac{exp(A_j(x,t))}{\Sigma_{j=1}^{N} exp(A_j(x,t))}, j = 1, \dots, N \tag{4.1.1}$$

and

$$A_j(x) = \alpha_j + \sum_{k=1}^{n} \beta_{jk} x_{jk} \tag{4.1.2}$$

Where, t is time, N corresponds to the total number of activities, n the number of predictors, and x the predictors describing the observation, each associated with a slope coefficient $\beta_j k$, and α_j is the intercept. As the probabilities are only dependent on time it is possible to generate a 10 by 24 matrix giving the probability of performing each activity at a given hour; though the corresponding model can also be re-called within the hour for sub-hourly simulation timesteps. Models can also be estimated for subpopulations of the time use survey dataset from which they are derived, to give probabilities that depend for example on age, employment status, season or day of the week. No-MASS accounts for diversity through a choice of sub-models, for the activity model a range of occupants defined through social demographical characteristics can be chosen, these alter when the where the agent will be present in a household. A retired person will spend more time at home during the day then a professional below the age of 60. The sub-model used for each agent is assigned when the agent population is generated. This model is pre-processed, assigning a state to each timestep within the simulation. This is achieved by drawing a random number for each timestep for each agent. Where that number falls within the range of probabilities for that hour, the corresponding activity is assigned to the relevant agent. These are then stored for retrieval at run time.

Note that this process is only considered for residential buildings and does not apply to the simulation of non-residential buildings, for which the corresponding time use survey data is not available. Jaboob (2015) also provides predictions for secondary activities, for example audio/ visual while cooking. However these have not yet been integrated; currently No-MASS infers location, clothing level and metabolic rate from the activity model, thus there would be no energy related consequences of the second action.

Agent States

During a simulation our (residential) agents are assigned one of ten activity-dependent states. No-MASS uses a state machine, which is a way of defining programmatically the states an agent may be in and the possible transitions from each state to the next. Dependent on the state, values of certain parameters are modified; in this thesis the values in Table 4.1 are used for all agents, although they can be changed as required depending on simulated building configuration and occupant type. These values are taken from ISO 7730 (ISO, 2005). In the case of the more constrained environments of non-residential buildings a metabolic rate is set to 116 and clothing level (clo) to 1.

State	Location	Clothing level (clo)	Metabolic rate (W/m2)
Sleeping	Bedroom	2.55	46
Passive	Living Room	0.7	58
Audio/Visual	Living Room	0.7	70
IT	Office	0.7	116
Cooking	Kitchen	0.7	116
Cleaning	Kitchen	0.7	116
Washing self	Bathroom	0.3	116
Washing appliance	Kitchen	0.7	116
Metabolic	Living Room	0.7	93
Absent	NA	NA	NA

Table 4.1: Agent states with corresponding locations and values

Presence and Location

Within No-MASS there are two methods for calculating presence within a building, the choice of which depends on the type of building. For residential buildings, presence (or rather absence) is predicted directly by the activity model (as noted above). Furthermore, based on the activity being performed [or the agent's state], a location can be inferred. For example, if the agent is in the sleeping state it can be assumed that they are in their bedroom. But this may not always hold true. For example, if an agent is predicted to sleep during the day (and retired folk do this with relatively higher probability) they may do so in the living room. Thus, in the future there may be need for archetype-dependent assignment probabilities to account for such eventualities. To allocate agents to a zone within EnergyPlus an external schedule of occupancy for each zone is defined in the EnergyPlus configuration file. EnergyPlus then assumes that a value for each schedule will be received by its external interface at each time step. For the simulation of non-residential buildings a presence model (Page et al., 2008) predicts when an occupant is present within their office, based on an inhomogeneous Markov chain, using a mobility parameter μ and a time-dependent profile of the probability of presence P(t) as input. Since this model uses no environmental parameters, it may be run as a pre-process, generating a sequence of presences and absences for each agent. These are deduced by calculating the transition probability at each time step, either from absent to present:

$$T_{01}(t) = \frac{\mu - 1}{\mu + 1} \cdot P(t) + P(t+1) \tag{4.1.3}$$

Or present to present:

$$T_{11} = \frac{P(t) - 1}{p(t)} \cdot \left[\frac{\mu - 1}{\mu + 1} \cdot P(t) + P(t+1) \right] + \frac{P(t+1)}{p(t)} \tag{4.1.4}$$

Where the mobility parameter is by default held constant at an empirically determined value of $\mu = 0.11$ (though this should in the future be assigned from a distribution that depends on the type of workplace; as in the case for p(t)). The other two transitions, present to absent and absent to absent are simply $T_{10} = 1 - T_{11}$ and $T_{00} = 1 - T_{01}$. Long term absences due to illnesses, vacations or work related business trips are not presently stochastically predicted. Although the TUS data (Gershuny and Fisher, 2013) has data on people while at work, at the current point in time there no empirically validated model using this data for presence in a non-residential building. The TUS data is a snapshot in time and does not include longitudinal sickness, vacation or work trip data over the course of a year.

Metabolic gains

Metabolic gains are calculated using Fanger's PMV model, as described in ISO 7730 (ISO, 2005), based on the standard physical (air temperature, radiant temperature, relative air velocity and relative humidity) and personal (clothing level and metabolic rate) parameters. With the exception of an assumed relative air velocity of 0.1 m/s, the physical parameters are supplied by EnergyPlus; whereas the state-dependent personal parameters are as defined in table 4.1 (with external work taken to be 0W). As EnergyPlus takes a single metabolic rate for all agents within a zone, an average for all agents present in the zone is calculated. This is set within EnergyPlus through the zone activity schedule, which multiplies this average gain by the number of present occupants to determine the total metabolic heat gain for all occupants.

Window Actions

Haldi and Robinson (2009) compared twelve window opening models and found the most effective at predicting window openings in offices to be a hybrid model, this was later validated against residential buildings Schweiker et al. (2012). This hybrid model first predicts transitions in opening status using a presence-dependent Markov chain and then in the cases of transitions to the open state, predicts the duration for which the window stays open using a Weibull distribution. The probability of transition in window opening state, from i to $j(i, j = 0, 1)$ is calculated using a logistic regression model of the form:

$$P_{ij}(x_1, \ldots, x_n) = \frac{exp(\alpha + \Sigma_{k=1}^{n} \beta_k x_k)}{1 + exp(\alpha + \Sigma_{k=1}^{n} \beta_k x_k)} \tag{4.1.5}$$

The predictors x at arrival are indoor temperature, outdoor temperature, absence duration and rain presence. During occupancy they are indoor temperature, outdoor temperature, length of presence and rain presence, whilst at departure the predictors are indoor temperature, daily mean outdoor temperature, length of departure and a dummy parameter to represent whether the window is on the ground floor (which reduces the probability of windows being left open at night). With the exception of the presence-related parameters and the ground floor parameter the values of these predictors are supplied by EnergyPlus via the Functional Mockup Interface. No-MASS calculates the occupants' presence as well as the future presence and past absence durations when needed. Absence duration is computed within No-MASS by rewinding the array of chains of presence and absence from the current timestep; and the opposite for the duration of departure. The ground floor parameter is given as an input to No-MASS at runtime as a boolean value per zone, set to true if the zone is at ground level otherwise false. This is defined in an simulation configuration file described in implementation section (3.2) as it can not be supplied by EnergyPlus. By default the regression coefficients α, β_k in equation 4.1.5 are estimated from data relating to an aggregate population (i.e. using all empirical data for all members of the population surveyed), so that each agent uses the same values to predict the probability of a transition. The parametric probability density functions of the Weibull distribution describing window opening survival time takes the form:

$$\lambda\alpha(\lambda t)^{\alpha-1} exp(-(\lambda t)^{\alpha}) \tag{4.1.6}$$

where the shape is $log(1/\alpha) = 0.871$ and the scale is:

$$\lambda = 1/exp(a + b\theta_{out}) \tag{4.1.7}$$

and $a = 2.213$, $b = 0.173$ and θ_{out} is the outdoor temperature supplied by EnergyPlus.

At any given time-step, determine whether a zone is unoccupied. If so, the window opening status is unchanged; otherwise five further cases are considered: 1) If the occupant arrives and the window is closed a random number is drawn. 1a) If this is greater than the probability for opening it the window is kept closed; otherwise the window is opened. 1b) Upon opening survival duration is calculated (the period of time that the window will remain open) from the Weibull distribution. This is decremented at each subsequent time step until the survival duration reaches 0 and the window is closed. 2) If the occupant arrives and the window is open, follow the same steps as in case 1b. 3) For intermediate presence a random number is drawn from a uniform distribution and if this is greater than the corresponding probability for opening, keep the window closed. Otherwise, a survival duration is calculated and step 1b is implemented. 4) If occupants vacate the zone and the window is closed, predict whether the window will be opened as they leave. On the other hand, if the window is open, predict whether the window will be closed. In each case a random number is drawn, if this is below the corresponding transition probability (open to closed / closed to open), the state is changed; otherwise the state stays the same. Within EnergyPlus an external schedule for windows is created, setting the value to be either 1 for fully open or 0 for fully closed for each time step (in the future it would be useful to include predictions of opening proportion, based for example on the model described in Schweiker et al. (2012)).

External Shading Actions

The shading action model (Haldi and Robinson, 2010) predicts lowering and raising probabilities, which are also based on Markov chains. Upon an agents' arrival the first step in this model is to determine the probability with which a raising or lowering action will take place:

$$P_{act}(E_{in}, B_L) = \frac{exp(a + b_{in}E_{in} + b_L B_L)}{1 + exp(a + b_{in}E_{in} + b_L B_L)} \tag{4.1.8}$$

Where E_{in} is the indoor illuminance supplied by EnergyPlus, at a suitable daylight reference point within the zone and B_L is the unshaded fraction at the previous time-step. If the shade is lowered or raised then predict whether the shade is fully raised or lowered:

$$P_{fullact}(E_{gl,hor}, B_L) = \frac{exp(a + b_{out}E_{gl,hor} + b_L B_L)}{1 + exp(a + b_{out}E_{gl,hor} + b_L B_L)} \quad (4.1.9)$$

The regression parameters (b_{in},b_L and b_{out}), taken from Haldi and Robinson (2010), are also estimated using data for an aggregate population. If the shades are only partially raised or lowered, their fractions are drawn from a Weibull distribution:

$$f(\Delta B) = \lambda\alpha(\lambda\Delta B)^{a-1}exp(-(\lambda\Delta B)^{\alpha}) \quad (4.1.10)$$

where $\alpha = 1.708$ and

$$\lambda = \frac{1}{exp(-2.294 + 1.522B_{L,init})} \quad (4.1.11)$$

Otherwise, shading remains unchanged. A similar process but with different probabilistic models occurs whilst occupants are present. Although the model relates to a specific building design, group of occupants and shading system, the work conclusively found that the driving variables for the shading actions are local stimuli on the workplane, which directly links visual comfort, visual variables and actions. Thus should be applicable to a range of buildings where occupants are affected by visual discomfort. The outcomes from these models allow us to set the shading fraction in EnergyPlus. The current version of EnergyPlus (8) only allows shades to be either fully open or fully closed. As such, it does not provide a function to overwrite an external shade fraction value from an external interface such as No-MASS. The EnergyPlus source code was altered to provide a function that reduces the radiation transmitted through the window in proportion to the fraction that the shade was closed, this function can now be accessed from an external interface at each timestep (see Appendix B for the source code changes).

Lighting

The prediction of the use of lights within No-MASS is based on the Lightswitch-2002 algorithm (Reinhart, 2004). Indoor illuminance E of the zone is taken from EnergyPlus for the current time step and compute the probability of turning the lights on when the agent arrives or whilst they are present, and thus whether this action takes place. The switch on probability is calculated as:

$$P = \frac{a + c}{1 + exp[-b(log10(E) - m)]} \quad (4.1.12)$$

where for arrival $a = -.0175, b = 4.0835, c = 1.0361, m = 1.8223$ and while present $a = 0.0027, b = 0.017, c = -64.19$ and $m = 2.41$. When all agents vacate the zone predict whether the lights will be turned off, as a function of the anticipated duration of their absence, calculated by forward-winding from the time of departure until the time of return. In the range $0.5 < D \leq 12$, where D in the duration of departure in hours, the probability of turning lights off at departure is described to a good approximation by:

$$P = 0.268 * ln(D) + 0.259 \tag{4.1.13}$$

For absences below 0.5 hours assume that lights remain on, where as for absences exceeding 12 hours the lights are assumed to be turned off (Pigg et al., 1996). The consequent lighting status (on-off) is set within EnergyPlus at each timestep as a lighting schedule for each zone within the building.

4.2 Case Study

To demonstrate the application of No-MASS and its coupling with EnergyPlus two different buildings in two locations are examined. A hypothetical house and shoe box office are located in both Geneva, Switzerland and in Nottingham, UK. Results from No-MASS are compared to the results arising from standard deterministic schedules and rules for the relevant house/ office typology (or template) used by the DesignBuilder interface. The layouts of the buildings are shown in Figure 4.1.

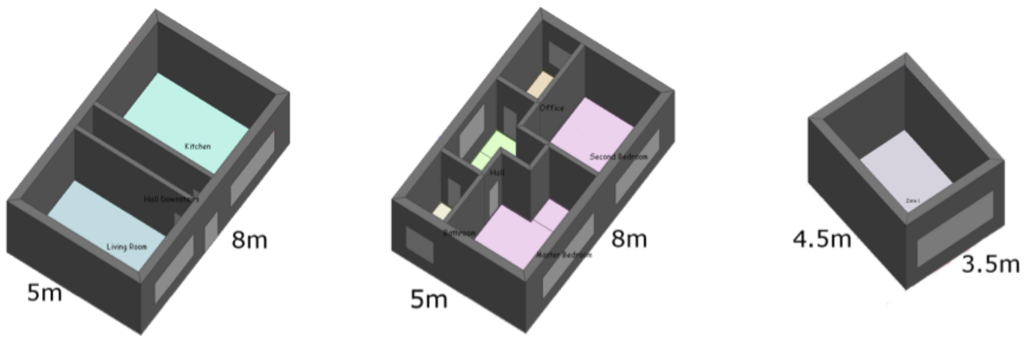


FIGURE 4.1: Residential ground floor (left), Residential 1st floor (middle), Office (Right)

Details such as heating set-points, glazing ratios, etc. are given in Tables 4.2 and 4.3. For simplicity, both buildings adopt the same constructions as in Table 4.4. The weather files are taken from DesignBuilder giving the locations Geneva, Switzerland $(+46°25', 6°13')$ and a location about 40 miles Nottingham, UK $(+53°28', -1°0')$ called Finningley. The stochastic models for windows and shading integrated into No-MASS have also been hard-coded within CitySim and tested with Geneva weather data (Haldi and Robinson, 2011). The results of which provided a visual comparison for initial testing. The weather data was taken from EnergyPlus weather files, of which the closest to the University Of Nottingham is the weather data at Finningley airport, UK. Thus providing two different climates to simulate No-MASS against. The fractional occupancy schedule for the office on weekdays are: [0.0] 00:00 until 07:00, [0.25] until 08:00, [0.5] until 09:00, [1.0] until 12:00, [0.75] until 14:00, [1.0] until 17:00, [0.5] until 18:00, [0.25] until 19:00, [0.0] until 24:00. Interactions with external shades and lights operate on the same schedule; the windows open when occupants are present and the indoor temperature exceeds 24° C. In Nottingham only heating demand is considered, whereas in Geneva cooling demand is also considered. Yearly simulations are performed throughout this thesis, unless otherwise stated.

Zone	Area [$m2$]	Volume [$m3$]	Gross Wall Area[$m2$]	Glazing ratio%	Lighting [$W/m2$]	Setpoint Temp[c]
Livingroom	13	46	36	7	7.5	21
Halldownstairs	4	15	6	0	5	20
Kitchen	15	52	39	8	15	18
Bathroom	3	10	12	11	7	18
Hall	4	19	10	0	5	18
Residential Office	3	12	13	10	5	22
Second bedroom	9	34	22	15	5	18
Master bedroom	10	37	24	16	5	18
Attic	37	26	7	0	0	-
Total	101	255	172	68	5	-

Table 4.2: Residential Building Zone Details

Zone	Area [$m2$]	Volume [$m3$]	Gross Wall Area[$m2$]	Glazing ratio%	Lighting [$W/m2$]	Setpoint Temp[c]
Office	11	39	47	6	20	21

Table 4.3: Non-Residential Building Zone Details

Location	Layer	Thickness (m)	Material
External Wall	Outer	0.1	Brick
External Wall	2	0.07	XPS extruded
External Wall	3	0.1	Concrete Block
External Wall	Inner	0.01	Gypsum Plaster
U-Value			0.37
Internal Partition	Outer	0.02	Gypsum Plaster
Internal Partition	2	0.1	Air Gap
Internal Partition	Inner	0.02	Gypsum Plaster
U-Value			2.86
Ground Floor	Outer	0.13	Urea Formaldehyde Foam
Ground Floor	2	0.1	Cast Concrete
Ground Floor	3	0.07	Floor Screed
Ground Floor	Inner	0.03	Timber Flooring
U-Value			0.26
Floor	Outer	0.10	Cast Concrete
U-Value			4.7
Pitched Roof	Outer	0.02	Clay Tile
Pitched Roof	2	0.02	Air Gap
Pitched Roof	Inner	0.005	Roofing Felt
U-Value			4.97

Table 4.4: Construction Materials

Repeated simulations help us to understand the likely distribution of the output parameters of interest and thus the corresponding robustness of alternative design proposals. But the extra simulation time needed for replicates can be seen as a weakness, especially with large models. It is important then to calculate the minimum number of simulations that need to performed to achieve a cumulative mean convergence graph (Stewart, 2004). To this end, the heating demand at each simulation is taken and added to a cumulative mean and plotted, with the results converging on the number of simulation replicates needed. Figure 4.2 suggests a convergence around 90-100 replicates. A t-test shows that the results do in fact converge at 100 replicates with a 95% confidence interval. At a 90% confidence interval the results converge at around 50-60 simulations.

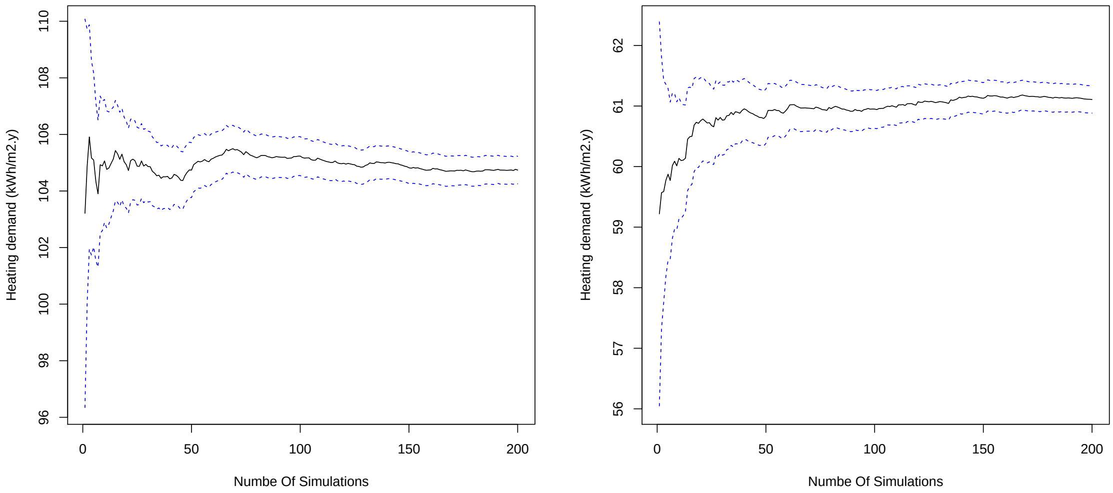


FIGURE 4.2: Convergence of mean heating demand: Geneva Office (left) and House (right), 95% confidence interval

4.3 Comparison of Deterministic Simulation and No-MASS

Depending upon the assumptions made in the choice of deterministic rules and schedules (taken from the defaults in the DesignBuilder tool given to the specified zone type i.e. kitchen, bedroom, etc.), performance results can deviate significantly from those arising from the stochastic representation of people. The predicted median heating demand for our office located in Nottingham obtained using No-MASS was $118.4kWh/m^2$, compared with $93.3kWh/m^2$ when assuming deterministic behaviours. Meanwhile, for Geneva there is a prediction of $103.9kWh/m^2$ (No-MASS) and $83.8kWh/m^2$ (deterministic) and $6.3kWh/m^2$ (No-MASS) and $5.6kWh/m^2$ (deterministic) for cooling.

Our predicted energy demands are considerably closer in the case of our house. For the house located in Nottingham for heating a prediction of $68.5kWh/m^2$ (No-MASS) and $66.1kWh/m^2$ (deterministic) is made, whilst for the Geneva house there is a predicted demand of $60.4kWh/m^2$ (No-MASS) and $58.3kWh/m^2$ (deterministic) for heating and $2.6kWh/m^2$ (No-MASS) and $4.9kWh/m^2$ (deterministic) for cooling.

In this case study the default deterministic rules and schedules assigned to the office building by DesignBuilder under predict (with respect to that predicted using our empirically derived stochastic models) the heating demand by 15 to $25kWh/m^2$. The use of windows during periods of heating causes the principle increase in demand; as the deterministic rules would not allow this to happen. The house encloses larger volume of space, so dampening the effects of occupants' interactions compared to the office, so that interactions per zone cause a smaller difference of approximately $2kWh/m^2$ between the deterministic and stochastic results. This close agreement also suggests that the deterministic representation of occupants' interactions in DesignBuilder is coincidentally close to that predicted using No-MASS models. No-MASS predicts a lower cooling demand in the house than in the deterministic representation but a marginally higher demand in the office. The value difference in both cases is approximately $2kWh/m^2$.

Repeated stochastic simulations enable the likely range of possible energy demands arising from occupant interactions (our type IIIb and type IV errors) to be quantified. A description such as in 90% of our test cases for the Geneva Office the heating demand ranges from $100kWh/m^2$ to $110kWh/m^2$ due to occupant interaction is of more value to designers. The range of results provided by No-MASS is shown in Figure 4.3, also plotted is the deterministic value as a point for comparison.

In line with the adaptive principle that upon experiencing discomfort people act in ways which tend to restore their comfort, it is assumed that in No-MASS our agents interaction are similarly motivated. Our agents sense their environment calling stochastic models that predict interactions that are motivated by the agents' (or strictly speaking the population from which these model were derived) desire to restore their comfort. It is interesting to determine whether our agent interactions have been successful. To this end, as aggregation of thermally discomforting stimuli is made by calculating the degree hours for which a threshold of 25° C has been exceeded, as an indicator of overheating risk (Robinson and Haldi, 2008). The prediction of which for the Nottingham office is 3% (No-MASS) and 3.5% (deterministic). Predictions for the Geneva office are 15% (No-MASS) and 14% (deterministic), which in both cases are similar. For the house the percentage of time above 25° C are lower for No-MASS compared to the deterministic case. For the UK a prediction of 0.01% (No-MASS) and 2% (deterministic), whilst for the Geneva house a prediction of 6% (No-MASS) and 8.5% (deterministic).

The improved performance here in the case of No-MASS suggests that our empirically derived stochastic models better emulates occupants' behaviours (they are more effective) than the assumed deterministic rules.

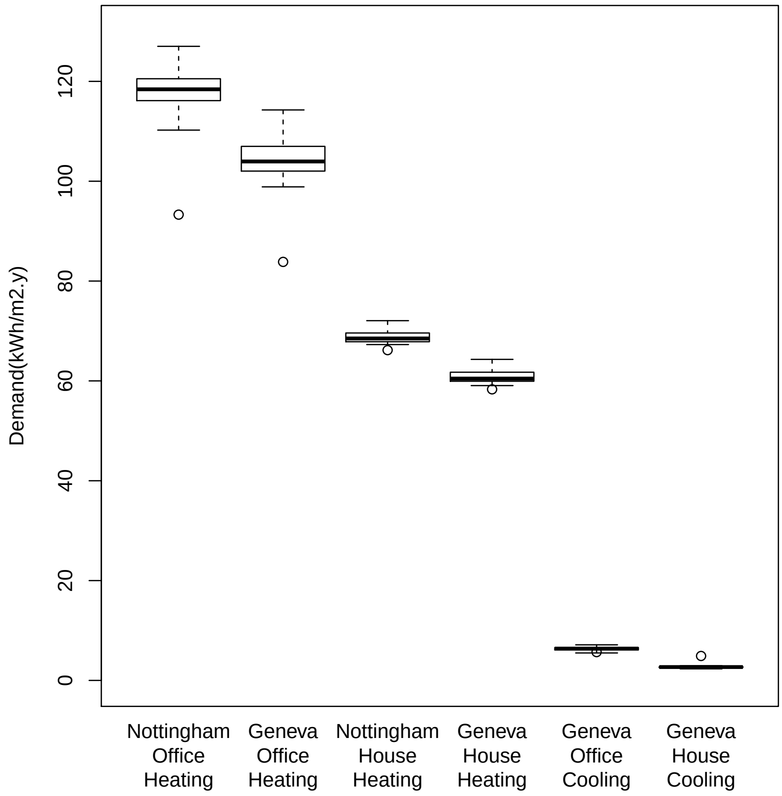


FIGURE 4.3: Simulation results for yearly heating demand (Boxplot) Stochastic agent platform 100 replicates, (Circle) Single deterministic simulation

Breaking the simulations down into monthly box plots[1] (Figure 4.4) enables us to further understand the variations in building performance arising from occupants' interactions over the course of a year. During January and December the heating demand is higher for the stochastic simulations, as occupants can interact with the windows at a range of temperatures (eg. to refresh the indoor air). During the summer months the deterministic simulations register no heating demands whereas, the stochastic simulations allow the temperature to drop below the heating setpoint as windows can now be left open over night and so may have inadvertently cooled the interior. This is shown in Figure 4.5 where during the summer, windows can be open for as much as 20% of the month. The monthly box plots for the rooms of the house (Figure 4.8) demonstrate that windows are proportionately more open during the summer months.

[1]A standardised way of displaying the distribution of data based on five values: minimum, first quartile, median, third quartile, and maximum

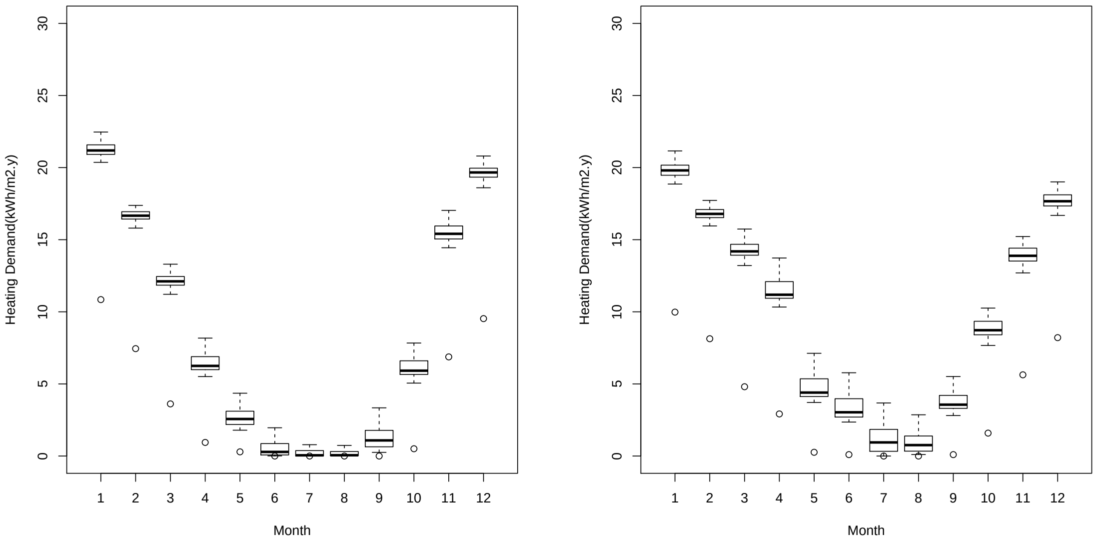


FIGURE 4.4: Simulation results for monthly heating demand, (Left) Geneva Office, (Right) Nottingham Office. (Boxplot) Stochastic agent platform 100 replicates, (Circle) Single deterministic simulation

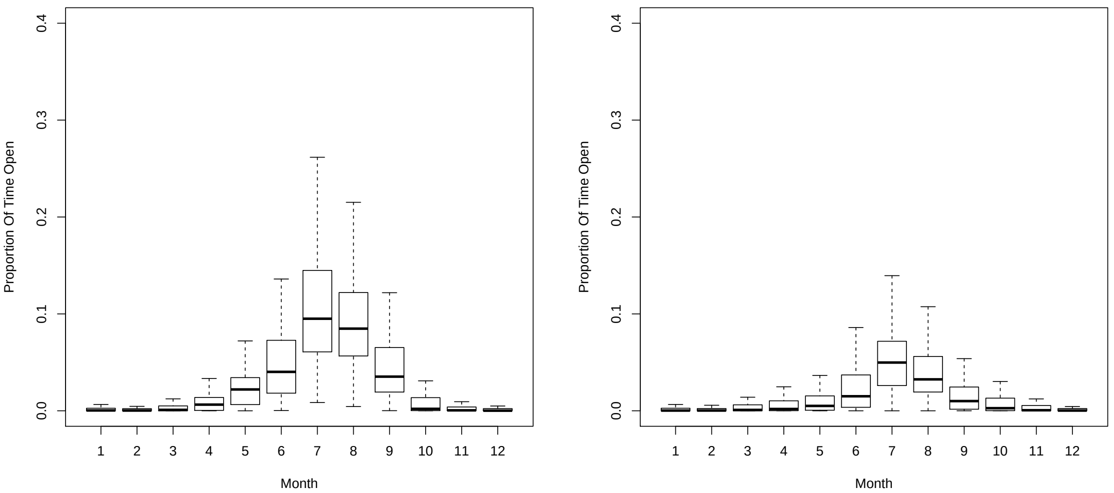


FIGURE 4.5: Monthly average window state, stochastic 100 replicates, Geneva Office (Left), Nottingham Office (Right)

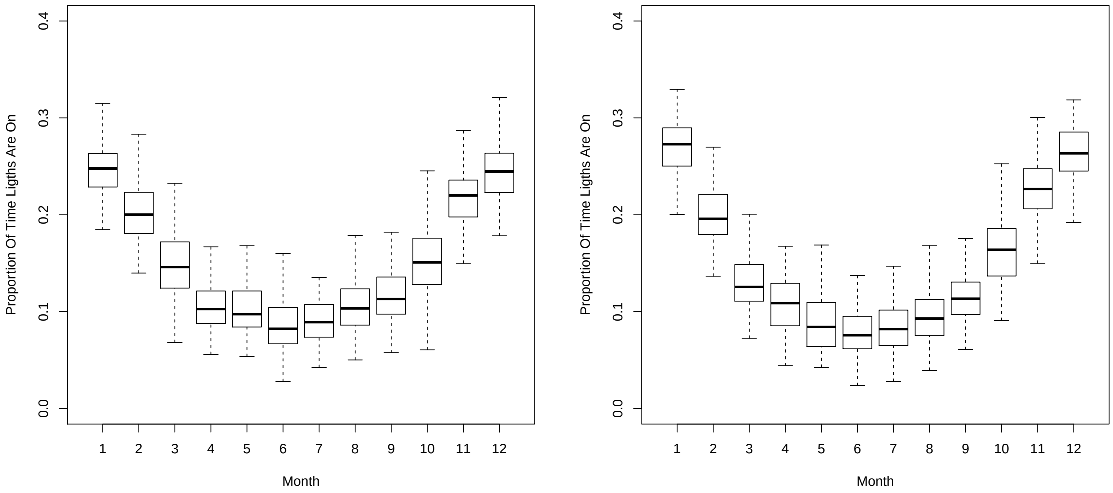


FIGURE 4.6: Monthly average light state, stochastic 100 replicates, Geneva Office (Left), Nottingham Office (Right)

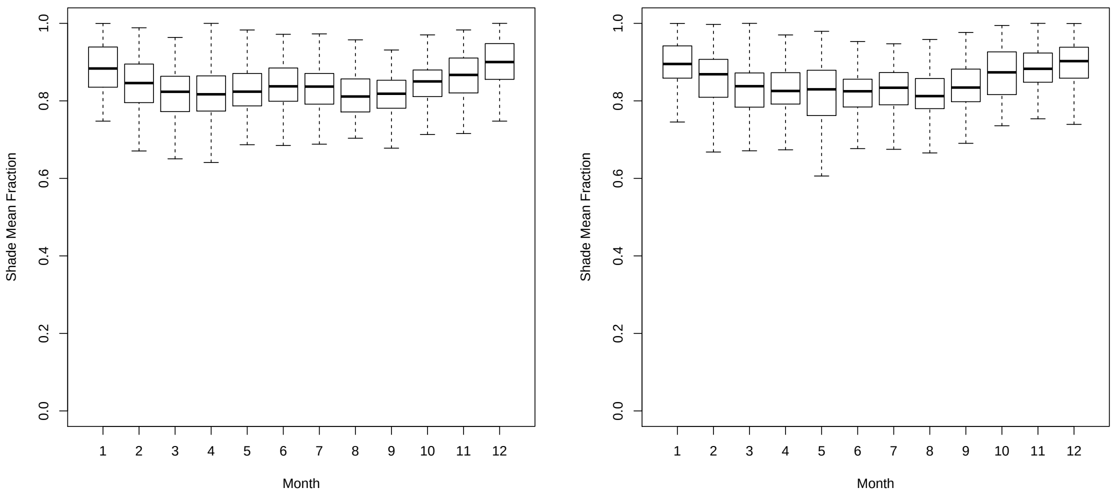


FIGURE 4.7: Monthly Average external shade opening fractions, stochastic 100 replicates, Geneva Office (Left), Nottingham Office (Right)

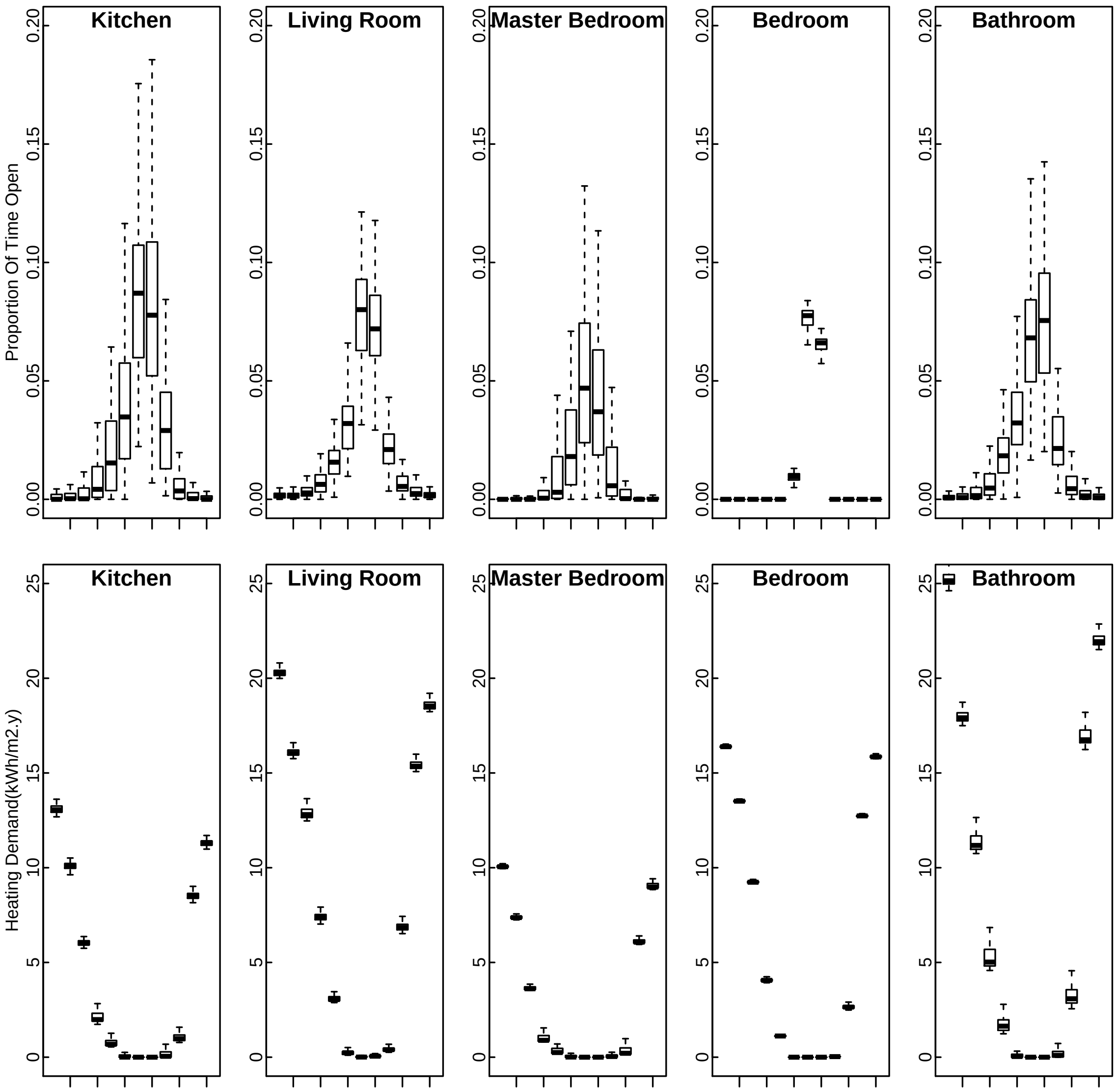


FIGURE 4.8: Monthly window and heating usage for the rooms of the Geneva House

4.4 Model Analysis

Relatively few of the stochastic behavioural models, that have been developed thus far, have been subjected to rigorous and comprehensive validation exercises. Of those that have, the focus has been on the models' ability to reproduce observed behaviours, expressed as discrete states. Through careful parameter selection, the best of these models have retained a parsimonious number of parameters and their coefficients estimated, for the purpose of emulating reality. But these models may be more complicated than they need to be, for the purposes of energy performance prediction or for design decision making support. It may be that for such purposes less complicated forms of model would suffice.

In this section seeks to determine firstly whether our stochastic models are useful (have a significant impact on building performance) and then whether these models really are parsimonious, as their authors claim them to be; for the purpose of building performance prediction.

Impact Of Stochastic Models

After performing 100 simulations of a base case simulation including only heat gains of the occupants from No-MASS, a model is added, either window, shading or lighting and 100 more simulations performed. To test in the first instance whether the models themselves are useful a t-test is performed between both sets of data, if a significant difference is observed then the model should be included within No-MASS. Table 4.5 shows that for both Nottingham and Geneva all models have a significant impact on the simulation results as the p-value in all cases are below 0.05. Each model should therefore be used in future building performance simulation. This is by no means an exhausted list of the occupant interactions that may take place; chapter 5 explores other occupant interactions that should developed into empirically validated stochastic models and included within No-MASS.

Case	Location	Tested Value	Simulation	t	p-value
House	Geneva	Cooling	Windows model	135	~ 0.0
House	Geneva	Cooling	Lights model	1790	~ 0.0
House	Geneva	Cooling	Shading model	-28.7	~ 0.0
House	Geneva	Heating	Windows model	87.2	~ 0.0
House	Geneva	Heating	Lights model	-1040	~ 0.0
House	Geneva	Heating	Shading model	-13.4	~ 0.0
House	Nottingham	Heating	Windows model	137	~ 0.0
House	Nottingham	Heating	Lights model	-1730	~ 0.0
House	Nottingham	Heating	Shading model	-13.6	~ 0.0
Office	Geneva	Cooling	Windows model	-238	~ 0.0
Office	Geneva	Cooling	Lights model	307	~ 0.0
Office	Geneva	Cooling	Shading model	-86.5	~ 0.0
Office	Geneva	Heating	Windows model	-43.9	~ 0.0
Office	Geneva	Heating	Lights model	-263	~ 0.0
Office	Geneva	Heating	Shading model	-17.8	~ 0.0
Office	Nottingham	Heating	Windows model	-51.8	~ 0.0
Office	Nottingham	Heating	Lights model	-350	~ 0.0
Office	Nottingham	Heating	Shading model	-22.2	~ 0.0

Table 4.5: Models included in simulation compared to the base case simulation (degrees of freedom = 99 and where p-value is ~ 0.0 it is less than 2.2e-16)

Model Input Parameter Analysis

In this section the form of our chosen models is tested, to determine whether they can simplified, by removing parameters and observing the affect this has on building energy demand. Removing parameters from the inputs of the model would mean that fewer variables are parsed between No-MASS and the building solver, the number of computations taking place within each model could be reduced, resulting in a decreased simulation time and that fewer input parameters will be required by them. The latter may be particularly helpful in the case of parameters that, whilst available in principle, may not be readily available in practice (eg. precipitation); or which require further data manipulation (eg. designation of ground floor windows).

To test the affect of removing model parameters on the overall simulation, once again the t-test is used, but now a base case simulation configuration is created for each full model and compared to the same configuration but with an input parameter removed. For example, create a base case configuration for the house and the office in the Nottingham and Geneva locations with each utilising the full window opening model and perform 100 simulations. The window model is altered by removing the ground floor input from the model, 100 more simulations are run. A t-test is then performed against the yearly heating energy demand and where required the cooling energy demands. If there is no significant difference (the null hypothesis cannot be rejected) then the input can be removed, simplifying future simulations.

Table 4.7 shows that for both buildings the input of rain into the window opening model has little affect on the heating or cooling of the building so that the rain input can be removed from future simulations. The duration an occupant is away from the zone for longer than 8 hours can also be removed as this also has no significant effect. The ground floor parameter does not have a significant affect on the heating of the house, but does affect cooling demand. Therefore, for houses that do not have cooling, the ground floor parameter could be removed. All other input parameters were significant for the window interaction model.

The shading model inputs and their significance are presented in table 4.8, which shows that for the office in Geneva heating demand is weakly sensitive to external illuminance but is significant in Nottingham; whereas workplane illuminance is significant in Geneva but has little effect in Nottingham. Cooling demand is strongly sensitive to both parameters within the house for Geneva and Nottingham. Table 4.9 shows that all inputs to the lighting model have a significant affect when removed. Both models have few environmental inputs and as each parameter affects energy demands in our case study (albeit not consistently) their full sets of parameters are retained.

Window Interactions on Office

Location	Tested Value	Removed Input	t	p-value
Geneva	Cooling	Future Duration	1.1	0.274
Geneva	Cooling	Ground Floor	3.08	0.0027
Geneva	Cooling	Indoor Temperature	-30.3	~ 0.0
Geneva	Cooling	External Daily Mean Temperature	-5.32	6.43e-07
Geneva	Cooling	External Temperature	-30.5	~ 0.0
Geneva	Cooling	Rain	0.264	0.792
Geneva	Heating	Future Duration	1.51	0.134
Geneva	Heating	Ground Floor	-1.67	0.0982
Geneva	Heating	Indoor Temperature	16.3	~ 0.0
Geneva	Heating	External Daily Mean Temperature	1.99	0.0496
Geneva	Heating	External Temperature	8.93	2.3e-14
Geneva	Heating	Rain	-0.327	0.744
Nottingham	Heating	Future Duration	-0.159	0.874
Nottingham	Heating	Ground Floor	-1.63	0.106
Nottingham	Heating	Indoor Temperature	17.7	~ 0.0
Nottingham	Heating	External Daily Mean Temperature	2.74	0.00736
Nottingham	Heating	External Temperature	13	~ 0.0
Nottingham	Heating	Rain	0.0572	0.954

TABLE 4.6: Window Input Parameter Sensitivity Analysis on non-residential building (degrees of freedom = 99 and where p-value is ~ 0.0 it is less than 2.2e-16)

Window Interactions on House

Location	Tested Value	Removed Input	t	p-value
Geneva	Cooling	Future Duration	-0.315	0.753
Geneva	Cooling	Ground Floor	2.77	0.00663
Geneva	Cooling	Indoor Temperature	-20.7	~ 0.0
Geneva	Cooling	External Daily Mean Temperature	-8.86	3.31e-14
Geneva	Cooling	External Temperature	-25	~ 0.0
Geneva	Cooling	Rain	-0.543	0.589
Geneva	Heating	Future Duration	1.72	0.0879
Geneva	Heating	Ground Floor	0.429	0.669
Geneva	Heating	Indoor Temperature	6.61	1.97e-09
Geneva	Heating	External Daily Mean Temperature	2.72	0.00777
Geneva	Heating	External Temperature	6.47	3.82e-09
Geneva	Heating	Rain	0.723	0.471
Nottingham	Heating	Future Duration	0.718	0.474
Nottingham	Heating	Ground Floor	-1.16	0.248
Nottingham	Heating	Indoor Temperature	8.55	1.54e-13
Nottingham	Heating	External Daily Mean Temperature	2.52	0.0132
Nottingham	Heating	External Temperature	6.36	6.26e-09
Nottingham	Heating	Rain	-0.00445	0.996

TABLE 4.7: Window Input Parameter Sensitivity Analysis on residential building (degrees of freedom = 99 and where p-value is ~ 0.0 it is less than 2.2e-16)

Shade Interaction

Case	Location	Tested Value	Removed Input	t	p-value
Office	Geneva	Cooling	External Illuminance	2.69	0.00849
Office	Geneva	Cooling	Workplane Illuminance	-5.01	2.41e-06
Office	Geneva	Heating	External Illuminance	0.928	0.356
Office	Geneva	Heating	Workplane Illuminance	2.69	0.00842
Office	Nottingham	Heating	External Illuminance	-4.61	1.21e-05
Office	Nottingham	Heating	Workplane Illuminance	-1.3	0.197
House	Geneva	Cooling	External Illuminance	-28.9	$\sim$ 0.0
House	Geneva	Cooling	Workplane Illuminance	-8.55	1.55e-13
House	Geneva	Heating	External Illuminance	57.9	$\sim$ 0.0
House	Geneva	Heating	Workplane Illuminance	9.93	$\sim$ 0.0
House	Nottingham	Heating	External Illuminance	46.4	$\sim$ 0.0
House	Nottingham	Heating	Workplane Illuminance	10.6	$\sim$ 0.0

TABLE 4.8: Shade Input Parameter Sensitivity Analysis (degrees of freedom = 99 and where p-value is $\sim$ 0.0 it is less than 2.2e-16)

Lights Interaction

Case	Location	Tested Value	Removed Input	t	p-value
Office	Geneva	Cooling	Workplane Illuminance	-60.3	$\sim$ 0.0
Office	Geneva	Heating	Workplane Illuminance	53.9	$\sim$ 0.0
Office	Nottingham	Heating	Workplane Illuminance	61.8	$\sim$ 0.0
House	Geneva	Cooling	Workplane Illuminance	-94.9	$\sim$ 0.0
House	Geneva	Heating	Workplane Illuminance	97.9	$\sim$ 0.0
House	Nottingham	Heating	Workplane Illuminance	113	$\sim$ 0.0

TABLE 4.9: Lights Input Parameter Sensitivity Analysis (degrees of freedom = 99 and where p-value is $\sim$ 0.0 it is less than 2.2e-16)

4.5 Conclusion

Coupled with EnergyPlus two test cases were studied, with the range of results being compared to deterministic representations. Shown through these applications (a single occupied office building and a house occupied by two adults who do not have children) No-MASS provides a convenient, comprehensive and rigorous basis for representing occupants stochastic behaviours in EnergyPlus (and other software using FMI); providing designers with means to evaluate the performance of their designs in response to the range of expected behaviours and thus to evaluate the robustness of their design solutions that is not possible using current simplistic deterministic representations.

In terms of the usefulness of the models integrated into No-MASS and their composition; the window, lighting and shading models can significantly effect the simulation results, although the window model within No-MASS can be simplified. The input parameters of rain presence and duration of absence from a room at departure and, in the absence of cooling systems also the ground floor identifier, can each be removed from the window opening model. This simplifies the calculations taking place, and the processing of inputs to No-MASS. The other models are already parsimonious in their form for building performance simulation purposes; so removing parameters would adversely effect simulation results.

A single simulation with No-MASS can add between 1% and 30% onto the annual simulation time depending on the choice of occupant behaviour models used. The number of occupants will also be a factor in determining the simulation time. Simulating replicates carries a computational penalty in proportion to the number of replicates, however this can be handled through hardware acceleration. To improve the accessibility of No-MASS, it has been integrated into the DesignBuilder graphical user interface used with EnergyPlus. This will allow users to quickly set up their stochastic occupancy representations for future simulations with No-MASS, and enables users to take advantage of the simulation manager that allows for repeated simulations, either locally or in the cloud.

Multi agent simulation also provides a quick and easy way to simulate diverse populations; it is possible to quickly change input coefficients for each model allowing the ability to test multiple occupant use cases. This can be done automatically at each simulation iteration, enabling users to more fully explore the robustness of their design to uncertain future populations of occupants and the diversity in behaviours amongst members with similar characteristics.

Chapter 5

Theory Driven Models

As described in the preceding chapters, No-MASS has a core set of rigorous and validated models, and has been developed into a proof of principle platform to test the stochastic nature of occupants on building performance simulation. However, this does not account for occupant interactions that do not have stochastic models developed to test their effect on building performance. There remain many gaps in our ability to model occupants' stochastic behaviours, based on a lack of corresponding data. But this need not preclude us from employing pragmatic rules that would help us to identify types of interactions that merit further study; or indeed whether simplified agent behaviours may in themselves suffice. For example what if it were possible to test the effects that occupants' closing of shades or switching off lights while watching television? Would this impact on the performance of the building? Performing a simulation within No-MASS with this interaction included would be less costly than collecting the data, building a model and then empirically validating it. This chapter introduces a belief-desire-intention (BDI) framework as a research tool to test such what-if scenarios within No-MASS, extending its current features in a pragmatic way. A selection of what-if scenarios are chosen and implemented within No-MASS demonstrating the approach, highlighting how sensitive the performance of the building is to each set of rules, thus illustrating the importance or otherwise of continued development of stochastic models that are not currently available for use within building performance simulation.

5.1 Theory

A highly popular methodology for representing an agents reasoning mechanism is the BDI architecture developed by Bratman (1987). An agent's beliefs encode the agent's understanding of the world, themselves and other agents. This does not have to be a truth, merely a suitable representation of the agent's perception of their environment. The agent's desires are what they would like to achieve in the world, expressed as goals, worked out through an internal deliberation and filtering process. An agents intentions are the desires that an agent has to some extent acted on and structured into plans. An agent might believe the indoor environment to be warm for example, and correspondingly desire it to be cooler, leading to the intention to open a window to make this environment cooler.

BDI assumes an agent commits to a plan (also referred to as a recipe) of multiple intentions, this can include a long term goal such as choosing to go to a meeting later in the day or a short term goal to seek shelter when outside in rainfall. This plan needs to be fairly rigid and in line with the agent's intentions, it can also be partial. For example a plan to go to a meeting does not have to include how to get to the meeting, because the environment that an agent is within may change over time. A changing environment would make rigid and complete long term plans fairly useless. Georgeff and Lansky (1986) used an approach built on a BDI architecture that was called the procedural reasoning system (PRS), it was developed for the reaction control system of a NASA space shuttle. The system sensed the environment and had beliefs about the environment, an interpreter would look at the desires of the system and make plans, that would then be pushed to a process stack. They chose this system as it allowed for partial plans; the system could then try to implement a plan and react to any changes to the environment as it went along. Although BDI and PRS are an effective way of agent representing reasoning, there was no formal definition of how the abstract concepts should be defined within an agent, meaning there have been many different variations in implementing this approach (D'Inverno et al., 1998). One approach to implementing BDI theory within an agent is called agent-oriented programming (Shoham, 1993). This has been developed as an extension to the ideas of object-orientated programing, but instead of unconstrained objects there are agents that have parameters that make up an agent's beliefs, commitments, capabilities etc. Kinny and Georgeff (1996) expands on a similar methodology to support the design and implementation of agent systems. This work appears to have more depth, explaining how inheritance and encapsulation can be used to allow groups of beliefs, goals and plans to be shared across agents, making it easier to analyse the interactions between plans in the design phase. With encapsulation it is easy to implement sets of agents with both similar and differing sets of attributes without re-definition. This approach was used in the implementation a complex air traffic control systems, with Kinny et al. (1996) arguing that their methodology allows for fine-grained analysis of systems, as it allows for agent boundaries that are flexible. There are other agent architectures built on the BDI concept, INTERRAP (Fischer et al., 1995), JAM (Huber, 1999), JACK (Busetta et al., 1999) and Mora et al. (1998); each with there own incremental improvements. For a detailed overview of early agent architectures see Wooldridge and Jennings (1995). Of most interest in the present context is the INTERRAP frameworks; which includes multiple layers of BDI facilitating co-operative planning, local planning and behaviour. Other systems tend to use a single BDI system, controlling all types of planning and behaviour, whereas the layered approach could lead to simpler systems, with the logic separated for each part, improving modularity.

A BDI framework was also used by Rao and Georgeff (1995) to model an air traffic control system, where multiple agents represented the planes and a global agent the air traffic controller. However, Rao and Georgeff found the system too abstract to use in practice and the system did not describe all the processes taking place. They also found that the option generator (which actions are available) and deliberation phases (which actions to take) were too slow for use in a real system. For such a system they found it hard to search for the optimal solution, due to the number of options available for controlling the planes and the possible solutions available to any situation.

Aside from the complexities of BDI systems, another weakness is their inability to describe social interactions and the use of learning. However Georgeff et al. (1998) argues that there is nothing within BDI that specifically makes it poorly suited to social interactions or agent learning. It is just that up until this point the frameworks did not specifically mention interactions between agents. BDI has therefore been expanded in ways to include social interaction, with one such attempt involving an obligation aspect (Broersen et al., 2001). Beliefs-Obligations-Intentions-Desires (BOID) creates a feedback loop where obligations can override desires or vice-versa. An obligation is described as a social norm, or the outside motivational attitudes of the agent. In contrast a desire is the internal attitude of the agent. An obligation and desire can be at conflict or in co-operation depending on the type of agent. For example self motivated agents are likely to chose their own beliefs over that of the group.

Guerra-Hernández (2005) demonstrated how agent learning can be linked to BDI. Given a plan with a library, an agent will learn what event will be satisfied by the given plan; meaning that the agent's beliefs, desires and intentions do not have to be linked before the system is used. This could lead to unexpected results with agents choosing plans not initially designed for the situation, but which achieve the goal. This may be interesting from a research perspective, but in the case of air traffic control or a NASA space shuttle may not be desired (for example, a suitable learned plan to stop two planes colliding would be for them to fly very close without touching, this would be an valid solution but would make the passengers very uncomfortable). Phung et al. (2005) also introduced a learning component, drawing from knowledge and history to learn a set of virtual beliefs about the environment instead of which plans reach each goal; further demonstrating that BDI can be extended by adding new features.

5.2 No-MASS BDI Architecture

Agents within No-MASS in general handle stochastic phenomena using statistical models that are estimated from real world empirical data. No-MASS uses a BDI approach to handle interactions where such data, and thus the corresponding models are not currently available. In these cases the BDI approach allows agents within No-MASS to sense the environment though input parameters. Then, when a No-Mass agent is given a set of beliefs, it can make a plan about how to influence the environment to it's benefit. This section outlines our vision for the BDI architecture within No-MASS; a set of beliefs or knowledge about what the agent will understand about the environment and themselves is developed; a set of desires that the agent may have about what they want to achieve is created, finally the intents and plans that an agent can take in the environment are described. This BDI architecture allows the sensitivity of buildings performance to be tested against different agent interaction types, creating a rationale for collecting future data to build empirical models that can replace or indeed reinforce our BDI plans. As No-MASS has been developed in C++, an object oriented programming language, it is only appropriate to follow a methodology that combines object oriented understanding of systems with an agent implementation. Building on the early work of BDI and PRS, Kinny and Georgeff (1996) and Kinny et al. (1996) set out a methodology and modelling technique for BDI agents, known as the agent-oriented methodology. Their methodology describes agents in terms of objects (a definition of a variable that has a set of attributes and methods). Encapsulation and inheritance is used to define multiple agent types. The method describes how to go from a set of goals an agent may wish to achieve to the plan that will describe the agent interactions. No-MASS uses their work as a guide, as it is not language specific and presents a methodology rather than a specific set of rules to follow; enabling us to adapt these principles and roles for implementation in No-MASS.

Using the agent oriented approach, agents can be viewed from two viewpoints, internal and external. The external view consists of the agent model and an interaction model.

- The agent model represents the view of the agents, their instances and when they come into effect. The current implementation of No-MASS has occupants as agents, each occupant is an instance of the agent class and the role it plays begins when the simulation starts.

- The interaction model describes the responsibilities of the agents, the services that they provide, their interactions and communication between the system and the agents.

The interaction model is more complex within No-MASS than the agent model, as the services they provide and the responsibilities that they have are complex. The responsibilities that No-MASS agents holds are; monitoring the environment, responding to certain events with actions, maintaining comfort (through both proactive and reactive actions) etc. So far No-MASS includes models for window, shading and lighting interactions, a presence model for offices and an activity model for residential buildings. Our agents are therefore responsible for monitoring the inputs of the models and calculating their responses. However, other considerations are needed, as these models do not cover all the No-MASS responsibilities. The BDI approach should work in conjunction with the stochastic models. For example adding the ability to open the window when cooking in response to a build up of air irritants. For models that are not included, a view of how occupants interact with the building environment needs to be included. In computer science a use case diagram is often used to show the interactions a user may perform with a system. The same technique is used to highlight how occupants may use a building. In our case these diagrams will illustrate where occupants may interact with the building (for interactions that are not yet addressed through stochastic models) and their underlying reasons for that interaction. Any missing responsibilities can then be defined using this diagram. Occupants have different needs depending on the type of building in question, a residential building use case diagram is displayed in Figure 5.1 and a non-residential building in Figure 5.2, the symbology of which is in Table 5.1. For simplicity, there is a focus on interactions that impact on building energy performance and, as already mentioned, for which No-MASS has no stochastic models.

Within No-MASS when simulating residential building the stochastic activity model is used, it has nine activities while present: cooking, cleaning, washing oneself, sleeping, audio/visual, passive, IT, washing appliance and metabolic. The activities cooking, cleaning, washing oneself and sleeping can involve reactions from occupants to air irritants. For example, cooking and cleaning both cause odours that occupants may find unpleasant. While sleeping an occupant exhales carbon dioxide and odours that can increase in concentration over time. When washing one's self steam can build up. These irritants can influence the occupant's decision to open of windows. Since the current stochastic window model does not consider the air quality of a zone as an input, and no alternative model exists that does, the BDI approach is employed as to better understand its importance.

Similarly the use of lights and shading by occupants can be influenced by the activity they are performing at the time. For example, while sleeping an occupant will desire to have the lights off and the shades closed so they are not disturbed by light during the early morning hours. During the activity audio/ visual, occupants may wish to close the curtains due to glare from external sources and perhaps to dim lights if it is night. Since the stochastic models within No-MASS were developed from data collected from offices, there is no privacy factor consideration, which can be significant in home environments. An occupant of a residential building will often close a shading devices during the evening and night hours to maintain privacy. Similarly while another privacy consideration is during the washing one's self (particularly while bathing), windows and maybe shades closed for modesty reasons.

Our final category of occupant interactions within a residential space related to HVAC systems and hot water usage. Both are difficult to model with simple rules due to the complexity of the interaction. First there are both temporal and spacial influences on the choice of heating setpoints: how soon before an occupant arrives in a zone does the heating need to be turned on and to achieve the setpoint to satisfy so their comfort requirements. Similarly after which a duration of anticipated absence should the heater be turned off or the setpoint reduced by how much? Hot water usage is similarly complex. How much is used cooking and cleaning? It is possible to link the usage to cooking and washing (both personal and using appliances) activities, but not easily the quantity used.

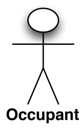
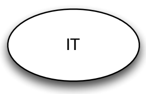
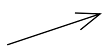
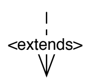
	An actor that has a influence on the system.
	An use case of the building, ie. how the occupant will use the building.
	An association that an actor can have on a use case.
	An extend adds functionality to the use case, for example during sleeping the user may have a light irritation.

Table 5.1: Use Case Diagram Symbology

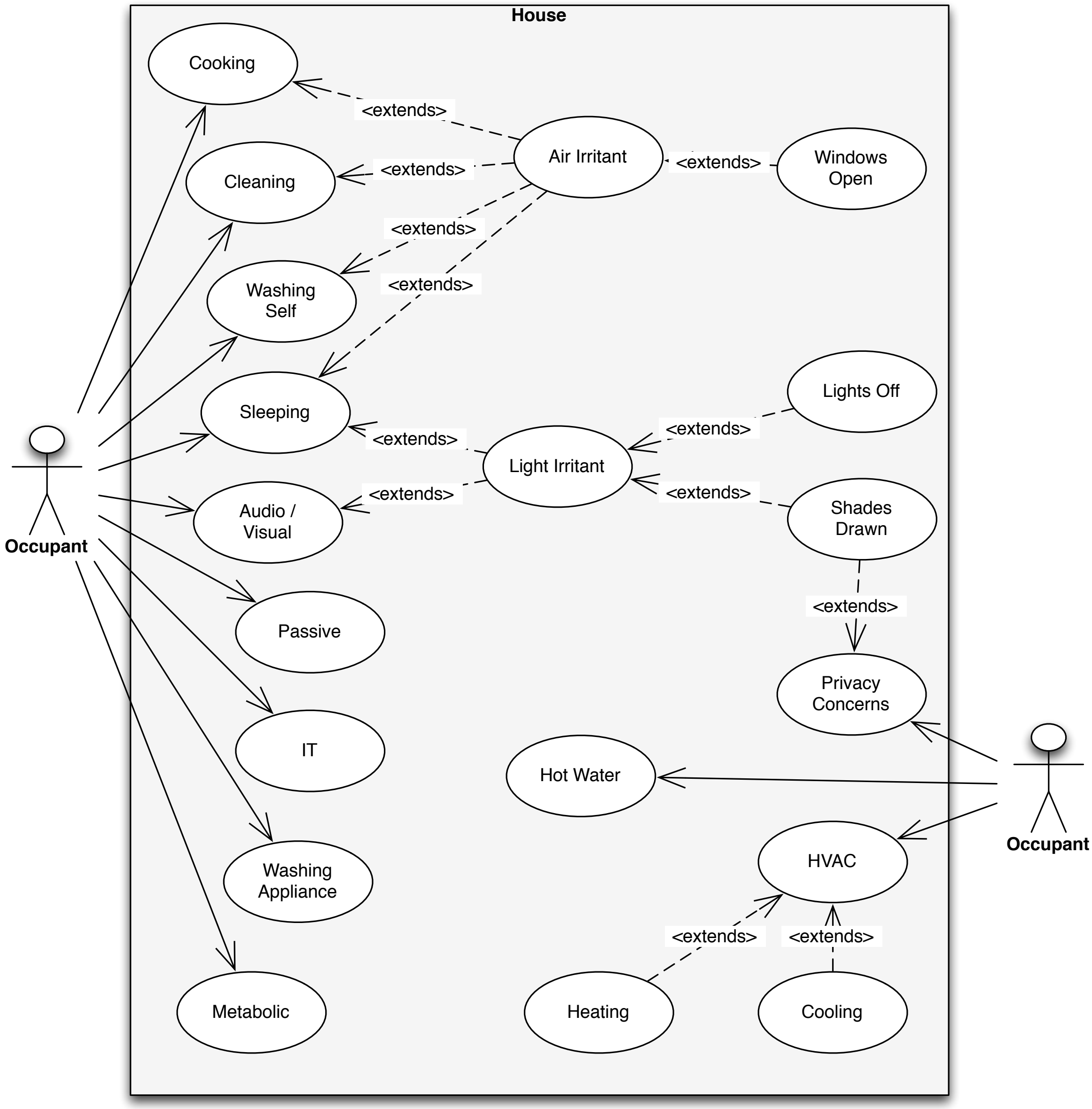


FIGURE 5.1: Use case diagram of occupant interactions in residential building

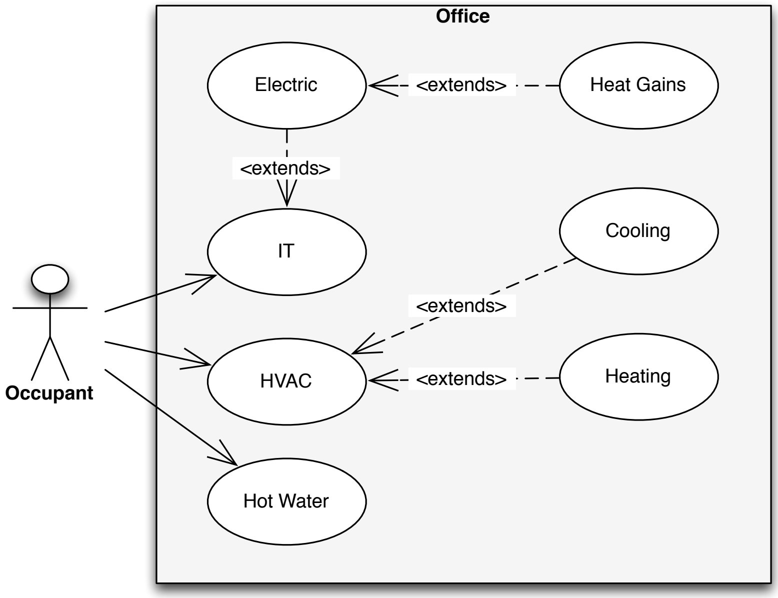


FIGURE 5.2: Use case diagram of occupant interactions in non-residential building

Non-residential buildings such as offices have a reduced set of interactions that can take place within them. Occupants are either present or absent, so activity dependent BDI rules are not possible. While present within an office occupants make use of IT equipment, for which there is a heat gain and a corresponding influence on building energy demand. Use of these devices does not always mimic presence; often a computer is switched on at arrival, remaining on during intermediate absences, and turned off at departure. It is possible to link the devices to an agents first arrival time and their final departure time of the day but it is less straight forward to predict the fluctuating power demands of the devices between these times. Occupants may also interact with the HVAC system, but as with residential building these can be complex. Using a methodology such as BDI to represent a generic way framework for interacting with HVAC systems would require definition of a large set of plans that attempt to model all the cognitive processes involved. Complex plans would tend to render them specific to a given scenario and would not be generic enough to be applicable beyond these cases. [1]

[1]A better approach that would take into account these complexities for each zone would be agent learning. This would allow the agents to learn how to heat the environment to minimise some indicator of their discomfort. This is discussed further in Chapter 6.

Regarding our BDI framework the following agent responsibilities are defined:

1. To monitor their own state.
2. To monitor the environment.
3. To understand when an activity may cause irritation (discomfort).[2]
4. To react and communicate any ways in which they wish to reduce discomfort.
5. To understand their presence and communicate any IT equipment used.

Kinny and Georgeff (1996) specifies that once responsibilities are defined, agents should be decomposed to the service level. Services are activities (not related to the activity model) that make up the responsibilities and cannot be decomposed further. The services give understanding as to the information that will be needed in order to run it, the types of communication with the system, the events that need to be noticed and the actions that need performing. These are broken down in Table 5.2.

[2]See Baker and Standeven (1996) who eloquently describe the link between irritability and adaptive behaviour

Responsibility	Service	Noticed Events	Actions	Communication
1	Monitor Activity State			
2	Monitor Lights		Request current state	Light State
	Monitor Window		Request current state	Window State
	Monitor Shade		Request current state	Shade Fraction
	Monitor Environmental Conditions		Request current conditions	Environmental conditions (Air temp, humidity, etc.)
	Monitor Time	Time of Day	Request current state	System Time
3	Calculate Discomfort	Change of Activity		
4	Communicate Desires		Light Interaction	Light State
			Window Interaction	Window state
			Shade Interaction	Shade fraction
5	Use Equipment	Presence	Equipment Interaction	

Table 5.2: Agent Services

To achieve the first responsibility the agent needs to understand the state they are in at any given time. The second responsibility requires an understanding of the environmental conditions (air temperature, humidity, etc.) and the state of the device they can interact with (windows, shades, etc.) and the current time. These values are requested and values returned through communication with the system. The third responsibility the agent must know if they are in a state of discomfort based on their activity, eg. in the event of (attempting to) sleep and the light is on, an agent is uncomfortable and will wish to turn off the light. The fourth responsibility to communicate their desires back to the system is in the form of an interaction with a device. Agents could also communicate their beliefs (i.e. that they are over heating) and then as group decide on a course of action that would resolve the discomfort for the majority of the population. As No-MASS agents communicate the actions taken from the stochastic models, for simplicity the same is done here. Finally, while present the use of appliances needs to be simulated, this is linked to the presence of the agent within an office.

Once the services have been broken down it is possible to start defining the internal viewpoint of an agent. Agents have mental attitudes, these are the agent's beliefs, desires and intentions. These reflect the informational, motivational and deliberative states of the agent. With the agent-oriented approach an agent is made up of three theoretical models, the belief, goal and plan models. The belief model is the agent's knowledge about the environment and the actions an agent can perform, they are made up of a belief set. A belief state is an instance of a belief set (ie. for No-MASS agents the current state of the environment, a belief state, would be that the window is open, the indoor temperature is high, etc.). The goal model defines what the agent wants to achieve, for example to have the lights off while sleeping. The plan model describes how an agent would achieve a goal, eg. switch light off when entering the activity sleeping.

The Belief Model

The belief model is a belief set and a belief state. The belief state is an instance of a belief set. The belief set is a set of predicates and functions which are derived from a belief set diagram and its associations, see Figure 5.3. The BDI framework within No-MASS required that agents have knowledge of the windows, shades, lights, their current activity and time of day these are parsed from EnergyPlus and the stochastic models. The agents have an understanding of the first arrival of the day and which departure is the last of the day, calculated by forward winding the predicted presence in a zone. Their activity can be one of the nine activities previously defined by the activity model. From here they know what their activity was in the previous timestep and if they are not in a given activity, allowing an agent to work out if there has been a change in the activity event. A shade belief is a fraction of the current shade state, and the current irradiance allowing the agent to lower the shade at a given threshold to maintain privacy. The understanding of a light is either on or off and a window is either open or closed. The window can also have an opening duration allowing the agent to set a survival time for how long it will be in its current state. This requires the Agent to have knowledge of the time. The agents will only make an interaction a proportion of the time, the agents therefore need to have knowledge of the percentage of time that an interaction will take place. For example an agent will not always open the window while cooking, but they may do 10% of the time. The percentages should be empirically validated against real world data for the building, however for now the percentage is used to test how sensitive the building's performance is to the rule. The derived predicates and functions are in Table 5.3. Each activity will have an associated interaction, meaning that a transition to an activity can cause an interaction with either the window, light or shade.

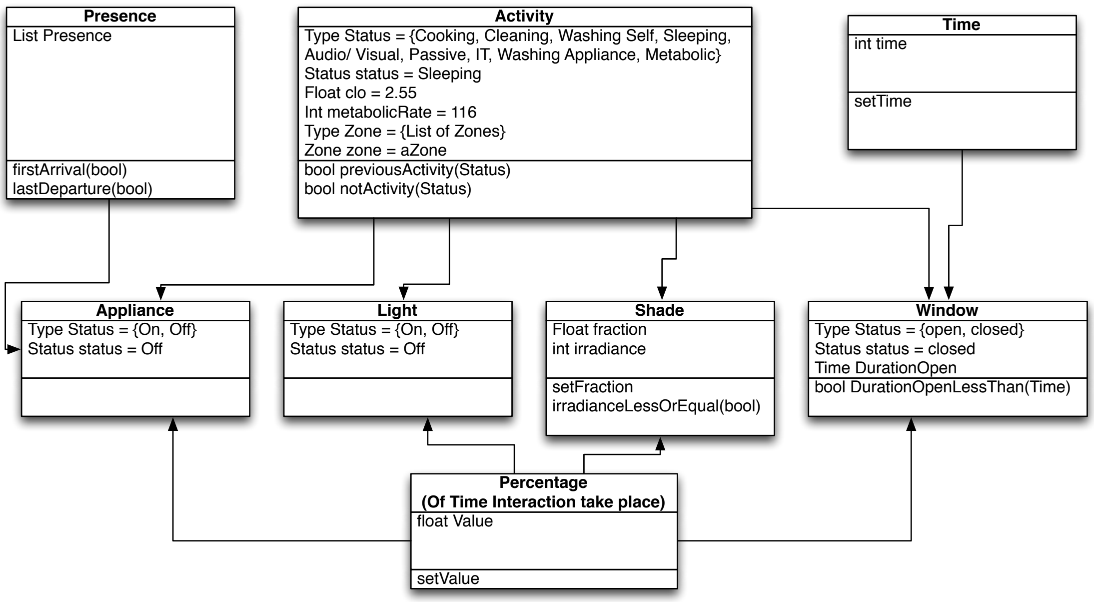


Figure 5.3: No-MASS belief set diagram

Predicates	Functions
status(Time, Status)	Status status(Time)
Percentage(Percentage, Value)	Value setValue(Percentage)
status(Window, Status)	Status status(Window)
status(Light, Status)	Status status(Light)
Fraction(Shade, Fraction)	Fraction fraction(Shade) Bool irradianceLessOrEqual(Shade)
status(Activity, Status)	Status status(Activity)
status(Presence, Status)	Bool firstArrival(Presence) Bool lastDeparture(Presence)
status(Appliance, Status)	
associated(Percentage, Window)	
associated(Percentage, Shade)	
associated(Percentage, Light)	
associated(Percentage, Appliance)	
associated(Activity, Window)	
associated(Activity, Shade)	
associated(Activity, Light)	
associated(Activity, Appliance)	
associated(Presence, Appliance)	
associated(Time, Window)	

Table 5.3: Agent predicates and corresponding functions

The Goal Model

The Goal model consists of a goal set and the goal states. Within No-MASS there are a number of interactions that will be tested based on the agent's belief of the environment. An assumption is made within No-MASS, that an agent wishes to minimise their discomfort with regards to light irritants, air irritants and privacy concerns and they may want to use a computer while working. All goals have a percentage that limits the number of times that an action will take place.

During the activity sleeping it would be sensible to turn off the lights. Our goals with regards to light irritants would have a goal state consisting of a set of predicates that have the following values:

$$\begin{aligned} &status(Light, Off) \\ &status(Activity, Sleeping) \\ &Percentage(Value, \%) \end{aligned} \tag{5.2.1}$$

During the sleeping activity, it may also be sensible to shut the shades

$$\begin{aligned} &setFraction(Shade, 0) \\ &status(Activity, Sleeping) \\ &Percentage(Value, \%) \end{aligned} \tag{5.2.2}$$

These goals can be combined into a single goal,

$$\begin{aligned} &status(Light, Off) \\ &setFraction(Shade, 0) \\ &status(Activity, Sleeping) \\ &Percentage(Value, \%) \end{aligned} \tag{5.2.3}$$

There are also privacy concerns when washing oneself and an air irritant with regards to humidity. An agent may wish to open the window, but only for a short time while the gases expel. A similar situation may occur while sleeping, if there is a build up of CO_2 or other pollutants. The goal set for both situations is therefore:

$$\begin{aligned} &setFraction(Shade, 0) \\ &status(Activity, WashingSelf) \\ &Percentage(Value, \%) \end{aligned} \tag{5.2.4}$$

$$\begin{aligned} &previousActivity(Activity, WashingSelf) \\ &notActivity(Activity, WashingSelf) \\ &status(Window, Open) \\ &DurationOpenLessThan(Window, 1\, timestep) \\ &Percentage(Value, \%) \end{aligned} \tag{5.2.5}$$

$$\begin{aligned}
&previousActivity(Activity, Sleeping)\\
&notActivity(Activity, Sleeping)\\
&status(Window, Open)\\
&DurationOpenLessThan(Window, 1\,timestep)\\
&Percentage(Value, \%)
\end{aligned} \tag{5.2.6}$$

The next goals are for the window opening during cooking due to odours and gases and then closure of the shade during audio/visual to reduce glare while watching television during the daytime.

$$\begin{aligned}
&status(Activity, Cooking)\\
&status(Window, Open)\\
&Percentage(Value, \%)
\end{aligned} \tag{5.2.7}$$

$$\begin{aligned}
&setFraction(Shade, 0)\\
&status(Activity, Audio/Visual)\\
&Percentage(Value, \%)
\end{aligned} \tag{5.2.8}$$

Finally, within homes there is a privacy concern in the evenings and over night where it is desirable to close the shades. The threshold for the privacy concern is defined to be an irradiance of $50W/m^2$. The following goal sets are defined within No-MASS.

$$\begin{aligned}
&irradianceLessOrEqual(Shade, 50W/m^2)\\
&setFraction(Shade, 0)\\
&Value(Percentage, \%)
\end{aligned} \tag{5.2.9}$$

These sets of goals have been defined for residential simulations. Within the non-residential settings there is a need for a different set of goals. This will cover the use of appliances while at work. They will be turned on at the first arrival and stay on until the last departure. Most occupants will leave their equipment on while they attend meetings or go for lunch. Our presence belief has a function *firstArrival* allowing us to ensure the at first arrival any appliances are turned on. As No-MASS sets all present occupants within non-residential building to the activity IT, the goal set does this also, see Goal Set 5.2.10. On the final departure of the day the agent turns the appliances off, see Goal Set 5.2.11.

$$\begin{aligned}&firstArrival(Presence, True)\\&status(Activity, IT)\\&status(Appliance, On)\\&Value(Percentage, \%)\end{aligned} \tag{5.2.10}$$

$$\begin{aligned}&lastDeparture(Presence, True)\\&status(Appliance, Off)\\&Value(Percentage, \%)\end{aligned} \tag{5.2.11}$$

The Plan Model

The plan model consists of sets of plans that are organised using plan diagrams (Kinny et al., 1996). The symbols used are given in in Table 5.4 and are based on Harel's (1987) state charts.

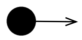
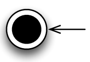
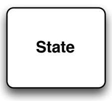
	The start of the state sequence, usually triggered by some event such as moving into an activity state
	The exit of the state sequence. When an agent has finished the plan the agent leaves the state
	A state of a particular class, that an agent could transition to
	A transition event, when an event occurs the agent moves from one state to another
	Denotes that a state has a history of its previous status, for example in Figure 5.4, the privacy state would remember the substate between timesteps, the shade would stay closed

TABLE 5.4: State Chart Symbology

In the privacy plan described in Figure 5.4 an agent will enter the state "no light influenced privacy concern", and at the event of an irradiance level lower or equal to $50W/m^2$ and a the user defined probability occurs the agent would move into the "light influenced privacy concern" state. The user define probability threshold is set before the simulation, a random number is drawn and if the random number is greater than the user defined probability the event takes place. This mechanism limits the events to only occur a percentage of the time. If the event was not triggered the agent would stay in the same state. After the transition into "light influenced privacy concern" state the agent sees that the shade is open and attempts to move the state of the shade to closed. The agent will remember their desired window state until the agent sees the irradiance level is greater than $50W/m^2$ at which point the exit parent state is triggered; opening the shade.

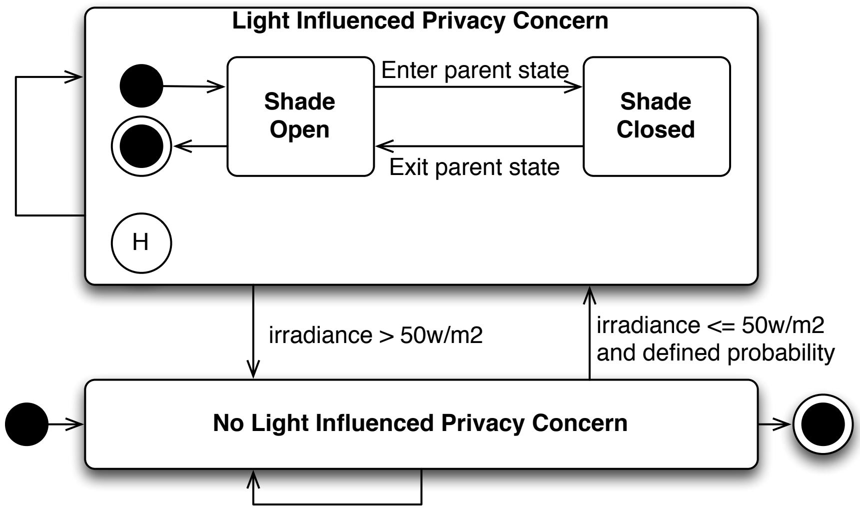


FIGURE 5.4: The light influenced privacy plan, used within the household to achieve goal 5.2.9

Figures 5.5 and 5.6, present plans that occur when the agent is sleeping and they have Goal Set 5.2.3 applied. Both plans execute in parallel to achieve the goal, removing the light irritants from the bedroom and achieving privacy. The sleeping window plan (Figure 5.7) is more complex, firing on the transition from the state sleeping to one of the other eight states. Once the event occurs the window state is set to closed for one timestep and the window is then closed. This will expel any unwanted gases from the room achieving Goal Set 5.2.6.

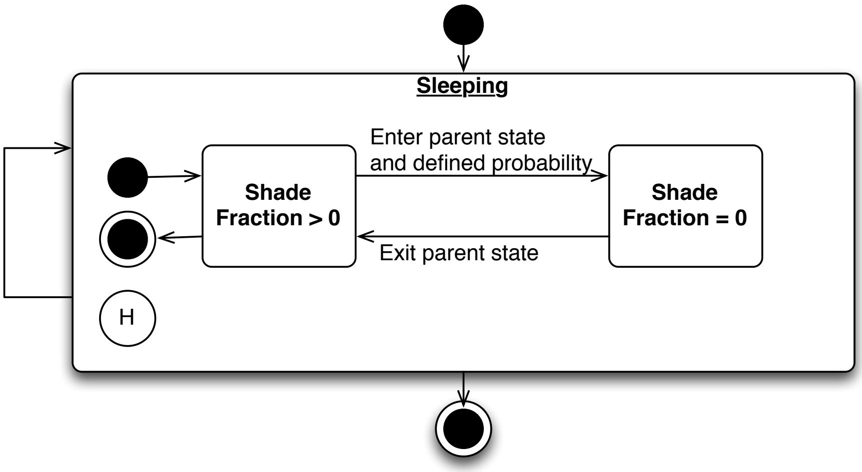


FIGURE 5.5: Sleeping privacy plan

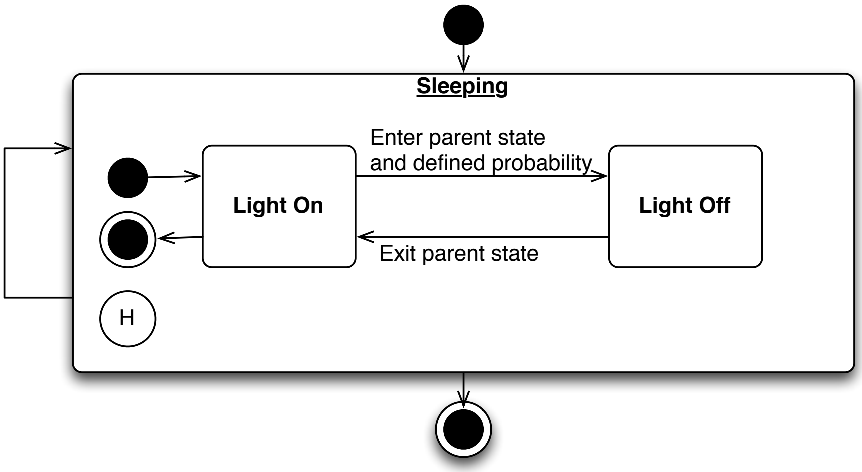


FIGURE 5.6: Sleeping lighting plan

Plans in Figures 5.8 and 5.9 achieve the goals given in Goal Sets 5.2.4 and 5.2.5 respectively. Shades are closed during the self washing activity. These plans could be joined together into a single plan, but separate plans will allow for greater flexibility in later iterations of No-MASS. Figure 5.10 provides the plan achieving Goal Set 5.2.7, opening the window during cooking and then 5.11 provides the plan for Goal Set 5.2.8, closing the shade during audio/ visual.

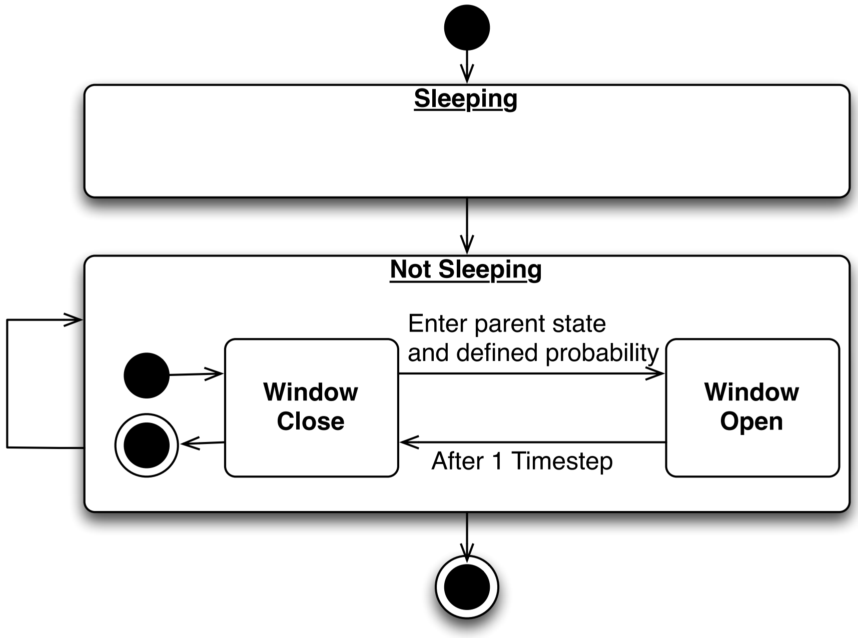


FIGURE 5.7: Sleeping window plan, the not sleeping state is used to depict any of the other eight states

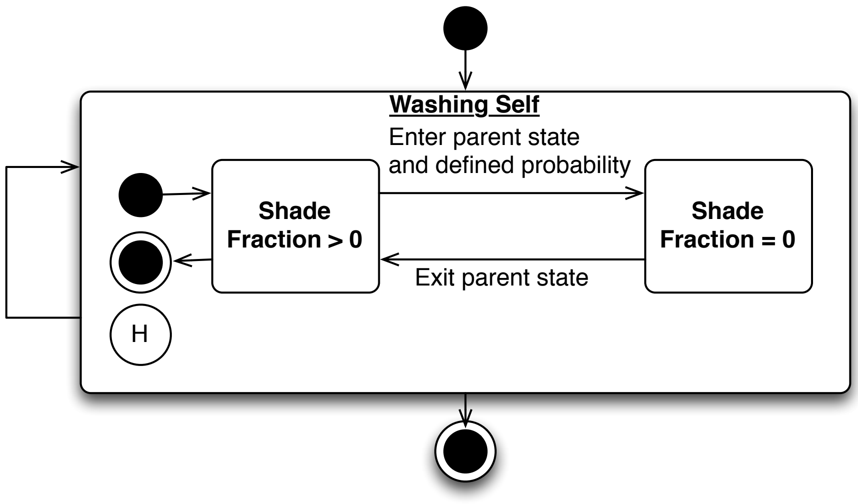


FIGURE 5.8: State diagram washing oneself with privacy consideration

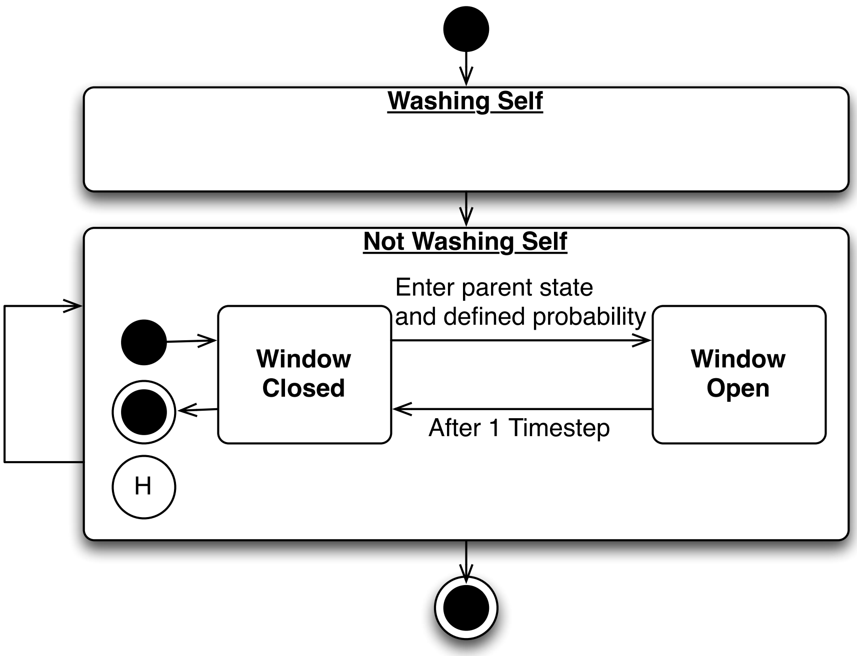


FIGURE 5.9: Washing self plan to open window once washing has completed, the not washing self state is used to depict any of the other eight states

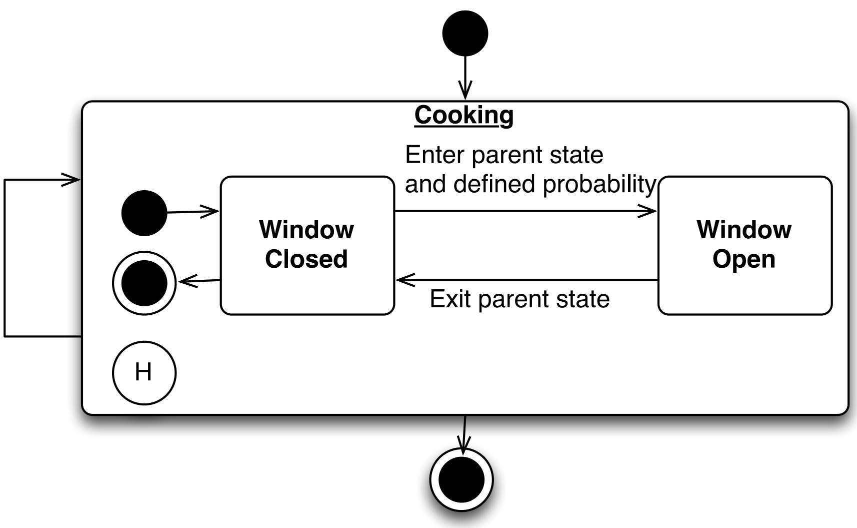


FIGURE 5.10: Window open during cooking plan to remove any odours from the air

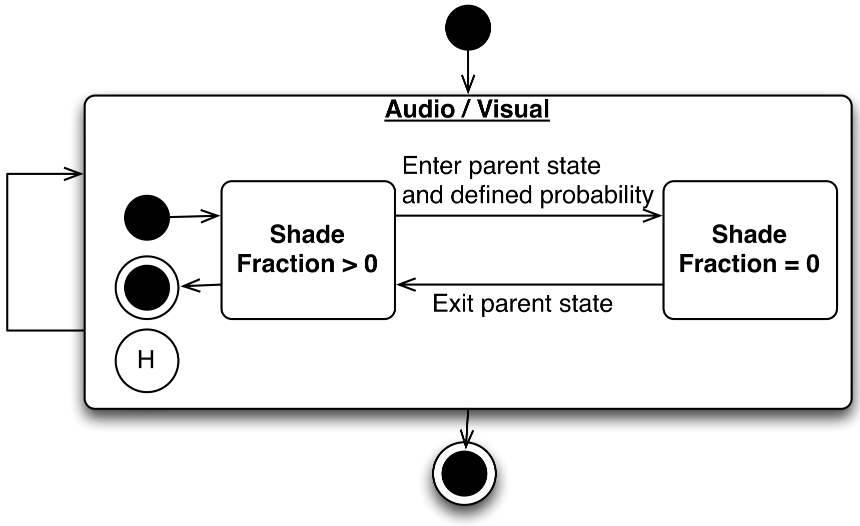


FIGURE 5.11: Plan for deploying shading during the audio/visual state to reduce glare on the televisions

Appliance use while at work in non-residential buildings requires a plan described by Figure 5.12. While present in the office the agent has the state IT, as this sets their clothing and metabolic rate for the activity. On entering this state at the first arrival of the day the agent will turn the appliance from off to on. The history state implies that the agent will remember this state over time. The agent can leave the office and the appliance will stay on, then on the final departure of the day the agent will turn all the appliances off, achieving Goal Sets 5.2.10 and 5.2.11.

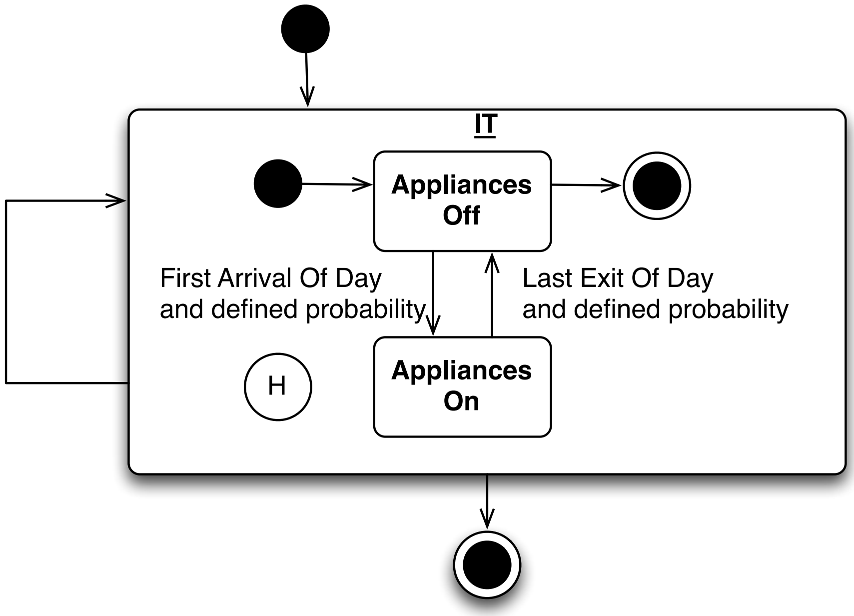


FIGURE 5.12: Plan for appliance usage during non-residential building use

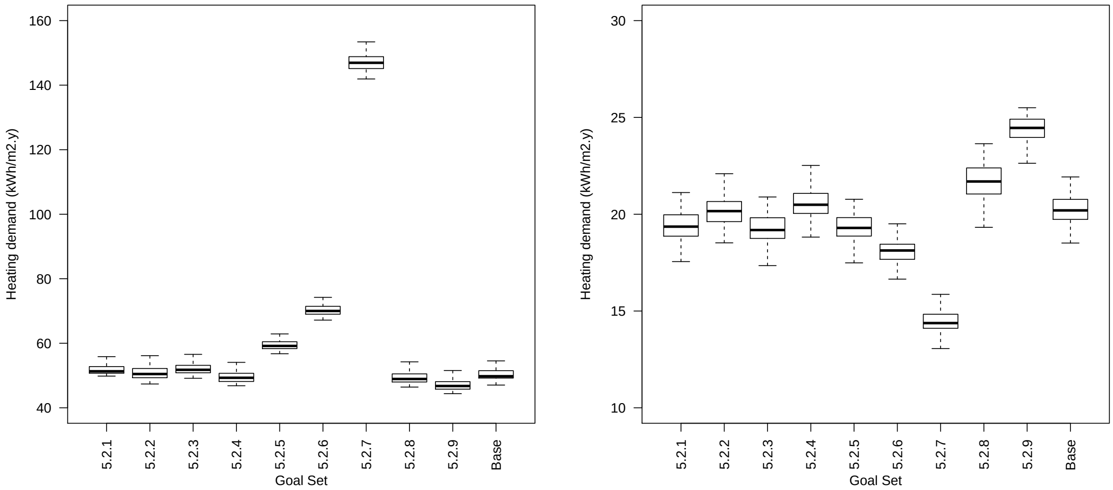
Integrating the BDI rules within No-MASS allows for the testing of occupant interactions within simulation software. They can answer *what-if* scenarios, showing the effects of different interactions. Sensitivity analysis performed against the different plans allows recommendations to be made about where to focus future efforts of data collection and empirically validated stochastic model development. It will highlight which plans have the greatest effect on the energy demands of the building and which can be discarded as they have no significant effect.

To test the sensitivity of building performance to each of the above plans, a plan is taken and executed every-time a goal can be achieved. The overall demands (heating and cooling) are compared for 100 replicates against a base case (where the plan never runs) using a t-test. If there is a significant effect the chance of the plan executing is reduced by 1% and tested again. This is repeated until it is possible to specify how often the plans need to be executed to influence the performance of the building. This allows descriptions such as while cooking the occupant can open the window for upto 10% of the time without influencing the buildings heating demands.

Using the both buildings in Figure 4.1 and set in the Geneva location, the effects of the rules on building performance are tested. Two occupants are included in each building and the stochastic models are enabled but can be overridden by the BDI rules.

5.3 The Influence Of Rule Based Models

The results show that when the plans execute 100% of the time they influence the heating and cooling demand, with the exception of having the shades drawn while sleeping which has an insignificant effect. The chance of sleeping during the daylight hours is small and high during the night, as such little irradiation is blocked during the day, and as there is little irradiation during the night there is little effect on heating or cooling demands. Figure 5.13b shows the influence of each goal and their associated plan. As these rules are effectively deterministic, the interquartile range of each boxplot is similar to that of running with and without the BDI rules. The rules which effect window openings have the most influence on building performance, with opening the window during cooking having the greatest effect.



(A) Heating demands (B) Cooling Demands

FIGURE 5.13: BDI Rules' Sensitivity Analysis, 100 replicates with the plans executing 100% of the time

All heating demands are the median value of the 100 simulations unless otherwise stated. The goal 5.2.1, turning the lights out while sleeping, increases the heating demand from the base case value of $49.7kWh/m^2$ to $51.3kWh/m^2$, this is due to the reduction in heat gains through lighting. Sleeping with the shaded closed (Goal 5.2.2) has a small increase on heating demand at $50.4kWh/m^2$; the occupant profiles used have the majority of the sleep occurring during the night hours, so that little solar radiation is intercepted during the day. With different profiles this might not be the case. In countries where siestas are common during the day this goal may have a greater effect. The combined goal (5.2.3) for sleeping with the shades closed and the lights off increases the heating demand to

$51.7kWh/m^2$. Closing the shades while showering to aid privacy (Goal 5.2.4) reduces the heating demand to $49.3kWh/m^2$ the small difference being due to the short time spent showering. On the other hand opening the window at the end of showering to expel the humid air (Goal 5.2.5) causes an increase in heating demand to $59.1kWh/m^2$. Although windows are not opened after every shower the sensitivity analysis will show how often they need to be, to influence building heating demand. The same is true for goals opening the window when waking (5.2.6) and opening the window while cooking (5.2.7) which increase heating demand to $70.0kWh/m^2$ and $146.9kWh/m^2$ respectively. Goal 5.2.8, closing the shades during the activity audio visual to reduce glare on the television, decreases heating demand to $48.9kWh/m^2$. Privacy concerns causing shades to be shut during the night (Goal 5.2.9) lowers the heating demand to $46.07kWh/m^2$. This is due to the shades being left lowered in the morning. An agent may not wake until after the sun rises and only then when they enter a zone do they raise the shade.

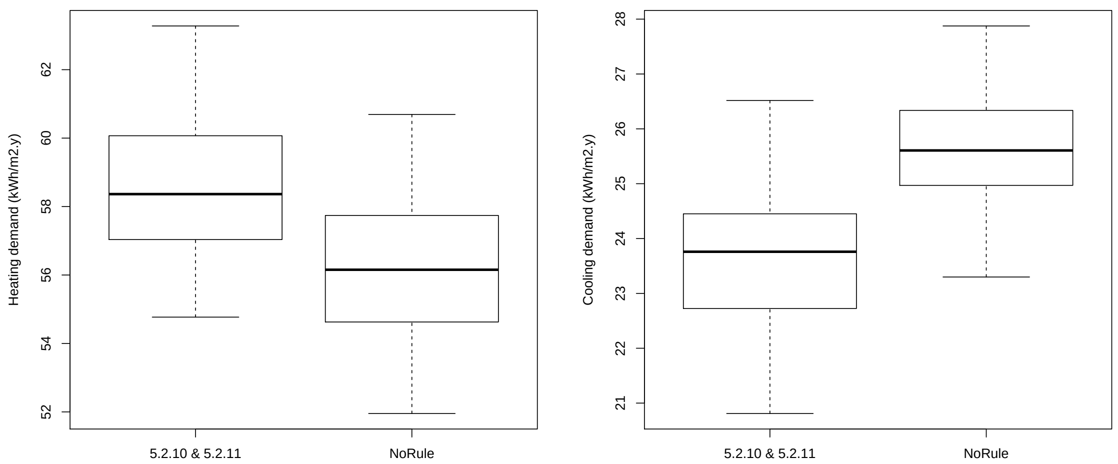
The influence of the BDI rules on cooling demands are more modest than on heating demands. When using no BDI rules the base case cooling demand is $20.1kWh/m^2$. Turning the lights off while sleeping (5.2.1) reduces cooling demands to $19.3kWh/m^2$, closing the shades while sleeping (5.2.2) has no effect with a cooling demand of $20.1kWh/m^2$ and the effect of the combined goal (5.2.3) is $19.1kWh/m^2$. Window and shade 5.2.4 ($20.4kWh/m^2$), 5.2.5 ($19.2kWh/m^2$) and 5.2.8 ($21.6kWh/m^2$) have cooling demands that are close to the base case. Opening the window after sleep (5.2.6) lowers the cooling demand to $18.1kWh/m^2$, cooling the bedroom with natural ventilation in the morning. The same is true for the goal opening the window after cooking (5.2.7), which causes a large reduction in cooling demand to $14.3kWh/m^2$. Privacy during the night (Goal 5.2.9) causes an increase in cooling demands to $24.3kWh/m^2$; shades are reducing the heat loss through the window during the night.

Sensitivity analysis of the goals show that a number of the plans need only be executed less than 5% of the time to have an effect on heating demand. As seen in Table 5.5 Window related plans (5.2.5, 5.2.6 and 5.2.7) only need to be executed 1% of the time to influence heating demand and at most 3% of the time to influence the cooling demand. Closing the shade while sleeping (5.2.2) has no significant effect in this case study. Although the combined goal of 5.2.3 requires it to be executed 14% of the time for a significant effect on heating and above 1% of the time for cooling. However this influence is probably due to the lighting part of the goal (5.2.1), which requires the light to be off at least 11% of the time to influence heating demand and 3% to influence cooling. Closing the shades while performing the audio/ visual activity (5.2.8) effects the heating demand when performed at least 22% of the time and at least 3% of the time for cooling. Privacy during the evening causing the shades to be closed for just 1% of the time effect for both cooling and heating.

Tested Value	Goal	Percentage of time plan needs to execute for significance	t	df	p-value (0.05)
Heating	5.2.1	11	-2.01	99	0.0473
Heating	5.2.2	100	-1.76	99	0.0811
Heating	5.2.3	14	-2.68	99	0.00853
Heating	5.2.4	100	1.19	99	0.236
Heating	5.2.5	1	-2	99	0.048
Heating	5.2.6	1	-8.02	99	2.2e-12
Heating	5.2.7	1	-6.14	99	1.75e-08
Heating	5.2.8	22	3.36	99	0.00109
Heating	5.2.9	1	10.3	99	<2.2e-16
Heating	5.2.10 & 5.2.11	2	3.93	99	0.000159
Cooling	5.2.1	3	2.15	99	0.0343
Cooling	5.2.2	100	0.834	99	0.406
Cooling	5.2.3	1	2.27	99	0.0254
Cooling	5.2.4	4	-2.13	99	0.0352
Cooling	5.2.5	2	2.61	99	0.0103
Cooling	5.2.6	1	4.81	99	5.37e-06
Cooling	5.2.7	3	3.13	99	0.00229
Cooling	5.2.8	3	-3.98	99	0.000132
Cooling	5.2.9	1	-43.2	99	<2.2e-16
Cooling	5.2.10 & 5.2.11	3	-4.95	99	3.06e-06

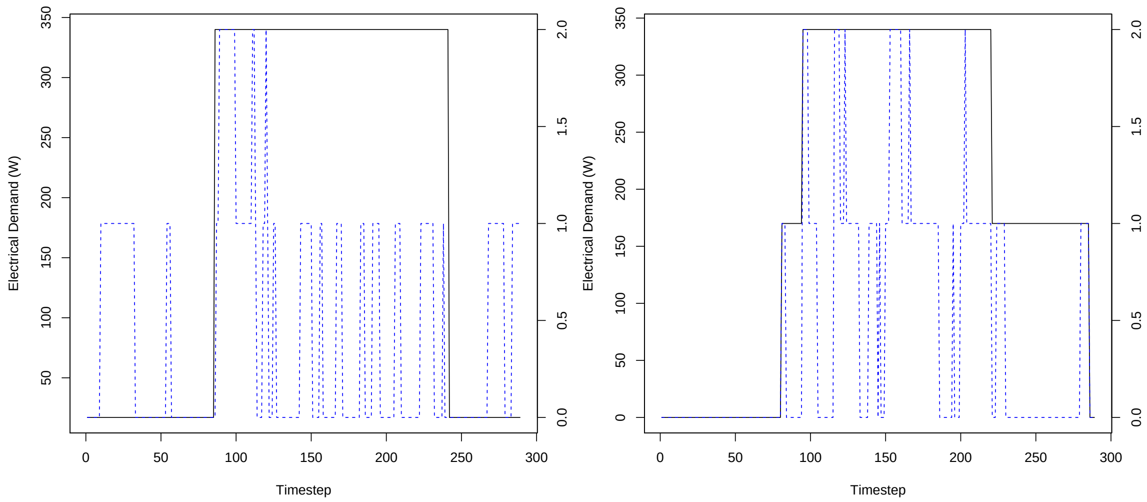
TABLE 5.5: BDI Rules' Sensitivity Analysis

BDI rules for office appliance interaction Goals 5.2.10 and 5.2.11, have a significant effect on both the heating and cooling demands of the office. When the rule is applied 100% of the time there is an increase of $2kWh/m^2$ in heating demand due to the lower usage of desktop computers, and a decrease of $2kWh/m^2$ in cooling demands. Figure 5.15 demonstrates the effectiveness of the BDI rules, in this instance without the BDI rule and using the appliance profile given in DesignBuilder, where from 7am till 8pm the computer is on; there are times when the stochastic agent is present but not consuming electricity. Although this may be the case, the BDI rule mimics the stochastic presence profile in a more realistic fashion. It is unlikely that a computer will use this constant demand throughout the day. With BDI rules it would be possible to build in a computer's energy saving profile, for example if an occupant is not present and 20 minutes has passed the computers power state could be reduced, effectively putting the computer into standby mode.



(A) Heating demands (B) Cooling Demands

FIGURE 5.14: Box plots of demands due to electrical appliance rules



(A) Without BDI rule (B) With BDI rule

FIGURE 5.15: Stochastic occupancy profile (dashed line) with electrical demand from computer(s) (solid line)

5.4 Conclusion

This chapter presents a methodology for including rule based models within No-MASS, for testing what-if scenarios with building performance simulation and providing a pragmatic basis for the modelling of the behaviours which there is insufficient data to develop stochastic models. A BDI methodology is used to develop a set of goals and plans that an agent must follow to influence the environment, given beliefs about current environmental conditions. These plans are implemented within No-MASS and tested to examine what effect the consideration of each individual plan has on the overall building performance predictions.

In the case of the privacy during the evening, the high t values in this case exemplify the strong influence this has on building performance when executed just 1% of the time (see Table 5.5 Goal 5.2.9). The other goal sets of note are the window models, which influence heating and cooling demand significantly even when only performed for a small percentage of the time. These results suggest the modelling of these behaviours should be explored further, through the collection of empirical data and the future development of validated models simulating them. With the exception of sleeping with the shade closed our results suggest that models of the other behviours should also be further studied, with the most behaviours only needing to take place less than four percent of the time to impact performance.

The office simulation results show the need of BDI rules when linked with stochastic models, as the generic rules used within EnergyPlus fail to address the stochastic presence of occupants. The electrical demand profiles fit the occupants much better when they are defined in terms of presence rather than on a schedule; this would also be the case for other types of appliances in offices. Other demands can be predicted in this way, for example modelling appliance transition to standby mode and water usage based either on presence in offices or activity in houses.

A limitation of the BDI rules implemented are that they are not based on empirical data and are therefore not a true reflection of reality. However they do give more scope in terms of the interactions taking place in response to the empirically driven stochastic models. For example within EnergyPlus it is not currently possible to set the window to open when someone has awoken based on the activity model within No-MASS. Another limitation is that large complex unwieldy models will increase simulation time and without empirical data wont be a true reflection of reality. It is therefore important to consider the trade off between what needs to be included for valuable results against the impact on simulation time.

The BDI methodology is an effective for testing what-if scenarios and should be used to ensure that occupant interactions have a significant effect on building performance before spending resources collecting data. It can be used to prove the need for empirical models, that reduce the gap between simulated and real world results.

Chapter 6

Social Interactions

An agent should be autonomous, have social ability, perceive and react to the environment and be proactive with their choices (Wooldridge and Jennings, 1995). So far No-MASS agents can perceive the environment and react to changes within it. They are autonomous and self interested, however a mechanism for handling conflicts between agents needs to be integrated. Occupant behaviours where there are no empirically validate models due to problems with access to data and thus our ability to reduce the performance gap between observed and simulated energy demands. One of these models is that of the social interactions between agents, conflicts are arise due to archetypal behaviours. Different types of occupants will want the windows, blinds or shades open during different conditions. These archetypical behaviours are not covered by the deterministic models and the stochastic models currently included within No-MASS do not consider social interactions. To resolve conflicts such as when to open a window, discussions will take place between the stakeholder parties until a resolution occurs.

6.1 Theory

Game theory has been used to explore different scenarios where different forces interact. In economic theory these forces can be market forces but the interactions can also be between sets of people. One early example is Nash Jr (1950), where a game with a finite set of actions must have an optimal strategy (the Nash equilibrium) for actions if the other players' set of actions are known and are not dynamic. Agent simulation has been used to model interactions between individuals to solve game theory scenarios such as the prisoner's dilemma scenario, a game to study whether you should cooperate or defect against your comrade. If you both defect you both lose, if you cooperate you both win, but if one defects and the other cooperates the defector wins. Here Axelrod and Dion (1988) simulated an iterative version of the prisoner's dilemma and found that the optimal simulation model configurations were the ones that employed a *tit-for-tat* strategy. Axelrod later went on to test this with agents using a genetic algorithm (Axelrod, 1997); finding that 95% of all populations evolved towards the optimal *tit-for-tat* strategy. The problem with the optimal *tit-for-tat* is that the decisions do not incorporate fairness into the decision making process. In response, Rabin et al. (1993) created a framework to include fairness into game theory decision making, where a player's payoff would be derived not just from their actions but also from their beliefs about the situation. For example, if a player thought that their opponent was going to defect they would be willing to sacrifice their reward to hurt the other player.

Computers have been used to simulate group decision making since the 1960's. Abelson and Bernstein (1963) worked on the computer simulation of a community referendum. They were able to run a number of different configurations of their simulation model to study differing assumptions and conditions. Computerised individuals were subject to channels (advertisements) and conversations which would influence the way in which they would vote. The individuals could influence others in their proximity based on their idealogical understanding. This understanding was based on what they had learnt during the previous cycle. The level an individual could be influenced was dependent on a set of predefined rules. Abelson and Bernstein (1963) ran scenarios to examine their effect on the results, ie. what if a mayor was introduced with more influence towards one side, what would happen if this mayor did something controversial, etc.

Researchers in the field of agent based social simulation have used a cognitive theory to model people, a classic example is the unified theories of cognition. Newell (1994) developed unified theories of cognition into an example framework, describing the immediate process of cognition and learning. This has been used as a method to create the Soar agent system (Wray and Jones, 2006), within which there are three levels; the knowledge level (a descriptive view of the agent's understanding), the symbol level (the representation of that knowledge), and finally the architecture (the fixed mechanisms that define the ways in which knowledge is accessed and acted on). The Soar agent system has been extended to include interactions between agents through STEAM (Tambe, 1997). STEAM uses the understanding of joint intentions (Cohen and Levesque, 1991) where agents who have a joint understanding of their current state and a shared goal can perform an action as a group; either one agent performs it or they perform sequential actions to achieve the goal. This model must adhere to the following requirements: the agents must have a joint goal, must be willing to co-operate, to agree on how to achieve the goal, and the agents must understand the viability of their actions (Jennings, 1995). The joint intentions approach has a weakness though. When an agent commits to a common goal, what happens when something goes wrong? Do all the agents continue on the same agreed path, even though the goal will not be achieved? What happens if one agent believes the goal has been met or will never be met? In an attempt to resolve this Jennings (1995) built into the model a concept of joint responsibility, requiring all agents to share why they have a lack of commitment to the persistent goal. This allows the other agents to assess the situation and decide on remedial actions. Related to the joint intentions method of collaboration is the SharedPlan approach (Grosz and Kraus, 1996). Agents using the shared plan have their own beliefs, intentions and goals. Any intentions that may affect other agents need to be communicated, and these intentions are then meshed into plans of action. This method allows for agents having partial information about another agent's understanding, but they must have sufficient information to allow consensus. Within the SharedPlan formalisation agents cannot hold intentions that conflict with each other.

The joint responsibility and shared plan methods assume there are common goals between agents or common actions that can be achieved. However, this is not always the case. There are times when an agents' goals are in direct conflict with those of another. In this case mediation is required to allow them to move towards an optimal solution for both parties. The PERSUADER program is a system that focuses on resolving labour disputes (Sycara, 1988). It works through proposals and modification of goals. Goal trade-offs are searched allowing the PERSUADER to make novel proposals. Agents continually re-assess their beliefs, so that what may not have been acceptable at one point may be acceptable later. An initial plan is made, and evaluated against previous plans. If it is acceptable it is proposed to the agents, who either agree or disagree. Disagreement allows for discussion between agents, then either modification of the plan or the process starts again with a new plan created. The argument stage works by changing the agents' belief structure. For example, with a wage dispute a company may request a lower wage for the union, this is against their goals, PERSUADER checks the possible alternatives, finding that unemployment will achieve the same goal for the company. This approach is worse for the union and the PERSUADER program highlights this to the union allowing them to reassess.

Over time the goals of an agent may change. Kraus et al. (1995) builds a time restraint into the decision making process, where resources are valuable when there is a disagreement and each agent has a period of time before the resource is no longer useful. The time constraints alter the decision being reached and can often determine the agreement. For example, if two agents want access to a resource within a period of time, they can negotiate an agreement based on the value of the resource and the cost of deliberation. Davis and Smith (1983) developed a method of negotiation to resolve distributed problem solving.

Called the contract net, agents bid for tasks agreeing that they will complete one for a given price and if a contract is agreed for the winner. The framework then distributes the tasks to the winner of the contract. This has been developed into a multi-agent system called TRACONET, which allows agents to announce bids for an available task, the lowest bid over a period of time requires that an agent is committed to completing the task (Sandholm, 1993). An agent calculates their bid based on a local calculation of the cost of performing the task; they combine multiple tasks to minimise their costs. The TRACONET system was developed for transportation agents, with the example of delivery companies. The cost of the task would be the time to make a delivery and the deliveries are chained to ensure they end near the start of the next delivery. TRACONET was further developed allowing agents to assign different levels of commitment to each task (Sandholm and Lesser, 1995); now agents can back out of a task for a cost if they see that another agent is committed to the task and they believe that they can get a better deal elsewhere.

Agents that can learn another's preferences have been demonstrated to find an optimal solution to conflict resolutions. In the example of a Bazaar, agents negotiated prices for products, moving towards a price they would sell or buy at (Zeng and Sycara, 1997). Both agents have conflicting goals but, if they had a price point that was acceptable to them both, they could eventually agree. If the buyer and seller could learn the other's preferences they came to an agreed price quicker than one that could not learn the other's preferences. Shoham and Tennenholtz (1992) presents a method of conflict resolution called social laws; constraints that should be built into the actions of the agent. They avoid conflicts and the unnecessary negotiations by removing conflicts from the system before a simulation takes place. An agent has a choice of actions that can be performed at a given state, but before taking an action the agent must consider a set of social laws. These define which actions the agent can take to ensure that no conflicts occur. The example used by Shoham and Tennenholtz (1992) is of a car turning a corner, the car can only turn if the driver understands the lights are green and no other car is in the location that the car will occupy once around the corner. This approach assumes all agents have the same set of social rules.

The modelling of multi-agent group decisions is either centralised or decentralised. In the centralised model an agent has a complete understanding of the environment and a decision is made to maximise global utility, ie. to keep everyone happy. In the decentralised model each agent communicates, gains an understanding of the environment from their perspective and makes a decision to maximise their local utility. One method of forcing decentralised communication is by adding cost to any communication taking place. To model a multi agent decision process between distributed systems Xuan et al. (2001)

developed a decision theoretic framework. This relied on each agent having the same goal to maximise the global utility. Each agent would have an incomplete understanding of the environment, the agents would only communicate up to the point where the reward would outweigh the cost of communication. This means that an agent's beliefs about what action to take to maximise utility can be changed through the communication process.

The Clarke Tax (Clarke, 1971) has been used to allow agents to come to a consensus and remove the *free rider* problem, where an agent may lie about the benefit of a resource to gain more at a discount. In Ephrati et al's 1991 system agents declare their preference for reaching a state from amongst a set of states. If their preferences are the deciding factor between the voters they are taxed the difference between the sum of preference of the first and second choice of states. This forces an agent into telling the truth of the true value of their vote. If they do not and they win through a higher vote they will be taxed higher for their choice. This method has its drawbacks, it does not consider how the resources are divided between the winning agents of the vote; on the other hand the voting strategy is a quick and effective method of arriving at a consensus.

As outlined many of the methods that are used to allow agents to reach a goal, either in cooperation or in conflict. Within No-MASS a simplified variation of the the voting mechanism to overcome conflict (Ephrati et al., 1991) is proposed. Then once the agents are in agreement over which action to take, No-MASS use's social laws (Shoham and Tennenholtz, 1992) to decide how the action will take place. For example, agents will vote on whether they want to open the blind and once votes are cast, they decide how far the blinds will be opened.

6.2 Implementation

Social interactions are an important consideration when modelling occupants. Decisions such as opening windows and interacting with shading devices are often discussed between occupants before an interaction is performed. The current convention of building performance simulation tools is to assume that all occupants within a building interact with the environment at a given set point of a particular variable, i.e. indoor temperature for windows and internal illuminance for shades. Therefore, there is no need for a conflict resolution mechanism. But in reality occupants have different beliefs and desires about how they wish the environment to be. Group interactions with the environment are often achieved through group mediation, with occupants voicing their concerns about their discomfort. No-MASS agents are self interested. They act based on their beliefs about their current state of discomfort. Given a set of decisions each agent decides if they would like to open or close the window to minimise their discomfort. An agent does not consider the state of the other agents at this point, if they intend to open a window they must negotiate with the group. But one agent's intent maybe in direct conflict with an other agent's goals and desires. A notable solution to conflict resolution would be a voting system such as the one employed by Ephrati et al. (1991), which has be demonstrated to be a quick and effective method for arriving at a consensus. However, as No-MASS agents behave rationally based on the outputs of the stochastic models our agents do not suffer from the *free rider* problem, Ephrati's system incorporates a tax system on choice to overcome this.

Occupants may have differing authority to make choices about the environment. Within No-MASS constraints can be placed on the actions that can be performed during a conflict; this is achieved through a biased voting system. Some agents can have larger voting rights than others, these voting rights are called *power* within No-MASS, these are social laws built directly into the actions. To demonstrate the mechanism three classes of group interaction have been chosen; these mimic possible scenarios in the real world; democratic, biased and authoritarian. Each will be explained and with a demonstration given of how they can be handled with this voting system.

Agent	Agent 1	Agent 2	Agent 3	Agent 4
Democratic	0.25	0.25	0.25	0.25
Biased	0.18	0.18	0.18	0.46
Authoritarian	0	0	0	1

Table 6.1: Agent Voting Power

Our agents and voting weights are given in Table 6.1 for scenarios with a four member group. The first scenario is the democratic environment in which each agent has equal voting power for the interactions that they wish to perform. If a single agent wishes to open a window and another two are present in the zone then the other two can choose to either side with this agent given their personal preferences, or they can veto the window opening. In this case there will need to be at least two agent suggesting the window stays closed to win the vote. In cases where the agents votes are tied a virtual coin toss is performed to decide the outcome. A random number is drawn, if it is above 0.5 then one action is performed, if not then the other choice is performed.

In the second scenario, the biased approach, one agent will have the majority of the voting power as may be found in hierarchical organisations where a group goes along with a supervisor's preferences. In most cases the supervisor's vote will win, however if the other agents disagree they can pool their voting resources together to veto the action.

Finally there is the authoritarian scenario where if present one agent has all the power, the others can perform actions when the leader is not present, but they do not have the ability to override the agent.

This approach works well in binary cases where the actions are on/ off, such as for the lighting model. However models such as the external shading interaction model predict a shade opening fraction. An agent has a choice of actions that they can take to either raise, lower or keep the shades as they are. Given a set of agents, one could choose to raise the shade to a percentage and the other lower the shade. In the first instance the voting mechanism can be used; agents can vote to raise, lower or do nothing to the shading device. To determine the percentage change that occurs, social laws (Shoham and Tennenholtz, 1992) are enforced on the agent, removing the need for time consuming negotiation. A set of two agents choose to raise the shade from its current position but they both choose to open it to different levels. Here a restriction is imposed on the agents that they must choose the average of the two. This will satisfy the agents' need to raise the blind and allows the simulation to move on (however it may please neither agent). Within No-MASS agents assess their personal preferences at each timestep for all the stochastic models, the agents will therefore have to resolve conflicts at each timestep. This methodology of processing votes does not increase simulation time significantly and provides a first instance of agent negotiation within buildings; the effects of which will be discussed later in this chapter.

6.3 Social Interactions And Building Performance

To examine the effectiveness of No-MASS's simple social interactions framework, a residential building and a non-residential building in Geneva is simulated, as described in Chapter 4. The residential building is a hypothetical house and the office is a simple shoe box design. Unless simulated by No-MASS through stochastic models all building characteristics are chosen from templates in DesignBuilder that fit the activity taking place in a given zone. For all simulations a timestep length of five minutes was chosen. For simulations testing the social interaction framework the three scenarios already mentioned are considered, authoritarian, biased and democratic. Each scenario is tested with sets of agents of sizes between one and four.

Group decisions between agents implies that there are differences between what occupants desire. No-MASS uses the window model taken from Haldi and Robinson (2009) which includes coefficients for an aggregate model of a population. Using the aggregate view used in Chapter 4, the window opening model would not show variation between occupant archetypes. However Haldi (2010) presents coefficients estimated for a range of individual occupants. Integrating these within No-MASS so they can be chosen at run time allows for the testing of different occupants' window opening behaviours. It is then possible to test the effectiveness of our social interaction model against these diverse behaviours. A box plot presenting the simulated heating demand of each agent window opening model is given in Figure 6.1. Each simulation consisted of a single person in the non-residential building, meaning that there are no social interactions taking place. The variation in results highlights the effects that different occupants could have on a building. There is a difference of $26kWh/m^2$ between the largest median heating energy demand (model 13) and the smallest (model 18). The social interaction framework within No-MASS is used to demonstrate the effects of social interactions between occupants each allocated their own behaviour model. One hundred replicates are performed. At the start of each replicate agents' window models are randomised, testing our case study models against many possible combinations of occupant. Each agent is given a voting power corresponding to the simulation scenario; authoritarian, biased and democratic. The scenarios are repeated up to four times each, with a greater number of occupants occupying the space.

With two agents there is a difference of $4kWh/m^2$ for the median heating demand between the two scenarios, with the democratic approach the larger of the two. This could be due to some indecision between the agents, allowing the windows and shades to be left on longer over the course of the year. Three agents reduce the variation between scenarios to $2kWh/m^2$, the democratic ($84.9kWh/m^2$) heating demand is again the highest, the authoritarian ($84.6kWh/m^2$) is almost equal. The biased ($82.8kWh/m^2$) scenario is the lowest. With four agents in the office there is a difference of $2kWh/m^2$, similar to the three agent scenario. The heating demands for each scenario are; authoritarian ($80.7kWh/m^2$), biased ($79.8kWh/m^2$) and for the democratic ($78.2kWh/m^2$). In this scenario the democratic result has the smallest heating demand, possibly due to indecision amongst the four agents.

Taking the extreme models from Figure 6.1 and assigning these to two agents. The extreme agents with the model of 18 ($81kWh/m^2$) negotiates a reduced window opening and therefore lower heating demand than that of model 13 ($109kWh/m^2$). Where model 18 has the greater voting power the median demand ($93.8kWh/m^2$) is slightly reduced than if model 13 has the greater power ($96.4kWh/m^2$). With equal voting power the result almost matches the scenario where 13 has the higher power ($96.7kWh/m^2$). When the agents have equal power the heating demand tends towards that for the occupant that has the more extreme behaviour: with greater interaction frequency being preferred they still exert a greater influence than their counterpart. In both, the random model assignment and the mediation between the extreme cases there is not a large variation. The agents successfully negotiate window opening behaviours that represent a compromise between the two cases. The authoritarian scenarios do move the results towards the more powerful agent models, however because of interactions taking place when these agents are not present their impacts are reduced. Consider that when a powerful agent who prefers it cooler leaves the room with the window open, the other agent(s) who prefer it warmer close the window (while it is cooler outside) at the next timestep. This reduces the overall heating demand to a choice that is between the two extremes.

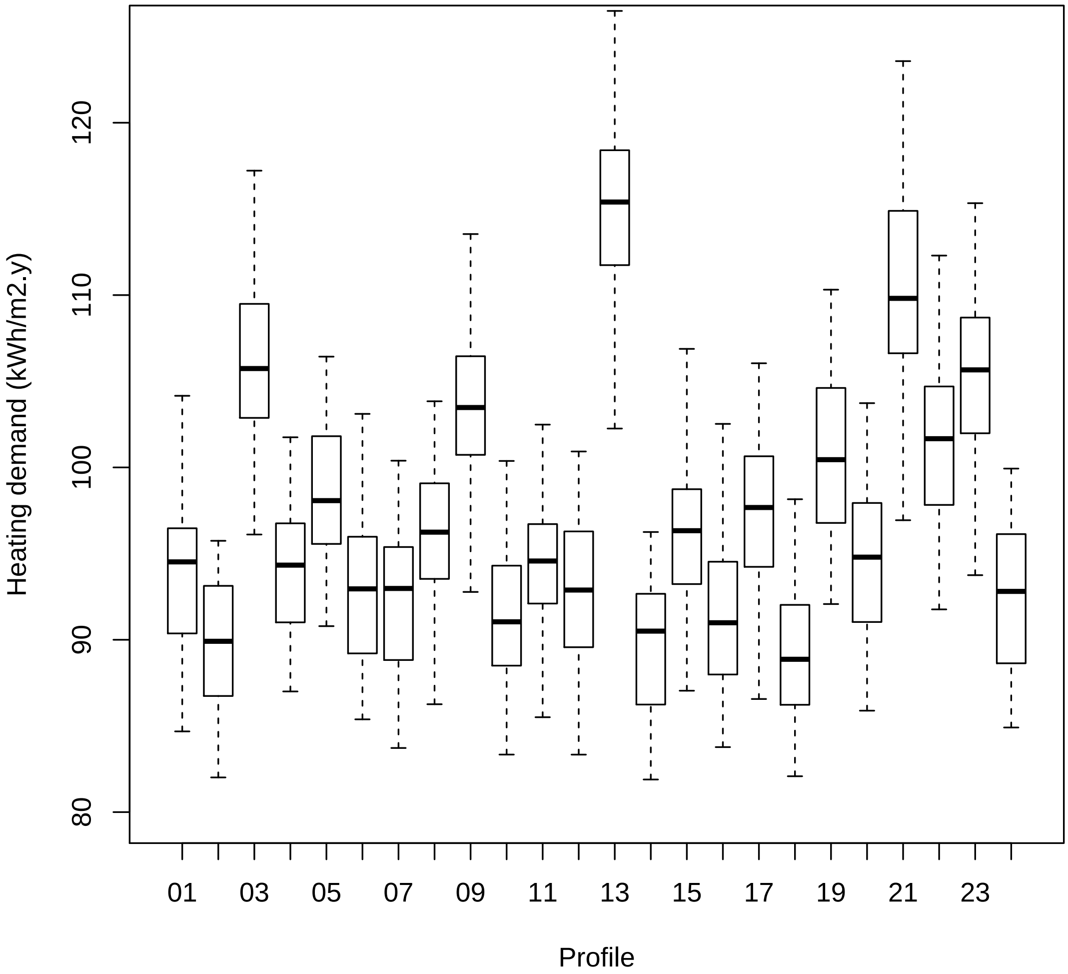
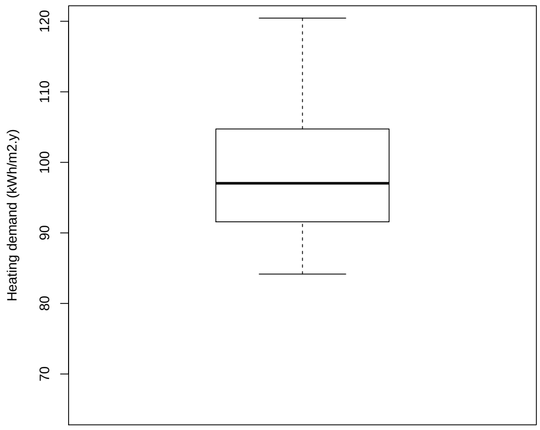
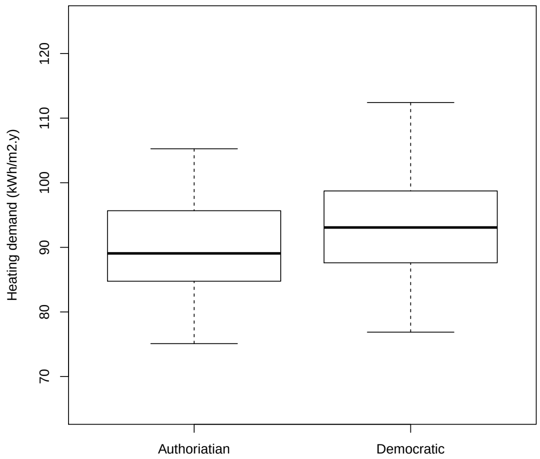


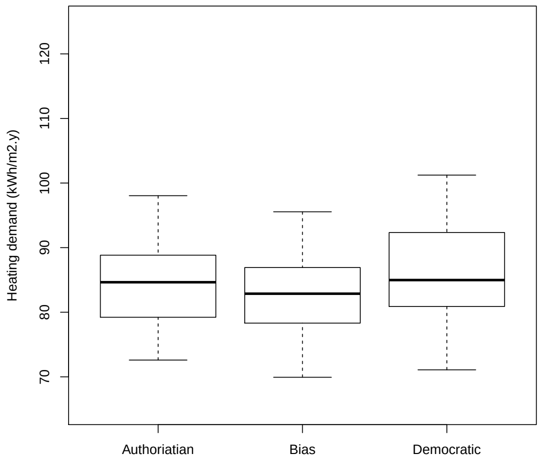
FIGURE 6.1: Heating demand in a non-residential for each individual window opening model with coefficients estimated for a range of individual occupants (Haldi, 2010) and assigned to an agent profile



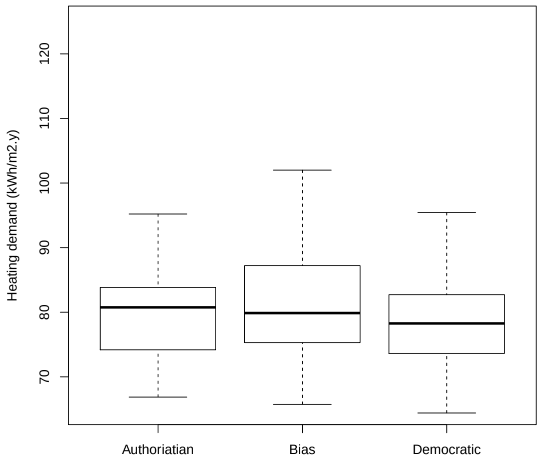
(A) One Agent



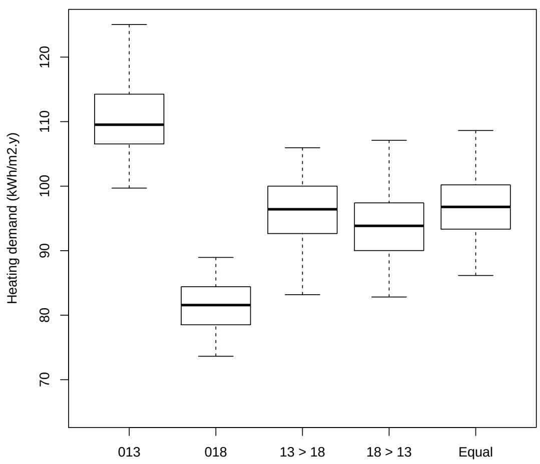
(B) Two Agents



(C) Three Agents



(D) Four Agents



(E) Heating demand for extreme cases in the office, all scenarios have two agents, left to right; model 13 equal power, model 18 equal power, model 13 has greater power than 18, 18 has greater power than 13 and both 13 and 18 have equal power.

For the residential building the same window models are assigned to a single occupant occupying the building. The results for each model are given in table 6.3. There is a smaller spread in the results as expected in the house (as noted in the previous chapter, the effect of occupants' interactions are dampened by the larger building volume). As with non-residential office the largest heating demand arises from the agent that is allocated model 13; on the other hand model 18 is no longer the smallest, model 14 is now marginally lower.

The random assignment of models to the three scenarios yields behaviours similar to those of the non-residential building, except in a few key areas. First with the increasing number of occupants the authoritarian variation is greater than that of the biased and democratic results; due to the extreme occupants having the power to execute their beliefs and dismiss the results of the other agents. In the fourth occupant scenario the median heating demand are almost equal; authoritarian ($45.4kWh/m^2$), bias ($45.1kWh/m^2$) and democratic ($45.2kWh/m^2$). Comparing extremes cases of 13 and 14, the results are again very similar, between the two cases. Once again performance under the equal power scenario tends towards that of the occupant with the larger heating demand.

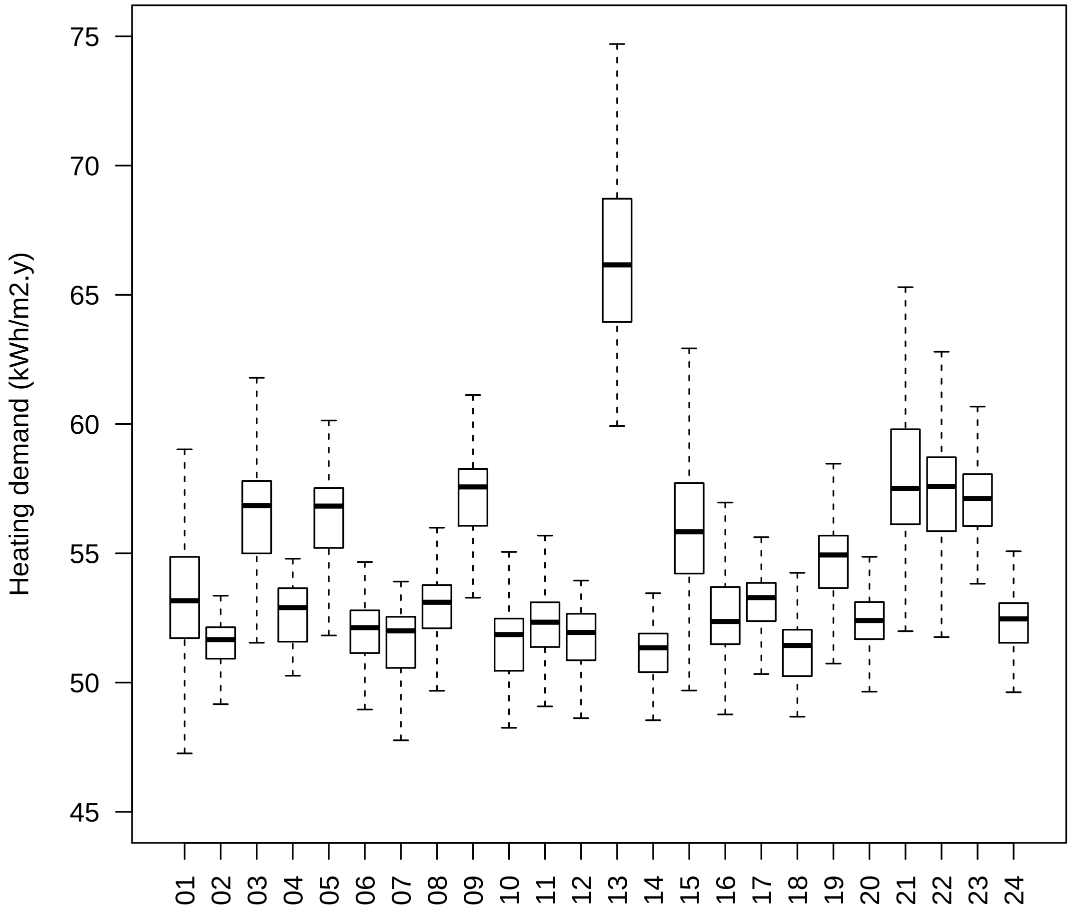
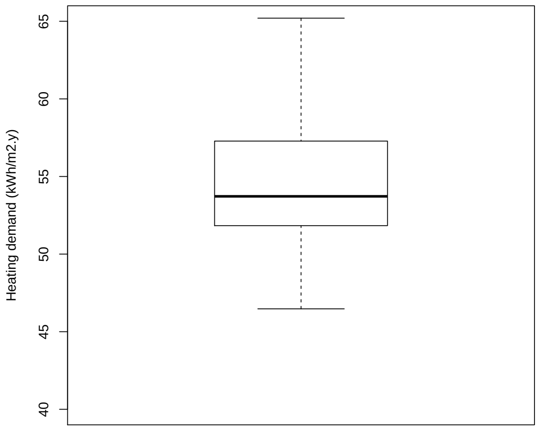
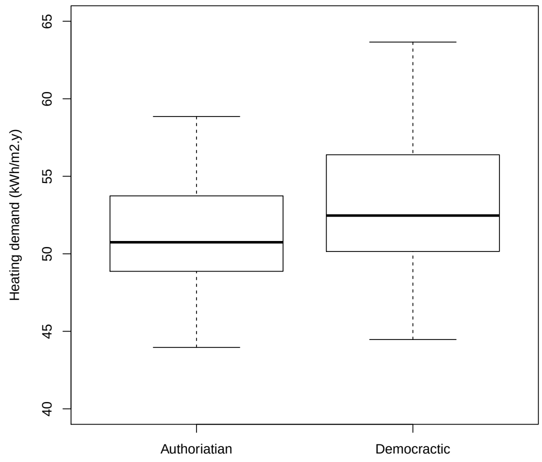


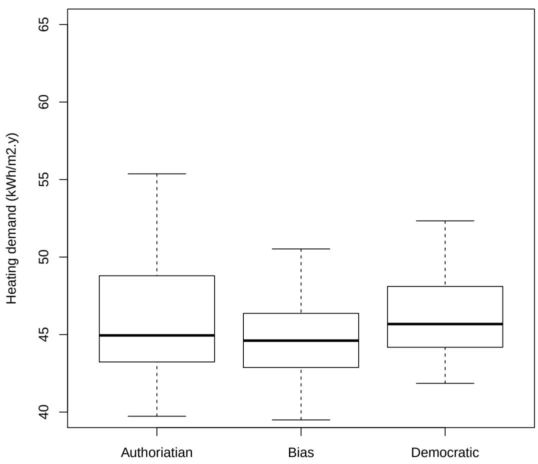
Figure 6.3: Heating demand in a residential for each individual window opening model with coefficients estimated for a range of individual occupants (Haldi, 2010) and assigned to an agent profile



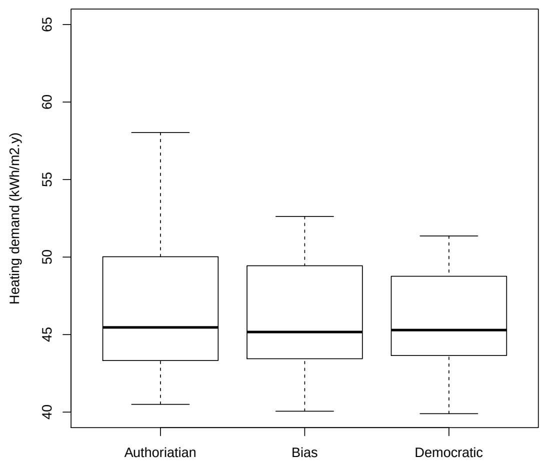
(A) One Agent



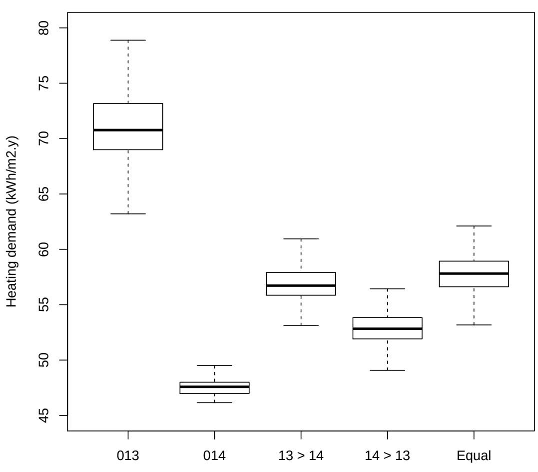
(B) Two Agents



(C) Three Agents



(D) Four Agents



(E) Heating demand for extreme cases in the office, all scenarios have two agents, left to right; model 14 equal power, model 13 equal power, model 14 has greater power than 13, 13 has greater power than 14 and both 14 and 13 have equal power.

6.4 Conclusion

This chapter has demonstrated a method for solving conflicting agent goals through the use of a vote casting system, where each agent is given a voting power that they can use to request their desired state of the environment. The social interaction model appears to perform effectively in mediating the different agent preferences converging to a median behaviour and corresponding energy performance that is between the two. No-MASS can now simulate populations of competing occupants in a number of different scenarios. In the scenarios tested, no major differences are predicted, but this may not be the case if the agent population was scaled up; particularly in a large open plan office environment where there may be many competing desires.

By allocating unique stochastic models to individual agents rather than using only aggregated models as in Chapter 4, it is now possible to assign individual models to each occupant for the activity model, window model, lighting model and the presence model. It is also possible to define for each occupant the percentage of time each occupant uses a BDI rule. The social interaction framework handles the interactions between occupants, either allowing them to agree on a desired outcome or to resolve any conflicts.

However this social interaction framework is not based on empirical evidence. To this end a dedicated field survey would be useful, to better understand group decision making dynamics to regulate the indoor environment and to encode this understanding into No-MASS. If possible this field study should be culturally diverse, the social interactions taking place in a western country for example may be different to those of an eastern country.

Chapter 7

Reinforcement Learning

As previously noted developing and rigorously validating stochastic models of occupants bahaviours is data intensive. Although a BDI framework is a useful companion to such models in the absence of the data, its use should be restricted to relatively straight forward types of interaction. For more complex interactions, agent learning is a promising alternative whereby agents to learn from past experiences to take actions in the present that will effect their comfort in the future. Reinforcement learning is related to dynamic programming, especially where the problems are defined as Markovian Decision Processes (MDPs) (Sutton and Barto, 1998). Dynamic programming requires breaking a problem into sub problems and saving the results. Later, when the same sub problem arrives, rather than calculate the result the saved value is retrieved. If these problems can be resolved recursively and if there is a link between the result of the total problem and that of the sub problems, the function is known as the Bellman equation (Bellman, 1957). MDPs and dynamic programming allowed researchers to study many optimisation problems (Howard, 1960), where given a state s, an agent may choose an action a, from a set of actions and at the next state s_1, the agent will receive a reward. This allows the agent to take an action and then make an assessment on the performance of the action at the next timestep based on the received reward.

7.1 Theory

Reinforcement learning is a trial and error strategy, where an agent in a given state, takes an action based on its expected reward, arrives at the next state and updates the reward of the previous state. The reward can be negative if the result is not desired or positive if it is desired (Minsky, 1961). There are many algorithms that use this methodology, one of which is a *temporal-difference* learning algorithm called $TD(\lambda)$ (Sutton, 1988). $TD(\lambda)$ looks at multi-step prediction problems where the predictions over a number of steps predict a final outcome. At each step it is possible to update predictions regarding the final outcome at the end of the episode, calculated from the difference between the reward expected and the reward received. Another is the is Q-learning algorithm developed by Watkins (1989), where an agent will choose an action in a given state based on a Q quantity, which is a weighted reward based on the expected highest long term reward. This method is considered an *Off-Policy* method, meaning its Q values are updated assuming that the best action was chosen, even if the best action was not chosen. Another *temporal-difference* learning algorithm is called SARSA. This method is an *On-Policy* version of the Q-learning algorithm (Rummery and Niranjan, 1994). Here, instead of updating the Q quantity based on the optimal greedy policy at the previous state, the Q value is updated based on the current Q value; meaning that the agent continues along the same policy. SARSA tends to learn the safer policy, where there is a lower risk (least negative reward), whereas Q-learning learns the optimal policy, the path to the highest reward. R-Learning is another *On-Policy* method (Schwartz, 1993). Where instead of considering the total reward, the goal is to maximise the timestep reward. It adapts the Q-learning method, but instead of discounting the current Q value based on future rewards it takes the average of the current reward and the previous maximum Q value as approximators. Schwartz (1993) argues that this allows the algorithm to converge faster that the Q-learning method for some cases. Alternatively Tesauro (1995) used a neural network which trained itself, playing the board game Backgammon to expert level. A neural network consists of a number of networked nodes that take an input, in this case the position of the board, and predicts the next best move. This works through a number of hidden weighted network nodes that can be understood as generic non-linear function approximator. The weights are updated based on the received reward with a back propagation algorithm. The neural network methodology has an advantage over Q-learning and other methods, as it can learn much more complex interactions, since Q-learning is limited by the table space of the Q table state/action predictions. To overcome this issue, Mnih et al. (2015) developed what they call a deep Q-network (DQN) where many layers of nodes are used to approximate the Q-value for a given state and action. This method has allowed agents to learn how to play many classic computer games at human level or above. These learning algorithms have been used extensively in the agent simulation literature. For a review of the different modelling multi agent learning methods please see Babuška et al. (2008).

The Q-learning algorithm has been applied to agents to test the effects of agents sharing either sensory data or learnt policies based on the learning rate (Tan, 1993). Using a predator-prey scenario the agents that shared their experiences learnt faster than independent agents, they were able to catch their prey in few steps, but they had the extra cost of the communication and required a larger table space. Nolfi et al. (1994) used a neural network coupled with a genetic algorithm to allow agents to walk across a grid collecting food. Instead of supplying a reward at each timestep at each simulation run or episode the best agent (the one that had collected the most food) would be chosen as the base agent for the next episode. This allowed the agents to evolve naturally towards agents that collected the most food. An altered version of the Q-learning algorithm was used to simulate elevator performance (Crites and Barto, 1996). Here, peak rush hour times were modelled, where each agent was subject to the random arrival times of people in a continue state space. Crites and Barto (1996) found that this algorithm was able to learn the optimal solution to reducing the square wait time of a person. It out performed the other state of the art elevator algorithms at the time. Claus and Boutilier (1998) developed agents that considered the joint actions of multiple other agents in a co-operative Q-learning strategy but found that it did not guarantee convergence towards an optimum in their test cases; but they found that with care of design it was possible in some simpler cases. It is also important to consider competing learning agents as with Littman (1994), who found that the Q-learning algorithm was a poor choice for the zero sum Markov game modelled after soccer. This is because Q-learning finds deterministic policies and every move in the game had a perfect defence. Littman (1994) suggests a minimax-Q algorithm for probabilistic situations. Agents using difference approaches were found to converge towards different polices while playing an iterative version of the prisoner's dilemma. Q-learning agents were found to learn more co-operative strategies over neural network learners that learned non-cooperative strategies (Sandholm and Crites, 1996).

7.2 Implementation

Q-learning (Watkins and Dayan, 1992) allows agents to learn a response from a reward, to an action. This allows agents to develop an understanding of their preferences over time. In a Markovian domain an agent learns the best action in a given state, this is achieved by trying every action in a state and updating the expected reward with the actual reward for that action. This can be computationally expensive, however it is useful in building performance simulation as the same methodology can be applied to areas where models are missing due to lack of data and where there is a clear link between an action and a driving stimulus. For example, does a chosen action cause comfort or discomfort, increase the reward if comfort is maintained and decrease if not. The Q Value at state s_t for action a_t is given by the function:

$$Q_t(s_t, a_t) = Q_t(s_t, a_t) + \alpha(r + \gamma * max(Q(s_{t+1}, a)) - Q_t(s_t, a_t)) \tag{7.2.1}$$

Where the reward is r, the learning rate, α, is $0 < \alpha \leqslant 1$ and the discount Factor, γ is $0 < \gamma \leqslant 1$. The discount factor specifies how soon the agent cares about the reward, ie. if an agent is myopic, they care about the near term gains, otherwise the agent would prefer long term rewards. Long term rewards are set closer to 1. The weakness of this approach is that if there are a large number of states and actions then the size of the table space is large, meaning that the learning process could become long or the model will not converge. The table space is the mapping of state to action. Consider indoor temperature to a heating setpoint, at each possible indoor temperature state there is associated heating setpoint action. Each indoor temperature from 0 - 30°C would have a heating setpoint action from 0 - 30°C that could be taken. The action chosen is the cell with the highest value, see Table 7.1. A large table space would make it difficult for an agent to learn and assess all possible states and corresponding actions. Avoiding this it may be better to replace the table with a Neural Network as a function approximator as in Tesauro (1995). However Q-learning is a tried and tested method for single agents who learn by themselves, and it converges towards an optimal policy making it a suitable choice for No-MASS. A large table space can be avoided in most spaces, removing the main weakness of the model.

Indoor Temperature °C	Heating Setpoint Action °C				
	...	20	21	22	...
...	...	...	...	...	...
10	...	0.1	0.2	0.1	...
11*	...	0.1	0.4	0.1	...
12	...	0.1	0.2	0.1	...
...	...	...	...	...	...

TABLE 7.1: Example Q-table table space, (*) indicates current state at 11°C where the largest value is an action of 21°C.

Heating Set Points

Q-learning requires a map from state to action. In building simulation heating setpoints (for which there is a lack of high quality longitudinal data) are time based allowing a different setpoint to be set for the time of day. This gives the model its first constraint, the timestep intervals within building simulation. EnergyPlus simulation timesteps are often set to sub hourly intervals, however the table space would be large if the agents learnt the heating setpoint for every time step. Since Q-learning requires that each action in each state is evaluated, this would make the learning process computationally expensive. To keep the table space small a second constraint is placed on the model, limiting the learnt states to hourly time intervals. The set of states has now been defined, the hours of the day, but not every day is the same; the heating set points for the working week may differ from those of the weekend. Rather than learn the optimal action for everyday of the year, No-MASS agents learn for weekdays and weekends. The final constraint placed on the states is that heating demand is seasonal, it changes over time based on the season. To overcome this issue No-MASS agents are set to learn the best action for each month. No-MASS Q-learning states are therefore the hours of a working weekday and the weekend for each month. No-MASS now has a set of states that an agent can be in at a given point in time, next a set of actions that an agent can perform at each state. This is the heating setpoint, which are constrained to be between the heating setback temperature and, if relevant, the cooling setpoint temperature.

With the states and actions defined a method of rewarding or punishing the agents when they perform an action needs to be considered. As the heating set point is linked to an agent's comfort, the sensible solution is to reward an agent based on the outputs of a thermal comfort model; thus allowing them to learn the heating setpoint values that minimise discomfort at a given point in time. No-MASS currently has a mechanism to calculates the agent metabolic gains based on ISO:77302005, which also provides the source code to calculate the Predicted Mean Vote (PMV) of each agent, a thermal sensation scale from -3 to 3: cold (-3), cool (-2), slightly cool (-1), neutral (0), slightly warm (+1), warm (+2) and hot (+3). Our agents must also aim to be efficient and not wasteful, agent should be punished if PMV values are above 0. Conversely if the agent is too cold $PMV < -0.5$, the agent is punished once again for having selected a discomforting strategy, so that a $-0.5 < PMV <= 0$ obtains a high reward. It is necessary to restrict the heating while the agent is absent, otherwise an optimal learning policy would be to wastefully leave the heating on to maintain a temperature that is satisfactory at all times. To overcome this agents are punished if the heating is above the setback temperature for more than an hour while the agent is not present. Thus our reward function is:

$$r = z * (1 * c + 0.1 * a - 0.1 * b - 0.1 * e) + (1 - z) * (d * 0.1 + (1 - d) * -0.1) \tag{7.2.2}$$

a, b, c, d, e and z are binary operators where a is 1 when $pmv > 0$ and the heating setpoint equals the heating setback, b is 1 when $pmv < -0.5$, c is 1 when $pmv >= -0.5$ and $pmv <= 0$ and d is 1 when the heating setpoint equals the heating setback and the absence length > 1 hour. e is 1 when $pmv > 0$ and the heating setpoint temperature is greater than the setback temperature.

For the Q-learning equation (7.2.1) our learning rate α is set to 0.1 as the environment is non deterministic, as suggested by Sutton and Barto (1998). The discount Factor γ is 0.1, making the agent short sighted, so that they prefer a rewards that are short term for discomfort to be reduced in the near future. This is not unlike reality; occupants want to be warm in the near term if they are cold. However an occupant may be long-sighted and pre-heat an environment, this is accounted for by the d parameter in the reward function.

The Q-learning method described so far has no exploration over time; choosing a methodology of $\epsilon - greedy$ which enables the agents to explore the parameter space. An $\epsilon - greedy$ policy is the chance of random action taking place over the optimal, Sutton and Barto (1998) suggest that a value of 0.01 is slower at learning random actions taken but converges to a better policy than $\epsilon = 0.1$. Random actions are taken on based to ensure that as the agent learns all possible actions are tested over time; what may have been optimal at the start of the learning period may not be at the end. For No-MASS's heating setpoint learning a value for ϵ of 0.01 is set, meaning that 1% of the actions taking place are a random action.

Window Opening Setpoints

A second approach to test its effectiveness would be to compare it to an existing stochastic simulation method. Comparing the stochastic window model to one already integrated within No-MASS is an effective way of evaluating how effective it will be in other situations. Learning window opening setpoints is more complex than learning heating setpoints. First the window opening as highlighted in Haldi and Robinson (2009) has more significant inputs than just the operative temperature that is used to determine the heating setpoints. Using all the inputs of the stochastic window model would make the number of states an agent could be in too large for effective convergence. Choosing the internal temperature to begin with is a good starting point. However when a window is opened the external temperature will effect the PMV of the agent. This could alter the reward either way, if the external temperature is lower than the internal, it may lower the PMV, if the external is higher it may increase the PMV. Including the external temperature as well as the internal is a rational bases for the agent to learn. With the states of indoor temperature limited to $9 < x < 30$, external temperature limited to $9 < y < 30$, gives 400 possible states. There are two possible window actions, open or closed. With this larger table space (compared to the heating setpoint model) it may be that all possible combination of interactions can be considered, but the model may not converge to an optimal solution in all states.

The reward strategy is again based on the PMV of the agent. If the agent is too hot and the window is closed then a punishment occurs, otherwise a reward. Conversely when an agent is cold and the window is open the agent is punished, otherwise rewarded. The reward function is therefore:

$$r = (0.1 * a - 0.1 * b) \tag{7.2.3}$$

Where a is a binary operator: if $pmv > 0.5$ and the window is open or if $pmv < -0.5$ and the window closed, $a = 1$, otherwise it is equal 0. b is also a binary operator: if $pmv > 0.5$ and the window is closed or if $pmv < -0.5$ and the window is open $a = 1$, otherwise $a = 0$. The other parameters needed in the Q-learning equation (7.2.1) are kept the same, α, γ are 0.1 and ϵ is again 0.01.

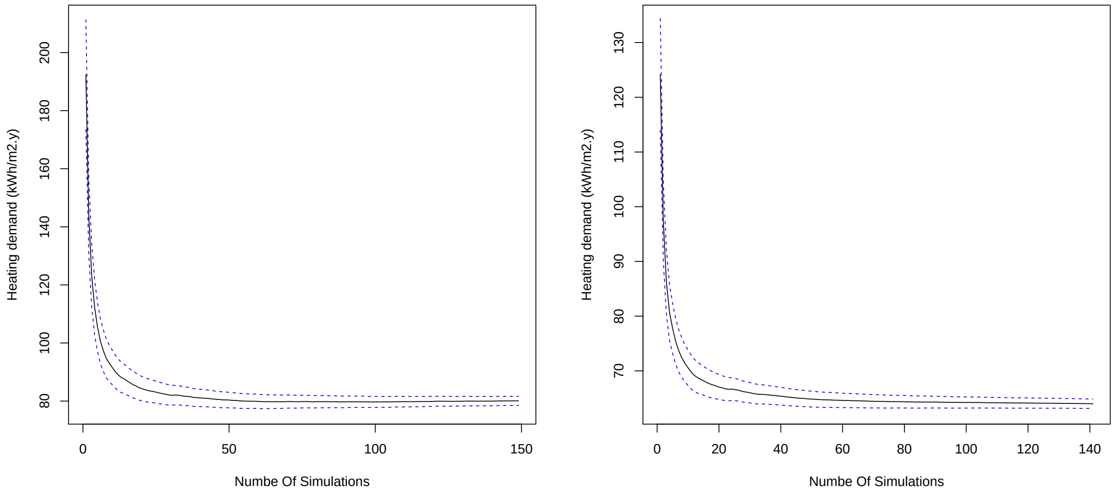
The residential building and non-residential building discussed in Chapter 4 are used once again as a case study. However for simplicity only Geneva is considered. Cooling is not used as it would influence the learnt heating demand. Operative temperature is used to set the heating setpoints, in contrast to Chapter 4 where air temperature was used. However air temperature does not take into account radiant gains, causing learnt temperatures to be higher than would be expected.

7.3 Agent Learning And Building Performance

Heating Setpoints

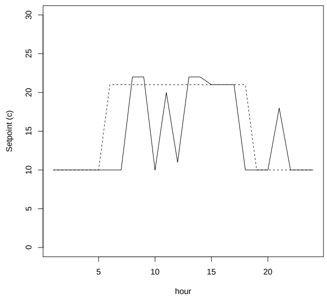
Agents should learn the optimal heating setpoints of a building after a period of time. Choosing a standard learning period, a t-test (P value = 0.05) is performed, comparing the median value of 100 replicates to the previous years' 100 replicates until the median heating demand is no longer significantly different, this occurs at 19 training years. The running mean for 150 years of simulation is given in Figure 7.1.

After a period of 19 training years a No-MASS agent learns the heating demand profiles presented in Figure 7.2. For the cooler months, November to March, similar profiles are learnt. At the end of this heating season in April and October the agents have learnt three distinctive profiles: heating first thing in the morning, reducing heating, raising around lunch, reducing after lunch and increasing again before departure. During the summer months and September there is no desire by the agents to enable heating.

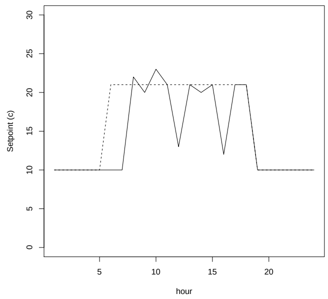


(A) Non-residential (B) Residential

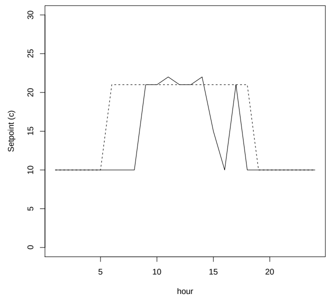
FIGURE 7.1: Heating demand mean convergence for the learning heating setpoint within the non-residential building



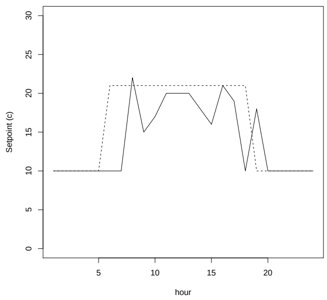
(A) Janurary



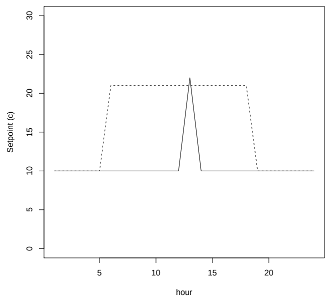
(B) Feburary



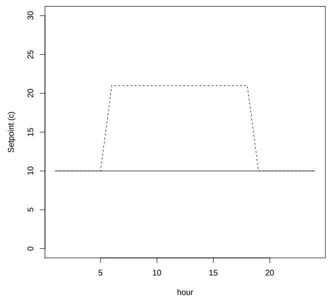
(C) March



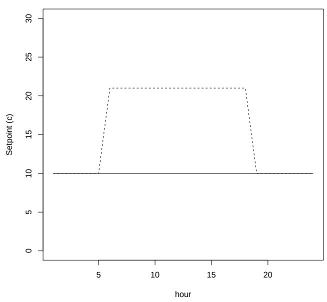
(D) April



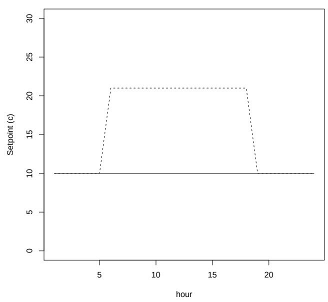
(E) May



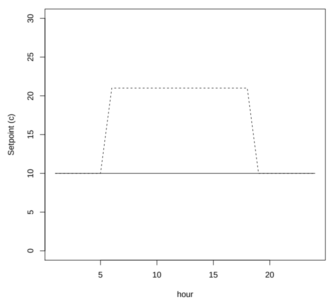
(F) June



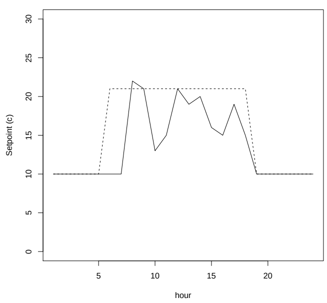
(G) July



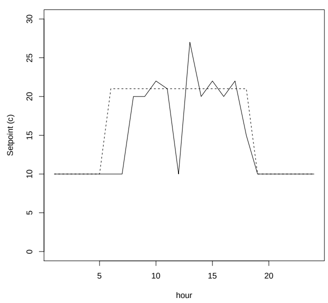
(H) August



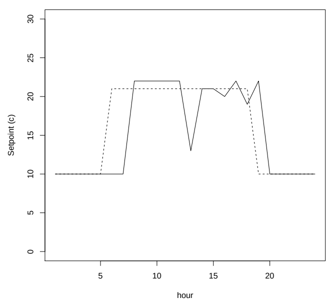
(I) September



(J) October



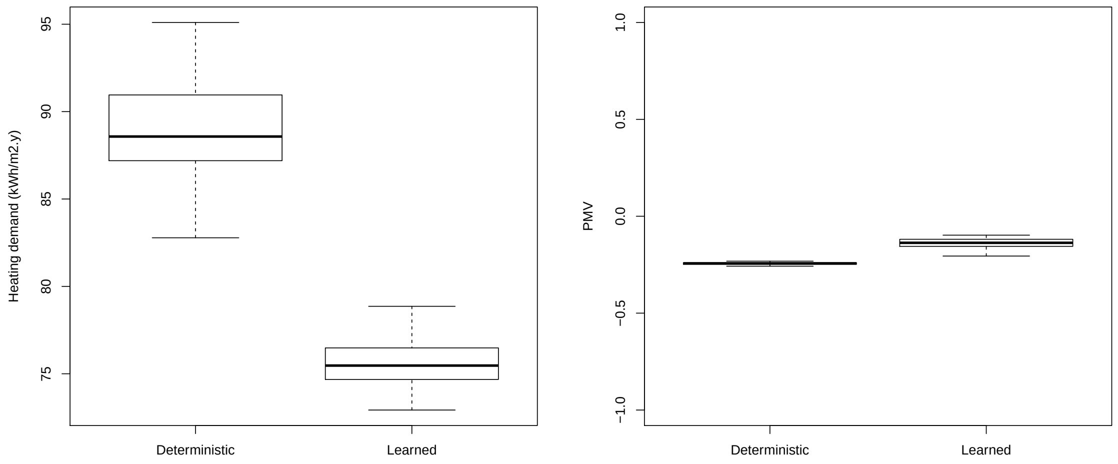
(K) November



(L) December

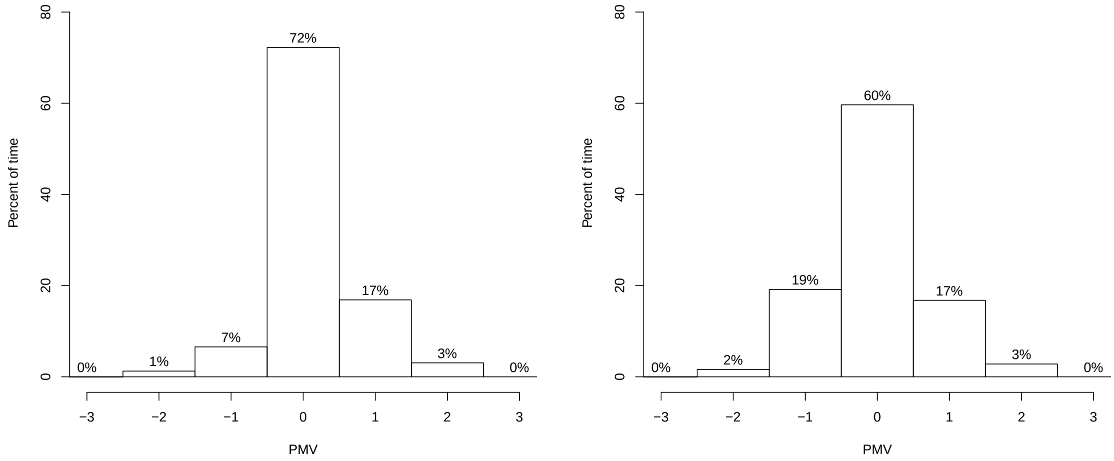
FIGURE 7.2: Learnt monthly heating setpoint profiles from 100 replicates within the non-residential building

Comparison between the heating demand of the learnt profile and the deterministic heating setpoint schedules are given in Figure 7.3a. The deterministic setpoint schedules simulated a median heating demand $5kWh/m^2$ larger than the learnt schedules. It would be expected that agents using the learnt schedules would not have their thermal comfort met as they are using less energy to heat the building. However in Figure 7.3b observe that this is not the case. Using the learnt profiles the yearly mean percentage of people dissatisfied (PPD) is within half a percent of the deterministic setpoint schedules with the mean PMV being closer to zero. The learnt profiles effectively meet the agents energy and comfort aspirations.



(A) Heating Demand (B) Yearly Mean PMV

FIGURE 7.3: Box plots of 100 replicates for learning heating setpoints within the non-residential building



(A) Deterministic heating set points (B) Learnt heating set points

FIGURE 7.4: Density plot of PMV for learning heating setpoints within the non-residential building

In the residential building the learnt heating setpoints are similar to those of the deterministic case, but for significantly reduced heating demands. This is expected as the learning mechanism considers agents' presence. This time after 25 simulations the agents have learnt a profile for which t-test declares the heating demand has not changed. The heating profiles for each room at the 25th simulation is given in Figures C.1-C.5. The agents have learnt that the kitchen and the office rooms do not need to be heated. Looking at the monthly air temperatures for the rooms (Figure 7.5), the mean air temperature does not go below 15°C during January in the kitchen. With typical figures for the month of January used in the PMV calculation; a metabolic rate at $116W/m^2$ for cooking, a clo of 1, humidity at 20%, radiant temperature and air temperature at 15 °C the PMV would be -0.17; which is in the comfortable range. However the metabolic rate is possibly too high in this scenario for cooking. The high random peaks that can be see are from exploration by the agents. For example in the kitchen the agents set the temperature to 27 °C at 7pm in the kitchen in September. For the living room the agents choose a high temperature setpoint for the winter months, 23/24 °C, due to the low metabolic rate while in the activity audio/visual or in the activity passive. During the months June to September the heating setpoints are low, with the winter months showing heating turned on from morning to late at night. The bathroom is heated from around 6/7am till 3pm and then from 9pm till midnight. The bedroom is heated during the night, albeit to maintain a relatively low setpoints thanks to the insulation afforded by the duvet.

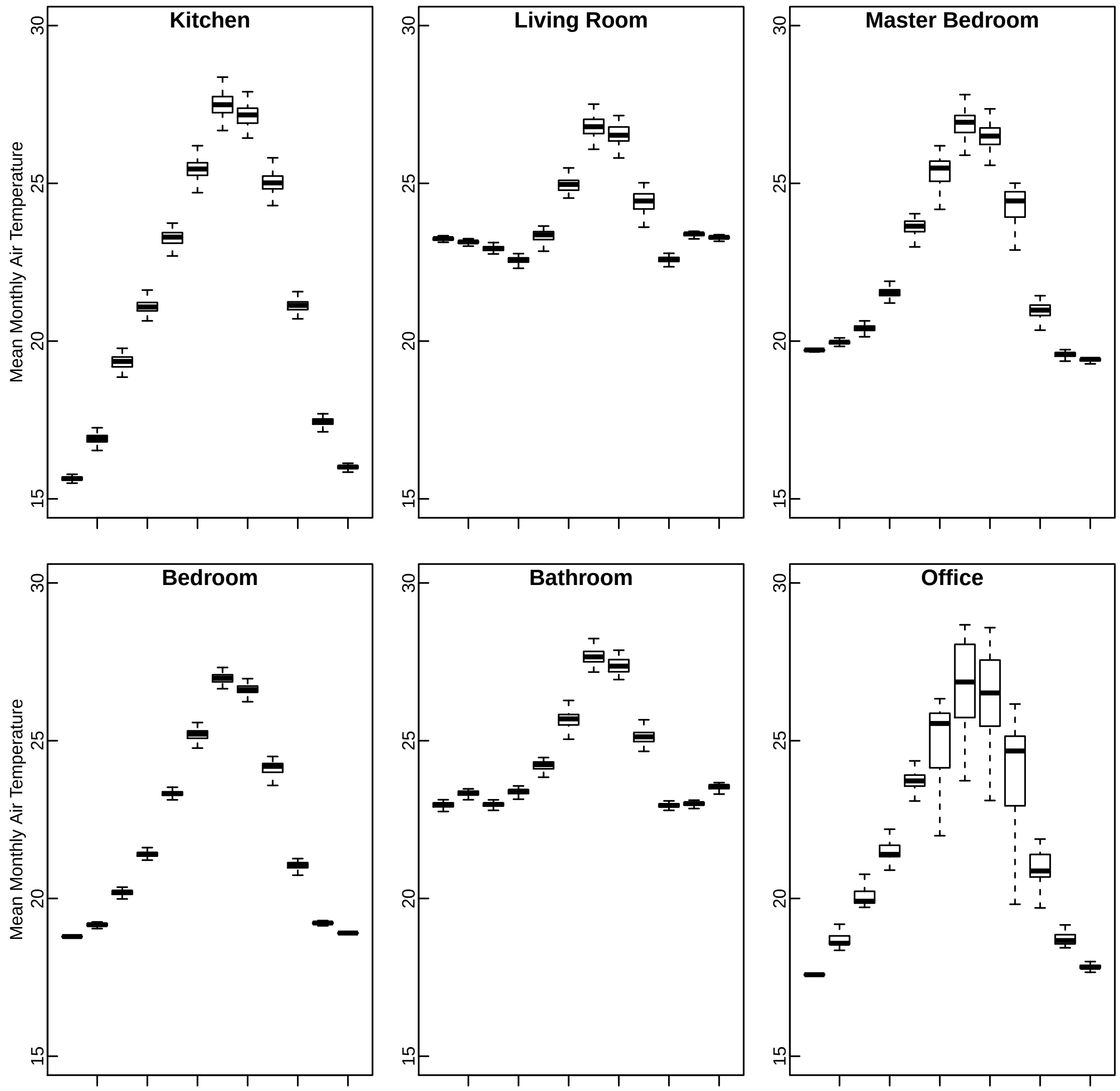
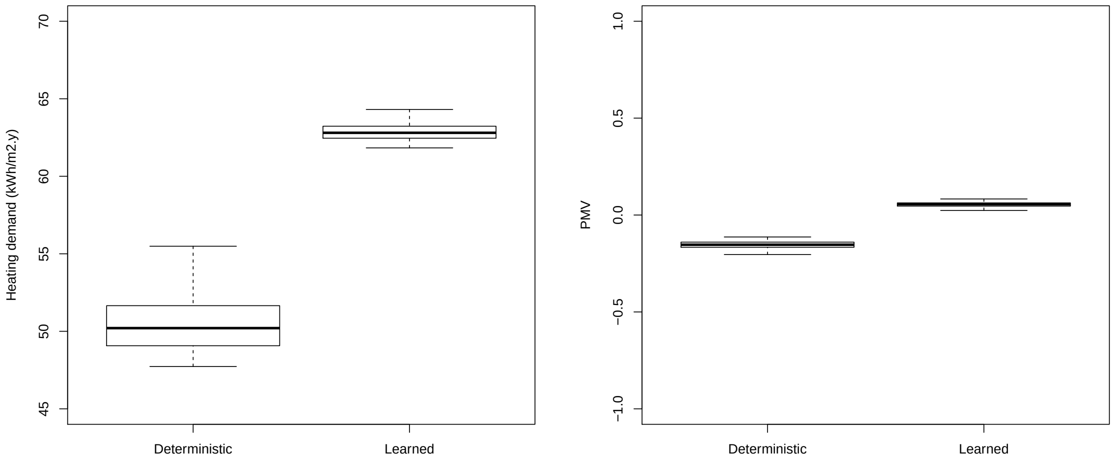


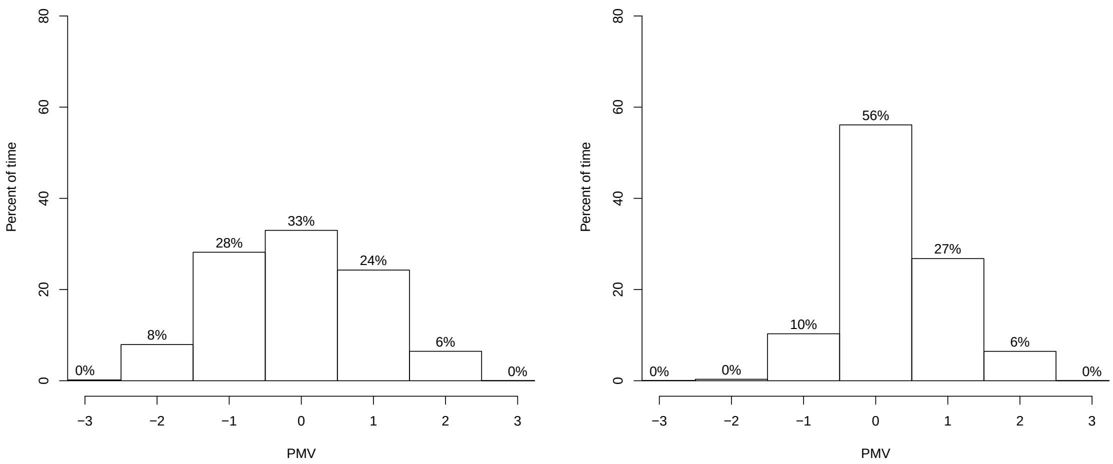
FIGURE 7.5: Box plot of monthly mean air temperature (°C) for the learning heating setpoints for the residential building

The monthly average air temperature appears to show that the mean air temperature is kept at an acceptable level for the building occupants. Observations of the time spent at the different levels of PMV for the year (Figure 7.7b) show that although heating demand has increased by $14kWh/m^2$, the agent is only slightly cool 10% of the time and rarely cooler. Looking at the deterministic schedule the agents are cool 28% of the time and cold 8% of the time. These schedules are therefore a significant improvement for this type of occupant.



(A) Heating Demand (B) Yearly Mean PMV

Figure 7.6: Box plots of 100 replicates for learning heating setpoints within the residential building

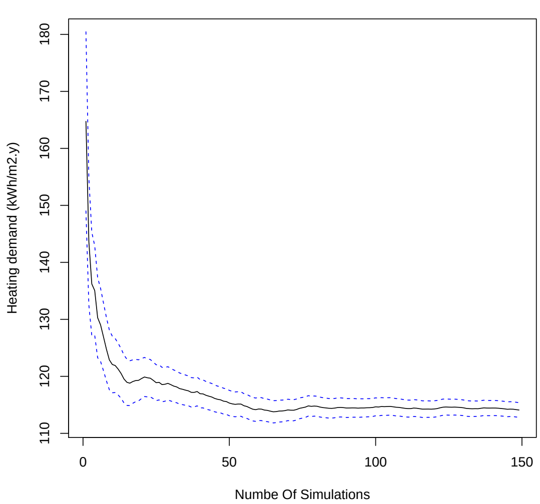


(A) Deterministic heating set points (B) Learnt heating set points

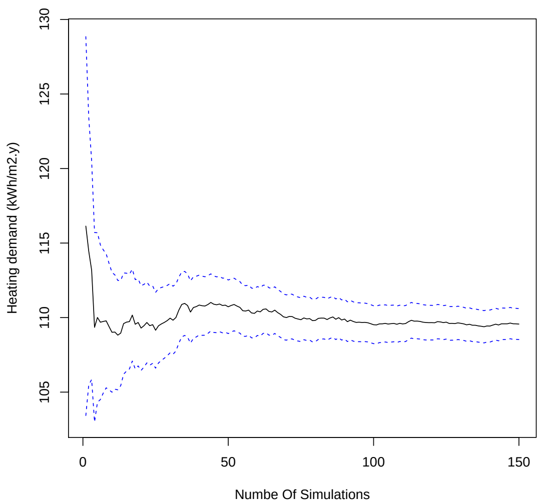
Figure 7.7: Density plot of PMV for learning heating setpoints within the residential building

Learning Window Opening Interactions

Comparison of the Q-learning window opening model and that of the stochastic window opening model shows the effectiveness of the learning methodology as a potentially useful model in place of stochastic models if no data is available. Figure 7.9a shows that there is a significant difference between the heating demands using the Learnt model and the stochastic model for the non-residential building. The mean PMV is almost equal in both bases. The density plots of PMV (Figure 7.10) show that the Learnt model lowers the proportion of time spent above a PMV of 0.5 by 11% compared to that of the stochastic model. However the Learnt model is over estimating the proportion of time that the windows are open over the course a year. This may be because the PMV model used assumes an airflow of $0.1m/s$ which will not be the case when the window is open. The airflow could be much higher resulting in a lower PMV. As EnergyPlus does not calculate local air velocity but takes a user specified value and No-MASS currently has no method of calculating air velocity, our corresponding PMV predictions will be error. Since window openings are overestimated it is unsurprising that heating energy demand is corresponding over estimated; but only by approximately 6%, in the case of the non-residential building. On balance, the Q-learning model performs well in this case.

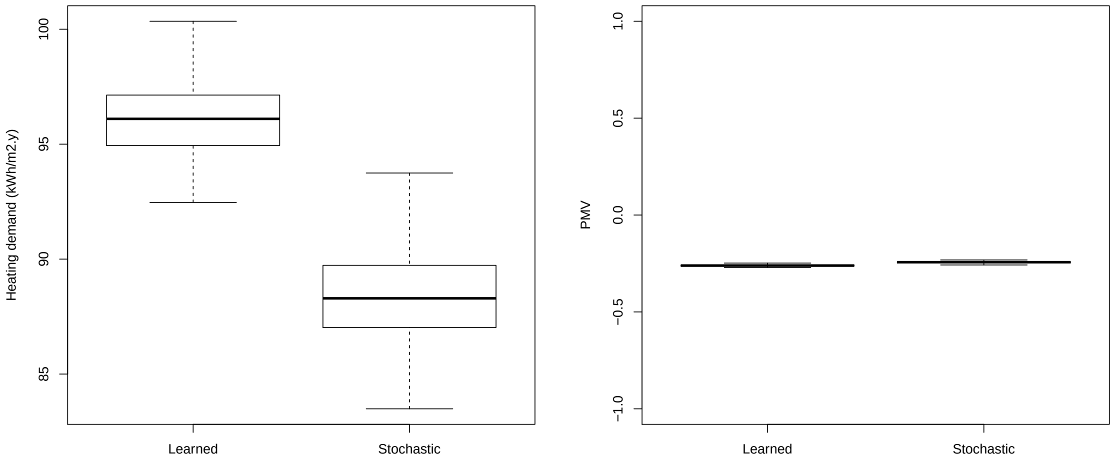


(A) Non-residential



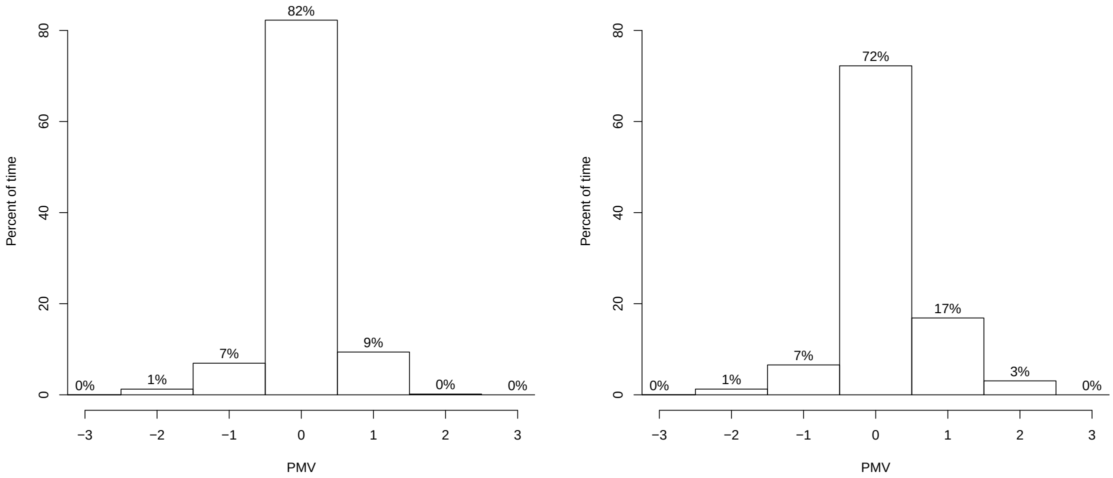
(B) Residential

FIGURE 7.8: Heating demand for mean convergence for learning window interactions



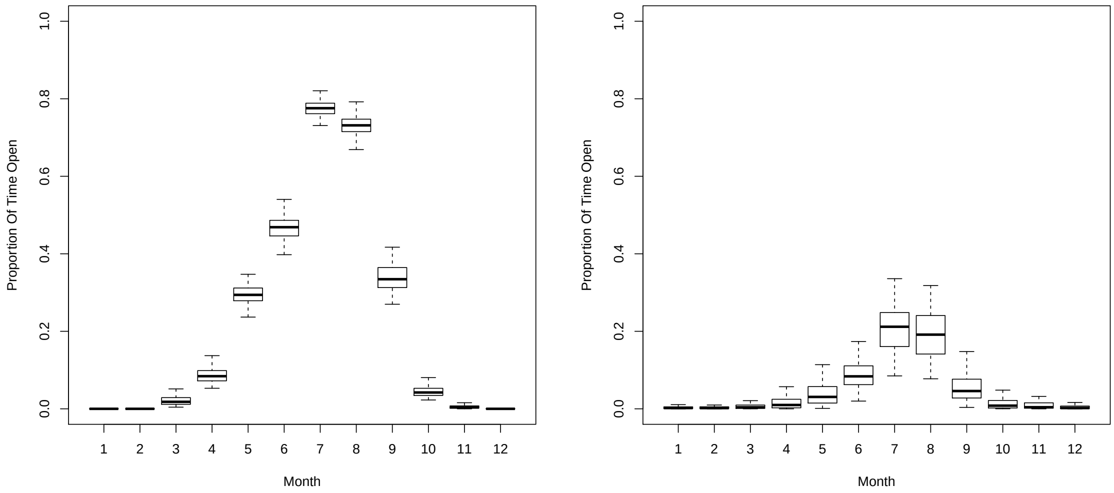
(A) Heating Demand (B) Yearly Mean PMV

FIGURE 7.9: Box plots of 100 replicates for learning window interactions within the non-residential building



(A) Learnt Window Openings (B) Stochastic Window Openings

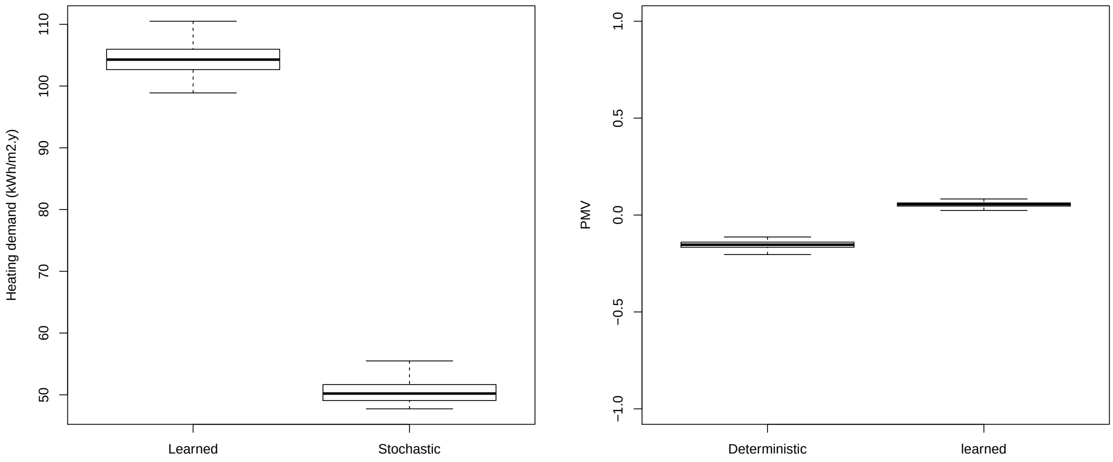
FIGURE 7.10: Density plot of PMV for learning window interactions within the non-residential building



(A) Learnt Window Openings (B) Stochastic Window Openings

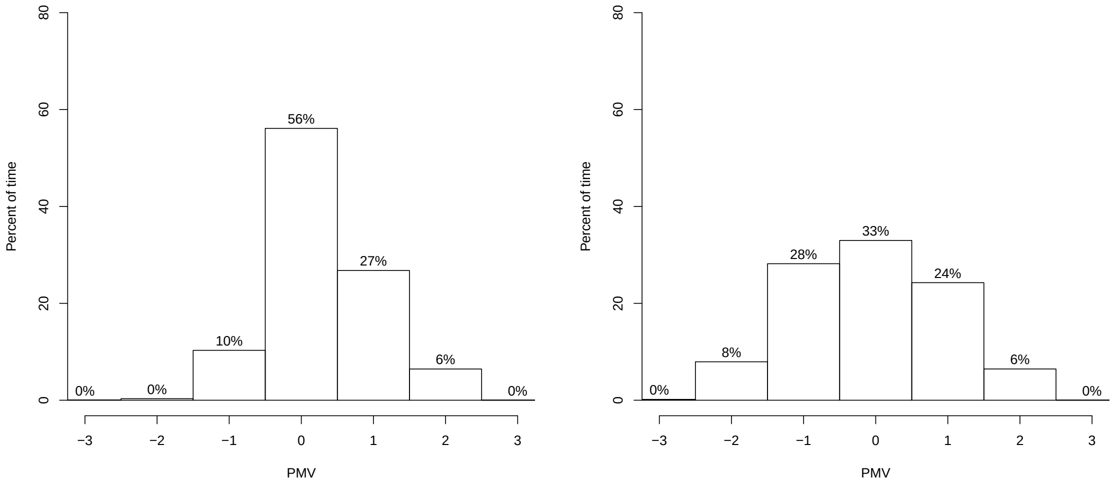
FIGURE 7.11: Box plots of the percentage of time the window is open per month for 100 replicates in the non-residential building

In contrast, the residential building's learnt window opening interactions cause the heating demand to double in value from $50kWh/m^2$ using the stochastic model to $104kWh/m^2$ using the learnt interactions. However, this increase in heating demand does results in agents spending more time at the PMV neutral state (Figure 7.13). But the boxplots of monthly demand for each room (Figure 7.14) show that for all zones there is a considerable overestimate in window usage. This is especially true in the kitchen, where even during the winter months the windows are open at least 30% of the time; suggests that this methodology needs further reinforcement. For example it appears that in the case of the kitchen the agent learns that the windows can be left open and that the heating systems will nevertheless try to maintain the temperature. To resolve this punishing the agent when the window is open and the heating is on seems sensible.



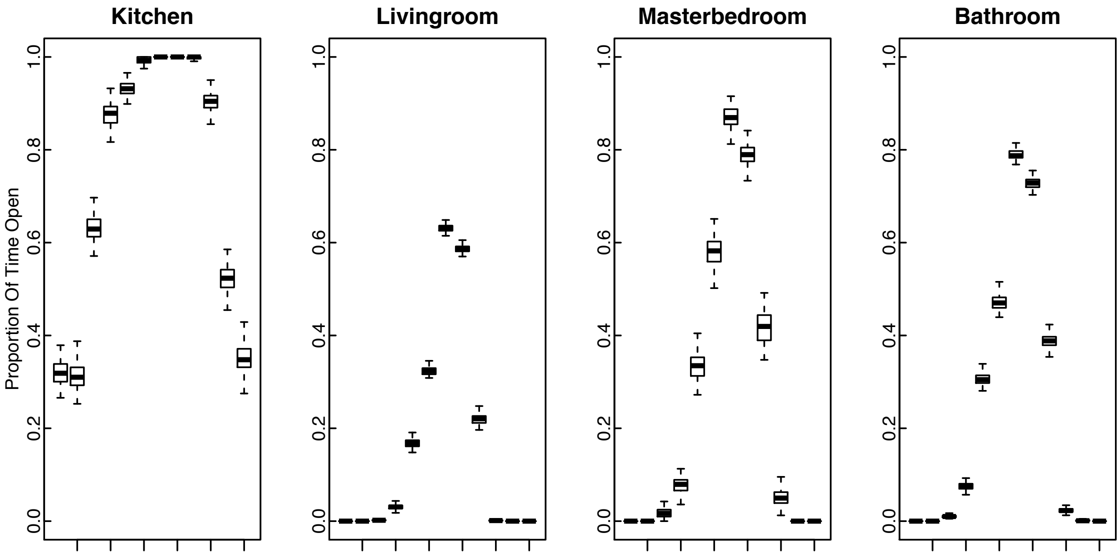
(A) Heating Demand (B) Yearly Mean PMV

FIGURE 7.12: Box plots of 100 replicates for learning window interactions within the residential building

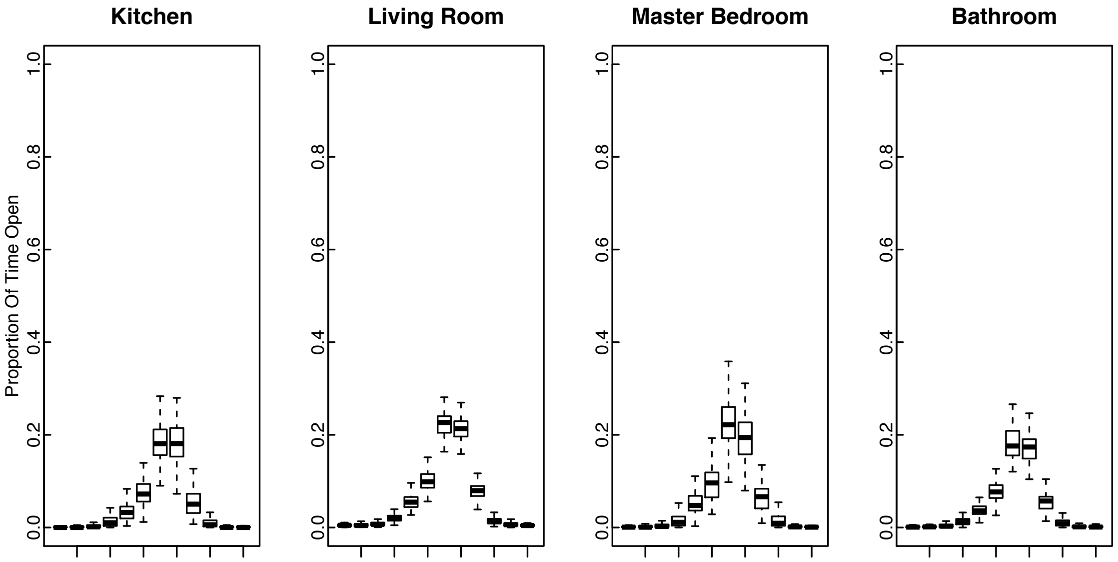


(A) Learnt Window Openings (B) Stochastic Window Openings

FIGURE 7.13: Density plot of PMV for learning window interactions within the residential building



(A) Learnt Window Openings



(B) Stochastic Window Openings

Figure 7.14: Box plots of the percentage of time the window is open per month for 100 replicates in the residential building

7.4 Conclusion

Agent learning appears to be an effective way of developing rules for emulating occupants' comfort stochastic behaviours for cases where data is limited or unavailable. As this work demonstrates, however, in buildings where occupancy may vary over time and with the season, a learnt set of heating setpoints may improve the fidelity of predicted building performance. A good example would be schools or universities, where not all rooms are used throughout the day but are sporadic, it would be unwise to simulate constant heating demands over the day, a better profile would be one learnt based on teaching schedules.

In the relatively simple case of the non-residential building our agent learning strategy also effectively simulate the use of windows; but this use is over estimated in the residential case; windows are open far longer than would be expected. The large table space and the low likelihood of all states occurring multiple times during a simulation limits the learning rate and leads to over estimation. Examining the stochastic models the parameters considered in the model (eg. indoor temperature, external temperature, length of time departure, etc.) show that there are a number of factors that are significant to consider, which are not in No-MASS's Q-learning model. Combining these for different models for arrival, presence and at departure make it almost impossible to model with the Q-learning methodology. A better model would be to use a neural network, taking in these parameters to learn how they effect the reward. This would also overcome the weakness of having to reach every state with just internal and external temperature as in the Q-learning method.

Nevertheless is would be useful to improve the quality of the reward calculation in this methodology, wither using an adaptive comfort model, or by adding finesse to the use of the PMV model, for example to penalise conflicting strategies (eg. of windows open and heating on) and better representing agents activity metabolic rather and clothing values (eg whilst cooking and sleeping).

Chapter 8

Conclusion and Recommendation

In Chapter 1 it was hypothesised that an effective modelling strategy for simulating occupants in building performance simulation is multi-agent simulation. The No-MASS framework used agents to integrate a hybrid system of stochastic models, BDI theory, a social interaction framework and machine learning to model people in buildings. In Chapter 3 the No-MASS framework was described, highlighting how it is a generic interface that can be linked to other simulation tools that use the open FMI standard for co-simulation. The effects of the stochastic nature of occupants was then highlighted in Chapter 4, showing how they influence simulated building performance using models of external shading, window opening, lighting, presence and activities, achieving aim 1. To achieve aim 2, a mechanism was described for using different occupant profiles for the activity model, allowing No-MASS to handle diversity between occupants. Chapter 5 achieves aim 3, since No-MASS now also uses a BDI framework to model behaviours for which there is no empirical data to build validated stochastic models. The sensitivity of simulated performance to these rules show that in some cases there is a clear need for data collection efforts; but also that the BDI approach is a sound intermediate solution. Agent social interactions and the effects they cause are described in Chapter 6 for aim 4, also observing the effects of agent diversity through the use of different window profiles. Finally, aim 5 is achieved in Chapter 6 with the development of machine learning techniques to solve occupant interactions that do not have data and are too complex to be solved through generic rules.

Employing these complementary techniques to support the comprehensive simulation of occupants presence and behaviour, integrated within a single platform that can readily interface with a range of building (and urban) energy simulation programs is the key contribution to knowledge from this thesis. Nevertheless there is significant scope to extend this work to further reduce the performance gap between simulated and real world buildings.

Validating this work against a real world data is difficult since it is difficult to conduct experiments that isolate individual energy flow pathways in buildings. Nevertheless, it would be useful to conduct a study similar to that of Blight and Coley (2013), where No-MASS is tested against a building designed to the passive house standard. These buildings have stringent designs that must be kept to, reducing our Type II errors, hence any variations should be due to occupants and or weather conditions.

The BDI models shows that there is a need for more stochastic models, requiring more data collection especially regarding of window opening behaviours in response to pollutants; likewise with respect to the lifestyle (eg privacy) related impacts on the use of windows, lights and shading devices. In the office, as already mentioned, other models that could included, such as long term absences due to sickness, business trips and holidays; these can also be applied to the household, as most longer holidays will be away from the home and sicknesses will be in the home. Models for large appliances and aggregated small appliances should also be included within No-MASS. Rather than define the agents at each simulation it would be better to build a system that generates a synthetic population based on the social demographics of the building to be simulated. This would provide a convenient basis for evaluating the robustness of a design to diverse populations of occupants.

This work created a social interaction model to demonstrate how conflicting interactions could be managed, however a field study studying the different types of social interactions and their frequency in different settings would allow the model to be based on real world data. To our knowledge no such data collection exists with building performance in mind and it could be used to replace or supplement the social interaction model implemented.

No-MASS is currently generalised for a single building but their is no reason why it cannot be extended to handle multiple buildings to support integration with tools like CitySim (Robinson, 2011). Further the same multi agent stochastic simulation methodology could be used in the simulation of smart grids, with appliances becoming agents, their demand profiles simulated with stochastic models where data is available and BDI/ agent learning when not. The appliance could communicate when to turn on using the social interaction model. The agent appliances would learn the optimal demand profiles for themselves for a day or week based on machine learning, either Q-learning or neural networks.

Finally the ability to use No-MASS from within the DesignBuilder interface tool is of great benefit for our work due to the feedback gained from the use of practitioners; however a study into the effectiveness of the interface and the integrated models by industry experts would validate the usability of this work.

Appendix A

DesignBuilder Interface Workflow

A Building is first designed in the DesignBuilder simulation program. This involves using the controls to sketch out the vertices of the building, selecting the parameters for the construction materials and laying out the internal zones. Windows, shades, lighting and HVAC systems can be altered, however defaults are used for quick simulations. Once designed the No-MASS agents can be enabled through the modelling options dialogue. See Figure A.1.

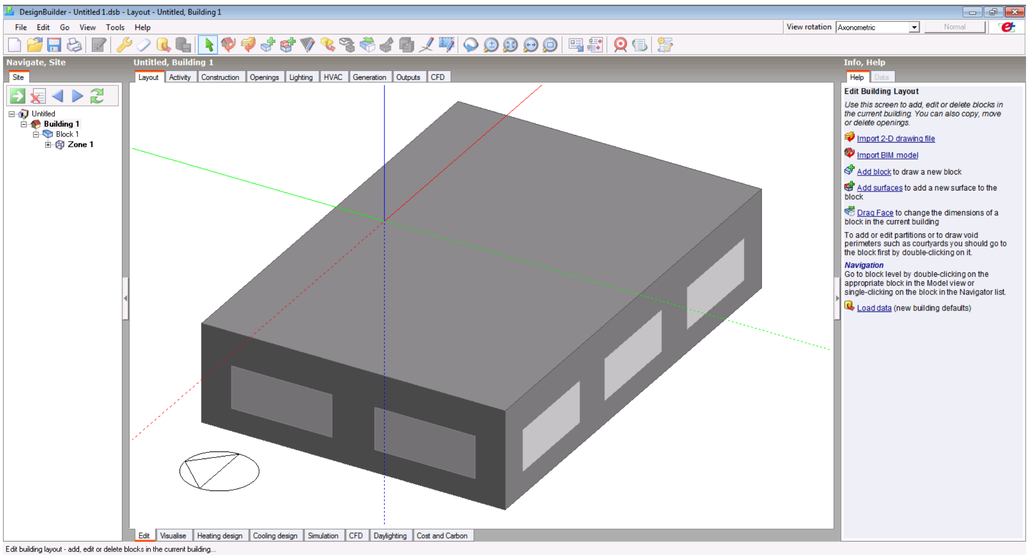


FIGURE A.1: Design a Building

The modeling options dialogue allows more advanced options in DesignBuilder to be accessed. This included the No-MASS agents. It is possible to enable the individual stochastic models and set a seed value for controlling the randomness of the simulations. Once enabled, in the main dialogue there is an extra selection box where the occupant parameters can be altered. See Figure A.2.

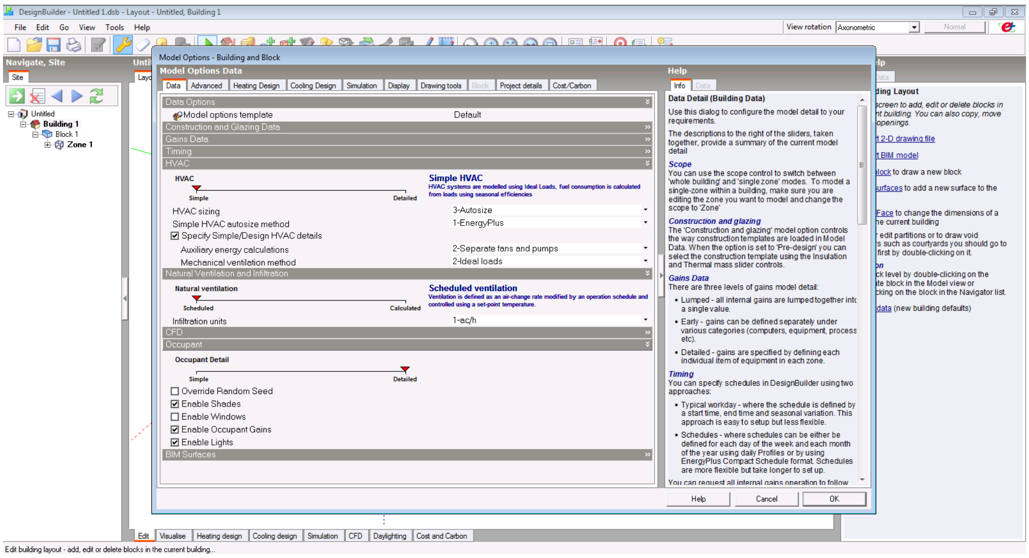


FIGURE A.2: Enabled detailed occupancy

The occupant detail selection box allows the building occupants to be selected from a database of available templates for occupants. It is possible to choose on the occupant templates such as a family of four or an office worker. Over time the user can add to these creating their own templates for the different buildings they work with. See Figure A.3.

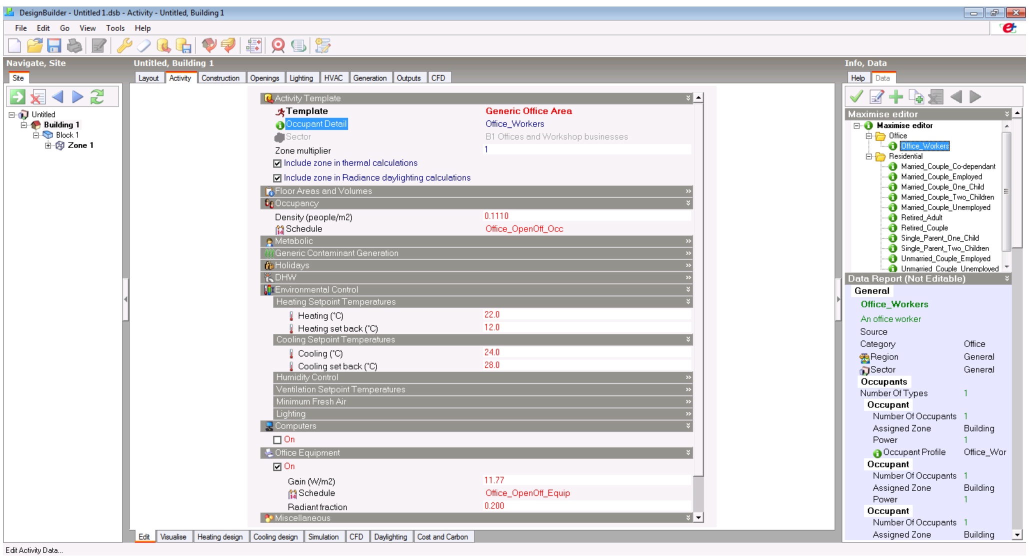


FIGURE A.3: Select the occupancy profile

Editing the occupant templates allows the user to choose the number of occupants that inhabit the building. They can then assign different profiles to the different occupants. There are some readily available profiles in the database. For example that of a retired elderly person, this sets the activity model more time at home during the day. The occupants power for use in the social interaction framework can also be specified. The occupant is assigned a zone, either their bedroom in a residential for sleeping, or the office they work in for a non-residential building. See Figure A.4.

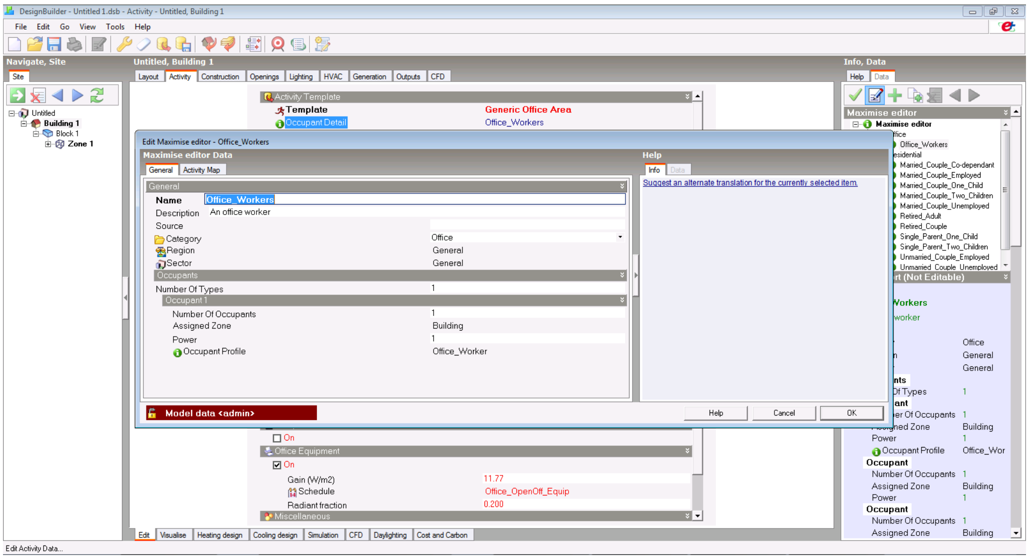


FIGURE A.4: Edit the occupancy profile

Editing the individual profiles allows the user to choose from the different parameters for the activity model, shading model and window model. Giving complete control to the end user on how the occupants will behave, however at this point expert knowledge would be needed as to what the values of the coefficients should be. See Figure A.5.

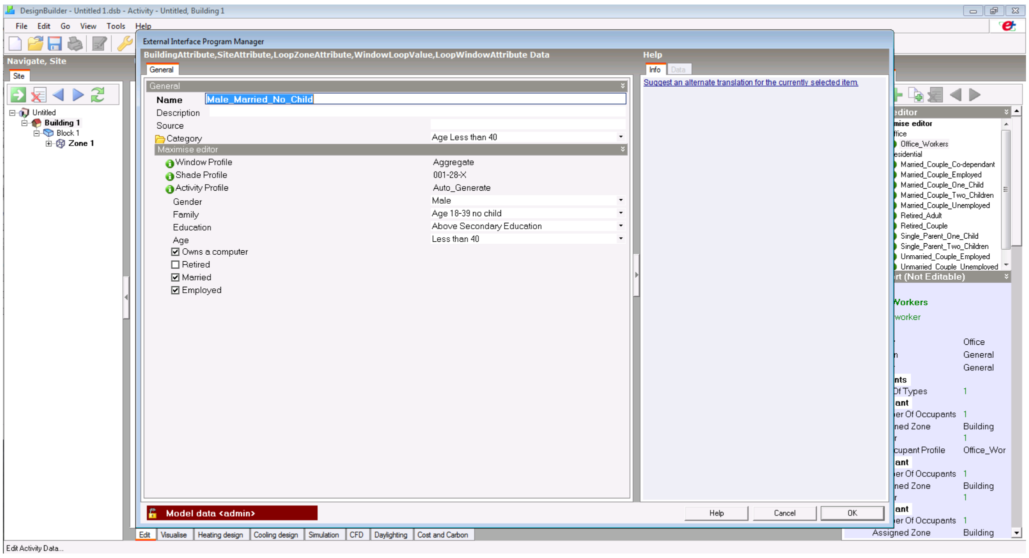


FIGURE A.5: Edit individual occupants profiles, selecting windows, shades, etc..

Finally, once the user is happy with the occupants defined through the interface a simulation can be performed. Result are now influence by the No-MASS agents. See Figure A.6.

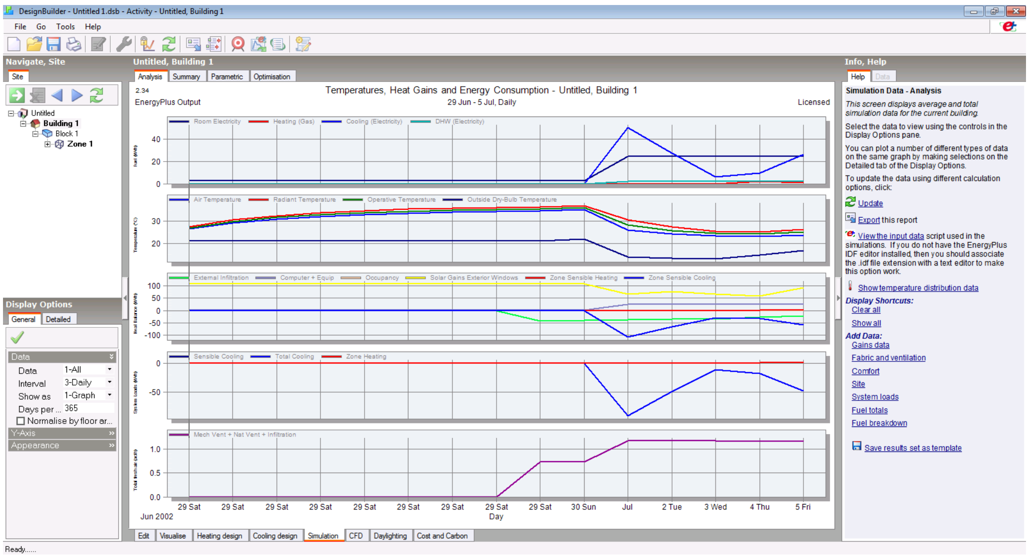


FIGURE A.6: Run a simulation and view the results of stochastic occupancy

Appendix B

EnergyPlus Source Code Changes

EnergyPlus differences for allowing shading interactions.

```
--- EnergyPlus/src/EnergyPlus/DaylightingManager.cc
+++ EnergyPlusNoMass/src/EnergyPlus/DaylightingManager.cc
@@ -5744,7 +5745,10 @@
    }
+
+   if (SurfaceWindow(IWin).ShadingFractionEMSOn){
+    VTRatio = VTRatio * SurfaceWindow(IWin).ShadingFractionEMSValue;
+   }

--- EnergyPlus/src/EnergyPlus/SolarShading.cc
+++ EnergyPlusNoMass/src/EnergyPlus/SolarShading.cc
@@ -5180,6 +5197,16 @@
     CosInc = CosIncAng( TimeStep, HourOfDay, SurfNum2 );
     SunLitFract = SunlitFrac( TimeStep, HourOfDay, SurfNum2 );

+    //! Set trans to shading fraction
+    //! EMS Actuator Point: override setting if ems flag on
+    if (SurfaceWindow(SurfNum).ShadingFractionEMSOn){
+     SunLitFract = SunLitFract - ( 1 - SurfaceWindow(SurfNum).ShadingFractionEMSValue);
+     if(SunLitFract < 0.0){
+      SunLitFract = 0.0;
+     }
+    }
+

@@ -9534,6 +9563,77 @@

+ voidComputeWinShadeAbsorpFactorsFor(int SurfNum)
+ {
+  int WinShadeCtrlNum; // Window shading control number
+
+  int ConstrNumSh; // Window construction number with shade
+  int TotLay; // Total layers in a construction
```

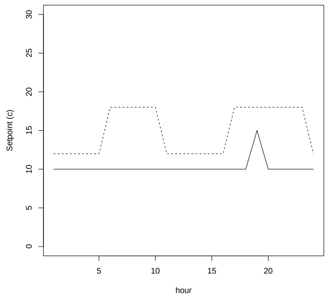
```
+  int MatNumSh; // Shade layer material number
+  Real64 AbsorpEff; // Effective absorptance of isolated shade layer (fraction of
+  //  of incident radiation remaining after reflected portion is
+  //  removed that is absorbed
+
+  if ( Surface( SurfNum ).Class == SurfaceClass_Window && Surface( SurfNum ).WindowShadingControlPtr > 0 ) {
+   WinShadeCtrlNum = Surface( SurfNum ).WindowShadingControlPtr;
+   if ( WindowShadingControl( WinShadeCtrlNum ).ShadingType == WSC_ST_InteriorShade
+      || WindowShadingControl( WinShadeCtrlNum ).ShadingType == WSC_ST_ExteriorShade
+      || WindowShadingControl( WinShadeCtrlNum ).ShadingType == WSC_ST_BetweenGlassShade ) {
+    ConstrNumSh = Surface( SurfNum ).ShadedConstruction;
+    TotLay = Construct( ConstrNumSh ).TotLayers;
+    if ( WindowShadingControl( WinShadeCtrlNum ).ShadingType == WSC_ST_InteriorShade ) {
+     MatNumSh = Construct( ConstrNumSh ).LayerPoint( TotLay ); // Interior shade
+    } else if ( WindowShadingControl( WinShadeCtrlNum ).ShadingType == WSC_ST_ExteriorShade ) {
+     MatNumSh = Construct( ConstrNumSh ).LayerPoint( 1 ); // Exterior shade
+    } else if ( WindowShadingControl( WinShadeCtrlNum ).ShadingType == WSC_ST_BetweenGlassShade ) {
+     if ( Construct( ConstrNumSh ).TotGlassLayers == 2 ) {
+      MatNumSh = Construct( ConstrNumSh ).LayerPoint( 3 ); // Double pane with between-glass shade
+     } else {
+      MatNumSh = Construct( ConstrNumSh ).LayerPoint( 5 ); // Triple pane with between-glass shade
+     }
+    }
+    //! Set trans to shading fraction
+    //! EMS Actuator Point: override setting if ems flag on
+    if (SurfaceWindow(SurfNum).ShadingFractionEMSOn){
+     Material(MatNumSh).Trans = SurfaceWindow(SurfNum).ShadingFractionEMSValue;
+    }
+
+    AbsorpEff = Material( MatNumSh ).AbsorpSolar / ( Material( MatNumSh ).AbsorpSolar
+       + Material( MatNumSh ).Trans + 0.0001 );
+    AbsorpEff = min( max( AbsorpEff, 0.0001 ), 0.999 ); // Constrain to avoid problems with following log eval
+    SurfaceWindow( SurfNum ).ShadeAbsFacFace( 1 ) =
+        ( 1.0 - std::exp( 0.5 * std::log( 1.0 - AbsorpEff ) ) ) / AbsorpEff;
```

```
+     SurfaceWindow( SurfNum ).ShadeAbsFacFace( 2 ) = 1.0 - SurfaceWindow( SurfNum ).ShadeAbsFacFace( 1 );
+    }
+   }
+
+ }

--- EnergyPlus/src/EnergyPlus/WindowManager.cc
+++ EnergyPlusNoMass/src/EnergyPlus/WindowManager.cc
@@ -2886,6 +2890,9 @@

   if ( ShadeFlag == IntShadeOn || ShadeFlag == ExtShadeOn
     || ShadeFlag == IntBlindOn || ShadeFlag == ExtBlindOn
     || ShadeFlag == BGShadeOn || ShadeFlag == BGBlindOn
     || ShadeFlag == ExtScreenOn ) {
    nglfacep = nglface + 2;
+
+   EnergyPlus::SolarShading::ComputeWinShadeAbsorpFactorsFor(SurfNum);
+
    ShadeAbsFac1 = SurfaceWindow( SurfNum ).ShadeAbsFacFace( 1 );
    ShadeAbsFac2 = SurfaceWindow( SurfNum ).ShadeAbsFacFace( 2 );
    AbsRadShadeFace( 1 ) = ( SurfaceWindow( SurfNum ).ExtBeamAbsByShade +
     SurfaceWindow( SurfNum ).ExtDiffAbsByShade ) * ShadeAbsFac1 +
     ( SurfaceWindow( SurfNum ).IntBeamAbsByShade +
     SurfaceWindow( SurfNum ).IntSWAbsByShade ) * ShadeAbsFac2;
```

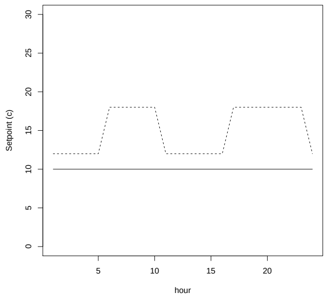
Appendix C

Learnt Heating Setpoint Profiles For Residential Building

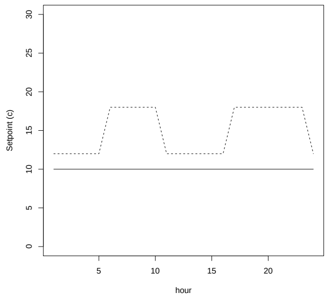
After a training period of 25 years the results converge enough for two years to have the same heating demand. The setback temperature set to 10°C. The kitchen has low values, as the average temperature is approximately 16 °C. With the high heat gains in the zone due to the activity cooking, there is no need for heating during the winter. The office is also rarely used hence the low heating setpoints.



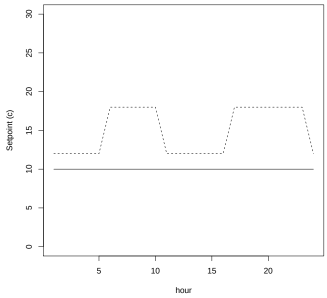
(A) Janurary



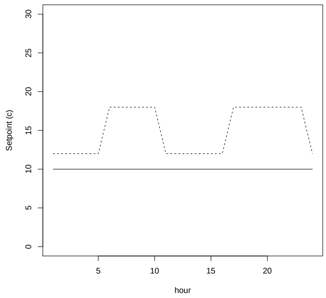
(B) Feburary



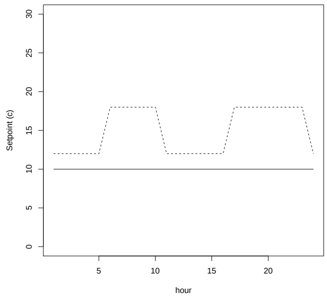
(C) March



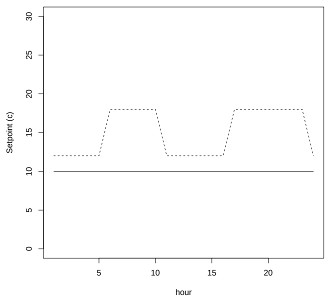
(D) April



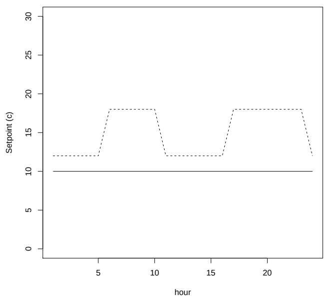
(E) May



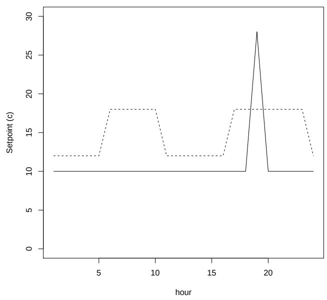
(F) June



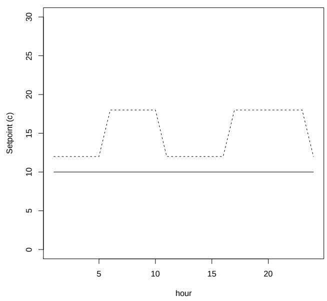
(G) July



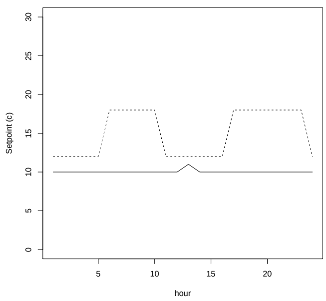
(H) August



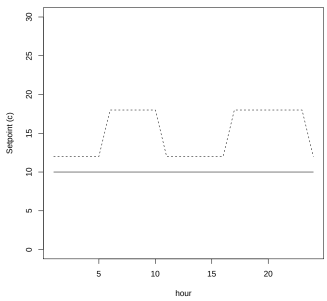
(I) September



(J) October

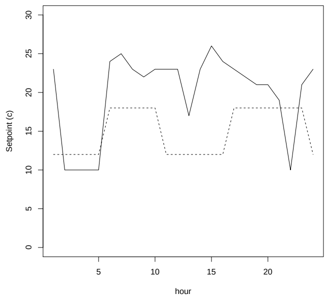


(K) November

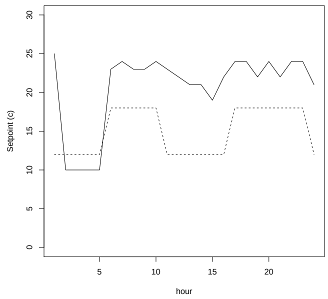


(L) December

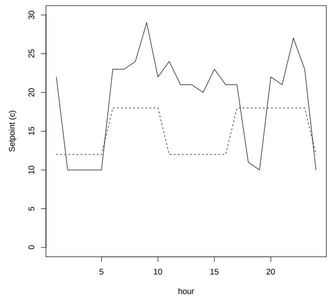
FIGURE C.1: Learnt monthly heating setpoint profiles from 100 replicates, Kitchen



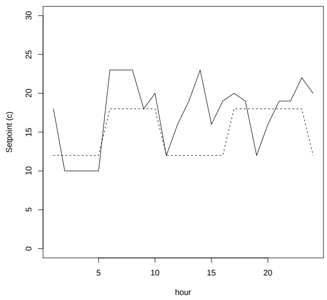
(A) Janurary



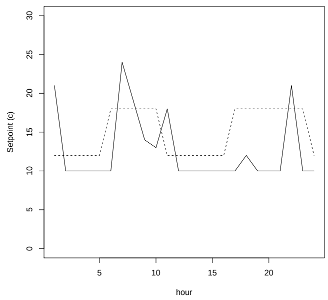
(B) Feburary



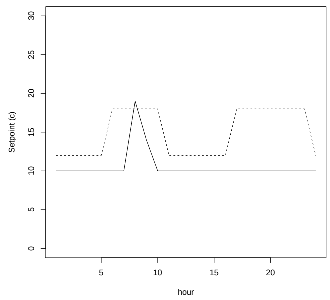
(C) March



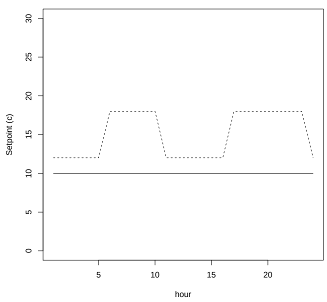
(D) April



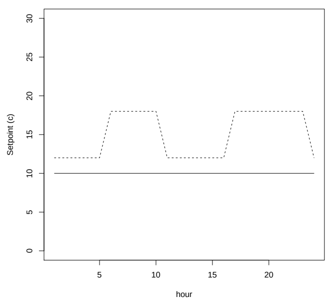
(E) May



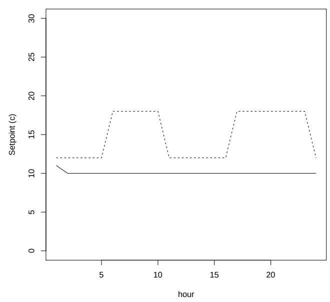
(F) June



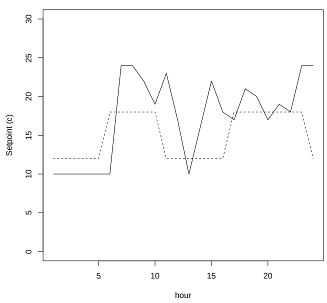
(G) July



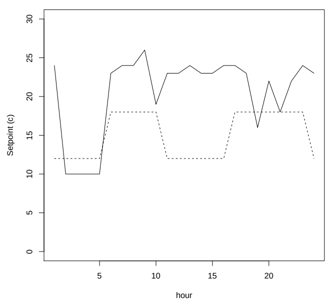
(H) August



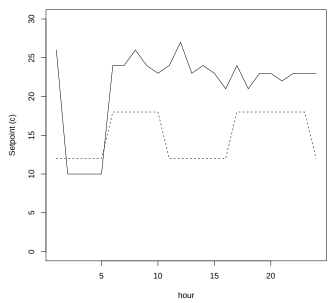
(I) September



(J) October

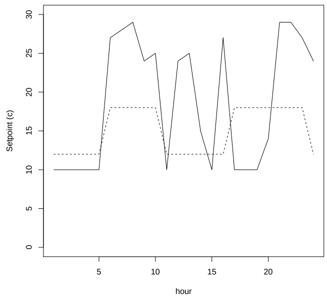


(K) November

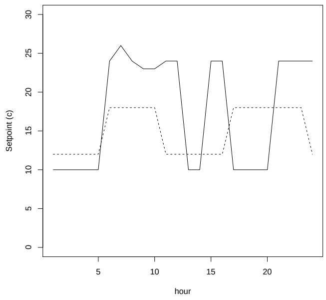


(L) December

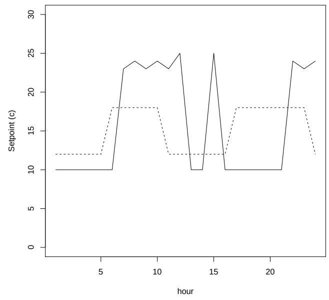
FIGURE C.2: Learnt monthly heating setpoint profiles from 100 replicates, Living Room



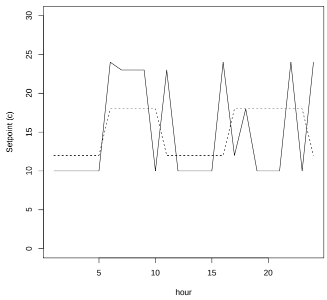
(A) Janurary



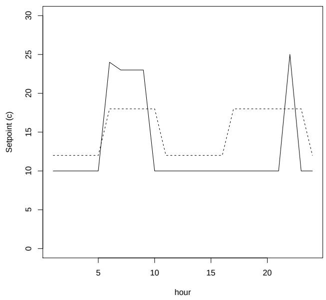
(B) Feburary



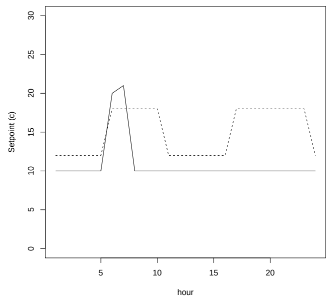
(C) March



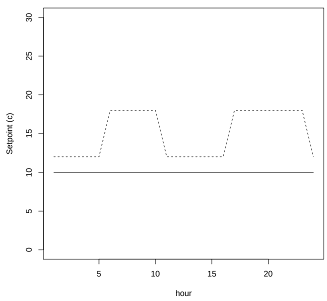
(D) April



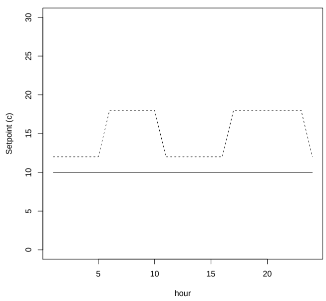
(E) May



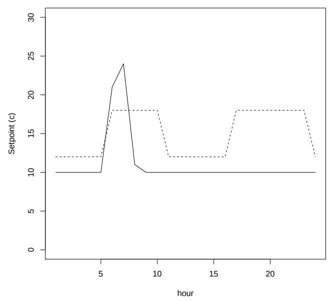
(F) June



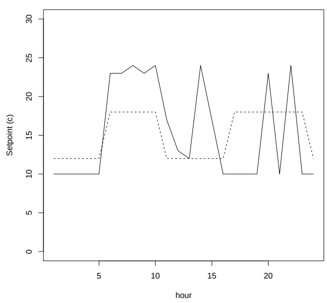
(G) July



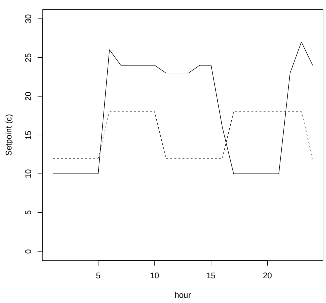
(H) August



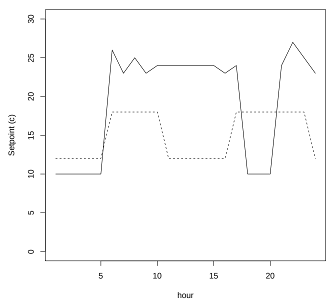
(I) September



(J) October

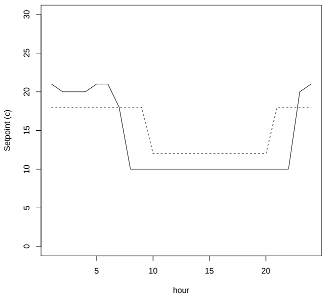


(K) November

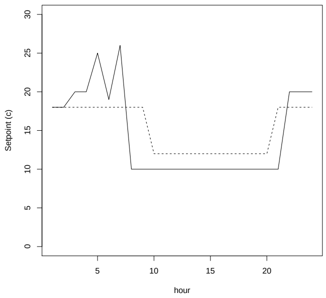


(L) December

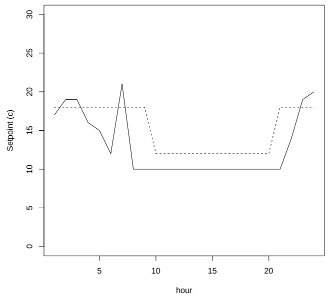
Figure C.3: Learnt monthly heating setpoint profiles from 100 replicates, Bathroom



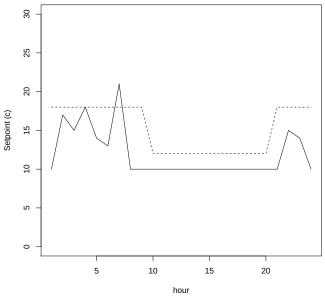
(A) Janurary



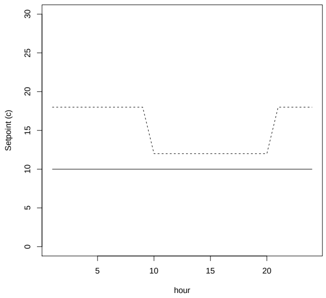
(B) Feburary



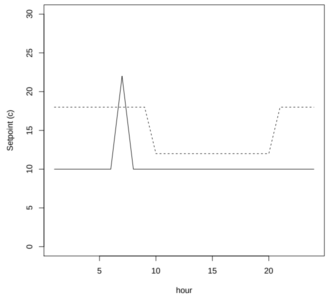
(C) March



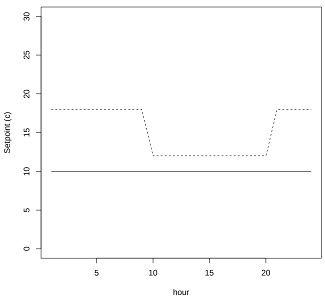
(D) April



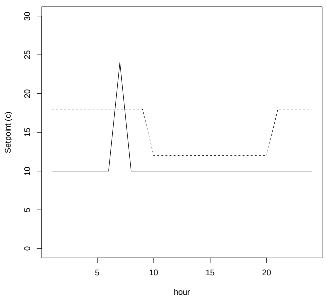
(E) May



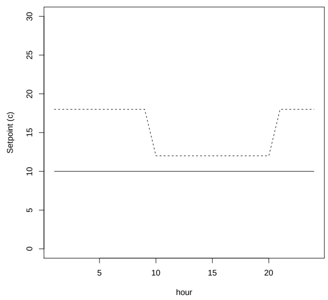
(F) June



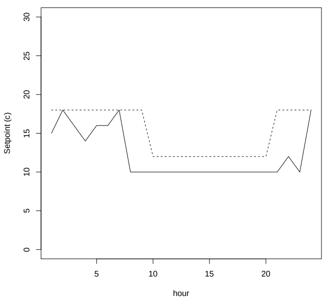
(G) July



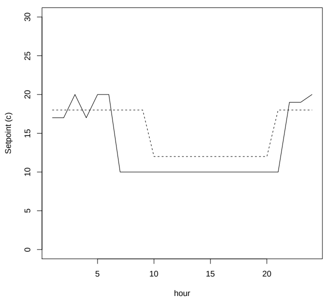
(H) August



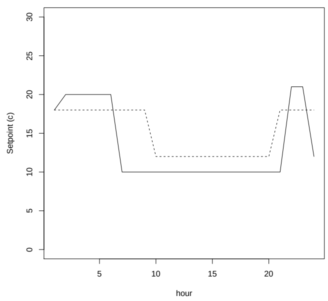
(I) September



(J) October

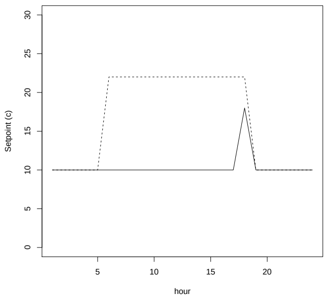


(K) November

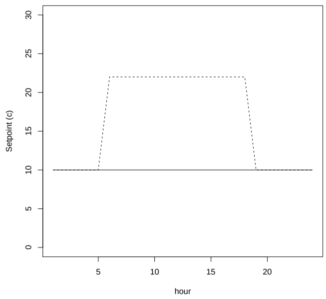


(L) December

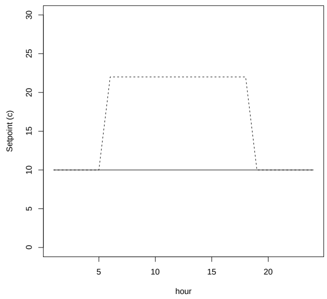
FIGURE C.4: Learnt monthly heating setpoint profiles from 100 replicates, Master Bedroom



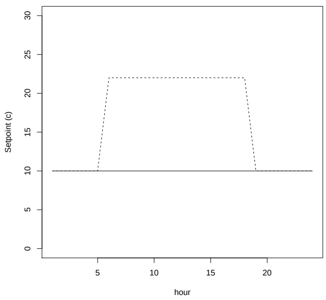
(A) Janurary



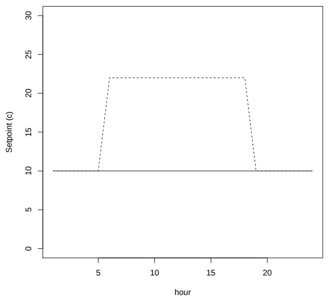
(B) Feburary



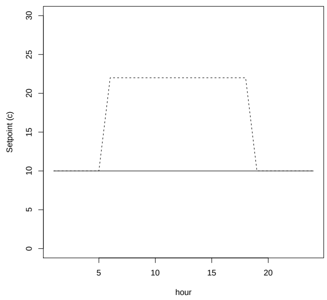
(C) March



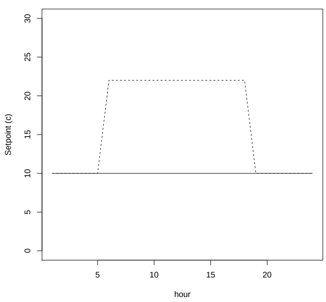
(D) April



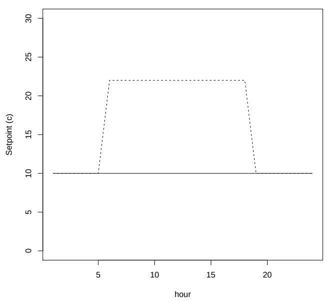
(E) May



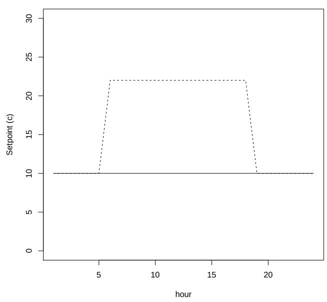
(F) June



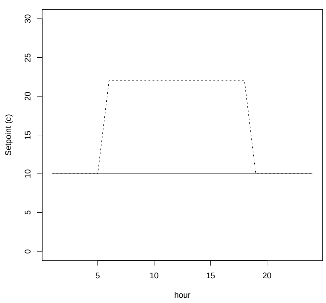
(G) July



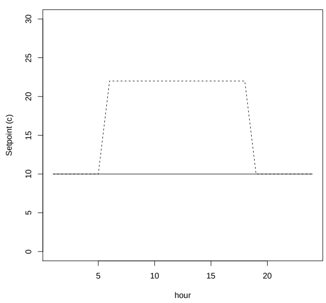
(H) August



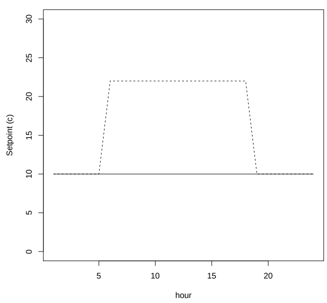
(I) September



(J) October



(K) November



(L) December

FIGURE C.5: Learnt monthly heating setpoint profiles from 100 replicates, Residential Office

Bibliography

Abelson, R. P. R. and A. Bernstein (1963). A computer simulation model of community referendum controversies. *Public Opinion Quarterly 27*(1), 93–122.

Andersen, R. V., B. Olesen, and J. Toftum (2007). Simulation of the Effects of Occupant Behaviour on Indoor Climate and Energy Consumption. In *Proceedings of Clima 2007*.

Andrews, C. J., D. Yi, U. Krogmann, J. A. Senick, and R. E. Wener (2011). Designing buildings for real occupants: An agent-based approach. *IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans 41*(6), 1077–1091.

Axelrod, R. (1987). The Evolution of Strategies in the iterated Prisoner's Dilemma. *The dynamics of norms*, 1–16.

Axelrod, R. and D. Dion (1988). The further evolution of cooperation. *Science (New York, N.Y.) 242*(4884), 1385–1390.

Axelrod, R. and W. D. Hamilton (1984). *The Evolution of Cooperation*. New York: basic books.

Axelrod, R. M. (1997). *The complexity of cooperation: Agent-based models of competition and collaboration*. Princeton University Press.

Axtell, R. (2000). Why agents?: on the varied motivations for agent computing in the social sciences. *Center on Social and Economic Dynamics Washington, DC*.

Babuška, R., B. D. Schutter, and L. Buşoniu (2008). A Comprehensive Survey of Multiagent. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 38*(2), 156–172.

Bahaj, A. S., L. Myers, and P. A. B. James (2007). Urban energy generation: Influence of micro-wind turbine output on electricity consumption in buildings. *Energy and Buildings 39*(2), 154–165.

Baker, N. and M. Standeven (1996). Thermal comfort for free-running buildings. *Energy and Buildings 23*(3), 175–182.

Baker, N. and K. Steemers (2003). *Energy and environment in architecture: a technical design guide*. Taylor & Francis.

Balmer, M., K. Axhausen, and K. Nagel (2006). Agent-based demand-modeling framework for large-scale microsimulations. *Transportation Research Record: Journal of the Transportation Research Board 4*, 1–17.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Blight, T. S. and D. A. Coley (2013). Sensitivity analysis of the effect of occupant behaviour on the energy consumption of passive house dwellings. *Energy and Buildings 66*, 183–192.

Bonabeau, E. (2002). Agent-based modeling: methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America 99 Suppl 3*, 7280–7.

Bonte, M., F. Thellier, and B. Lartigue (2014). Impact of occupant's actions on energy building performance and thermal sensation. *Energy and Buildings 76*, 219–227.

Bordass, B., R. Cohen, M. Standeven, and A. Leaman (2001). Assessing building performance in use 3: energy performance of the Probe buildings. *Building Research & Information 29*(2), 114–128.

Bourgeois, D., C. Reinhart, and I. Macdonald (2006). Adding advanced behavioural models in whole building energy simulation: A study on the total energy impact of manual and automated lighting control. *Energy and Buildings 38*(7), 814–823.

Bratman, M. (1987). *Intentions, Plans, and Practical Reason*. Center for the Study of Language and Information.

Broersen, J., M. Dastani, J. Hulstijn, Z. Huang, and L. van der Torre (2001). The BOID architecture: conflicts between beliefs, obligations, intentions and desires. In *Proceedings of the fifth international conference on Autonomous agents*, pp. 9–16.

Busetta, P., R. Rönnquist, A. Hodgson, and A. Lucas (1999). JACK Intelligent Agents - Components for Intelligent Agents in Java. *AgentLink News Letter 2*(1), 2–5.

Chang, W. K. and T. Hong (2013). Statistical analysis and modeling of occupancy patterns in open-plan offices using measured lighting-switch data. *Building Simulation 6*(1), 23–32.

Chapman, J., P. Siebers, and D. Robinson (2016). On The Multi Agent Stochastic Simulation Of Occupants In Buildings. *Manuscript submitted for publication.*

Chen, X. and F. B. Zhan (2006). Agent-based modelling and simulation of urban evacuation: relative effectiveness of simultaneous and staged evacuation strategies. *Journal of the Operational Research Society 59*(1), 25–33.

Clarke, E. H. (1971). Multipart Pricing of Public Goods. *Public Choice 11*(1), 17–33.

Clarke, J. A. (2001). *Energy simulation in building design.* Routledge.

Claus, C. and C. Boutilier (1998). The Dynamics of Reinforcement Learning in Cooperative Multiagent Systems. *Proceedings of the National Conference on Artificial intelligence/Innovative Applications of Artificial Intelligence*, 746–752.

Cohen, P. R. and H. J. Levesque (1991). Teamwork. *Nous 25*(4), 487–512.

Crawley, D. B., J. W. Hand, M. Kummert, and B. T. Griffith (2008). Contrasting the capabilities of building energy performance simulation programs. *Building and environment 43*(4), 661–673.

Crawley, D. B., L. K. Lawrie, F. C. Winkelmann, W. Buhl, Y. Huang, C. O. Pedersen, R. K. Strand, R. J. Liesen, D. E. Fisher, M. J. Witte, and J. Glazer (2001). EnergyPlus: creating a new-generation building energy simulation program. *Energy and Buildings 33*(4), 319–331.

Crawley, D. B., C. O. Pedersen, L. K. Lawrie, and F. C. Winkelmann (2000). EnergyPlus: Energy Simulation Program. *Ashrae 42*(April), 49–56.

Crites, R. and A. Barto (1996). Improving elevator performance using reinforcement learning. *Advances in Neural Information Processing Systems 8*, 1017–1023.

Davis, R. and R. G. Smith (1983). Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence 20*(1), 63–109.

D'Inverno, M., D. Kinny, M. Luck, and M. Wooldridge (1998). A Formal Specification of dMARS. *Intelligent Agents IV Agent Theories, Architectures, and Languages 1365*, 155–176.

Ephrati, E., E. Ephrati, J. S. Rosenschein, and J. S. Rosenschein (1991). The Clarke Tax as a Consensus Mechanism Among Automated Agents. *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 173–178.

Epstein, J. M. and R. Axtell (1996). *Growing artificial societies: social science from the bottom up.* Brookings Institution Press.

European Energy Agency (2015). Final energy consumption by sector and fuel.

Feng, X., D. Yan, and T. Hong (2015). Simulation of occupancy in buildings. *Energy & Buildings 87*, 348–359.

Fischer, K., J. P. Müller, and M. Pischel (1995). A pragmatic BDI architecture. In *International Workshop on Agent Theories, Architectures, and Languages.*, pp. 203–218. Springer Berlin Heidelberg.

Georgeff, M., B. Pell, M. Pollack, M. Tambe, and M. Wooldridge (1998). The Belief-Desire-Intention Model of Agency. In *International Workshop on Agent Theories, Architectures, and Languages*, pp. 1–10. Springer Berlin Heidelberg.

Georgeff, M. P. and A. L. Lansky (1986). Procedural knowledge. *The Proceedings of the Institute of Electrical and Electronics Engineeris 74*(10), 1383–1398.

Gershuny, J. and K. Fisher (2013). Multinational Time Use Study.

Gill, Z. M., M. J. Tierney, I. M. Pegg, and N. Allan (2010). Low-energy dwellings: the contribution of behaviours to actual performance. *Building Research & Information 38*(5), 491–508.

Gimblett, H. R. (2002). *Integrating geographic information systems and agent-based modeling techniques for simulating social and ecological processes.* Santa Fe Institute studies in the sciences of complexity. Oxford University Press.

Grosz, B. J. and S. Kraus (1996). Collaborative plans for complex group action. *Artificial Intelligence 86*(2), 269–357.

Guerra-Hernández, A. (2005). Learning in BDI multi-agent systems. *Computational Logic in Multi-Agent Systems 3259*, 218–233.

Haldi, F. (2010). *Towards a Unified Model of Occupants' Behaviour and Comfort for Building Energy Simulation.* Ph. D. thesis, École Polytechnique Fédérale De Lausanne.

Haldi, F. and D. Robinson (2009). Interactions with window openings by office occupants. *Building and Environment 44*(12), 2378–2395.

Haldi, F. and D. Robinson (2010). Adaptive actions on shading devices in response to local visual stimuli. *Journal of Building Performance Simulation 3*(2), 135–153.

Haldi, F. and D. Robinson (2011). The impact of occupants' behaviour on building energy demand. *Journal of Building Performance Simulation 4*(4), 323–338.

Harel, D. (1987). Statecharts: a visual formalism for complex systems. *Science of Computer Programming 8*(3), 231–274.

Howard, R. A. (1960). *Dynamic Programming and Markov Processes.* MIT Press.

Huber, M. J. (1999). JAM: A BDI-theoretic Mobile Agent Architecture. In *Proceedings of the third annual conference on autonomous agents*, pp. 236–243.

Huhns, M. N. and M. P. Singh (1998). *Readings in agents.* Morgan Kaufmann.

ISO (2005). *ISO 7730: Ergonomics of the thermal environment Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort criteria*, Volume 3.

Jaboob, S. (2015). *Stochastic Modelling of Occupants' Activities and Related Behaviours.* Ph. D. thesis, The University of Nottingham.

Janssen, M. and E. Ostrom (2006). Empirically based, agent-based models. *Ecology and Society 11*(2).

Jennings, N. R. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence 75*(2), 195–240.

Kashif, A. (2014). *Modélisation du comportement humain réactif et délibératif avec une approche multi-agent pour la gestion énergétique dans le bâtiment.* Ph. D. thesis, Université Grenoble Alpes.

Kinny, D. and M. Georgeff (1996). Modelling and Design of Multi-Agent Systems. *Proceedings of International Workshop on Agent Theories, Architectures and Languages*.

Kinny, D., M. Georgeff, and A. Rao (1996). A Methodology and Modelling Technique for Systems of BDI Agents. In *Proceedings of European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pp. 56–71. Springer Berlin Heidelberg.

Kraus, S., J. Wilkenfeld, and G. Zlotkin (1995). Multiagent negotiation under time constraints. *Artificial Intelligence 75*(2), 297–345.

Langevin, J., J. Wen, and P. L. Gurian (2014). Including occupants in building performance simulation: integration of an agent-based occupant behavior algorithm with EnergyPlus. In *Proceedings of Building Simulation Conference.*

Liao, C., Y. Lin, and P. Barooah (2012). Agent-based and graphical modelling of building occupancy. *Journal of Building Performance Simulation 5*(1), 5–25.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, Volume 157, pp. 157–163.

Maier, T., M. Krzaczek, and J. Tejchman (2009). Comparison of physical performances of the ventilation systems in low-energy residential houses. *Energy and Buildings 41*(3), 337–353.

Menezes, A. C., A. Cripps, D. Bouchlaghem, and R. Buswell (2012). Predicted vs. actual energy performance of non-domestic buildings: Using post-occupancy evaluation data to reduce the performance gap. *Applied Energy 97*, 355–364.

Minsky, M. (1961). Steps Toward Artificial Intelligence. In *Proceedings of The IRE.*

Mnih, V., K. Kavukcuoglu, D. Silver, A. a. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis (2015). Human-level control through deep reinforcement learning. *Nature 518*(7540), 529–533.

Mora, M., J. Lopes, R. Viccariz, and H. Coelho (1998). BDI models and systems: Reducing the gap. In *Proceedings of International Workshop on Agent Theories, Architectures, and Languages*, pp. 11–27. Springer Berlin Heidelberg.

Nagel, K., R. Beckman, and C. Barrett (1999). TRANSIMS for urban planning. In *Proceedings of Computers in Urban Planning and Urban Management.*

Nash Jr, J. F. (1950). The bargaining problem. *Econometrica: Journal of the Econometric Society 18*(2), 155–162.

Newell, A. (1994). *Unified theories of cognition.* Harvard University Press.

Nolfi, S., D. Parisi, and J. L. Elman (1994). Learning and Evolution in Neural Networks. *Adaptive Behavior 3*(1), 5–28.

Norford, L., R. Socolow, E. Hsieh, and G. Spadaro (1994). Two-to-one discrepancy between measured and predicted performance of a 'low-energy' office building: insights from a reconciliation based on the DOE-2 model. *Energy and Buildings 21*(2), 121–131.

Nouidui, T., M. Wetter, and W. Zuo (2013). Functional Mock-Up Unit Import in EnergyPlus For Co-Simulation. In *Proceedings of International Building Performance Simulation Association.*

Page, J., D. Robinson, N. Morel, and J.-L. Scartezzini (2008). A generalised stochastic model for the simulation of occupant presence. *Energy and Buildings 40*(2), 83–98.

Phung, T., M. Winikoff, and L. Padgham (2005). Learning Within the BDI Framework: An Empirical Analysis. *Proceedings of Knowledge-Based and Intelligent Information and Engineering Systems*, 282–288.

Pigg, S., M. Eilers, and J. Reed (1996). Behavioral aspects of lighting and occupancy sensors in private offices: A case study of a university office building. In *Proceedings of ACEEE Summer Study*, pp. 161–170.

Rabin, M., U. Growth, T. Anatomy, U. C. Author, W. J. B. Source, T. American, and E. Review (1993). Incorporating fairness into game theory and economics. *The American Economic Review 57*(3), 1281–1302.

Rao, A. S. and M. P. Georgeff (1995). BDI agents: From theory to practice. In *Proceedings of ICMAS95*, pp. 312–319.

Reinhart, C. (2004). Lightswitch-2002: a model for manual and automated control of electric lighting and blinds. *Solar Energy 77*(1), 15–28.

Robinson, D. (2011). *Computer modelling for sustainable urban design: Physical principles, methods and applications*. Routledge.

Robinson, D. and F. Haldi (2008). Model to predict overheating risk based on an electrical capacitor analogy. *Energy and Buildings 40*(7), 1240–1245.

Robinson, D., U. Wilke, and F. Haldi (2011). Multi Agent Simulation Of Occupants' Presence And Behaviour. In *Proceedings of building simulation*.

Roetzel, A., A. Tsangrassoulis, U. Dietrich, and S. Busching (2011). Context dependency of comfort and energy performance in mixed mode offices. *Journal of Building Performance Simulation 4*(4), 303–322.

Rummery, G. A. and M. Niranjan (1994). *On-Line Q-Learning Using Connectionist Systems*. University of Cambridge.

Sandholm, T. (1993). An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations. *Proceedings of the National Conference on Artificial Intelligence AAAI*, 256–262.

Sandholm, T. and V. Lesser (1995). Issues in automated negotiation and electronic commerce: Extending the contract net framework. *Proceedings of ICMAS95*, 328–335.

Sandholm, T. W. and R. H. Crites (1996). Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems 37*(1), 147–166.

Schwartz, A. (1993). A Reinforcement Learning Method for Maximizing Undiscounted Rewards. In *Proceedings of Machine Learning*, pp. 298–305.

Schweiker, M., F. Haldi, M. Shukuya, and D. Robinson (2012). Verification of stochastic models of window opening behaviour for residential buildings. *Journal of Building Performance Simulation 5*(1), 55–74.

Seligman, C., J. M. Darley, and L. J. Becker (1978). Behavioral approaches to residential energy conservation. *Energy and Buildings 1*, 325–337.

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence 60*(1), 51–92.

Shoham, Y. and M. Tennenholtz (1992). On the synthesis of useful social laws for artificial agent societies (preliminary report). *Proceedings of AAAI92*, 276–281.

Siebers, P.-O. and U. Aickelin (2011). A first approach on modelling staff proactiveness in retail simulation models. *The Journal Of Artificial Societies And Social Simulation 14*(2), 1–25.

Stewart, R. (2004). *Simulation: the practice of model development and use.* John Wiley & Sons.

Sutton, R. S. (1988). Learning to Predict by the Method of Temporal Differences. *Machine Learning 3*(1), 9–44.

Sutton, R. S. and A. G. Barto (1998). *Reinforcement Learning: An introduction.* MIT Press.

Sycara, K. (1988). Resolving Goal Conflicts via Negotiation. In *Proceedings of AAAI-88*, pp. 245–249.

Tambe, M. (1997). Towards Flexible Teamwork. *Journal of Artificial Intelligence Research 7*, 83–124.

Tan, M. (1993). Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents. *Proceedings of the Tenth International Conference on Machine Learning*, 330–337.

Tesauro, G. J. (1995). Temporal difference learning and TD-Gammon. *Communications of ACM 38*(3), 58–68.

Turner, C. and M. Frankel (2008). Energy performance of LEED for new construction buildings.

United Nations (2015). Paris Agreement. In *the United Nations Framework Convention on Climate Change.*

Vorger, E. (2014). *Etude de l'influence du comportement des habitants sur la performance énergétique du bâtiment.* Ph. D. thesis, Ecole Nationale Superieure des Mines de Paris.

Wang, C., D. Yan, and Y. Jiang (2011). A novel approach for building occupancy simulation. *Building Simulation 4*(2), 149–167.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards.* Ph. D. thesis, University of Cambridge.

Watkins, C. J. C. H. and P. Dayan (1992). Q-learning. *Machine Learning 8*(3-4), 279–292.

Widén, J., M. Lundh, I. Vassileva, E. Dahlquist, K. Ellegård, and E. Wäckelgård (2009). Constructing load profiles for household electricity and hot water from time-use data-Modelling approach and validation. *Energy and Buildings 41*(7), 753–768.

Widén, J., A. Molin, and K. Ellegård (2012). Models of domestic occupancy, activities and energy use based on time-use data: deterministic and stochastic approaches with application to various building-related simulations. *Journal of Building Performance Simulation 5*(1), 27–44.

Wilke, U. (2013). *Probabilistic Bottom-up Modelling of Occupancy and Activities to Predict Electricity Demand in Residential Buildings.* Ph. D. thesis, École polytechnique fédérale de Lausanne.

Wooldridge, M. and N. R. Jennings (1995). Intelligent agents: Theory and practice. *The knowledge engineering review 10*(02), 115–152.

Wray, R. E. and R. M. Jones (2006). Considering Soar as an agent architecture. *Cognition and multi-agent interaction: From cognitive modeling to social simulation 33*, 53–78.

Xuan, P., V. Lesser, and S. Zilberstein (2001). Communication Decisions in Multi-agent Cooperation: Model and Experiments. *Proceedings of the fifth international conference on Autonomous agents*.

Zeng, D. and K. Sycara (1997). Benefits of learning in negotiation. In *Proceedings of the fourteenth national conference on artificial intelligence and ninth conference on Innovative applications of artificial intelligence (AAAI'97/IAAI'97)*, pp. 36–41.

Zhang, T., P.-O. Siebers, and U. Aickelin (2011). Modelling electricity consumption in office buildings: An agent based approach. *Energy and Buildings 43*(10), 2882–2892.