Design of a smart occupancy agent that provides information about the number of people in a room

Keune, R.

Award date:
2018

Unit Building Physics and Services
Department of the Built Environment
Den Dolech 2, 5612 AZ Eindhoven
P.O. Box 513, 5600 MB Eindhoven
The Netherlands
www.tue.nl

Design of a smart occupancy agent that provides information about the number of people in a room

Author
Student: R. (Roel) Keune

Student ID:
0791976

Graduation Committee
prof. ir. W. Zeiler
ir. G. Boxem
dr. T.M. Labeodan (msc.)

Date
December 2017

Design of a smart occupancy agent that provides information about the number of people in a room

Keune, R (Roel)

Award date:
December, 2017

Technische Universiteit Eindhoven
Unit Building Physics & Services

Author name: Roel Keune
Student number: 0791976
First supervisor: prof. ir. W. Zeiler
Second supervisor: ir. G. Boxem
Third Supervisor: dr. T.M. Labeodan (msc.)

Design of a smart occupancy agent that provides information about the number of people in a room

Date
21-12-2017

ABSTRACT

The build environment is responsible for 30% of the energy consumption. When no sustainable measures will be taken the energy consumption is expected to rise up to 50% between 2010-2050. One key element to reduce the energy consumption is to increase the energy efficiency of buildings. This can be done by decreasing the occupant related energy consumption of a building. A possibility to improve the relation between the user related energy consumption and the energy efficiency of a building is to exchange fine-grained occupancy information with the Building Energy Management System (BEMS).

The exchange of fine-grained occupancy information with the BEMS can take place with the help of agents. Which is an autonomous piece of software which is capable to work towards a predetermined goal. The goal of this research is to develop an occupancy smart occupancy agent which is capable to obtain fine grained occupancy information and share this information with outside actors and agent, while safeguarding the privacy of the occupants

The occupancy agent is designed with help of the SYSMOD method by Weilkiens (2006). This approach starts with a concept plan and works through a sequence of logical steps to obtain fine a grained design, which displays the different aspects of an occupancy agent.

Different experiments are conducted to establish the reliability of the system. During the experiments the effect of entrance and exit speed, groups of people entering the room after each other and the signal disruption have been tested. Based on these experiments the PIR based system had a reliability 100%-90%, the pressure based system 100%- 74% and the thermal based system had a reliability 58% - 11%.

The result of this study is a smart occupancy agent which is capable to provide fine-grained occupancy information to outside actors and agents. The occupancy agent is capable of keeping track of the number of people in the room in a single way passage, while safeguarding the privacy of the occupant. The occupancy information can be exchanged with outside agents and actors with help of the middle-ware software program JADE.

PREFACE

This report presents my master thesis "Design of a smart occupancy agent that can provide information about the number of people in a room". The thesis is the last step in obtaining the master degree of building physics and services at the Eindhoven university of technology.

This thesis is a further exploration in the field of demand driven control in the built environment. A research field which offers many chances to create a more sustainable and comforTable environment in the future for building occupants. It is therefore my hope that this thesis can have a contribution to the knowledge in this topic.

Writing this master thesis was a challenging task in which I received a lot of help to get to the final product. First of all, I would like to thank my graduation committee for the opportunity they gave me to work on this topic. But even more, I would like to thank them for their guidance and support throughout the whole process. They were always there when I had questions or needed advice, and were the most critical readers of my work. This not only helped to improve the quality of the thesis significantly, it also helped me to grow as a professional person. For this I am very grateful.

Also, I would like to thank the BPS laboratory staff for helping me with the measurement setup and the use of equipment, and the skilled hands necessary to make it all work.

Last but certainly not least, I would like to thank the company Sensor Development (SD) international for giving me the opportunity to rent their equipment for my study. In particularly, I would like to thank Ronald and Remko for their hospitality and introduction to the companies people counter.

This leaves me with a final note to you, the reader of this thesis. I want to thank you for your interest in my thesis, and I hope you will enjoy on its content.

Eindhoven, December 2017

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

The Intergovernmental Panel on Climate Change (IPCC) has measured a fast increase of the number of CO_2 particles over the last century in the outdoor air. This increase of CO_2 particles is one of the leading causes of global warming. The most likely reason for the rise in CO_2 is anthropogenic energy consumption. The build environment is with 35%, one of the largest energy consumption sector in the world (Intergovernmental Panel on Climate Change [IPCC],2014). Half of the total energy consumption in the built environment is used for building services to provide comfort to the occupants inside the building (see Figure 2). According to the report of the International Energy Agency (IEA) "transition to sustainable buildings 2013"(IEA, 2013), the expected energy demand in buildings will further increase up to 50% between 2010-2050 in case no sustainable measures are implemented.

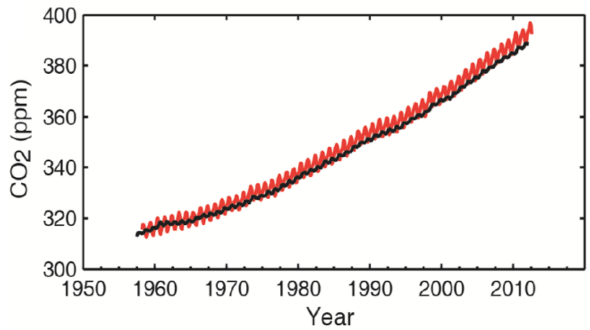


Figure 1: CO_2 in the atmosphere source; (Stocker et al., 2013)

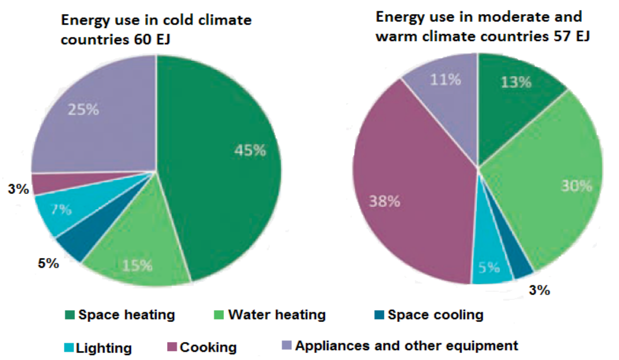


Figure 2: Buildings end-use energy consumption, 2010 source: IEA,2013

One element to reduce the energy consumption is to increase the energy efficiency of buildings. Energy efficiency is defined by the IEA as "*a way of managing and restraining the growth in energy consumption. Something is more energy efficient if it delivers more services for the same energy input, or the same services for less energy input.*" (IEA, 2016). By enhancing the energy efficiency of a building, the energy demand of the building is reduced, and the energy that is needed is used as efficient as possible. This corresponds to the first and third step of the trias energetica (Rijksdienst voor Ondernemend Nederland, 2013)

1.1 Occupants influence on energy

Also, the behaviour of occupants has an important influence on the energy consumption of a building, and is therefore a key element to increase the energy efficiency of a building. An example of the influence of occupants actions on the energy consumption of a building are the actions who people have taken in their indoor surroundings. These actions can be switching on the light, adjusting the temperature or opening a window (Feng, Yan, & Hong ,2015).The difference in behaviour and wishes from occupants towards the indoor environment can have a large influence on the energy consumption of the building. Because of these differences, it is difficult to predict the influence of occupant behaviour on the total energy consumption. Research by Hong & Lin (2013), showed that occupants who are pro active in saving energy, reduce the energy consumption in a private office by 40%. The study defines the difference between work styles by a combination of behaviours. For example, the use of heating and cooling set points, Heating Ventilation And Cooling (HVAC) controls, daylight control, and adaptive opportunities. People with a wasteful work style (occupants who are not intrested in energy use) made less use of the opportunities to change their environment for energy savings. The unwillingness of occupants to activly adopt energy saving measurements in the surrounding is demonstrated in the research of Tiedemann, Sulyma, Sahota, & Hydro (2008). In their study they interviewed 279 occupants if they changed HVAC or lighting settings in their environment, and compared the results with the opportunities the occupants had to change their surroundings. The questionnaires show that a marginal amount of people change the HVAC settings. This study indicates that there is room for improvement to reduce the energy related consumption of occupants in buildings. This is also confirmed in the research of Janda (2014) who states that there is a relation between the user and the energy consumption of a building. This supports an interdisciplinary approach between user and energy to increase the energy efficiency of a building.

A possibility to improve the relation between the users energy related consumption and the energy efficiency of a building is to exchange occupancy presence information with the Building Energy Management System (BEMS) (Macarulla, Casals, Forcada, & Gangolells, 2017). The occupancy information provides an opportunity for the BEMS systems to react on the presence or amount of people in the room, and at the same time the building still can fulfill one of its main purpose, providing comfort to the occupants. This type of BEMS control creates a situation in which the use of energy can be limited to the amount of time that the occupant demands energy from the system. This type of control is labelled as "demand driven control" by Labeodan, Zeiler, Boxem, & Yang (2015).

An example of such a cooperation between occupancy and BEMS systems can be found in the study of Peruffo, Pandharipande, Caicedo, & Schenato (2015), who used a network of lighting sensors to provide a minimum level of illuminance on the working space. The goal of the system was to provide a minimum level of illuminance on their workspace, and to save energy by switching off the artificial lighting when the minimum of illuminance was reached on the desk of the user. A study of Goyal, Barooah, & Middelkoop (2014) demonstrates another use of occupancy information which uses PIR (Passive Infrared) sensors to determine the presence of occupants in a single room at the university of Florida. The occupancy presence information was used by the HVAC control system and led to a reduction of 39% in energy consumption of the room. Both studies only viewed if the room was occupied or not. This boolean vision of the room can be made more fine-grained by counting the number of people in the room, which in turn could lead to a further reduction in energy consumption by improving management of air, heat and cooling in the building.

For a demand driven control to work, the BEMS systems need to be informed about the presence and number of people in a room in order to react. The communication between the occupancy detection system and the HVAC system can be provided by an autonomous piece of software, also called an agent. This agent is capable of receiving the signal from an occupancy detection system and process the signal into information that is readable by other systems or users of the system (actors). A system that makes use of multiple agents which can communicate to each other and their environment is called a Multiple Agent System (MAS) (Wooldridge, 2002).

Because of the potential that agents and MAS can offer to reduce energy consumption by counting the number of people, this research attempts to build an occupancy agent. The occupancy agent has the capability to count the number of people entering and leaving the room in a single way passage. The ability of an agent to receive information from its environment with help of multiple detection methods provides the capability to collect occupancy information. The agents social capabilities make it possible to communicate the occupancy information with other agents.

An important boundary condition of the agents is that it does not has to be able to determine the identity of the occupant. Each occupant is a person that has the right to be protected against violation of their own privacy (General Data Protection Regulation (GDPR) s 1.4 (Eu)). It is for this reason that the occupancy agent should not be able to track the identity of the occupant who is measured.

1.2 Research questions

This research will focus on answering the following questions:

Can a smart occupancy agent be developed to obtain fine grained occupancy information and share this information with outside actors and agent?

In order to answer the main research question the following sub questions are investigated:

1. Which key components can be identified for detecting occupants and sharing occupancy information?

2. How can an occupancy agent be designed that has the capability to obtain occupancy counting information, and shares this information to outside agents and actors?

3. What is the reliability of the developed smart occupancy agent?

1.3 Report thesis outline

To answer the research questions, a systematically approach is needed to get from a concept to a fine-grained design of the occupancy agent. For this, the model-based systems engineering approach SYSMOD is used by Weilkiens, (2006). The goal of SYSMOD approach is to use models to design and communicate the different aspects of a designed product. In this case step 1-5 (Figure 3) are used to design the occupancy agent.

The following step is performing experiments (step 6) on the occupancy agent to gain insight into its reliability. During the experiments, the effect of different variables are investigated to determine their influence on the occupancy agent.

In step 7, the occupancy agent is evaluated based on the outcomes of the experiments. The discussion provides the possible explanation for the behaviour of the occupancy agent and its performance. Based on this reflection the research question is answered, and recommendations for future research are given.

1.4 Methodology

Step 1: context/ background
A literature study in the field of demand driven control is conducted to provide context for this study. With this literature research, insight is gained in previous work on this subject, and how this relates to this thesis.

Two key systems are discussed which together form the basis for a demand driven system. The occupancy detection method, that shows the importance of counting occupants in a room, and the agent.

The first system is the occupancy detection system, this is the part of the occupancy agent that recognizes an occupant from its environment. A sensor is needed to distinguish a person from its environment. The literature study examines multiple sensors used in research done in the field of occupancy detection, which could be used to distinguish a person from its environment. This information can be used to determine which sensor has the most potential to count the number of people in a room, and can therefore be used in this research.

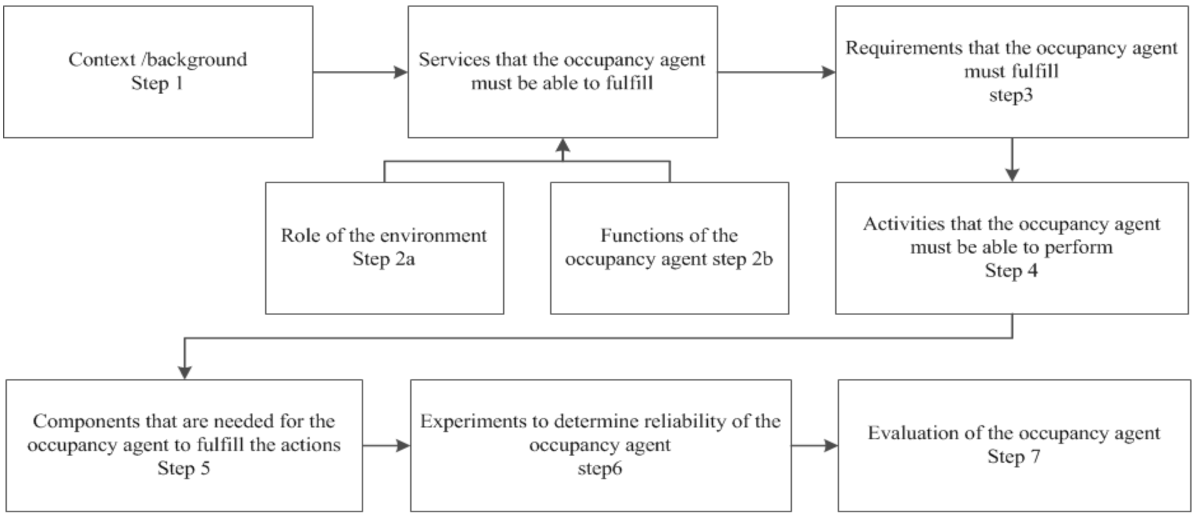


Figure 3: Design methodology steps

The second key system is the agent. The agent is responsible for communication between the detection system and other agents. An occupancy agent has the potential to improve the collection and communication of occupancy information. In order to gain knowledge on this potential a literature study is conducted, which investigated the different types of agents and their properties.

Previous research shows that there is a potential for the use of demand driven control application to reduce the energy consumption of a building (Nguyen & Aiello, 2013). However, previous research has focused on presence detection, counting the number of people in the room would provide more detailed occupancy information. This in turn can be used to decrease the energy consumption even more. For this reason an occupancy agent is built, which makes use of multiple occupancy detection methods to determine the potential use of occupancy counting. An agent is used as a communication tool between the detection system and outside agents. The occupancy agent's potential is examined by several experiments.

Step 2a: Context analysis
The environment is important to be taken into account, for the occupancy agent to function correctly . This relation is explored with help of a context diagram. The context diagram creates insight in variables have an effect on the performance of the occupancy agent.

Step 2b: Services analysis
Besides the environment in which a system needs to operate, it is also important to know what kind of services the system needs to be able to provide. The services are needed to get the desired output of the system to other entities who want to retrieve this information. The use case diagram is used to provide insight in the different services and for whom these services are.

Step 3: Requirements
The goal of the requirements is to take the results from the environmental analysis and the service analysis and translate those results to requirements, which the occupancy agent must fulfill. The requirements are the base for the next steps in the process, in which each step is reflected back if it fulfills the requirements.

Step 4: Activity diagram
The goal of the activity analysis is to translate the requirements into actions, which the occupancy agent must be capable to perform. This is done with the help of the activity diagram. The result of the activity diagram is an overview of the main flow and alternative flow of the system.

Step 5: Component diagram
The component diagram translates the different functions to components that are needed to perform the different actions described in the activity diagram, and the relationships the components have with each other. Besides the component diagram, more detailed information is given on the selection of the different sensors with help of the Kesselring method (see §3.5.1), the microprocessor and the possibilities of the JADE agent. With this information the component diagram provides insight on how the occupancy agent functions on a component level.

The final step in the design of the occupancy agent is the physical connection between the different components. In contrast to the component diagram, the circuit diagram displays the location of the wires, resistances and other components instead of their relation. This is especially done for the connection between the sensor and the microprocessor.

Step 6: experiments
In the sixth step, experiments are performed to test the reliability of the different detection systems. The occupancy detection system of the occupancy agent is responsible for counting the number of people in the room. It is for this reason that this part of the system is subjected to a number of experiments, in which the effect of the following variables are tested:

1) The effect of entrance and exit speed.
2) The effect of groups entering and leaving the room.
3) The effect of signal disruption.

Each of the different variables is tested multiple times for each occupancy detection system. The outcome of the experiment is described in chapter 4.

The information on the reliability of the occupancy detection systems is used by the agent to determine the reliability of a new measurement.

Step 7: Evaluation of the occupancy agent
When the occupancy agent is built and tested an evaluation can be made in which the performance of the occupancy agent can be assessed. Subsequently, the evaluation answers the research questions. The evaluation of the occupancy agent is divided into five different aspects:

1) State the key findings obtained from the experiments.
2) Errors which were detected during the experiments.
3) Combination of multiple occupancy detection systems.
4) Limitations of the occupancy agent.

With these points in mind, it is possible to make recommendations for possible future research and determine if the used approach has led to answer the research questions.

To answer the research question, this study has built an occupancy agent which has the capability to inform other actors on the occupancy of a room. In the first step, a review on the previous research is made, which is done into the field of demand driven control. This displayed the importance of occupancy counting and formed the context for the occupancy agent design. The agent was further developed with the help of the tailored made SYSMOD approach. Next, each occupancy detection system was tested for different aspects to get insight into the reliability of each system. Finally, the design and the results of the experiments are evaluated in a discussion, so it becomes clear to what extend the research goal is met, and where more research is needed to improve the occupancy agent.

CHAPTER 2: BACKGROUND

The goal of the literature study is to gain insight into the current state of the field of demand driven control. This provides context for which environment the occupancy agent should function. This chapter is divided into two different section, namely §2.1 and §2.2 .

§2.1 focusses on the applicability of sensors. The sensor has to be able to distinguish an occupant from its environment, and needs to determine the direction of the occupant. In literature there are multiple sensors used to detect people in a room. In the segment below, further detailed information is given on the different sensors, which have the ability to detect a person form its environment.

§2.2 provides more information on the agent's communication of the information obtained from the sensors to outside entities (e.g. other agents or actors). The first part of this section focusses on what the properties of an agent are. Where the second part discusses the different types of agents that can be distinguished. The obtained knowledge on the agent paradigm can be used to determine what role agents might play in this study.

At the end of the chapter the importance of occupancy detection is know. And, the possible role which an agent can fulfill to collect and communicate the occupancy information to other agents.

2.1 sensors

A sensor is necessary to distinguish people from their environment. In this section an overview is given on the most used sensors in research and the different studies which are done using those sensors. This information is used to determine which sensor can be used in the development of the occupancy agent in chapter 3. The previous research also displays the role that occupancy detection can have on the reduction of energy consumption in the built environment.

2.1.1 passive infrared (PIR) sensor

The first method to detect a person is with the use of infrared. An infrared sensor detects the change in infrared radiation in its field of view. Because all heat sources sent infrared radiation a filter is placed on the infrared sensor so it only detects the infrared pattern from living organisms, like animals and humans (Erden, Ziya, & Enis, 2015). When the movement of a person takes place the sensor can only respond with ON or OFF signal.

This principle of measurement is used in multiple studies in the field of occupancy detection. An example is the research of Duarte, Van Den Wymelenberg, & Rieger (2013) in which they used a PIR sensor to detect the occupancy of an 11-story commercial office building. In their research, they found that the occupancy of the building was between 0%-60%, indicating that there is a potential to save energy when a room is not used. The research of Wahl, Milenkovic, & Amft (2012) investigates the energy potential when using a PIR-based counting system in an office, and concluded that PIR sensor can be used to count the number of people in the room.

2.1.2 Ultrasonic sensor

Often used in combination with the PIR sensor is the ultrasonic sensor. This sensor sends out ultrasonic sound which is reflected by objects. The time it takes for the ultrasonic sound to travel from the object and back determines the distance from the object.

This technology is used in the study of Hnat, Griffiths, & Dawson (2012) in which they used an ultrasonic sensor, placed on the ceiling near a door to determine the height of people as they were entering and leaving the room. With this technology, they reached a detection rate of 90%.

However, the research of Labeodan, Zeiler, et al. (2015) mentions that this sensor type has the disadvantage that it easily reacts to disturbance in the room like moving air or the movement of non-human objects like a piece of paper.

2.1.3 Vision-based sensor

The vision based sensors use cameras to detect people in the room. Different methods to discriminate an occupant from other objects are used. In the research of Benezeth, Laurent, Emile, & Rosenberger (2011), the background was subtracted from each image, so the remaining (moving) objects can then be traced throughout the field of view of the camera. This led to a detection rate of 83%.

Another used technique to distinguish occupants with help of a vision based sensor is described in the research of Diraco, Leone, & Siciliano (2015). In their study, a point cloud technique is used to detect people in a room with a 3D depth sensor. This technique led to a detection rate of 97% and an energy saving of 27%.

The results of both studies were reported to be reliable to determine the number of people present in an image of the camera. Although the positive result obtained by this technique, the researchers in each of the studies reported that because the image of the occupant was recorded the privacy of the occupants might form a problem with this method of detection.

2.1.4 Pressure based

The principle of a pressure based system is to measure a person's weight in order to detect their presence in the room. An example in which a pressure sensor is used can be found in the research of Fernandez-Luque, Martínez, et. al., (2013) in which the goal of the research was to detect if a person is lying in a bed. As soon as a predetermined weight was put on the pressure sensor, it would send a signal that a person was lying in his bed.

Another study done is the research of Labeodan, Zeiler, et al., (2015). In their study they used a pressure sensor to detect people sitting in a chair. The performance of the chair was evaluated in a conference room and showed a reliance of the system up to 100%.

Both studies show that there is a possibility to use a pressure sensor as a method to measure the presence of occupants inside of a room.

2.1.5 CO_2 based sensor

Another method to determine the number of people in the room is to measure the CO_2 concentration, which is produced by people when they exhale. A higher concentration of CO_2 in the room indicates more people in the room, and a lower CO_2 concentration indicates a low occupancy of the room.

Gruber, Trüschel, & Dalenbäck (2014) concluded in their study that CO_2 measurement could be a well-suited technique for estimating the number of occupants in a room. Goyal et. al. (2014) found that when the HVAC control was combined with the occupancy measurement of the CO_2 measurement this led to 40% energy saving.

However, the research of Dedesko, Stephens, Gilbert, & Siegel (2015) illustrates that occupancy detection, with help of CO_2 measurements, can lead to a large deviation of the number of people that enter and leave the room. Besides, the research of Labeodan, Aduda, Zeiler, & Hoving (2016) show that in order to count the number of people in the room information should be available on: the supply air flow rate in the room, CO_2 generation by the occupants, the CO_2 concentration of the supplied air and the concentration of CO_2 in the room. Furthermore the response time to calculate the occupancy in the room is taking a long time. This is because the CO_2 exhaled by the occupant needs to pass the sensor (Dedesko et al., 2015; Labeodan et al., 2016). These aspects makes CO_2 measurements a less suited technique to measure occupancy in a room.

2.1.6 Occupancy detection on the basis of appliances

Besides detecting the occupants in a room, it is also possible to measure the applications which the occupants use to detect their presence. Early examples focus on the use of telephones and workstation equipment to discover the presence of occupants in a room. More recently the focus is shifted towards the use of laptop and mobile phones, which are connected to the Wi-Fi of the building. Melfi, Rosenblum, Nordman, & Christensen (2011) used the IP addresses from applications used by occupants to determine their the presence and direction. This led to an accuracy of occupancy detection between 31%-94%.

The reason for false measurements was, among others, the problem of overlapping Access Points (AP). An additional error encountered in the research was that the AP's were not bound to physical boundaries like walls. They also discovered that false measurements occurred due to the inconsistency of the mobile internet signal. Another problem when using Wi-Fi signals to detect people is the assumption that each person has only one device which is requesting Wi-Fi, while this does not have to be true for all the occupants.

2.1.7 Temperature sensor

The temperature of people is higher than the temperature of the background (Katić, Li, Kingma, & Zeiler, 2017). This principle can be used to distinguish an occupant from its environment. So far, most research focussed on the use of temperature measurement to develop a personal cooling and heating system. An example can be found in the research of Veselý, Molenaar, Vos, Li, & Zeiler (2017). In their research the thermal system is used to measure the temperature of the hands for a longer period of time in the field of view of the sensor.

In another research M Veselý & Zeiler (2014) managed to track the movement of the fingertips of a person while the temperature remained above 30°C. These studies implicate that there is a possibility that this technology can be used to count the number of people in a room.

2.1.8 Radio-frequency identification (RFID)

Finally, there is the possibility to track occupants by radio-frequency identification(RFID). Each occupant has to be equipped with an RFID tag that sends out a radio-frequency signal which identifies the occupant in the room, for this sensor to function. In the research of Zeiler, Labeodan, Boxem, & Maaijen (2014) RFID tags were used to measure the activities of personnel in a commercial office building. The result was a detection rate of 85%.

The main reason for measurement faults was that the people in the research sometimes forgot their RFID tag which led to false measurements. One of the disadvantages of the RFID tag is that it needs to be worn by the occupant while visiting the room. This means that the system misses occupants who visit the office, enhancing the error made by the system. Also, each occupant location can be identified through his RFID tag, which raises privacy problems

Table 1 summarizes the results found for the different sensors that are used to distinguish people from their environment, towards the capacity of the sensors to measure occupants by their own.

Table 1: Occupancy detection sensors overview

Sensor	Complications when used for occupancy counting
PIR based	Can only measure the presence of an occupant
Ultrasonic	Is easily distracted
Vision based	Privacy problems
Pressure based	Is only tested for the presence of occupants
CO_2 based	Low response time
Application based	Can only measure an occupant when the application is used
Temperature sensor	Has potential to track hand, but is not tested to count occupancy
RFID	Privacy problems, and only detects people wearing a RFID tag

2.2 Agent definition

When a change in occupancy is detected by the occupant detection sensor, the information needs to be shared with outside actors, which is done with the help of agents. This section will give more elaborate information on what an agent is and what type of agents can be identified. The first part of this section will describe what the characteristics are of an agent in general. In the second section more detailed information is given on the different type of agents that can be distinguished.

An agent is described as "*a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.*". Wooldridge (2002). To reach the design objectives, the agent needs to be aware of its environment and needs to react to that environment towards a predetermined goal. A third premise that is not included in the definition by Wooldridge is that the agent also needs to have social capabilities, such as the capability to interact with other agents (Jennings & Wooldridge, 1995). An overview for the different properties of an agent can be found in Table 2.

Table 2: Premises what an agent is according to (Jennings & Wooldridge, 1995; Wooldridge, 2002)

1	A computer system that has predetermined goals
2	Is capable of autonomous action in the environment
3	Has to be able to have social capabilities to interact with other agents

With the knowledge of what an agent is Labeodan, Aduda, Boxem, & Zeiler (2015), make a distinction in their research between different types of agents. The first type of agent is the simple reflex agent (see Figure 4). This type of agent receives its information about the environment form sensors. Based on the measurement of the sensor an actuator gets triggered which performs an action in that environment. More complicated agents also have a data management layer in between, which can make decisions by comparing options on the basis of different goals, for example comfort or economical basis. The different type of agent give understanding in the structure which an agent can have.

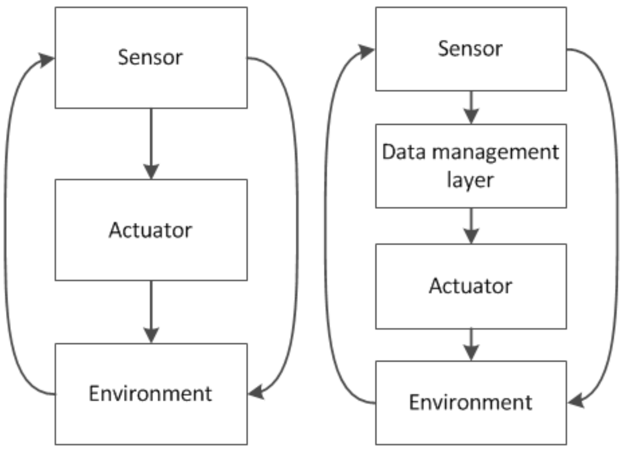


Figure 4: Simple react agent (l) agent with data management layer (r)

2.3 Discussion

This chapter describes several sensors which can be used to detect occupants in a room or building. When these sensors were combined with the HVAC-L (Heating Ventilation and Cooling - lighting) system of a building they have the potential to reduce the energy consumption of the building.

Previous research that is done on the use of coarse grained information, which only measured if the room was occupied or not. However, the use of more fine grained occupancy information can lead to an improved use of BEMS in a building. More fine grained information could be obtained by knowing the number of people in the room. (Labeodan, Zeiler, et al., 2015)

For this reason, an occupancy agent is developed that has the capability to count the number of people entering and leaving a room through a single way passage. The ability of an agent to receive information from its environment provides the capability to collect occupancy information with help of multiple detection methods. And, the agents social capabilities make it possible to communicate the occupancy information with other agents.

3 OCCUPANCY AGENT DESIGN

The occupancy agent is developed with help of a tailor-made SYSMOD approach, which is visualized in Figure 5 (Weilkiens, 2006). This Figure describes the sequence of steps taken in this chapter to develop a functioning agent. The Figure is further elaborated in the section below.

The first step in the design process is to examine the environment in which the occupancy agent must function. This is done with the help of a context diagram.

The second step is to analyse the different services, which the occupancy agent must be able to perform. For this step, a use case diagram is used that displays the different function and the relationship with other services.

In the third step, the requirements of the occupancy agent are determined using a requirement diagram.

The fourth step is to break down the requirement into actions which are needed to fulfil those requirements. An activity diagram is used to display which actions are needed to be performed, and in which sequence these actions need to take place.

One of the basic functions of the occupant agent is to detect people, which can be done with the help of sensors. In step 5A, a pre-selection is made between the different sensors which are fit for this purpose with help of the Kesselring diagram. This is a tool to determine the optimum between the functional aspects and realization aspect of each sensor.

Besides the sensor, there are other components needed to fulfill the different actions. A component diagram is used to visualize the different components which are needed in order to make the occupancy agent function. The component diagram also displays the relationship among each component. Thereby giving a clear insight in the occupancy agent on a detailed level (step 5B and 5C).

3.1 Step 1: Role of the environment on the occupancy agent

The environment wherein an occupancy agent must function is partly determined by the goals that the agent needs to achieve. The occupancy agent must be able to obtain information from the environment, and use this information to deliver the desired occupancy information to outside agents or actors. The context diagram (Figure 6) displays the relationship which the occupancy agent has with its environment.

An example of a relationship where the occupancy agent needs to obtain information from the environment is when occupants enter the room, the occupancy agent needs to be able to detect the occupants in its environment.

The occupancy agent can distinguish an occupant by measuring unique properties of a person in the environment. For instance, to recognize an occupant on the basis of temperature, the occupancy agent needs to obtain information on the temperatures which belongs to an occupant, as well the temperatures belonging to the environment. The environmental temperature is lower than that of the occupant. When a higher temperature is

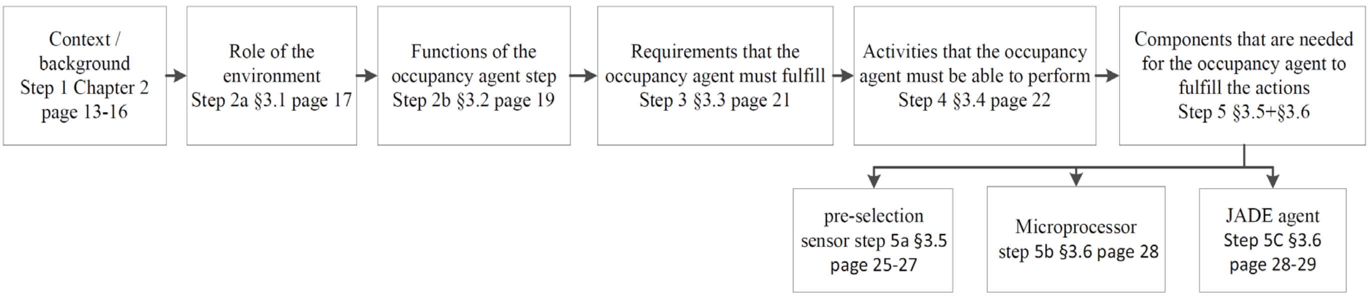


Figure 5: Design methodology steps (1-5) that describe the deign of the occupancy agent. step 6 and 7 are described in chapter 4 and 5

measured the agent should report that a person is present in the field of view. This process is only possible when occupancy agent retrieves information from the environment.

In the same way, the occupant can be detected on the basis of different other variables. Examples of such variables could be: the difference in pressure, in line of sight, CO_2 level in the room, the equipment which is used in the room or infra-red radiation. With help of the input from the environment, it is possible for the occupancy agent to use this information to deliver the desired output.

Besides the input the occupancy agent needs from its environment, it also needs to be capable of providing specific output to outside agents and actors. Figure 6 shows a context diagram that presents an example of a relation between an occupancy agent and a lighting agent. This relation is a reference for the relation which the occupancy agent has with other agents.

The output of the occupancy agent are different messages which the occupancy agent must be capable of sending to other agents outside the system. The messages which the occupancy send are:

-Inform other agents on the number of people in the room.
-Inform other agents on the average number of people in the room.
-Inform agent on the RMSE of the number of people in the room.
-Inform other agents on the time of measurement.
-Inform other agents on the reliability of the measurement.

Based on these messages the occupancy agent is capable of delivering fine-grained information to other agents.

In order to build a functioning agent, the analysis of the context diagram indicates the need for multiple context dependent requirements. Table 3 shows a summary of these requirements.

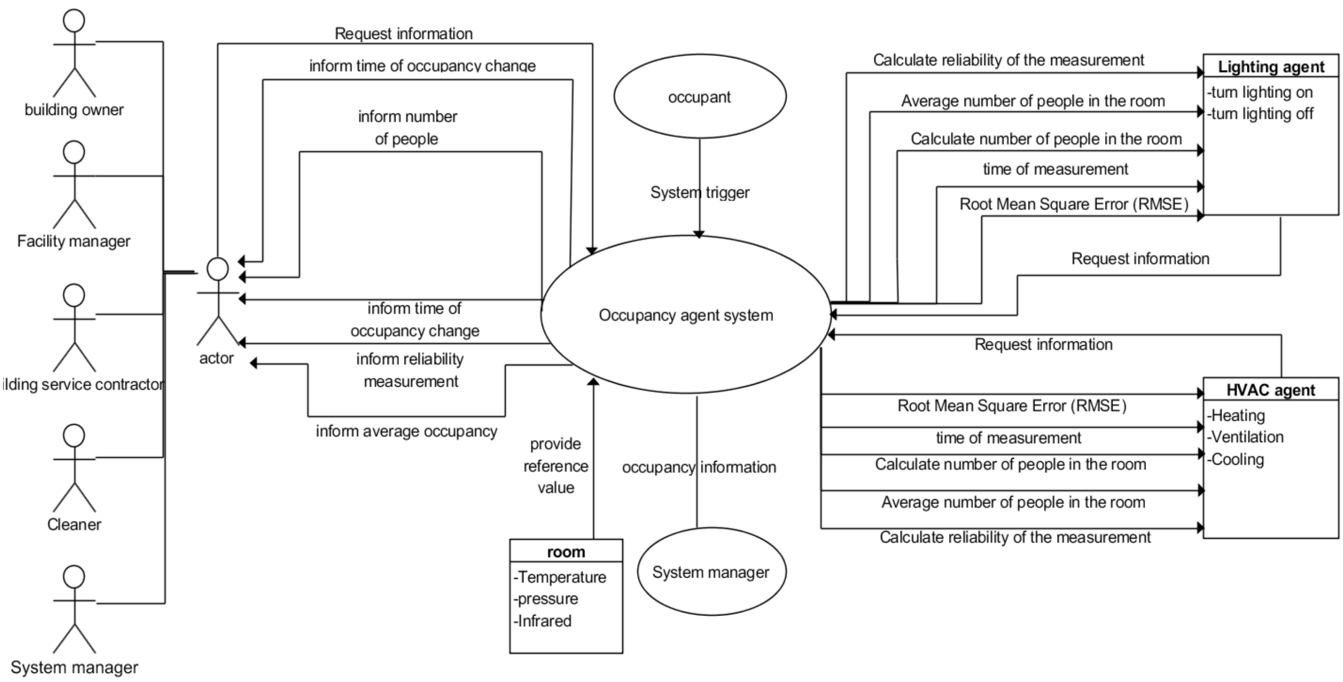


Figure 6: Context diagram, which displays the interaction between the environment and the occupancy agent.

Table 3: Context depended diagram requirements

Relation with between the occupancy agent and the environment	
1	The occupancy agent must be able to receive input from its surroundings.
2	The occupancy agent must be able to distinguish a person from its surrounding.
3	The occupancy agent must be able to translate the received signals from the surroundings into occupancy information.
4	The occupancy agent must be able to communicate information with other agents and actors.

3.2 Step 2: Analysis services which the occupancy agent has to be able to perform

The context diagram displayed the interaction between the occupancy agent system and the environment it must function in. Figure 7 shows the use case diagram and displays the services that the occupant agent has to provide, in order to establish the desired interaction the occupancy agent system has with its environment.

The services, which the occupancy agent must fulfill are depicted as an ellipse, also called a use case. All of the use cases can be found inside the occupancy agent represented as a square box. The different services are performed for outside actors, the use case diagram displays which action communicates with what outside actor or agents. Because these actors and agents are not part of the occupancy agent system, they are placed outside of the system.

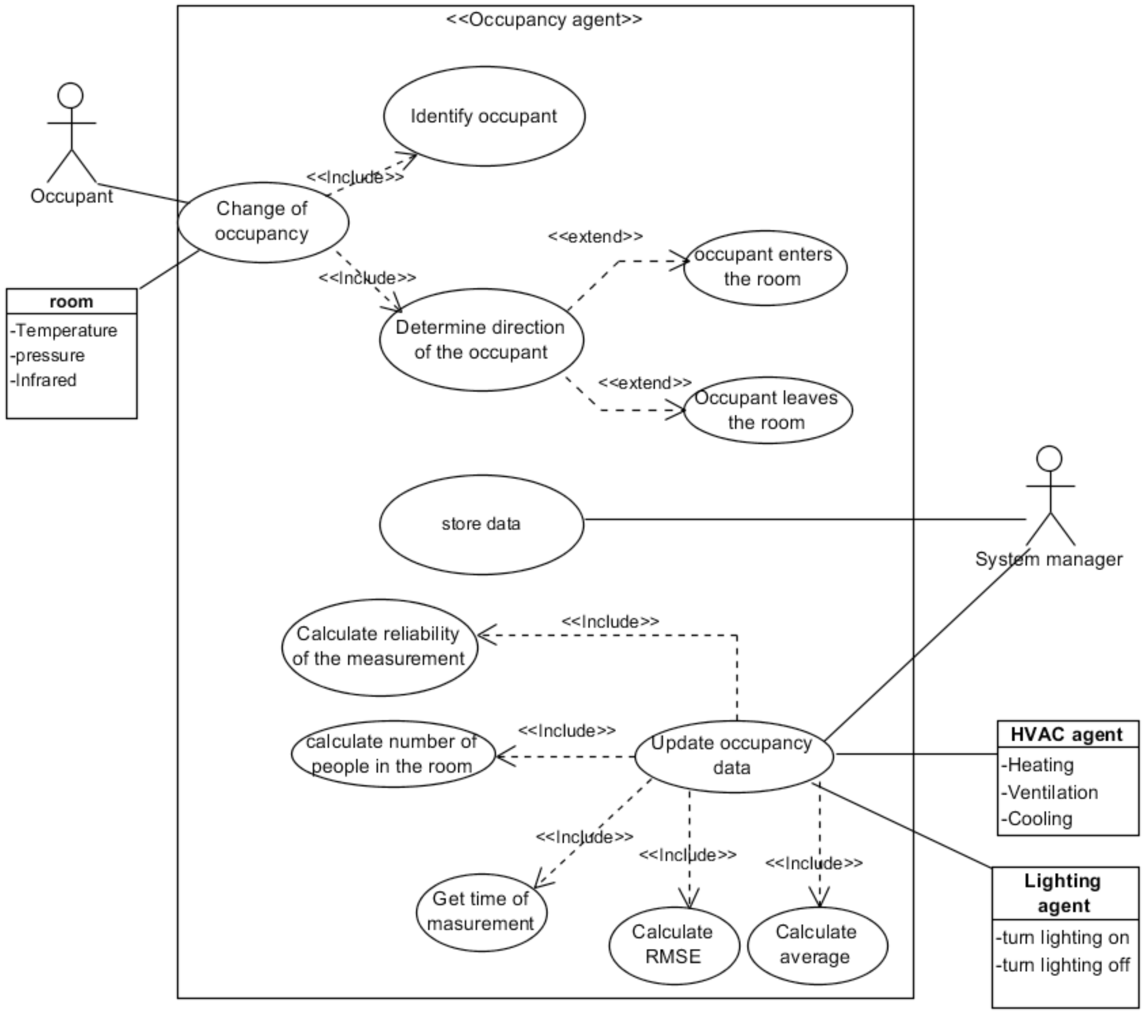


Figure 7: Use case diagram, displaying the services that the occupancy agent must fulfill

The use case diagram can be divided into two different parts. The first part displays the input which the system needs to be able to receive (Figure 8). And, the second part looks at what output the occupancy agent must be capable of producing (Figure 9). More information on both parts can be found in the section below.

Figure 8 shows the movement of the occupant as an input for the occupancy agent which triggers the change of occupancy use case. Table 4 describes the corresponding use case essence.

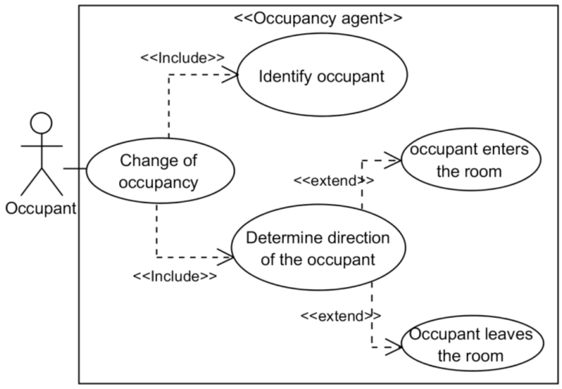


Figure 8: Input use case diagram

When the occupancy agent has identified that a person has entered or left the room the input use case diagram (Figure 8) is finished. This leaves a situation in which the occupancy agent should process this information, and share the information with other occupants as output.

The update occupancy data use case is described in Figure 9, together with its corresponding use case essence in Table 5. This use case is responsible for the output of the occupancy agent. The output use case diagram takes place when the occupancy agent has determined that a person entered or left the room, and ends when an update has been send to the outside agents and actors.

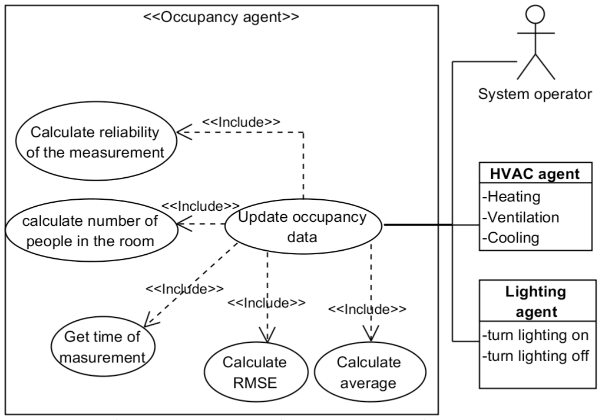


Figure 9: Update occupancy data use case diagram

The update from the occupancy agent can be used by outside actors. Based on the occupancy information, the actors perform their own actions. An

Table 4: Use case essence: change of occupancy

Title:	Change of occupancy
Description:	When a person is entering a room it needs to be identified as a user of the room. To identify a person entering or leaving the room a sensor is needed to translate a physical signal, that is given off by a person, into an electrical signal
Actors	Occupant
Basic flows	
Trigger:	User is entering a room
Step 1:	Sensor measures the occupant (see table 1)
Step 2:	The signal of the occupant is received by the system (see table 1)
Step 3:	The systems identifies the incoming object as a person
Step 4:	The occupancy agent determines which direction the person is going
Post condition	
The movement of the occupant is determined, this information can be used to transfer to other agents.	

Table 5: Use case essence: update occupancy data

Title:	Update occupancy data
Description:	When the occupant is identified and the direction in which the occupant is moving is determined, the occupancy information needs to be updated. This update exists of several steps which, (after the update) need to be communicated with entities who want to receive the information.
Actors	Entities who want to receive occupancy information.
Pre-condition	
An occupant is detected, and the direction which the occupant is going is determined.	
Basic flows	
Trigger:	Change of occupancy use case
Step 1:	Obtain time of measurement
Step 2:	Update number of people in the room
Step 3:	Calculate average number of people in the room
Step 4:	Calculate RMSE of the room
Step 5:	Calculate reliability of the measurement (see §3.4)
Step 6:	Send information to outside entities
Post condition	
The entities who want to receive the occupancy information are informed	

example in the use case diagram is the HVAC agent, which can decide to change the ventilation of the room based on the new information. What an outside agent can do with the occupancy information is outside the scope of this research.
When the update is send to the outside actors and agents, the occupancy agent waits until the next occupancy change takes place. If a change in occupancy takes place, then the system will be triggered again, and the process repeats itself with a new update as a result.

3.3 Step 3: Requirement analysis

At the requirement analysis, the context analysis and services of the system are translated into requirements. The requirements form the bases on which the rest of the design process will be tested. For this analysis a requirement diagram is used (see Figure 10, Figure 11 and Figure 12).

The requirement analysis results in three types of requirements. The first type of requirements places demands on the detection of the system. The second type of requirement is that the system needs to safeguard the privacy of the occupants who are measured. And, the last type of requirements places further specifications on the communication between the system and the outside actors.

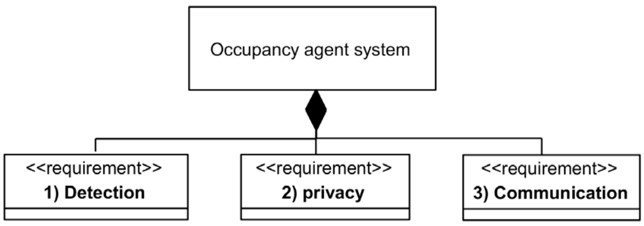


Figure 10: Requirement diagram

1) Detection requirements
The detection requirements are split up into three different requirements:

1.1) The system must be able to identify occupants from other objects.
1.2) The system should be able to determine the direction in which an occupant is moving.

1.3) The system must be able to keep track of the number of people which are in the room.

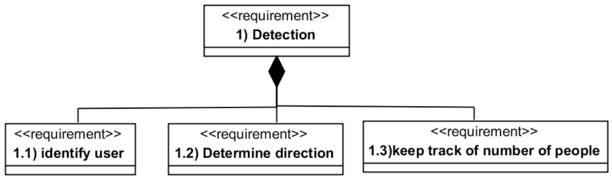


Figure 11: Detection requirements

2) Privacy requirement
One of the dangers of occupancy measurement is that personal information is gathered that contains private information, which can be mistreated by unwanted entities. This problem is already mentioned in several occupancy sensors studies, that track the movement of people. Li, Calis, & Becerik-gerber (2012) mention in their research on the possibility to use RFID about privacy "*such systems can raise serious privacy concerns, as occupants may resist the collection and analysis of their images*."

And, they are not alone in this view, Zeiler et al. (2014) also state that gathering occupancy information can lead to privacy issues when using a RFID sensors. The awareness of privacy is not only recognized by researchers but is even chaired by platforms like IEA rapport "repowering the market" (International Energy Association, 2016).

The aforementioned reason shows the need for an occupancy detection system to not only be reliable, but also safeguarding the privacy of occupants, who are measured by the system. This is achieved by ensuring that the applied occupancy methods used, are not traceable to a single person, but instead measure a general element which applicable for all people.

3) Communication requirements
The gathered information by the occupancy detection system must be communicated with other agents and outside actors. The use case diagram already displays the different aspects that the occupancy agent must be capable to communicate with outside actors. These use cases are converted to requirements displayed in the requirement diagram of Figure 12.

Figure 12 shows how several requirements have a "derive" relationship with other requirements. These requirements depend on the requirements from which they derive. An example is that the number of people depends on the previous amount of people in the room. This type of relation can also be seen in other requirements.

3.4 Step 4: Activities which the occupancy agent must fulfill

The first step to translate the requirements into a functioning occupancy agent is by analysing what actions are needed to fulfill the requirements. An activity diagram (Figure 13) is used to get insight in what actions need to be performed, and what the relation between the different actions are. This section describes, with help of the activity diagram, the different steps which the occupancy agent needs to take in order to fulfill the requirement of the system.

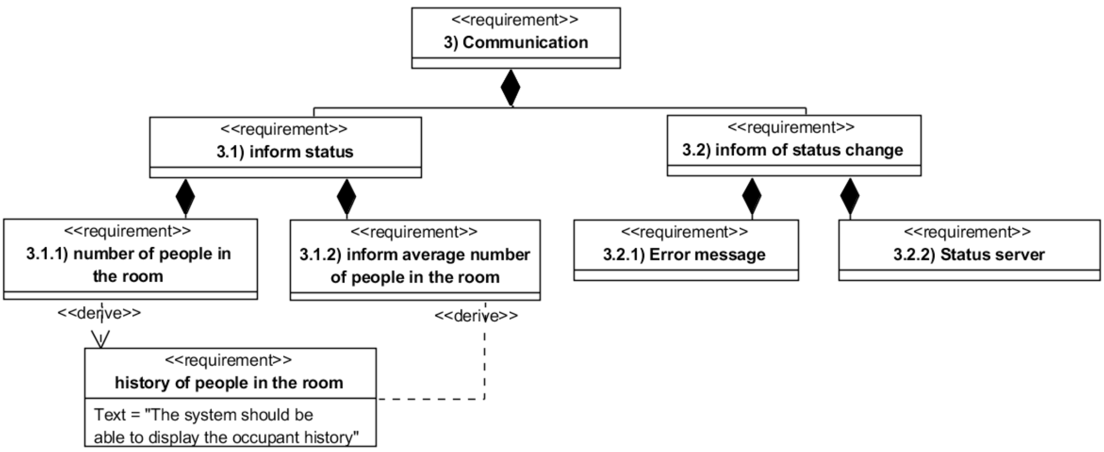


Figure 12: Communication requirements

The activity diagram starts in the upper right corner and ends in a situation in which an occupant is measured and the updated information is sent to the outside actors. This is done through a series of actions which the occupancy agent has to take in order to achieve the different requirements. All actions are completed when the final end node is reached at the bottom of the diagram. In this case, the systems waits for another measurement to occur. Some of the actions taken in the activity diagram are further elaborated in the text below and are denoted with a number in front of them.

1) Detect occupant with occupancy detection systems: When a change in occupancy occurs, the occupancy detection systems must determine if the passing object is a person. Each of the occupancy detection systems distinguishes a person by measuring their own specific property of a person.

2) Determine direction with occupancy detection systems: If the systems detect the object as a person, the direction of that person needs to be specified. This needs to be done to determine if the person is entering or leaving the room.

3) Determine success chance with occupancy detection systems: When a person is measured entering or leaving the room the reliability of the occupancy detection system is calculated for the next

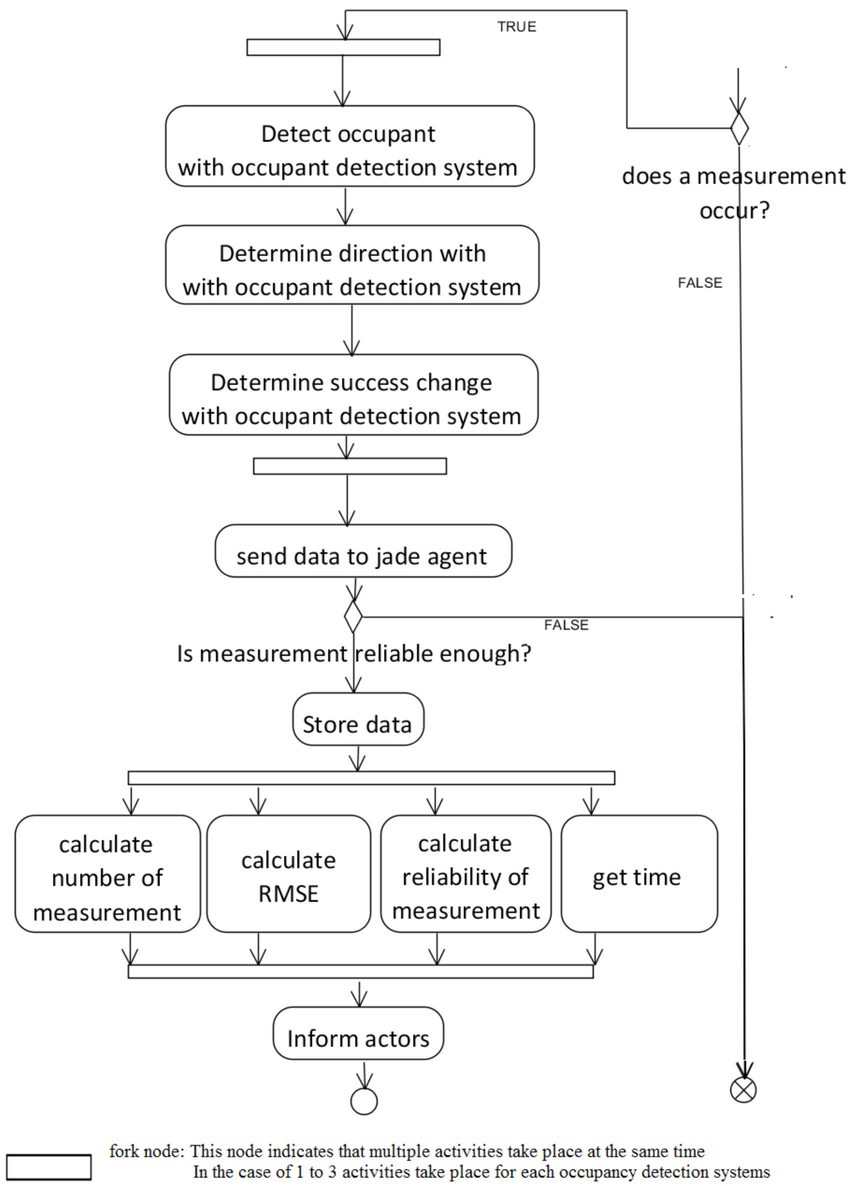


Figure 13: Activity diagram, displaying the different action that the occupancy agent must perform to fulfill the requirements

steps. This way the combination of different occupancy detection system can be used to increase the reliability of the occupancy agent. The reliability of each occupancy detection system is obtained through experiments.

4) Update occupancy information
When the occupancy detection systems measure a change in occupancy, then an update is needed to inform the outside actors of the new situation. Each update exists of the following actions:

A. Calculate the number of people in the room
When a person is entering or leaving the room, an update is sent of the changing number of people in the room. This input is used to calculate other occupancy information like the RMSE.

B. Root Mean Square Error (RMSE)
The Root Mean Square Error calculates the percentage by which the new measurement(s) deviate from the expected result. The goal of calculating the RMSE is to give insight how the number of people in the room relates to the expected number of people in the room. The RMSE is calculated with the help of formula 1.

$$RMSE[\%] = \sqrt{\frac{|(\bar{x} - x)|^2}{n}} * \frac{100 * n}{\sum x_{(n)}}$$

$\bar{x}$ = expected value
x = measured value
n = number of measurements

Formula 1

C. Calculate number of measurement
Part of the RMSE calculation is the expected number of people in the room. In this study, the expected number of people is resolved by using the average number of people in the room. The average is calculated on the basis of the total number of people in the room divided by the number of measurements. The expected number of people in the room can be optimized, however, this is beyond the scope of this study.

D. Calculate reliability of measurement
The reliability of the different occupancy detection systems are determined by experiments described in chapter 4. The result from those experiments make it possible to send this information to the outside actors and agents. There is also the possibility to combine the results from each occupancy detection system to gain a more reliable result. This can be done with the help of Formula 2, which calculates the reliability of a measurement when multiple occupancy detection systems are used.(Berkum & Bucchianico, 2009; Montgomery, 2003).

$$P_{A \to B} = 1 - \prod_{i=1}^{n} (1 - P_i)$$

Formula 2

$P_{A \to B}$ = measurement reliability
P = reliability detection system
i = number of detection system

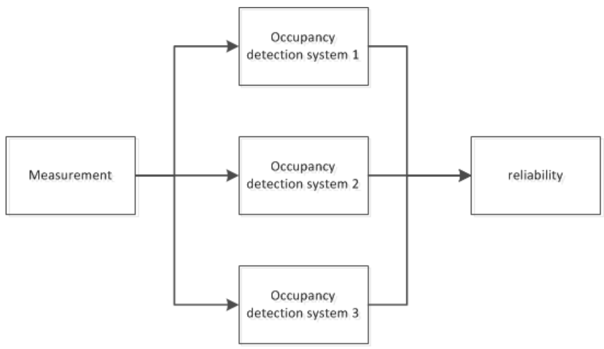


Figure 14: System probability calculation

When a change in occupancy occurs, each of the detection systems will measure this change. When a detection system does not measure a change in occupancy, then it will not send a signal. The consequence of a system that does not detect a person is that the reliability of the measurement will decrease (see also formula 2).

E. Get time of measurement
When a measurement takes place, the time of that measurement is recorded. The recorded time can be combined with the time of occupancy change to make predictions about the future occupancy in the room.

3.5 step 5a: Sensor pre-selection

One of the of the key activities of the occupancy agent is the detection of people entering or leaving a room. The system can use multiple sensors for this. A pre-selection is made on what sensors have the most potential for this activity. The Kesselring method is used as tool for this pre-selection . Experiments are performed on the selected sensor to determine how reliable these selected sensors are.

3.5.1 Kesselring method

The main goal of the Kesselring method is to make a pre-selection which sensors to use for the occupancy detection system, based on the requirements stated in §3.3. The Kesselring method is a decision support method in which different variants are graded on their functionality and realization aspects in an evaluation matrix (Zeiler, Quanjel, & Savanovic 2007).

3.5.2 Evaluation matrix

In the evaluation matrix, the different sensors are graded on different aspects, these are derived from the detection requirements (§3.3). Requirement 1.1 states that the occupancy agent must be able to detect people. A person can only be detected from its environment when the sensor is independent from its environment (grading point A) and needs to recognize a person from other objects (grading point B)

Table 6: Grading aspects for sensor pre-selection

	Points	1	2	3
A	Independent of environment	Sensor is easily distracted	Sensor can react to other objects then people	Sensor only reacts to people.
B	Occupant detection	Has an implicit method to detect people	Can mistake objects as people	Can recognize people only
C	Determine direction	Sensor cannot determine direction	Sensor can determine direction with additional software	Sensor can independently determine direction
D	Responds time	>120 seconds	Between 2 – 120 sec	< 2 seconds
E	Reliability	Detection rate < 80% in literature study.	Detection rate 80%-90% in literature study.	Detection rate >90% in literature study.

Table 7: Evaluation matrix (realization aspect) for sensor pre-selection

		Realization		
Sensor	Privacy	Independent of environment	Response time	Total
PIR (passive infrared)	y	2	2	4
Ultrasonic	y	2	3	5
Vision based	n	x	x	x
Application based	y	1	2	3
RFID	n	x	x	x
Temperature	y	2	2	4
CO_2 based	y	1	1	2
Pressure sensor	y	3	2	5

Table 8: Evaluation matrix (Functioning aspect) for sensor pre-selection

	Functioning			
Sensor	Occupant detection	Determine direction	Reliability	Total
PIR (passive infrared)	3	2	2	7
Ultrasonic	1	2	1	4
Vision based	x	x	x	x
Application based	2	2	2	6
RFID	x	x	x	x
Temperature	2	2	2	6
CO_2 based	2	1	2	5
Pressure sensor	2	2	3	7

Detection requirement 1.2 states that the occupancy agent needs to be able to determine the direction in which a person is going. This requirement can be found under grading point C. Besides determining the movement of an occupant, the occupancy agent must also be capable to determine multiple movements in a short period of time, this is graded in grading point (D).

The results of the evaluation matrix are visualized in the Stärke diagram (see Figure 15). This diagram provides a clear overview on how the different variables relate to each other. Based on the results from the Stärke diagram, the sensors are chosen that have the best score. The chosen sensors are tested with experiments to measure their reliability in chapter 4.

Table 7 displays the privacy aspect that is rated as yes(y) or no(n). This aspect follows from the requirement that the occupancy agent must safeguard the privacy of the occupants which are measured. In case the system does not meet this essential constraint then there is no further purpose of evaluating that sensor as a possible solution for the occupancy agent.

3.5.3 Stärke diagram

The next step in the Kesselring approach is to visualize the results from the evaluation matrix into the

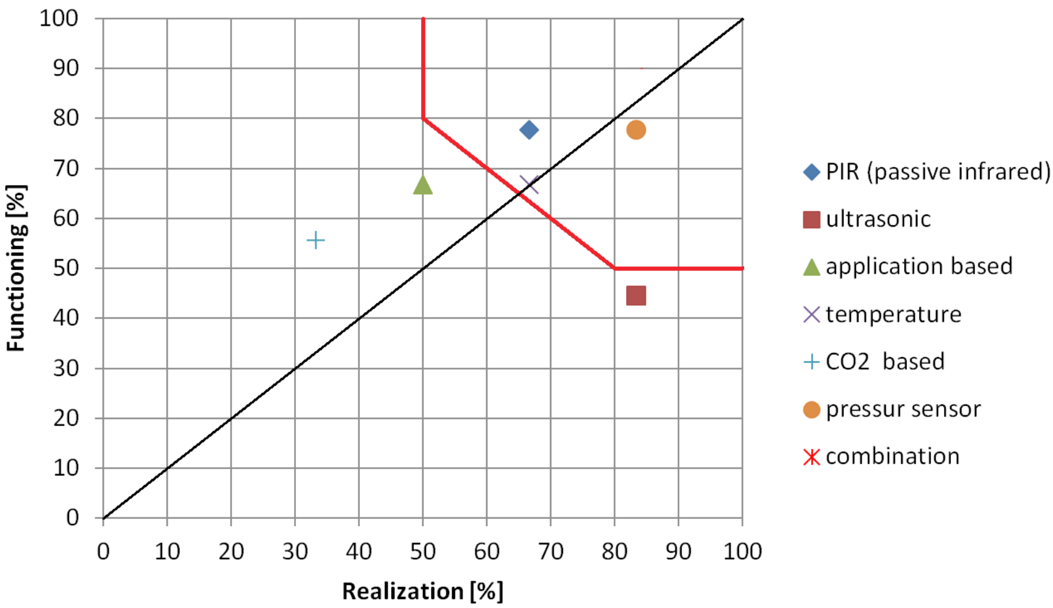


Figure 15: Stärke diagram, displaying the result of the Kesselring diagram method to pre-select sensors for the research

Stärke diagram. The goal of this visualization is to determine which sensors have the most potential to be used in the research. Figure 15 presents these results.

According to the guidelines of Zeiler et al. (2007), the selection area should be maintained at a border of (70% of 40%). However, in this research is chosen to move the selection criteria to (80% of 50%) because of the high demands required from the sensor.

The Stärke diagram illustrates that the most suitable sensors for this research are the pressure and PIR sensor. Both sensors have the same score on the functioning aspect of the spectrum, but score differently on their realization score. Which makes the pressure sensor the most suited sensor for the research. However, both sensor will be used in this research.

Another sensor that could be used in the research is the temperature sensor and the ultrasonic sensor. The ultrasonic sensor is almost every time used in combination with other sensors, like a PIR sensor (Hnat et al.,2012). This is done because the ultrasonic sensor is susceptible to false measurements like turbulent air and non-human objects like a piece of paper. (Labeodan, Zeiler, et al. 2015). This reduces the reliability of the sensors, and makes the temperature sensor more suitable to be used in this research.

3.5.4 improvements

The last step in the Kesselring method is to determine the improvement when multiple sensors are coupled with each other. In this case, a combination of the thermal, PIR and pressure sensor is explored (see Figure 16). The benefit of this combination is an increase in functionality compared to the pressure sensor. The advantage of adding a PIR sensor to the pressure sensor is that the PIR sensor can detect an occupant better than the pressure sensor (§2.1.1). On the other hand, the pressure sensor has no problem where the line of sight is blocked by any other objects.

The same is true for the thermal based sensor. The temperature based sensor also needs to have a line of sight in order for the occupant to be measured. The thermal sensor, on the other hand, measures a different property (temperature) of the occupant that enters or exits the room. This could increase the functioning relative to the pressure sensor.

Another benefit of coupling the thermal based sensor to the measurement is that when an occupant is measured by multiple systems at the same time, the reliability that a person has entered the room will increase (see formula 2).

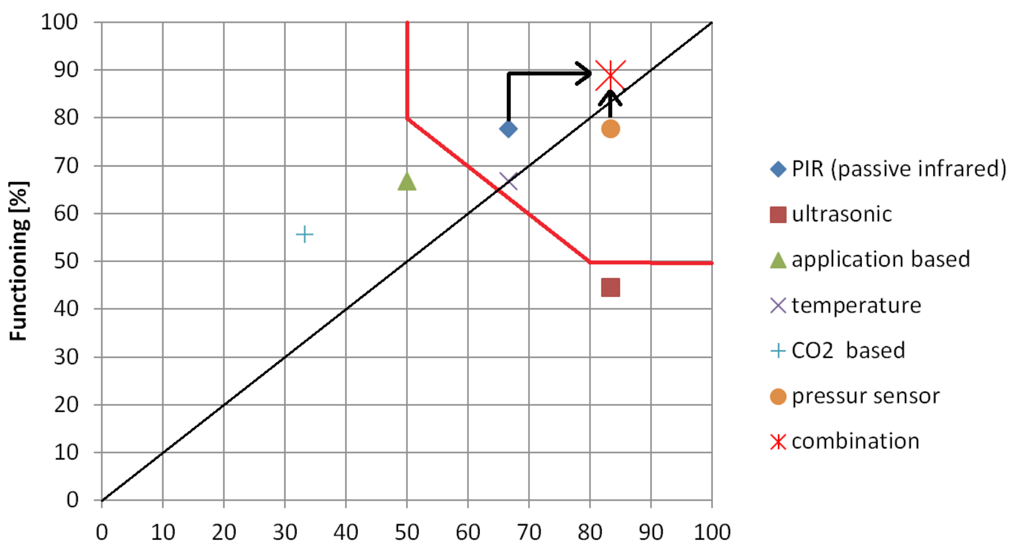


Figure 16: Stärke diagram, displaying the expected improvements when multiple sensors are used to determine the amount of people in the room

3.6 step 5b: Component responsible for Communication

Not only the sensor, but also other components are necessary for the occupancy agent to function. An overview of the different functions and the relation between the components is displayed in the component diagram (see Figure 17). This paragraph provides further information on each of the components described in the component diagram and its capabilities.

Microprocessor
As can be seen in the component diagram, the different used sensors are coupled to an Arduino microprocessor. The Arduino is responsible for fulfilling the detection requirements (see Figure 11). For this purpose, the Arduino receives the signals from the sensors. Based on these signals the Arduino needs to determines if:

1 The signal corresponds to a person
2 Determine the direction the person is going.
3 Count how many occupants are in the room.

The Arduino needs to be programmed to recognize the signals from the sensors to complete its tasks. Arduino has its own programming tool which is used to for programming the occupant detection systems, namely Arduino IDE ("Arduino - Home," 2016)

JADE/ JAVA
The communication between the occupancy agent and other outside agents takes place through the middle-ware software JADE (Java Agent DEvelopment framework). JADE is responsible for fulfilling the communication requirements (as indicated in Figure 12).

One of the goals of JADE is creating a framework which simplifies the implementation of multi-agent systems through a set of graphical tools (Telecom Italia SpA, 2016.). This framework is displayed to the user of JADE in the GUI (Graphical User Interface), and provides the user insight on the occupancy in the room.

One of those graphical tools is the introSpectorAgent. In the introSpectorAgent, the communi-

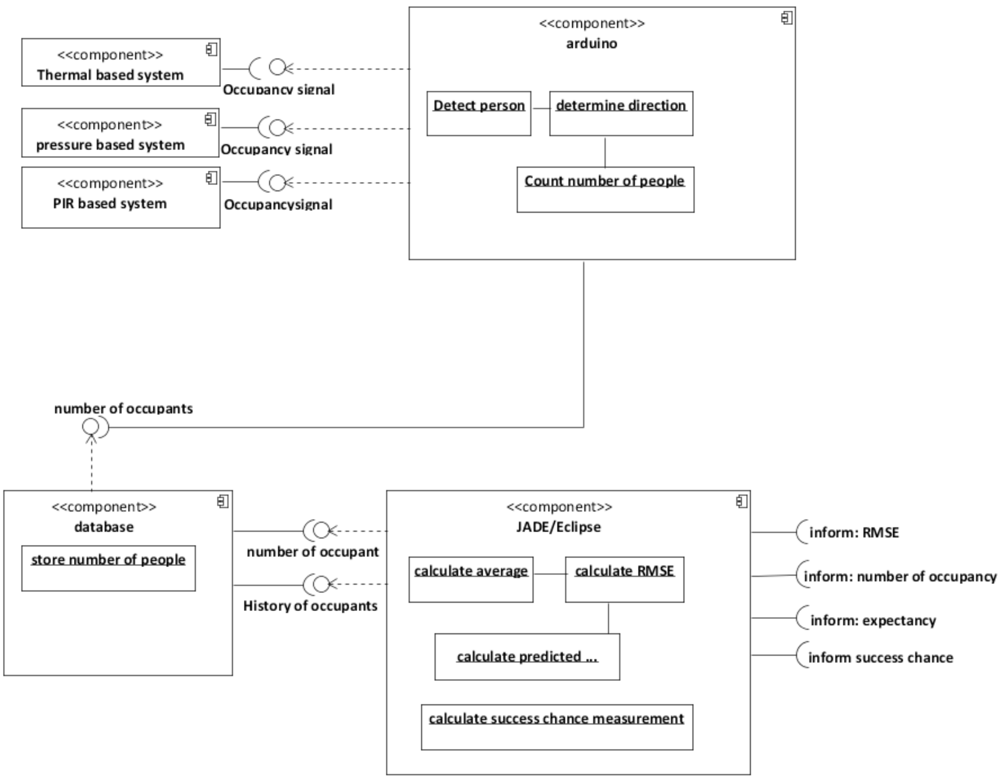


Figure 17: Component diagram displays the different components, and relation with each other, of the occupancy agent

cation of an agent can be monitored. Figure 19 displays an example of an agent which receives occupancy information of another agents.

An additional option of the IntroSpectorAgent is the possibility to get more detailed information of each message by pressing on the envelope. This will open a window with the information on the sender of the message, the receiver of the message and the content of the message.

The example in Figure 19 displays a message which is sent from the reliability agent to an example agent. The content of the message is the reliability of the different occupancy detection agents.

Another method to get insight into the communication between the different agents is the Sniffer option. The Sniffer option displays the communication between the agents with help of a sequence diagram. An example can be found in Figure 18, where the communication with an example agent and the number of people in the room agent is displayed. It is also possible to visualize the communication between other agents in a similar fashion. Just like the IntroSpectorAgent, it is also possible to get more detailed information of the communication. In the Sniffer option, this can be done by pressing on the message which is sent between two agents. This will open up a window similar to the IntroSpectorAgent with the same information.

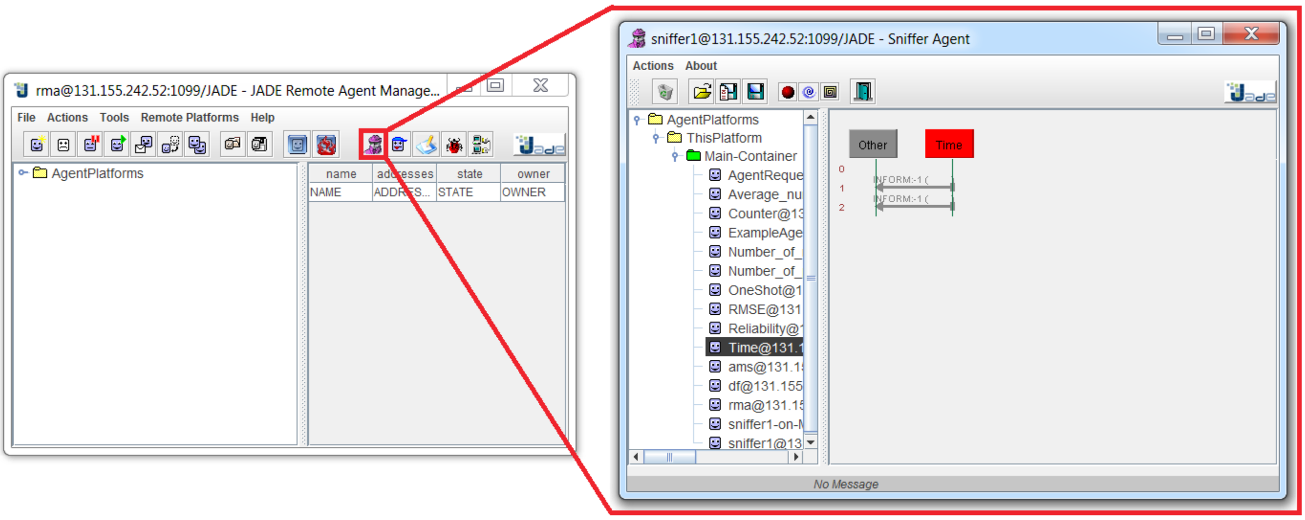


Figure 18: GUI containing the communication structure between the agents, in this case an example can be seen of communication between the example agent and time agent

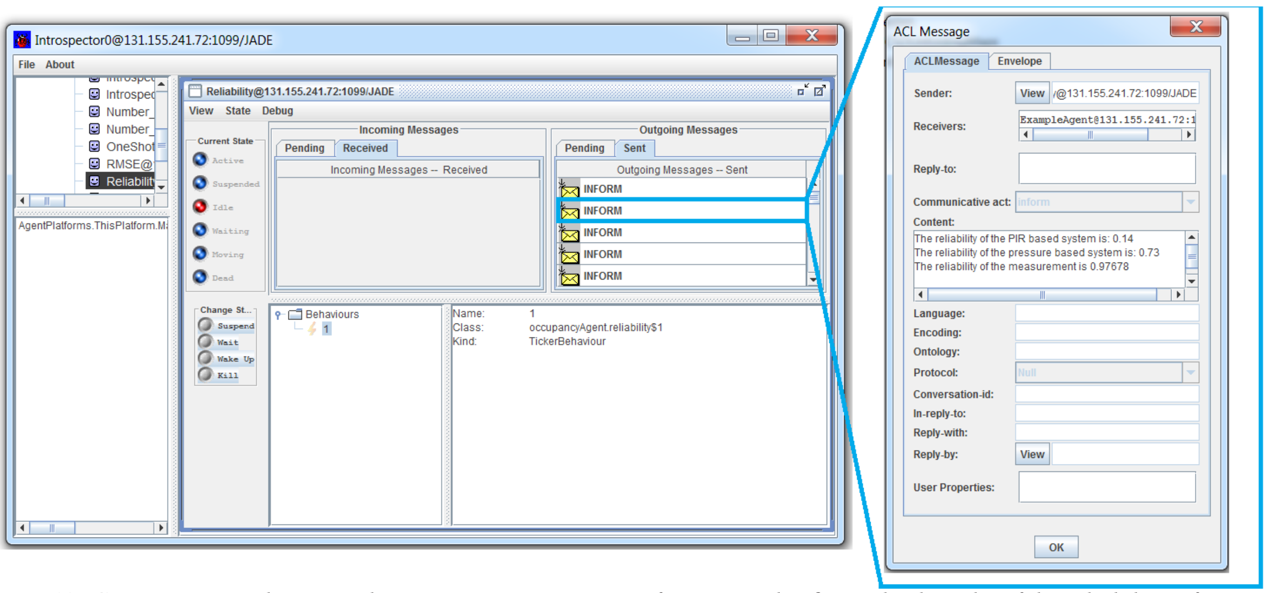


Figure 19: GUI containing the internal communication structure of an agent, this figure displays the of the reliability information to the example agent.

The communication of information between the occupancy agent and other actors or agents can go in two different ways. First, updates can be given in the form of one way communication, in which the information is sent to the actor or other agents on a predetermined time interval. The other form of communication is one in which an actor or agents request information from the occupancy agent. The occupancy agent will respond by sending the desired information.
The information sent to outside actors and agents are based on the communication requirements of Figure 12. Based on these requirements, the occupancy agent is able to send messages informing outside actors and agents on:

1) The number of people in the room.
2) The reliability of each measurement.
3) The average number of people in the room.
4) The RMSE of each measurement.
5) The time of measurement.
6) The number of measurements which are performed.

JADE provides the possibility to send these messages to other agents and makes this process visible in its own GUI so actors keep track of the process. With these possibilities the communication requirements are fulfilled.

Chapter recap
This chapter describes, the design process of the occupancy agent with help of the SYSMOD approach introduced by Weilkiens (2006). In the first step, the environment and the services were explored with help of a context diagram and use case diagram. On the basis of these analyses, the requirements were made which needed to be fulfilled by the occupancy agent.

On the basis of the requirements, the occupancy agent is built. In order to fulfill the requirements an activity diagram is used to break down the requirements into actions. Finally, the different components are described which are necessary to perform the different actions.

The Kesselring method was applied to determine which sensor has the most potential for the research. The result of the Kesselring diagram led to the selection of the PIR based, pressure based and thermal based occupancy detection systems. Other components that are necessary for the occupancy agent to function are also described in the component diagram.

The result is an occupancy agent which uses multiple detection methods, that has the capability to inform other agents of the amount of occupants in a room.

CHAPTER 4: EXPERIMENTS

The goal of the occupancy agent is to provide reliable occupancy information. The SYSMOD approach in chapter 3 is used to design the occupancy agent. The reliability of the occupancy agent is determined by conducting different experiments.

The experiments are conducted for each of the occupancy detection systems. This is done because this part of the system is responsible for determining the detection and direction of an occupant. An error in this part of the occupancy agent will influence the reliability of the whole occupancy agent.

During the experiments, the effect of different variables is tested, which are described in §4.1. Each of these variables is tested multiple times for each occupancy detection system. This is done in order to establish the effect they have on the reliability of the occupancy detection system, and thereby show how reliable the occupancy agent is.

The different experiments are performed in an office room at the Eindhoven University of Technology. More detailed information on the setup of the room can be found in §4.2.

The results of each experiment are described in §4.3. The outcome of the results are described using the following structure:
- Tested variable (experiment)
    - o PIR based system
    - o Pressure based system
    - o Thermal based

Besides the results of the experiments, this section will also provide further interpretation of the results were needed.

4.1 Tested variables

The reliability of the different occupancy detection systems is determined by testing the effect of the following variables:

1) The effect of entrance/exit speed
2) The effect of multiple people entering or leaving the room.
3) The effect of signal disruption

The effect of these variables on the different occupancy detection system is measured by subjecting each variable to a number of tests. The choice for these tests are further elaborated in the section below.

1) Entrance and exit speed
The aim of this experiment is to test what effect the speed on an occupancy agent has on the reliability of the occupancy detection systems. Each occupancy detection system must be able to measure an occupant, and his direction, in a short time interval. This time interval is determined by the speed of the occupant. To get insight in the effect of this variable the occupant detection systems is investigated for different entrance and exit velocities. Speeds are tested from saunter speed (0.2 m/s) to running speed (2.4 m/s). Speeds exceeding this value are not expected to occur in the built environment and are therefore not tested.

2) Number of people entering or leaving a room
In an office setting, there are situations where a group of people can enter and leave a room. An example of such a situation is when a meeting is held by several people. The occupancy detection system should accurately detect the number of people coming into and going out of the room to be reliable. The challenge for the occupancy detection system is that multiple people are entering and leaving the room after each other. This way the capability of the occupancy detection system to recognize multiple people in a short time interval is tested.

To test the reliability of each occupancy detection system, experiments are held where a predetermined amount of people enter and exit the test room multiple times. The number of people can vary from 4 to 6 persons. Each test is repeated at least 36 times to make sure that there is a minimal influence of the number of measurements on the result.

When a false measurement is observed, the type of error is noted (false positive or a false negative), and the cause for the error is investigated.

3) Signal disruption
The last variable which is examined is the effect of signal disruption on the reliability of the occupancy detection systems. Disruption is described as the effect of disturbance on the measured signal of the sensor by an outside source. For each detection method a different type of signal disruption source is used.

With this experiment insight is gained on the effect of the environment on the performance of the occupancy agent system. With this knowledge it is possible to determine if the occupancy detection system can perform in every environment, or that it can only work in an environment under certain conditions. For instance, if the PIR based system is effected by outside infrared radiation (like the heating system) it is important to incorporate this in the design condition of the system.

To test the effect of signal disruption a heater is placed in front of the PIR-based system and the thermal based system. The heater sends out infrared radiation, which could influence the capability of the system to correctly measure a person and the direction which a person is going.

In the second experiment, thick clothing is used to reduce the infrared radiation of an occupant. This creates a different disruption in which the signal is more difficult to measure. This in turn, might also lead to false measurements.

A dummy weight of 30.4 kg is used to distract the pressure based system. In the first test, the dummy weight was rolled over the pressure matt by itself, in the second test the dummy weight combined with an occupant is tested.

Table 9: Signal disruption

System	Disruption	Disruption
PIR based	Additional infrared from surrounding	Reduced infrared radiation occupants
Pressure based	Additional weight	Additional weight+occupant
Thermal based	Additional infrared from surrounding	Reduced infrared radiation occupants

4.2 Data analysis

Besides the detection rate which is calculated by dividing the total number of successful measurements through the total number of measurements, the uncertainty of the measurement sample plays an important role. This uncertainty is expressed by the margin of error. In this case the margin of error can be calculated with help of the Bernoulli distribution. Characteristic for this distribution is that the outcome of each test only exists out of two outcomes, in this case a correct or a false measurement. When multiple measurements are performed the margin of error can be calculated with help of formula 1 (Montgomery, 2003),(Berkum & Bucchianico, 2009).

$$\overline{X} - z_{\alpha/2}\sqrt{\frac{p\cdot(1-p)}{n}} < \mu < \overline{X} + z_{\alpha/2}\sqrt{\frac{p\cdot(1-p)}{n}} \qquad \text{formula 3}$$

$\overline{X}$ =Expected value
p = Detection rate
n = total amount of measurements

4.3 Measurement setup

Figure 20 shows the test setup for the different occupancy detection systems. The room is situated at the Building Physics and Services laboratory of Eindhoven university of technology.

The PIR based sensor system is placed on an elevation near the door to detect people that enter or exit the room. The system is placed at a height of 96 cm, following the recommended height described by the supplier of the system.

The supplier of the system describes the optimum height to be around the hip. According to the DINED Anthropometric database, the average height of male and the female hip at an age between 20-60 years is 960mm in the Netherlands (TU Delft, 2016).

The pressure sensor is placed in front of the entrance and exit of the door on the ground. There is a small distance between the two pressure mats to prevent false measurements from other objects.

The Omron thermal sensor is placed above the door on the inside of the room. The reason to place the sensor inside is to shield it from outside activities.

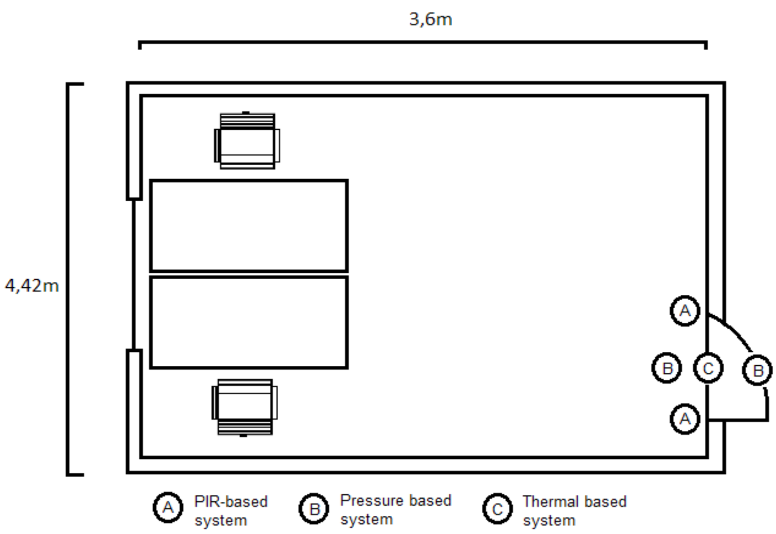


Figure 20: Measurement setup

4.3.1 step 5b:Component responsible for detection

The different sensors used in this research all measure a different property of a person in order to distinguish him/ her from the environment. This section will further elaborate on how a person is distinguished, and what the properties of each sensors are.

1) PIR based system
The PIR based sensor used in this research is a system from Sensor Development international. The working principle of the system is when an occupant passes the sensor it detects the change infrared radiation coming from a person (International B.V. Sensor Development, 2016).

Figure 21: PIR based occupancy detection system source: http://www.sd-international.nl/

2) Pressure based system
The pressure based system used in this study is developed by Arun electronics. The pressure mat is used in a variety of products like interactive toys, patient care or multimedia systems.

The model used in the research is the PM2 Standard Floor Mat 4 wires, with a dimension of 720 x 390 x 3 mm; which is triggered by a minimum weight of 16kg (“Pressure Mat Sensors - Data Sheet,” 2016). When a person is standing on the mat, the pressure based sensor will recognize this as a person.

3)Thermal based system
The thermal based sensor used in this research is the Omron D6T Mems thermal sensor. The Omron sensor registers the temperature in a field of 4x4 squares (“Omron D6T Mems thermal sensor,” 2016)(see Figure 22). A person that enters the room has a higher body temperature than the environment, this higher temperature is distinguished by the system as a person.

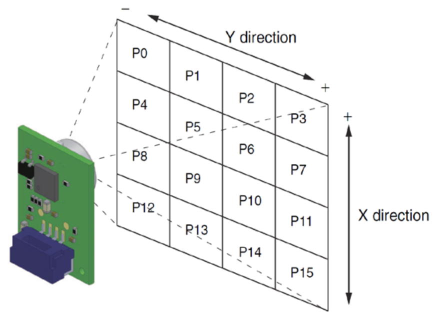


Figure 22: Omron thermal sensor Source: (“Omron D6T Mems thermal sensor,” 2016)

4.3.2 Determine direction

After the occupant is detected, the direction which he/she is going needs to be determined. Each occupant detection method is programmed,to determine the direction of an occupant in its own manner. This section will provide more information how this process takes place for each type of sensor.

1) PIR based system
The PIR based system determines the direction of an occupant with the help of a software algorithm. The exact structure of the algorithm is unknown, however, some details are possibly identified. Most likely, the system receives a change of infrared radiation of an occupant from multiple angles. The sequence in which the changes of infrared takes place determines the direction of the occupant.

2) Pressure based system
The pressure based systems consist of two different pressure mats working together. If a person is standing on the pressure mat a signal from the first mat is triggered. When the occupant resumes his way into, or out of the room, he will walk on the second mat which is a second trigger. The combination of these two triggers and timing determines if a person is leaving or entering the room (see Table 10 and Figure 23).

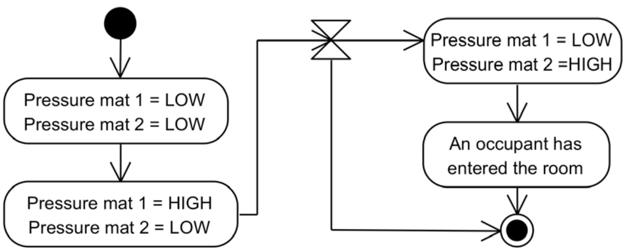


Figure 23: Operation pressure based diagram

Table 10: On (1) off (2) Table

Pressure mat 1	Pressure mat 2	Output
0	0	0
1	0	0
0	1	0
1	1	1

3) Thermal based system
The thermal based system can determine the direction of an occupant by the sequence of temperature exceeding levels. The Omron thermal sensor measures the temperature in sixteen different squares in its field of view (see Figure 24). Each square gives off a signal when the temperature of a square exceeds a predetermined threshold value. When an occupant enters the room each square is triggered after its previous square, while the previously passed square stops being triggered. In this way, the sequence of squares determines the direction which an occupant is going.

4.4 Measurement results

The outcome of the different experiments composes the reliability of the occupancy detection system. This depends on the amount of correct and false measurements divided by the total amount of measurements. The false measurement can be categorized into two different outcomes.

The first type of error is a false positive in which the agent measures an occupant movement, while no movement is taken place. The second type of error is a false negative measurement, this error is made when no occupant movement takes place while the occupancy agent measures a person entering or leaving the room.

Beside the outcome of each measurement, the result is also dependent on the number of measurements made. This is taken into account with help of the margin of error, which calculates the uncertainty of the outcome based on the number of measurements.

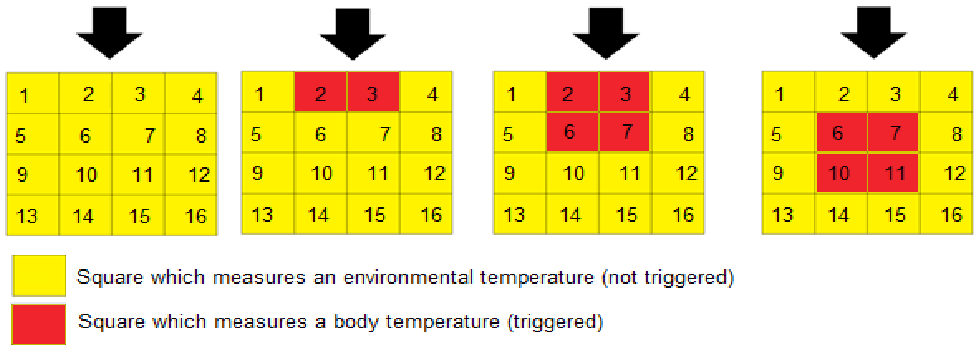


Figure 24: Determination of occupant for the thermal based system

4.4.1: The effect of entrance and exit speeds

The first variable taken into consideration is the effect of the speed at which an occupant enters or leaves the room. For this variable different speeds are tested from saunter (0.2 m/s) to running speed (2.4 m/s). Higher speeds are not expected to occur in the built environment and are for this reason excluded.

The result for each occupancy detection system is an overview of the outcome of each measurement done for the different velocities, with an interval 0.2m/s. The results of this experiment are displayed for each occupancy detection system individually in the following sequence:

1) PIR-based system
2) Pressure based occupancy detection system
3) Thermal based occupancy detection system

1) PIR based system
The result for the PIR-based system is visualized in Table 11. The reliability of the system is 100% for all the different velocities, except for 2.2m/s. In that case a person entering the room was not always measured.

The margin of error is in the range of 1% to 3%. This is due to the large amount of measurements. The reason why the margin of error is larger for a speed of 2.2 m/s is because an false positive error was observed at this velocity.

2) pressure based system
The reliability of the pressure based detection system for varying entrance and exit velocities can be seen in Table 11. The reliability of the pressure system in the test ranged from 100% to 73% .

During the measurements with the pressure based system, the most errors occurred at lower velocities (between 0.2 m/s and 0.8m/s), and at a speed of 2.4m/s.

The margin of error for the pressure based measurement ranged from 1% to 13%.The main cause of the highest margin of error is a combination of the low number of measurements, and the relatively large amount of false errors.

The amount of false positive and false negative measurements in total are the same, however noticeable is that the false positive errors only occurred for lower velocities.

3) Thermal based
The reliability of the thermal based system ranges from 9% to 51%, when tested for different velocities. The highest reliability is obtained with a velocity between 0.6m/s to 0.8m/s. While other velocities have a reliability of only 23% to 9%.

The margin of error is also larger when compared to the other systems. This is due to the lower number of measurements (see formula 3), and the lower detection rate of the thermal based system. However, even with fewer measurements it becomes clear that the detection rate is significantly lower compared to the other detection systems.

Table 11: Result speed measurements

Speed	0.2	0.4	0.6	0.8	1.0	1.2
PIR based system	100	100	100	100	100	100
Pressure based system	90	91	95	98	100	100
Thermal system		14	51	47	15	23

Speed	1.4	1.6	1.8	2.0	2.2	2.4
PIR based system	100	100	100	100	97.44	100
Pressure based system	98,33	94,74	100	100	100	73
Thermal system	11	17	22	27	9	

4.4.2 Group measurements

The second tested variable is the effect of multiple people entering and leaving the room after each other. The effect of this variable on the reliability of the different occupancy detection systems is tested with a group of 4-6 people entered and left the room multiple times.

The results noted in Table 12, 13 and 14 display the total number of people which entered and left the room. The reason for this is, because in one measurement multiple people could be missed, if the group as a whole was measured then this would lead to a misrepresentation of the results.

1) PIR based system
The result of the group measurement is presented in Table 11. A division is made between Sensor A which is people who enter the room, and Sensor B which is people who are leaving the room. The reliability of the PIR based system was between 99% to 97%, which made the combined average detection rate 98%.

The margin of error is varying in between 1% to 2%.The main reason the margin of error is the lowest when both measurements are combined is the total measurement being twice as large than for one direction. The highest margin of error is due to more errors made in sensor A then sensor B.

It should also be noted from Table 12 that sensor A made three false positive errors and two false negative errors. While, sensor B only made two false negative measurements and no false positive errors.

2) Pressure based
The reliability of the pressure based system is displayed in Table 13. The reliability of the pressure based system when multiple people entering and leaving the room is between 75%-56%.

The margin of error is in the range of 3% to 2%. The main reason why the margin of error is larger than the PIR based system is because more errors were measured using the pressure based system.

Table 12: The effect of groups on the reliability of the PIR based system

	Total measurement	Correct	False measurement	False positive	False negative	Detection rate[%]	Margin of error [%]
Both direction	322	315	7	4	3	98	1
Sensor A enter room	161	156	5	3	2	97	1
Sensor B exit room	161	159	2	0	2	99	1

Table 13: The effect of groups on the reliability of the pressure based system

	Number of measurements	Correct	Total both	Missed	Detection rate[%]	Margin of error [%]
Total amount	474	346	130	128	73	2
People enter room	240	179	61	63	75	3
People leaving room	234	167	67	69	71	3

Table 14: The effect of groups on the reliability of the thermal based system

	Number of measurements	Correct	Both	Wrong	Misses	Detection rate [%]	Margin of error [%]
Total amount	378	135	106	56	81	36	7
People entered	189	26	67	48	48	14	5
People left	189	109	39	8	33	58	7

A closer look at the behaviour of the pressure system reveals that when an occupant entered the room, the system could count multiple in and out movements from one person. When the occupant was in the room the pressure system displayed a correct final result. However, this anomaly was not the case for every measurement.

When overestimated counting is seen as an error the detection rate drops to a value between 59% to 52%. This means that 18% to 19% of the difference is caused by this error. Also the margin of error increased to 3%-2%.

3) Thermal based system
The thermal based system gives the least reliable result from the tested occupancy detection systems. Noteworthy is the large difference in detection rate between people entering (14%) and leaving the room (58%) in table 14. This indicates that direction can be an influential factor for the performance of the system.

In contrast to the previous experiment, the margin of error is lower although the thermal based system made more errors (see table 14). This is because the margin of error is a measure how likely the outcome of the next measurement is going to be. In this research this can be a correct measurement, or a false measurement. When a lot of false measurement occur, the chance that the next measurement will result in a false measurement is more likely to occur. Hence the margin of error decreases.

The people that entered the room and were measured by the thermal based occupancy detection system had only a detection rate of 14%. This means that the likelihood that in a next measurement a false measurement occurs is high. Hence, the margin of error is relativly small.

However, the people who left the room and were measured by the thermal based system had a detection rate of 58% and a margin of error of 7%. This higher margin of error occurs because it is harder to predict if the next measurement outcome will lead to a correct or false measurement.

4.4.3 Signal disruption

The last experiments reviews the performance of the different detection systems when the signal, which the sensor measures, is being disrupted. This is done with an additional source that could disturb the signal, or mask the signal of an occupant, that could cause a false measurement (see Table 15).

Furthermore, experiments are conducted in which the measured signal is blocked. This is done by wearing additional cloths. An overview of the different experiments can be found in Table 15.

Table 15: Tested disruption variables

System	Disruption	Disruption
PIR based system	Additional infrared from surrounding	Reduced infrared radiation occupants
Pressure based system	Additional weight	Additional weight+ occupant
Thermal based system	Additional heat from surrounding	

1) PIR based system
For the PIR based system two types of experiments are conducted. In the first experiment an additional infrared radiation was supplied with the use of a 1500W heater as depicted in Figure 25.

In the second experiment, the occupant wore three layers of clothing, one t-shirt and two sweaters

Figure 25: Heater placed in front of the PIR based system which sends infrared radiation and heat which could disrupt the PIR based and thermal based system

with a clo-value of 0.89 for the upper part of the body to block the infrared radiation coming from his body. When the heater was placed in front of the PIR based sensor this led to the results as can be seen in Table 16.

The average detection from 101 measurements was 99%. Which means that only one mistake was made from the 101 measurements. The margin of error for the research is 1%. On the basis of this data no or a small effect of additional radiation on the PIR based system has been found.

Table 16: Result disruption PIR based detection system

Test	Number of tests	Average detection rate [%]	Margin of error [%]
Added infrared	101	99	1
Blocking infrared	50	90	4

For the second experiment, the infrared radiation of a person was partly blocked by extra layers of clothing. The average detection rate from this experiment is 90%. Because less measurements were taken in comparison with the added infrared radiation the margin of error increased to 4%. On the basis of these data a small effect of additional clothing is discovered on the PIR based system.

2) Pressure based system
The pressure based system is triggered when a weight of at least 16 kg is placed on the mat. This weight does not have to be from an occupant, but could also be another weight source. These other weight sources could lead to false measurements of the system when the weight is rolled over the pressure mats.

To measure the effect of signal disruption on the pressure based system two types of experiments are performed.

1) The effect of a dummy weight on the pressure based system
2) The effect of a dummy weight together with an occupant.

In the first experiment, a dummy weight of 30.5 kg is placed on a hand cart as depicted in Figure 26. Only the dummy weight is rolled in and outside the room. The output of the pressure based system is checked if it recognized a movement from the handcart as a person entering or leaving the room.

In a second test, the same setup is used as in the previous experiment, only in this experiment an occupant is entering the room together with the dummy weight. Again the output of the system is analysed to view how the pressure based system reacted to the measurements.

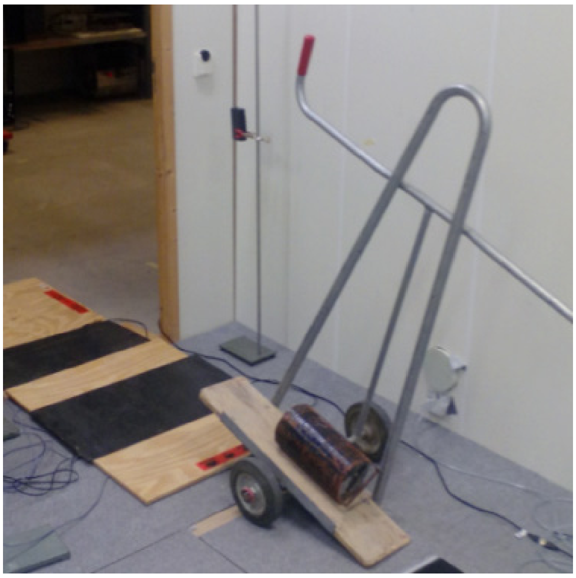


Figure 26: A hand cart is used as disruption signal for the pressure based system.

The results of both experiments are depicted in Table 17. When only the dummy weight entered and left the room, the detection rate, which describes the amount of correct measurements, was 91%. If an occupant entered the room together with the dummy weight the detection rate dropped to 73%.

Table 17: The effect of disruption on the pressure based system

Test	Number of tests	Detection rate [%]	Margin of error [%]
Dummy weight	34	91.2	5
Dummy weight + occupant	34	73.5	8

The margin of error differs in both experiments. This is because in the experiment in which an occupant entered the room together with a dummy weight more false measurements were made, which led to a margin of error of 8%. While on the other hand, when only the dummy weight was used, fewer errors were made and the margin of error reduces to 5%.

3) Thermal based system

The experiments done for the PIR based system are also done for the thermal based system. This means that likewise in front of thermal based system a heater is placed, which increases the temperature measured by the system.

The result of this experiment is that from the 34 measurements which were done only 24% of the measurements were correct, with a margin of error of 7%

Table 18: The effect of disruption on the thermal based system

Test	Number of tests	Detection rate [%]	Margin of error [%]
Added infrared	34	24	7

CHAPTER 5: DISCUSSION

The goal of this chapter is to reflect back on the research. To do so, a review of each chapter can be found in this discussion. Each chapter will be reviewed on possible points of improvement, and where possible further explanation is given on the outcomes of the each of the chapters.

5.1 Context

The literature study (§2.1) provided information on previously conducted studies which are done in the field of occupancy detection, demand driven control, and sensors which were used to detect a person. This state-of-art research also gave understanding on the potential of each sensor, and what the possible disadvantages are for using a particular sensor. This information is used to determine which sensor had the most potential for this research in chapter §3.5.

The literature study also provided insight on the possible use of agents as a tool for gathering and sharing information (§2.2). Through this way understanding is gained of what different properties an agent possesses and what types of agent can be recognized.

5.2 Occupancy agent design

With the obtained knowledge on the available sensors and the possible use of agents, the occupancy agent was designed. The SYSMOD method of Weilkiens (2006) is used as a guideline for designing the occupancy agent. This provides a structure to the design process, that made it possible to get from a global concept to a fine grained design.

The design of the occupancy agent displays the different levels and aspects of the occupancy agent. In the first part, the context (§3.1) and the services (§3.2) which the occupancy agent needed to accomplish were analysed. These analyses were translated to requirements in §3.3.

Based on those requirements the different aspects of the occupancy agent are displayed. An example is the activity diagram in §3.4.

With the insight what different actions were needed, the components are selected that would fulfill the requirements. For the detection requirements a sensor was needed. With help of the Kesselring method a pre-selection was made which sensor had the most potential to be used in the research (§3.5).

The information from the sensors needs to be translated and communicated with other agents and actors in order to meet the communication requirements. The translation of the information from the sensor to the occupancy agent takes place through an Arduino microprocessor. And, with help of JADE the information is communicated with outside agents and actors.

The JADE agent and the occupancy detection systems are coupled through a S.D.-card. The S.D.-card needs to be removed from the microprocessor and plugged into the computer. From the computer, the agent can (provided the correct pathway is entered) read the data from the S.D.-card and uses this information to calculate its own data, and share this information with other agents.

A possible solution to establish a live connection between the occupancy detection systems and the agent might be to use a Raspberry pi, which has better capabilities to connect to the internet as a medium to establish this live connection. Another solution might be to use a radio transmitter in combination with the Arduino.

5.3 Reliability experiments

To test the reliability of the occupancy agent, different experiments are conducted which gave insight to the reaction of an occupancy detection system towards predetermined variables. The methodology of conducting controlled experiments to find the reliability of an occupancy detection system can also be found in other studies. An example is the research of Benezeth, Laurent, Emile, & Rosenberger (2011), in which a set of predetermined reference situations was tested.

This thesis also uses different experiments to determine to what extent the design meets the occupancy counting requirements, and to make a comparison between the different occupancy detection systems. The goal of the experiment is to gain insight in the performance of each occupancy detection system. An indication for the reliability is the detection rate. However, at which detection rate value a system is considered reliable is subjective depending on the purpose for which the system is used.

During each experiment, every occupancy detection system made false measurements. Closer analysis of these errors showed that numerous causes presented why a measurement failed to be recognized by the occupant detection system. The section below provides a summary of the most viable causes why each occupancy detection system made a false measurement.

5.3.1 Thermal based results

Several experiments are conducted to test the reliability of the thermal based system. The result from the experiment in which the effect of different speeds are measured demonstrated that the reliability of the thermal based system is between 11% (at a speed of 1.4 m/s) and 51% (at a speed of 0.6 m/s), shown in Table 11. The detection rate when a group of people entered or left the room is 36% (see table 14).

Furthermore the effect of signal disruption is tested by wearing additional clothing. This experiment demonstrated that the reliability of the thermal based system is 26%. From each occupancy detection system,the performance of the thermal based system had the lowest relibaility. Further investigation was done to find the reasoning for measurements.

One of the reasons for a failed measurement was due to the low registration speed. the registration speed is the speed at which each measurement point is refreshed. If the speed of the occupant is higher than the threshold registration speed, the occupant can enter the room and measurement points will be kipped, as is indicated in Figure 27. The cause of this error can be found in the thermal based system or the code that is used to detect an occupant. The threshold registration speed can explain the low reliability speeds above 1 m/s indicated in table 11. During the experiment in which a group entered and left the room, the speed of the occupants was also above 1m/s explaining the relative same result as the speed experiment for higher speed (Table 11).

The result from the experiments seem to contradict former research that is done by Veselý et al. (2017), they found in their result that the thermal based system was reliable. In their research the thermal system is used to track the temperature of the hands for a longer period of time in the field of view of the sensor. The difference in this research however is that an attempt is made to track the movement of a person when the person enters the field of view of the sensor for a short period of time.

Another error made by the thermal based system, is when an occupant is passing through the system and the threshold temperature is not reached (see Figure 28). This could explain the false measurements made when the occupants wore additional clothing (see table 18).

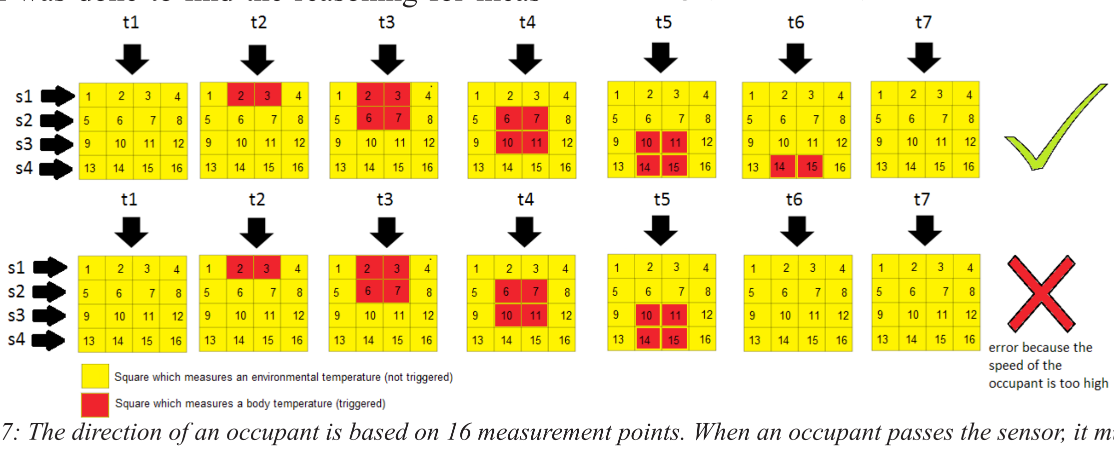


Figure 27: The direction of an occupant is based on 16 measurement points. When an occupant passes the sensor, it must passes a sequence of measurement points. The figure displays 7 consecutive time frames. In a correct measurement the occupant passes a measuement point until the occupant is inside. When the occupant is too fast it skips measurement points (see t7) this results in a false measurement

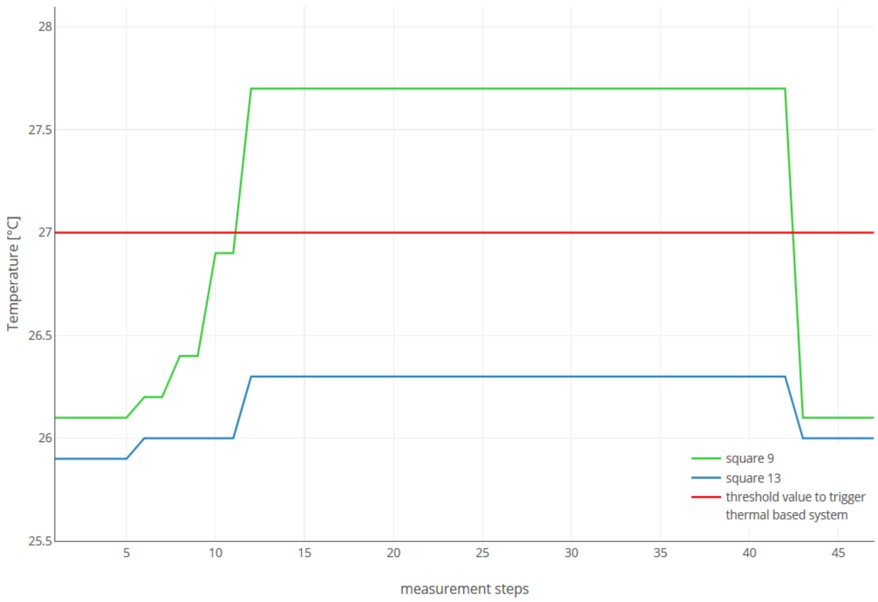


Figure 28: Temperature error of the thermal based system, in which the threshold temperature of 27°C is not reached. Because the trigger temperature is not reached the occupant is not counted

In the research Amin, Taylor, Junejo, Al-Habaibeh, & Parkin (2008) temperature is used to detect an occupant. However in their research they did not found a temperature of an occupant lower as 27°C.

A possible explanation is the angle at which the occupants were measured. In this research the occupant are measured from an downswards angle above the occupant. The hair of the occupant could cause a lower surface temperature and therefore decrease the measured temperature by the thermal based system.

When the face temperature of the occupant would be measured , the measured temperature would be higher. This will increase the reliability in this research. Another improvement for the system would be to use the difference in background- and occupant temperature as a trigger point, instead of a fixed temperature.

A possible solution for the low registration speed of the thermal based system would be increasing the threshold registration speed. This solution can be divided in two additional options; increase the measurement speed of the sensor; or increase the height of the sensor placement. According to the obtained results, the measurements of thermal based sensor has to be increased to approximately 12 measurement /second, assuming the same sensor height as the measurement. In this research, the thermal based sensor is placed 0.4 m above the person. Another solution would be to increase the height of the thermal based sensor. This will increase the measurement field and creates more measurement distance. The increased measurement distant will reduce the needed number of frames per second due to a larges measurement field.

All in all, the thermal based system can have potential to accurately count the number of people in the room when the above mentioned errors are resolved. Another possibility for the system to function is in a different kind of setting. As the research Veselý et al. (2017) already indicates, is that a more static image can be monitored by the sensor. This is also what this research found.

5.3.2 Pressure based results

The experiments done for the pressure based system are displaying a reliability for the speed measurements varying from 100% percent (at a speed of 1.8 m/s -2.2 m/s and 1.0 m/s -1.2 m/s) to 73% (at a speed of 2.4 m/s) (table 11). The experiment when a group of people enter and exit the room demonstrates that the reliability of the pressure based system is 73% (table 13). The effect of signal disruption (table 17) displayed that the reliability of the pressure based system is 74%.

This shows that the pressure based system could be used to accurately count the number of people in the room. However, The obtained results should be handled with caution, because there are some errors in the system

An observed error in the experiments was related to the dimension of the mat. If the occupant steps over one of the mats (a distance of 1.2 meter) as is depicted in Figure 29, the pressure based system recognizes a person passing the room, without entering the room. Hence, a false measurement takes place.

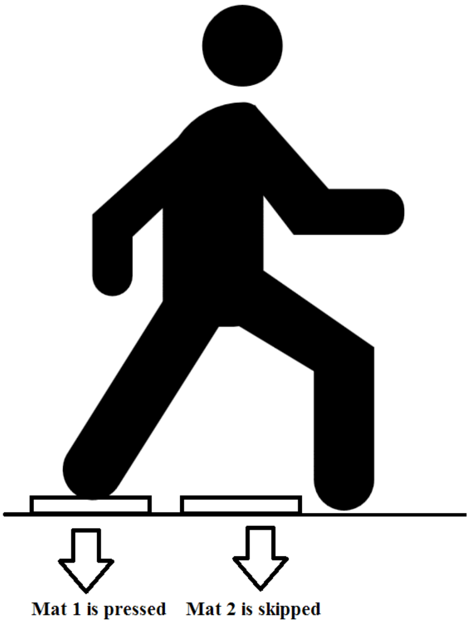


Figure 29: When an occupant steps over one mat the pressure based system does not count the person entering the room.

This error is one of the main causes of the lower detection rate at larger speeds. As a result of the increasing speed, the occupant increased his step width, thereby skipping one of the pressure mats.

Another observed error took place when both the pressure mats were occupied at the same time. When this situation occurred the pressure based system registered this as an event in which the occupant was walking in and out of the room at the same time.

The reason why this error occurred is because the pressure based system measures a person entering, or leaving the room when one mat is pressed, followed by the other pressure mat. When both mats are pressed at the same time, a situation occurs in which alternating boundary conditions for a person who is both entering the room, and leaving the room are met. This results in an output in the command window as displayed in Figure 30.

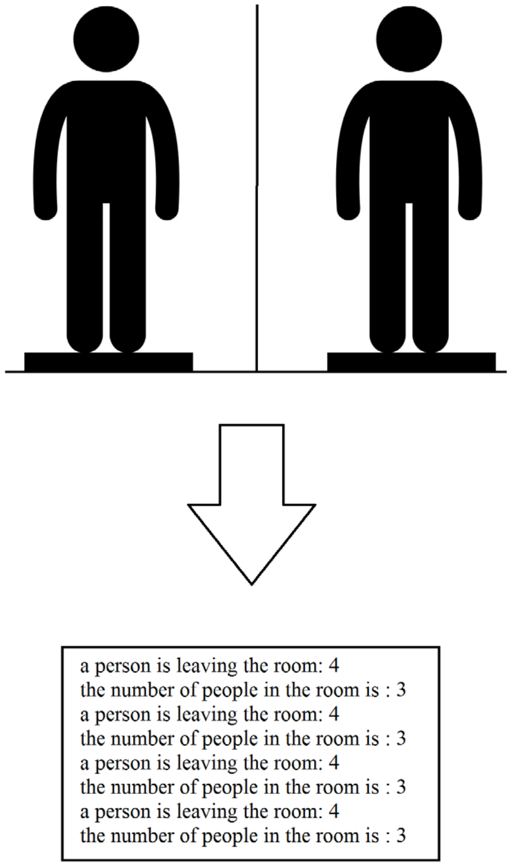


Figure 30: System output when an occupant is standing at both pressure mats at the same time

When a person entered or left the room after standing on both mats at the same time than this will result in a correct measurement. Because the last mat that is stepped onto determines the direction which that person is going. So ,in case one person presses both mats at the same time the outcome of the measurement is correct, although the system receives multiple signals of an occupant entering and leaving the room at the same time.

When multiple people enter the room, without a break in between than the pressure based system can miss occupants. This explains the people that were not measured in the group experiment table 13. But, it also explains the false measurements during the signal disruption experiment, where an occupant together with an object entered the room.

Both errors are the consequence of the play off between the both matts. The research Labeodan et al. (2015) and Fernandez-Luque, Martínez et al. (2013) used one single pressure sensor to determine the presence of an occupant, while this research explores the possibility to count the number of people.

All of the studies which made use of a pressure sensor (including this research) succeeded in detecting an occupant. But when the occupants need to be counted by two pressure points errors start to occur.

One possibility to prevent false measurements when multiple people are standing on both mats is by creating a grid of points that measure a person standing on the mat. Based on the shape and pattern of the pressed pressure point a person (or multiple people) could be recognized. Another option would be by enhancing the width of the pressure mat over 1.2 meter.

With the possible solution presented in this research, the system could have the potential to count multiple people entering and leaving the room at the same time. This makes the pressure based system, the system with the most potential to count the number of people in the room.

5.3.3 PIR based system

Based on the experiments from chapter 3, the PIR based system had the best performance from all of the occupancy detection systems. The experiment that tested the effect of entrance and exit speed on the reliability of the PIR based system showed that the system made only one mistake out of all the tests (see table 11). Also, the detection rate of the group experiment displayed a reliability of the PIR based system of 98% (table 12).

However, a marginal decrease can be seen when the occupants wore multiple layers of clothing. In this case, the detection rate decreased to 90% (see table 16). Cause of this error is that the extra layers of clothing are likely blocking the infrared radiation coming from the body of the occupant. This infrared radiation is the trigger for the system to react on.

However, this would not explain the false measurements that are observed in the group measurements (table 12). The false positive errors occurred in the group measurement when two people entered the room too close to each other. In this case, the PIR based system calculated only one occupant.

The probable cause of this error is the masking time. An example where this phenomenon also occurs is in the research of Wahl, Milenkovic, & Amft (2012) where they mention the PIR sensor had a masking time. The masking time is the time in which the output of the PIR sensor is forced off, during this time no measurements can take place (Micko, 2009). The system might be capable to reduce the masking time to a minimum, but this can still cause a problem when two people are walking to close to each other
A false positive error was also obsereved during

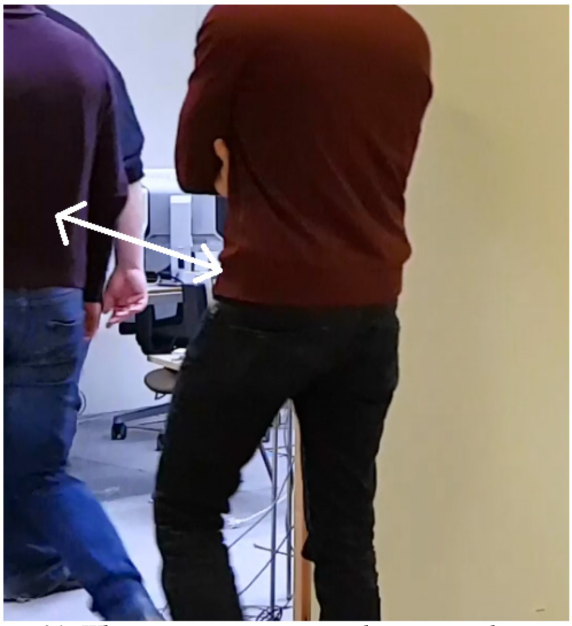


Figure 31: When two occupants enter the room to close to each other the PIR based system can recognize this as one person

a change in an hour. When a measurement took place between 11:59 and 12:01 the system made one false positive error, which coincides with the change of hour. Why this error is made is still unknown.

With the combination of the experiments done in this research, it can be said that the PIR based system is at this moment the best working system to count the number of people in the room, even compared to the research of Dedesko, Stephens, Gilbert, & Siegel (2015) and Wahl et al., (2012) However, a small decrease in reliablity is measured when an occupant is wearing thick clothing.

5.4 Occupancy agent

The question this project answered is: “can a smart occupancy agent be developed to obtain fine grained occupancy information and share this information with outside actors and agent, while safeguarding the privacy of the occupants?”

To answer this question three occupancy detection systems were tested for their ability to count the number of people in the room. Based on these experiments, it could be determined if the occupancy agent could be developed, and if so what combination of occupancy detection systems results in the highest accuracy.

Different occupancy detection systems can be combined. However, the experiments demonstrated that the PIR based system had the ability to produce reliable occupancy counting ability. This means that the other systems are not necessary to count the number of people in the room. However, as can be seen in table 16 the reliability of the PIR based system decreases somewhat when an occupant wears thick clothing. To prevent false measurements, it would be possible to combine the PIR based system with the pressure based system. This is also the reason why the pressure mat had a higher score in the Kesselring diagram as the PIR based system (see table 7).

When these two occupancy detection systems are combined they can reach a detection rate of 97%. This value is calculated with help of the system probability calculation (formula 3 at §2.1). For the formula a combination is used of the lowest detection rate of the PIR and pressure based system (see §4.3.3).

The thermal based system would not increase the reliability of the measurement because the system would have the same difficulty measuring occupants wearing thick clothing. This is demonstrated in the experiment where the same signal disruption is used as the PIR based system (see table 18).

Even with a high reliability, as demonstrated in this research, it would still be possible that the occupancy agent does make a mistake. In this case, the occupancy agent will provide false information to its environment. This could be prevented by providing more information to the occupancy agent. An example could be to measure the occupancy in the room with a CO2 sensor. Even though the sensor has a low response time (reference) the sensor might be used to check if an occupant is present or not.

All the of the researched detection systems are not revealing any data that could be traced back to the identity of the occupant entering or leaving the room in contrast to the research of Li, Calis, & Becerik-gerber (2012) and Benezeth et. al. (2011). Because no identity information is gathered the privacy of the occupant is no longer an issue as mentioned by Zeiler, Labeodan, Boxem, Maaijen, & Velden (2014).

All in all this research has demonstrated that it is possible to develop an occupancy agent that can obtain fine grained occupancy information, while safeguarding the privacy of occupants. Thereby making a further contribution by integrating the use of “demand driven control” with the build environment.

CHAPTER 6: CONCLUSION

The goal of this research is to answer the research question as defined in §1.2. In this chapter the different sub questions together with the main research question are evaluated. At the end of the chapter recommendations are given for further research.

6.1 The research question

To answer the research question several sub questions were defined. Each of the research questions can be found below together with the answer of these questions. At the end of this paragraph the research question is answered with help of the sub questions.

1. *Which key component can be distinguished in literature for detecting occupants and sharing occupancy information?*
In order to detect an occupant entering or leaving the room several sensors can be used. With help of the Kesselring method (§3.5) a selection is made which sensors have the most potential to be used in this research.

For the communication between the different agents and actors the middle-ware software program JADE is used. Communication in JADE takes place by different messages which are sent to outside actors or agents in two different ways. The agent can respond to a request form another occupant, or the occupancy agent can give an update at a predetermined time interval.

2.How Can an occupancy agent be designed which has the capability to obtain occupancy counting information, and shares this information to outside agents and actors?
The SYSMOD approach by Weilkiens (2006) is used to develop the occupancy agent. With this approach it is possible to build an occupancy agent based on multiple detection methods, which has the capability to communicate with outside actors and agents on the number of occupants in a room. The occupancy agent is optimized with the results of the reliability experiments and other occupancy information within single person passages.

3.What is the reliability of the developed smart occupancy agent?
When the occupancy agent was built, experiments are performed to determine the reliability of the different occupancy agents under varying conditions. Based on these experiments several measurements are not detected by the occupancy detection system. Closer analysis of these measurements leads to the insight that each of the occupancy detection system had their own flaws. These flaws and their possible solutions described in the discussion of this report (chapter 5)

How can a smart occupancy agent be developed to obtain fine grained occupancy information and share this information to outside actors and agent?

With help of the different sub questions the following conclusion for the research question is made:

•It is possible to determine the number of people in the room with the help of several occupancy detection systems, while the privacy of the occupants is guaranteed.

•The SYSMOD approach by Weilkiens (2006) is a suitable method for the development of a smart agent.

•The thermal based system as used in this research is not suited to count the number of people in the room since it has a reliability from the experiments between 58% and 11%.

•The PIR-based system performed the most accurate in the experiments with a reliability between 100% and 90%.

•The Pressure based system produced an accuracy from 100% to 74%.

•The combination between the pressure based system and the PIR based system increased the reliability up to 97.4%.

•This study demonstrated that it is possible for the occupancy agent to provide information about the occupancy to the other agents, by using JADE.

6.2 Recommendations for further research

This project succeeds in building an occupancy agent that is capable of communicating detailed occupancy information to other agents. However, there is still room for improvement on multiple levels. This last section is dedicated to possible future research which is closely connected to this research.

A possible future research could be focused on increasing the reliability of the detection systems. Each of the occupancy detection methods made errors, which creates a situation where the system gives information based on false measurements. When the error is not recognized by the occupancy detection system, it will continue its measurement with the error, creating false information on the occupancy of the room. For this reason, the improvement on this front is key for future applications.

Besides increasing the reliability of the occupancy agent, it could also be possible to research in techniques which could recognize an error. By doing so the agent can adjust its information when communicating with outside agents and actors.

Another possibility for future research is the use occupancy systems that can detect multiple people enter or leave the room at the same time. The detection systems investigated in this project only focus on one person, or multiple people following each other, entering the room. One of the promising systems, which could also have this function ,is the pressure based system. In the experiments, the pressure mats had only one surface from which the pressure was measured. If this surface is divided into multiple smaller measurement points it might be possible to also measure multiple people enter and leave the room at the same time.

The final recommendation focusses on the use of occupancy detection in the built environment. This research focuses on developing an agent that could communicate with other agents in the built environment. However, what the outside agent does with that information and what kind of adjustments are made in the building is not studied.

REFERENCES:

Amin, I. J., Taylor, A. J., Junejo, F., Al-Habaibeh, A., & Parkin, R. M. (2008). Automated people-counting by using low-resolution infrared and visual cameras. Measurement: Journal of the International Measurement Confederation, 41(6), 589–599. http://doi.org/10.1016/j.measurement.2007.02.010

Arduino - Home. (2016). Retrieved September 4, 2015, from https://www.arduino.cc/

Benezeth, Y., Laurent, H., Emile, B., & Rosenberger, C. (2011). Towards a sensor for detecting human presence and characterizing activity. Energy and Buildings, 43(2–3), 305–314. http://doi.org/10.1016/j.enbuild.2010.09.014

Berkum, E. E. ., & Bucchianico, D. (2009). Statistical Compendium.

Dedesko, S., Stephens, B., Gilbert, J. a., & Siegel, J. a. (2015). Methods to assess human occupancy and occupant activity in hospital patient rooms. Building and Environment, 90, 136–145. http://doi.org/10.1016/j.buildenv.2015.03.029

Diraco, G., Leone, A., & Siciliano, P. (2015). People occupancy detection and profiling with 3D depth sensors for building energy management. Sensors & Actuators: A. Physical, 92, 246–266. http://doi.org/10.1016/j.enbuild.2015.01.043

Duarte, C., Van Den Wymelenberg, K., & Rieger, C. (2013). Revealing occupancy patterns in an office building through the use of occupancy sensor data. Energy and Buildings, 67, 587–595. http://doi.org/10.1016/j.enbuild.2013.08.062

Erden, F., Ziya, A., & Enis, A. (2015). Infrared Physics & Technology A robust system for counting people using an infrared sensor and a camera. INFRARED PHYSICS & TECHNOLOGY, 72, 127–134. http://doi.org/10.1016/j.infrared.2015.07.019

Feng, X., Yan, D., & Hong, T. (2015). Simulation of occupancy in buildings. Energy & Buildings, 87, 348–359. http://doi.org/10.1016/j.enbuild.2014.11.067

Fernandez-Luque, F. J., Martínez, F., Domènech, G., Zapata, J., & Ruiz, R. (2013). EMFi-based low-power occupancy sensor. Sensors and Actuators, A: Physical, 191, 78–88. http://doi.org/10.1016/j.sna.2012.11.027

Goyal, S., Barooah, P., & Middelkoop, T. (2014). Experimental study of occupancy-based control of HVAC zones. Applied Energy, 140, 75–84. http://doi.org/10.1016/j.apenergy.2014.11.064

Gruber, M., Trüschel, A., & Dalenbäck, J.-O. (2014). CO2 sensors for occupancy estimations: Potential in building automation applications. Energy and Buildings, 84, 548–556. http://doi.org/10.1016/j.enbuild.2014.09.002

Hnat, T. W., Griffiths, E., & Dawson, R. (2012). Doorjamb : Unobtrusive Room-level Tracking of People in Homes using Doorway Sensors. SenSys.

Hong, T., & Lin, H. (2013). Occupant Behavior : Impact on Energy Use of Private Offices, (January).
IEA. (2013). Transition to sustainable buildings. Paris. Retrieved from http://www.iea.org/termsandconditionsuseandcopyright/

IEA (International Energy agency). (2016). Electricity Market Series Re-powering Markets Market design and regulation during the transition to low-carbon power systems. Paris. Retrieved from https://www.iea.org/publications/freepublications/publication/REPOWERINGMARKETS.pdf

International B.V. Sensor Development. (2016). SD international. Retrieved from http://www.sd-international.nl/

International Energy Agency (IEA). (2014). Energy Efficiency Indicators : Fundamentals on Statistics. International Energy Agency, Paris, 1–387. Retrieved from http://www.iea.org/termsandconditionsuseandcopyright/

IPCC. (2014). Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change.

Janda, K. B. (2014). Building communities and social potential : Between and beyond organizations and individuals in commercial properties. Energy Policy, 67, 48–55. http://doi.org/10.1016/j.enpol.2013.08.058

Jennings, N., & Wooldridge, M. (1995). Intelligent agents: theory and practice. The knowlegde and engineering review, vol. 10:2, 1995, 115-152

Katić, K., Li, R., Kingma, B., & Zeiler, W. (2017). Modelling hand skin temperature in relation to body composition. Journal of Thermal Biology, 69(July), 139–148. http://doi.org/10.1016/j.jtherbio.2017.07.003

Labeodan, T., Aduda, K., Boxem, G., & Zeiler, W. (2015). On the application of multi-agent systems in buildings for improved building operations, performance and smart grid interaction – A survey. Renewable and Sustainable Energy Reviews, 50, 1405–1414. http://doi.org/10.1016/j.rser.2015.05.081

Labeodan, T., Aduda, K., Zeiler, W., & Hoving, F. (2016). Experimental evaluation of the performance of chair sensors in an office space for occupancy detection and occupancy-driven control. Energy and Buildings, 111, 195–206. http://doi.org/10.1016/j.enbuild.2015.11.054

Labeodan, T., Zeiler, W., Boxem, G., & Yang, Z. (2015). Occupancy Measurement In Commercial Office Buildings For Demand-Driven Control Applications- A Survey And Detection System Evaluation. Energy and Buildings, 93, 303–314. http://doi.org/10.1016/j.enbuild.2015.02.028

Li, N., Calis, G., & Becerik-gerber, B. (2012). Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations. Automation in Construction, 24, 89–99. http://doi.org/10.1016/j.autcon.2012.02.013

Macarulla, M., Casals, M., Forcada, N., & Gangolells, M. (2017). Implementation of predictive control in a commercial building energy management system using neural networks. Energy and Buildings, 151, 511–519. http://doi.org/10.1016/j.enbuild.2017.06.027

Melfi, R., Rosenblum, B., Nordman, B., & Christensen, K. (2011). Measuring building occupancy using existing network infrastructure BT - 2011 International Green Computing Conference, IGCC 2011, July 25, 2011 - July 28, 2011.

Micko, E. (2009). PIR motion sensor. US Patent 7,579,595. Retrieved March 15, 2016, from http://www.google.com/patents/US7579595

Montgomery, D. C. (2003). Applied Statistics and Probability for Engineers Third Edition. Phoenix Usa (Vol. 37). http://doi.org/10.2307/1269738

Nguyen, T. A., & Aiello, M. (2013). Energy intelligent buildings based on user activity: A survey. Energy and Buildings, 56, 244–257. http://doi.org/10.1016/j.enbuild.2012.09.005

Omron D6T Mems thermal sensor. (2016). Retrieved May 25, 2016, from https://www.omron.com/ecb/products/pdf/en-d6t.pdf

Peruffo, A., Pandharipande, A., Caicedo, D., & Schenato, L. (2015). Lighting control with distributed wireless sensing and actuation for daylight and occupancy adaptation. Energy and Buildings, 97, 13–20. http://doi.org/10.1016/j.enbuild.2015.03.049

Pressure Mat Sensors -- Data Sheet. (n.d.). Retrieved May 28, 2016, from http://www.arun-electronics.co.uk/pdf/Pressure Mat Data Sheet - issue 1.pdf

Rijksdienst voor Ondernemend Nederland. (2013). InfobladTrias Energetica en energieneutraal bouwen (No. https://www.rvo.nl/sites/default/files/Infoblad%20Trias%20Energetica%20en %20energieneutraal%20bouwen-juni%202013.pdf).

Stocker, T. F., Plattner, G.-K., Tignor, M. M. B., Allen, S. K., Boschung, J., Nauels, A., Midgley, P. M. (2013). Climate Change 2013. http://doi.org/10.1017/CBO9781107415324.Summary

Telecom Italia SpA. (n.d.). Jade Site | Java Agent DEvelopment Framework. Retrieved April 4, 2017, from http://jade.tilab.com/

Tiedemann, K., Sulyma, I., Sahota, R., & Hydro, B. C. (2008). Behavior and Potential Energy Savings in Business Establishments. ACEEE American Council for an Energy-Efficient Economy, 327–335.

TU Delft. (2016). DINED. Retrieved June 2, 2016 from http://dined.io.tudelft.nl/en/database/tool

Veselý, M., Molenaar, P., Vos, M., Li, R., & Zeiler, W. (2017). Personalized heating – Comparison of heaters and control modes. Building and Environment, 112, 223–232. http://doi.org/10.1016/j.buildenv.2016.11.036

Veselý, M., & Zeiler, W. (2014). Tracking hand movement in an infrared image. Proceedings of 8th Windsor Conference: Counting the Cost of Comfort in a Changing World Cumberland Lodge, Windsor, UK, 10-13 April 2014, (April), 10–13.

Wahl, F., Milenkovic, M., & Amft, O. (2012). A distributed PIR-based approach for estimating people count in office environments. Proceedings - 15th IEEE International Conference on Computational Science and Engineering, CSE 2012 and 10th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2012, 640–647. http://doi.org/10.1109/ICCSE.2012.92

Weilkiens, T. (2006). Systems engineering with SysML/UML. Statewide Agricultural Land Use Baseline 2015 (Vol. 1). http://doi.org/10.1017/CBO9781107415324.004

Wooldridge. (2002). An introduction to MultiAgent Systems. JOHN WILEY & SONS, LTD. Liverpool.

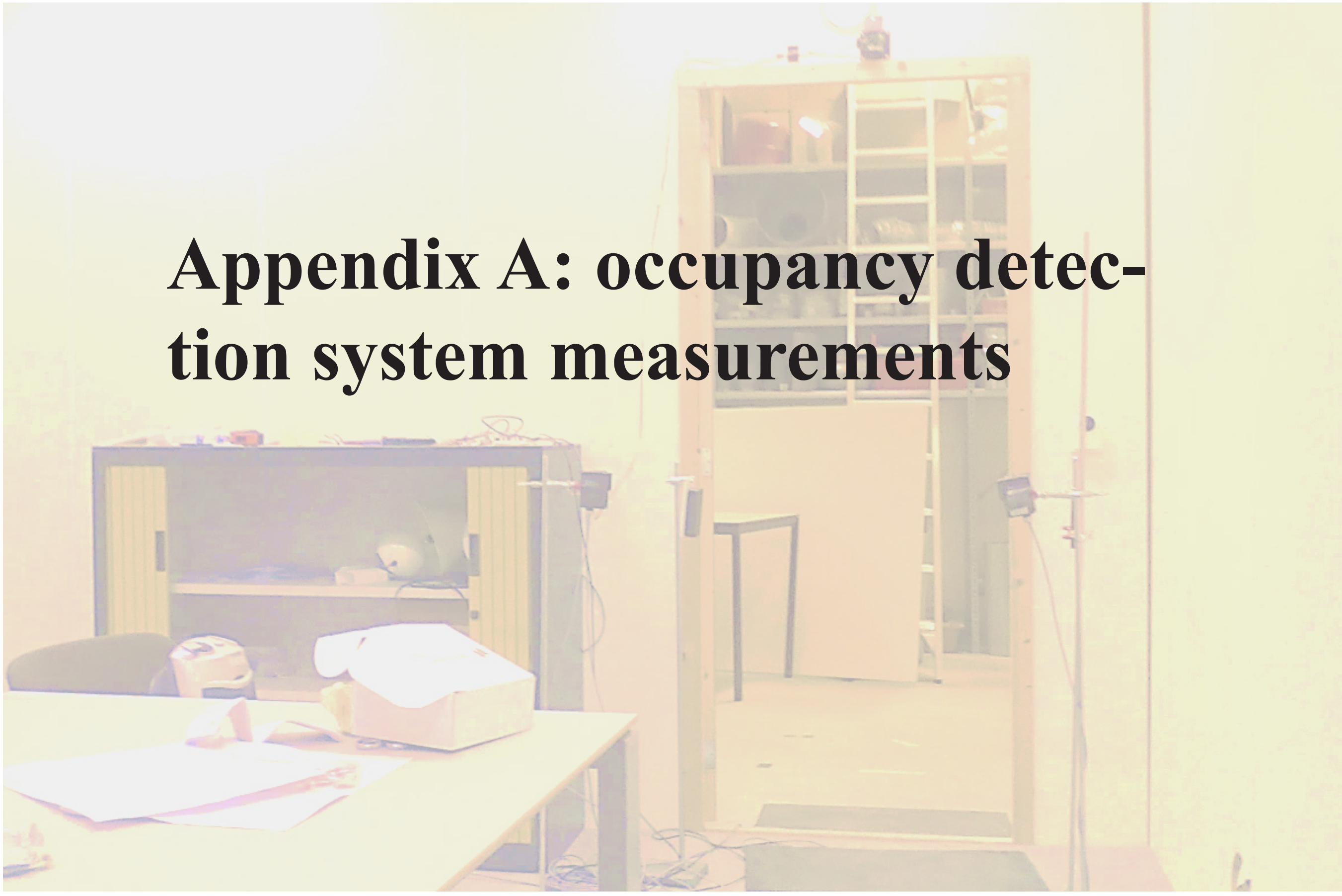
Zeiler, W., Labeodan, T., Boxem, G., Maaijen, R., & Velden, J. Van Der. (2014). Track and trace for optimal process control of energy flows within a building. Procedia Environmental Sciences, 22, 236–246. http://doi.org/10.1016/j.proenv.2014.11.023

Zeiler, W., Quanjel, E., & Savanovic, P. (2007). Design decision support for conceptual design. Iasdr (International Association of Societies of Design Research Conference), 1349–1350.

Appendices

Contents

Appendix A: occupancy detection system measurements



A1.1 RESULTS THERMAL BASED SYSTEM ENTRANCE SPEED

The tables below display the different measurement rounds that are done using the thermal based sensor. Each round is divided into two different columns, success and failure. The numbers indicate the time of the measurement in milliseconds, this is the time in which the occupant crossed a distance of 1.6 meter. If the time is situated under the success column then the detection system correctly measured a person entering or leaving the room. Else if the time is located under the false column, the detection system made a false measurement.

Measurement round 1 29 June 12:00	
success	failure
2923	1732
2676	2283
2281	2574
2409	2465
2226	2472
2487	2674
2636	2719
2506	2483
2580	1414
3169	1541
2742	1283
1580	1464
1549	1362
2199	1252
	1241
	1120
	1464
	1661
	2001
	1586
	1624
	3044
	2978
	1423
	2580

Measurement round 2 29 June 12:45	
success	failure
2964	1583
3000	2114
2941	3183
2444	2005
3017	1506
2103	1241
2551	1484
1400	1222
1413	1220
2743	1322
942	1448
2220	1024
979	1004
2679	1107
1243	1133
1019	1080
2883	991
1107	1006
1002	942
	3593
	2005
	1105
	1262
	1369
	2461

Measurement round 2 29 June 12:45	
success	failure
	1460
	1108
	1164
	1107
	1064
	1083
	1050
	1214
	1000
	1021
	1021
	2528
	2488
	3044

Measurement Round 3 12 July 10:20		Measurement round 4 12 July 11:00	
success	failure	success	failure
2690	2780	2659	2228
2560	5821	2164	2858
3224	5366	1002	2303
2741	5444	2549	941
2839	1724	1184	1003
3039	920	2482	1162
2723	902	2319	1161
880	1002		1380
909	844		953
6549	4863		2499
	804		3298
	6683		2241
	925		2566
	2324		960
			906
			883

A1.2 RESULTS THERMAL BASED SYSTEM GROUP MEASUREMENT

The table below displays the outcome of the group experiment done with the thermal based detection system. The measurements in which the occupants entered the room are displayed with an integer number, while the measurements in which the occupants left the room are displayed with,5 at the end of the measurement.

Measurement number	both directions	wrong direction	correct	misses
1	1	3	1	0
1,5	0	0	4	1
2	0	0	4	1
2,5	2	0	2	1
3	1	2	0	2
3,5	1	0	3	1
4	1	0	1	3
4,5	0	0	5	0
5	2	1	2	0
5,5	0	0	4	1
6	1	3	1	0
6,5	2	0	3	0
7	2	1	1	1
7,5	3	0	2	0
8	3	0	1	1
8,5	1	0	4	0
9	1	2	0	2
9,5	1	0	3	1
10	1	2	0	2
10,5	1	0	1	3
11	1	2	0	2
11,5	0	0	5	0
12	1	3	0	1
12,5	2	0	1	2
13	1	1	0	3
13,5	2	1	1	1
14	1	0	2	1
total	32	21	51	30

Measurement number	both directions	wrong direction	correct	misses
14,5	1	0	2	1
15	1	2	0	1
15,5	0	0	3	1
16	2	2	0	0
16,5	1	2	1	0
17	2	1	1	0
17,5	2	0	1	1
18	2	1	0	1
18,5	1	0	2	1
19	2	0	1	1
19,5	1	0	3	0
20	3	0	0	1
20,5	2	1	1	0
21	2	2	0	0
21,5	2	0	2	0
22	1	0	2	1
22,5	1	2	0	1
23	2	2	0	0
23,5	2	0	1	1
24	2	1	0	1
24,5	1	0	3	0
25	3	1	0	0
25,5	0	0	4	0
26	1	0	2	1
26,5	1	0	3	0
27	1	1	1	1
27,5	0	0	2	2
total	71	39	86	46

Measurement number	both directions	wrong direction	correct	misses
28	1	1	2	0
28,5	1	0	1	2
29	1	1	0	2
29,5	0	0	4	0
30	1	1	0	2
30,5	1	1	0	2
31	1	1	1	1
31,5	1	0	3	0
32	2	0	0	2
32,5	1	0	1	2
33	2	1	0	1
33,5	2	0	2	0
34	1	1	0	2
34,5	1	0	3	0
35	1	1	0	2
35,5	0	0	4	0
36	2	0	1	1
36,5	1	0	3	0
37	2	0	0	2
37,5	1	1	0	2
38	2	2	0	0
38,5	0	0	3	1
39	1	3	0	0
39,5	0	0	4	0
40	1	1	1	1
40,5	0	0	2	2
41	2	0	0	2
total	100	54	121	75

Measurement number	both directions	wrong direction	correct	misses
41,5	0	0	4	0
42	3	0	1	0
42,5	0	0	2	2
43	1	1	0	2
43,5	0	0	3	1
44	2	1	0	1
44,5	0	0	4	0
total	106	56	135	81

A1.3 RESULTS THERMAL BASED SIGNAL DISRUPTION

The table below displays the output from the system for each measurement during the signal disruption experiment.

1
#of people : 1
#of people (outgoing) : 0
2 (correct measurement)
#of people (outgoing) : -1
#of people : 0

3
Nothing

4(correct measurement)
#of people (outgoing) : -1
#of people : 0

5
#of people : 0
#of people : 1

6
#of people : 2
#of people (outgoing) : 1

7
#of people (outgoing) : 0
#of people : 1
#of people (outgoing) : 0
#of people : 1

8(correct measurement)
#of people (outgoing) : 0
#of people : 1

9
#of people : 2

10(correct measurement)
#of people (outgoing) : 1
#of people : 2

11
#of people : 3
#of people (outgoing) : 2
#of people : 3
12
#of people : 4

13
#of people (outgoing) : 3
#of people : 4
14
Nothing

15
#of people (outgoing) : -1
#of people : 0
#of people : 1

16
#of people : 2
#of people : 3

17
#of people (outgoing) : 2
#of people : 3
#of people : 4

18
#of people (outgoing) : 3
#of people : 4
#of people : 5

19
#of people : 1
#of people : 2

20
#of people (outgoing) : 1
#of people : 2
#of people : 3

21
#of people : 4
#of people : 5

22
#of people (outgoing) : 4
#of people : 5
#of people (outgoing) : 4
#of people : 5

23
#of people : 6
#of people : 7

24
#of people (outgoing) : 6
#of people : 7
#of people : 8

25(correct measurement)
#of people (outgoing) : 7
#of people : 8

26
#of people : 9
#of people : 10

27
#of people (outgoing) : 9
#of people : 10
#of people : 11

28(correct measurement)
#of people (outgoing) : 10
#of people : 11

29
#of people : 12

30(correct measurement)
#of people (outgoing) : 11
#of people : 12

31
#of people : 13
#of people : 14

32
#of people (outgoing) : 13
#of people (outgoing) : 12
#of people : 13

33
#of people (outgoing) : 12
#of people (outgoing) : 11

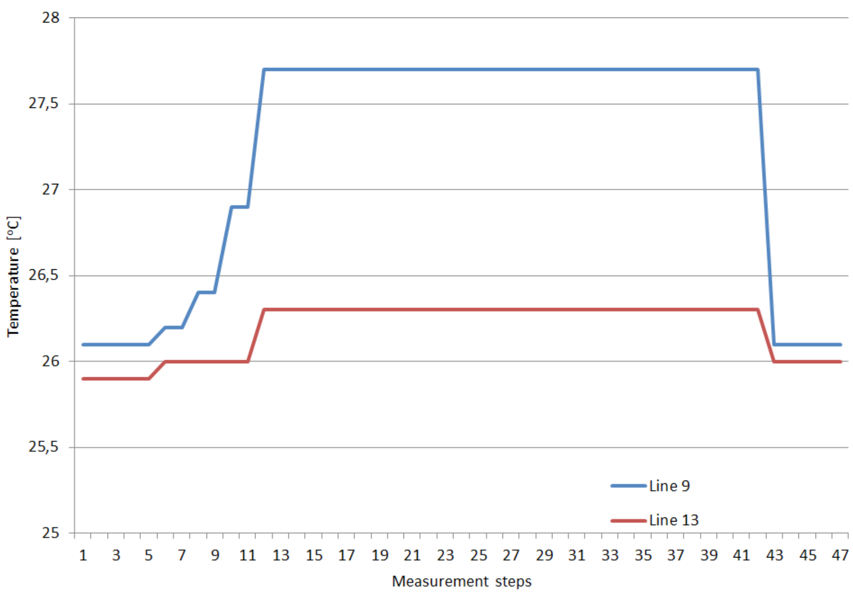
34
#of people (outgoing) : 11
#of people (outgoing) : 10

A3.3 RESULTS THERMAL BASED SIGNAL DISRUPTION WITH TEMPERATURES

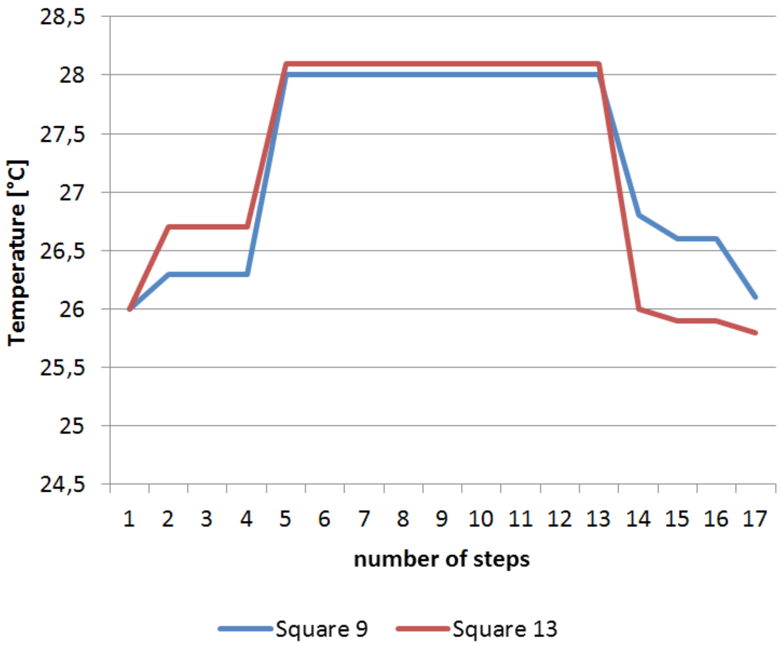
closer analysis of figure 28 (page 42 of the thesis) in the report different errors are identified. The graphs display the temperature over the number of measurements of two normative measurement points.

steps	Line 9	Line 13
1	26,1	25,9
2	26,1	25,9
3	26,1	25,9
4	26,1	25,9
5	26,1	25,9
6	26,2	26
7	26,2	26
8	26,4	26
9	26,4	26
10	26,9	26
11	26,9	26
12	27,7	26,3
13	27,7	26,3
14	27,7	26,3
15	27,7	26,3
16	27,7	26,3
17	27,7	26,3
18	27,7	26,3
19	27,7	26,3
20	27,7	26,3
21	27,7	26,3
22	27,7	26,3
23	27,7	26,3

steps	Line 9	Line 13
24	27,7	26,3
25	27,7	26,3
26	27,7	26,3
27	27,7	26,3
28	27,7	26,3
29	27,7	26,3
30	27,7	26,3
31	27,7	26,3
32	27,7	26,3
33	27,7	26,3
34	27,7	26,3
35	27,7	26,3
36	27,7	26,3
37	27,7	26,3
38	27,7	26,3
39	27,7	26,3
40	27,7	26,3
41	27,7	26,3
42	27,7	26,3
43	26,1	26
44	26,1	26
45	26,1	26
46	26,1	26
47	26,1	26



steps	Square 9	Square 13
1	26	26
2	26,3	26,7
3	26,3	26,7
4	26,3	26,7
5	28	28,1
6	28	28,1
7	28	28,1
8	28	28,1
9	28	28,1
10	28	28,1
11	28	28,1
12	28	28,1
13	28	28,1
14	26,8	26
15	26,6	25,9
16	26,6	25,9
17	26,1	25,8



A2.1 RESULTS PRESSURE BASED SYSTEM ENTRANCE SPEED

The table below displays the output of the pressure based system. Each (correct) measurements exists out of two numbers. The first number displays the number of occupants in the room (The negative number is because in the experiment the occupant only left the room). The second number displays the time that was necessary for the occupant to cover a distance of 1.6 meter.

Press 1 for Start/reset, 2 for elapsed time
Started...
-1,2118,
Started...
-2,2117,
Started...
-3,2248,
Started...
-4,2127,
Started...
-5,1974,
Started...
-6,1969,
Started...
-7,2005,
Started...
-8,1909,
Started...
-9,1976,
25167,
Started...
-10,1999,

// tweede ronde
Press 1 for Start/reset, 2 for elapsed time
-1,2054,
-2,2124,
-3,2210,
-4,1915,
-5,2060,
-6,2081,
-7,2199,
-8,2185,
-9,2108,
-10,2055,
-11,1998,
-12,2125,
-13,2151,
-14,2135,
-15,2264,
-16,1996,
-17,2084,
-18,2133,
-19,1802,
-20,1964,
-21,2181,
-22,2189,
-23,2180,
-24,2185,
-25,2259,
-26,2134,
-27,2121,
-28,2024,
-29,2175,
-30,2137,
-31,2225,
-32,2197,
-33,2102,
-34,2086,
-35,2154,
-36,2243,
-37,2221,
-38,2176,
-39,2304,
-40,2126,
-41,2184,
-42,2193,
-43,2363,
-44,2247,
-45,2214,
-46,1942,
-47,2230,
-48,2321,
-49,2130,
-50,2519,

// derde ronde
Press 1 for Start/reset, 2 for elapsed time
-1,1985,
-2,1814,
-3,2061,
-4,1926,
-5,1920,
-6,2086,
-7,2170,
-8,2193,
-9,1900,
-10,2204,
-11,1963,
-12,2024,
-13,2126,
-14,2191,
-15,2333,
-16,2072,
-17,2097,
-18,2183,
-19,2038,
-20,2314,
-21,2085,
-22,2050,
-23,2063,
-24,2061,
-25,2091,
-26,2088,
-27,12978,
-28,2188,
-29,2209,
-30,2161,
-31,2131,
-32,2051,
-33,2080,
-34,2139,
-35,2113,
-36,2408,

-37,2079,
-38,2278,
-39,2206,
-40,2153,
-41,2245,
-42,2112,
-43,2209,
-44,2147,
-45,2326,
-46,2009,
-47,2096,
-48,2074,
-49,2137,
-50,2294,

// ronde 4
Press 1 for Start/reset, 2 for elapsed time
-1,1470,
-2,1445,
-3,1297,
-4,1361,
-5,1196,
-6,1360,
11360,
910,
-7,1445,
-8,1452,
-9,1231,
-10,1330,
-11,1426,
-12,1493,
-13,1425,
-14,1419,
-15,1468,
-16,1338,
-17,1362,
-18,1386,
12640,
-19,1510,
-20,1418,
-21,1387,
-22,1410,
-23,1414,
-24,1303,
-25,1414,
-26,1415,
-27,1457,
-28,1235,
-29,1437,
-30,1508,
-31,1434,
-32,1377,
-33,1452,
-34,1451,
-35,1405,
-36,1448,
-37,1249,
-38,1358,
-39,1386,
-40,1406,
-41,1460,
-42,1476,
-43,1265,
-44,1435,
-45,1460,
-46,1402,
-47,1510,
-48,1213,
-49,1440,
-50,1389,

//ronde 5
Press 1 for Start/reset, 2 for elapsed time
-1,1418,
-2,1463,
-3,1432,
-4,1461,
-5,1212,
-6,1414,
-7,1443,
-8,1467,
-9,1417,
-10,1454,
-11,1391,
-12,1441,
-13,1353,
-14,1409,
-15,1456,
-16,1436,
-17,1405,
-18,1267,
-19,1395,
-20,1434,
-21,1399,
-22,1475,
-23,1465,
-24,1394,
-25,1445,
-26,1428,
-27,1457,
-28,1456,
-29,1398,
-30,1228,
-31,1440,
-32,1404,
-33,1414,
-34,1310,
-35,1280,
-36,1390,
-37,1461,
-38,1418,
-39,1410,
-40,1494,
-41,1421,
-42,1382,
-43,1476,
-44,1439,
-45,1211,
-46,1427,
-47,1484,
-48,1462,
-49,1442,
-50,1223,

// ronde 6
Press 1 for Start/reset, 2 for elapsed time
-1,3151,
-2,3800,
-3,3762,
-4,-3
,-4,4103,
-5,-4
,-5,3748,
-6,-5
,-6,3650,

-7,3814,
-8,4248,
-9,3648,
-10,3479,
-11,4192,
-12,3734,
-13,3681,
-14,3542,
-15,3587,
-16,3569,
-17,4119,
-18,3589,
-19,3560,
-20,3090,

// ronde 7
Press 1 for Start/reset, 2 for elapsed time
-1,3120,
-2,3466,
-3,3474,
-4,-3
,-4,3214,
-5,4089,
-6,3439,
-7,3348,
-8,3516,
-9,3815,
-10,3947,
-11,4175,
-12,3602,
-13,3756,
-14,3842,
-15,-14
,-15,3508,
-16,3510,
-17,3880,
-18,-17
,-18,3421,
-19,3378,
-20,3516,
-21,3696,
-22,3656,
-23,3755,
-24,-23
,-24,3694,
-25,3596,
-26,-25
,-26,3598,
-27,3274,
-28,4029,
-29,-28
,-29,3707,
-30,4251,

A2.1 RESULTS PRESSURE BASED SYSTEM GROUP EXPERIMENT

The table below displays the outcome of the group experiment done with the pressure based detection system. The measurements in which the occupants entered the room are displayed with an integer number, while the measurements in which the occupants left the room are displayed with,5 at the end of the measurement.

measurement	Correct	standing on both mats	missed	registration system error
1	4	1	1	0
1,5	6	0	0	0
2	6	0	0	0
2,5	2	2	2	3
3	3	2	1	2
3,5	1	2	3	3
4	0	2	4	4
4,5	4	1	1	1
5	3	1	2	3
5,5	3	1	2	1
6	6	0	0	0
6,5	2	2	2	2
7	1	1	4	3
7,5	2	1	3	3
8	2	2	2	3
8,5	2	2	2	3
9	1	1	4	4
9,5	6	0	0	0
10	3	1	2	2
10,5	3	1	2	2
11	3	1	2	2
11,5	3	1	2	2
12	4	1	1	1
12,5	3	1	2	1
13	2	1	3	4
13,5	4	1	1	1
14	4	1	1	1
14,5	2	2	2	2
15	4	1	1	1
15,5	3	1	2	3
16	3	1	2	1

measurement	Correct	standing on both mats	missed	registration system error
16,5	2	1	3	2
17	2	2	2	2
17,5	1	2	3	2
18	2	1	3	2
18,5	2	2	2	3
19	4	1	1	1
19,5	1	2	3	3
20	6	0	0	0
20,5	1	2	3	3
21	3	1	2	3
21,5	2	1	3	3
22	2	1	3	3
22,5	4	1	1	1
23	6	0	0	0
23,5	4	1	1	1
24	6	0	0	0
24,5	4	1	1	1
25	1	2	3	3
25,5	3	1	2	2
26	4	1	1	1
26,5	4	1	1	1
27	4	1	1	1
27,5	3	1	2	3
28	2	1	3	3
28,5	3	1	2	2
29	4	1	1	1
29,5	4	1	1	1
30	4	1	1	1
30,5	3	1	2	2
31	2	2	2	3
31,5	6	0	0	0
32	6	0	0	0

measurement	Correct	standing on both mats	missed	registration system error
32,5	4	1	1	1
33	6	0	0	0
33,5	3	1	2	2
34	5	0	1	0
34,5	4	1	1	1
35	4	1	1	1
35,5	6	0	0	0
total	232	73	115	117

A2.3 PRESSURE MAT SIGNAL DISRUPTION

The table below displays the output from the system for each measurement during the signal disruption experiment.

1
a person is leaving the room: 1
the number of people in the room is: 0
the number of people in the room is: -1
a person is leaving the room: 0

2
the number of people in the room is: -1
a person is leaving the room: 0

3
the number of people in the room is: -1
a person is leaving the room: 0

4
the number of people in the room is: -1
the number of people in the room is: -2
a person is leaving the room: -1

5
the number of people in the room is: -1
a person is leaving the room: 0
the number of people in the room is: -1
a person is leaving the room: 0

6
Press 1 for Start/reset, 2 for elapsed time
the number of people in the room is: -1
a person is leaving the room: 0
the number of people in the room is: -1
the number of people in the room is: -2
a person is leaving the room: -1

7
the number of people in the room is: -1
a person is leaving the room: 0

8
the number of people in the room is: -1
a person is leaving the room: 0

9
the number of people in the room is: -1
a person is leaving the room: 0

10
the number of people in the room is: -1
a person is leaving the room: 0

11
the number of people in the room is: -1
a person is leaving the room: 0

12
the number of people in the room is: -1
a person is leaving the room: 0

13
the number of people in the room is: -1
a person is leaving the room: 0

13
the number of people in the room is: -1
a person is leaving the room: 0

14
the number of people in the room is: -1
the number of people in the room is: -2
a person is leaving the room: -1

15
the number of people in the room is: -2
a person is leaving the room: -1

16
the number of people in the room is: -1
a person is leaving the room: 0

17
the number of people in the room is: -1
a person is leaving the room: 0

18
the number of people in the room is: -1
a person is leaving the room: 0

19
the number of people in the room is: -1
a person is leaving the room: 0

20
the number of people in the room is: -1
a person is leaving the room: 0

21
the number of people in the room is: -1
a person is leaving the room: 0
22
the number of people in the room is: -1
a person is leaving the room: 0

23
the number of people in the room is: -1
a person is leaving the room: 0

24
the number of people in the room is: -1
a person is leaving the room: 0

25
the number of people in the room is: -1
a person is leaving the room: 0
the number of people in the room is: -1

26
a person is leaving the room: 0
the number of people in the room is: -1

27
a person is leaving the room: 0
the number of people in the room is: -1
a person is leaving the room: 0

28
the number of people in the room is: -1
a person is leaving the room: 0

29
the number of people in the room is: -1
a person is leaving the room: 0

30
the number of people in the room is: -1
a person is leaving the room: 0

32
Press 1 for Start/reset, 2 for elapsed time
the number of people in the room is: -1
a person is leaving the room: 0
the number of people in the room is: -1
a person is leaving the room: 0

33
the number of people in the room is: -1
the number of people in the room is: -2
a person is leaving the room: -1

34
the number of people in the room is: -1
a person is leaving the room: 0

A3.1 PIR BASED SYSTEM

The result for the influence of speed on the reliability of the PIR based measurement are the same as the pressure based system.

A3.2 PIR BASED SYSTEM SIGNAL DISRUPTION WITH CLOTHING

Sd meting met kleren 10:47
21/20 Van 7 na 28 -> telt +1

Sd meting met kleren 10:47
21/20 Van 7 na 28 ->telt +1

Sd meting 11:26
Sensor A 10 ->20
Sensor B: 11 -> 23 sensor B telt +2

Sd meting 11:26
Sensor A 22 -> 27
Sensor B: 24 -> 29

Sd meting 11:44
Sensor A 27 -> 38 -> -1
Sensor B: 29 -> 39
Sd meting 11:55
Sensor A 38 -> 47 -> -1
Sensor B: 39 -> 48 -> -1

Sd meting 12:06
Sensor A: 0 -> 11 +1
Sensor B: 0 ->12 +2

Sd meting 13:05
Sensor A: 0 -> 11 +1
Sensor B: 0 -> 12 +2

Sd meting 13:17
Sensor A: 11 -> 20 -1
Sensor B: 12 -> 22

	Friday 2016-07-01	Saturday 2016-07-02	Sunday 2016-07-03	Monday 2016-07-04	Tuesday 2016-07-05	Wednesday 2016-07-06	Thursday 2016-07-07	Friday 2016-07-08	Saturday 2016-07-09	Sunday 2016-07-10	Monday 2016-07-11	Tuesday 2016-07-12	Wednesday 2016-07-13	Thursday 2016-07-14	Friday 2016-07-15	Saturday 2016-07-16
0:00 - 1:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1:00 - 2:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2:00 - 3:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3:00 - 4:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4:00 - 5:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5:00 - 6:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6:00 - 7:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7:00 - 8:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8:00 - 9:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9:00 - 10:00	0	0	0	0	4	1	0	0	0	0	0	1	0	0	0	0
10:00 - 11:00	0	0	0	0	7	0	0	19	0	0	0	52	0	0	3	0
11:00 - 12:00	0	0	0	0	3	2	1	24	0	0	0	48	0	5	0	0
12:00 - 13:00	0	0	0	0	3	2	0	2	0	0	0	71	5	1	7	0
13:00 - 14:00	0	0	0	1	3	1	7	2	0	0	0	37	0	3	4	0
14:00 - 15:00	0	0	0	0	1	0	9	3	0	0	41	0	0	7	2	0
15:00 - 16:00	0	0	0	2	1	0	25	1	0	0	27	0	0	11	2	0
16:00 - 17:00	0	0	0	0	2	0	36	1	0	0	28	0	11	79	2	0
17:00 - 18:00	0	0	0	0	5	0	12	7	0	0	23	2	1	1	1	0
18:00 - 19:00	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0
19:00 - 20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20:00 - 21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21:00 - 22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22:00 - 23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23:00 - 0:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	0	0	3	29	6	90	59	0	0	126	211	17	107	21	0
	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph

	Sunday 2016-07-17	Monday 2016-07-18	Tuesday 2016-07-19	Wednesday 2016-07-20	Thursday 2016-07-21	Friday 2016-07-22	Saturday 2016-07-23	Sunday 2016-07-24	Monday 2016-07-25	Tuesday 2016-07-26	Wednesday 2016-07-27	Thursday 2016-07-28	Friday 2016-07-29	Saturday 2016-07-30	Sunday 2016-07-31
0:00 - 1:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1:00 - 2:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2:00 - 3:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3:00 - 4:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4:00 - 5:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5:00 - 6:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6:00 - 7:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7:00 - 8:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8:00 - 9:00	-	0	-	0	0	0	0	0	0	0	0	0	0	0	0
9:00 - 10:00	-	0	15	14	0	0	0	0	0	0	0	6	0	0	0
10:00 - 11:00	-	0	94	11	0	0	0	0	27	28	0	61	0	0	0
11:00 - 12:00	-	0	50	3	0	0	0	0	43	95	0	141	4	0	0
12:00 - 13:00	-	2	17	1	0	0	0	0	42	29	0	80	2	0	0
13:00 - 14:00	-	43	3	5	0	0	0	0	44	45	0	148	1	0	0
14:00 - 15:00	-	36	4	1	0	0	0	0	6	0	0	0	1	0	0
15:00 - 16:00	-	30	0	0	0	0	0	0	4	0	0	4	0	0	0
16:00 - 17:00	-	105	5	3	0	2	0	0	0	0	0	0	0	0	0
17:00 - 18:00	-	1	8	1	0	0	0	0	2	0	0	2	0	0	0
18:00 - 19:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19:00 - 20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20:00 - 21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21:00 - 22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22:00 - 23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23:00 - 0:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	217	196	39	0	2	0	0	168	197	0	442	8	0	0
	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph

A3.3 PIR BASED SYSTEM SIGNAL DISRUPTION WITH HEATER

Sd international meting 10:45
20/20 7 to 27

Sd international meting 11:42
19/20 13 to 32

Sd international meting 12:29 sd 1
20/20 1 to 21

Sd international meting 12:40 sd 21
20/20 || 21 to 41

Sd international 13:10 sd 12
21/21 || 12 to 33

	Friday 2016-07-01	Saturday 2016-07-02	Sunday 2016-07-03	Monday 2016-07-04	Tuesday 2016-07-05	Wednesday 2016-07-06	Thursday 2016-07-07	Friday 2016-07-08	Saturday 2016-07-09	Sunday 2016-07-10	Monday 2016-07-11	Tuesday 2016-07-12	Wednesday 2016-07-13	Thursday 2016-07-14	Friday 2016-07-15	Saturday 2016-07-16
0:00 - 1:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1:00 - 2:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2:00 - 3:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3:00 - 4:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4:00 - 5:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5:00 - 6:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6:00 - 7:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7:00 - 8:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8:00 - 9:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9:00 - 10:00	0	0	0	0	4	1	0	0	0	0	0	1	0	0	0	0
10:00 - 11:00	0	0	0	0	7	0	0	19	0	0	0	52	0	0	3	0
11:00 - 12:00	0	0	0	0	3	2	1	24	0	0	0	48	0	5	0	0
12:00 - 13:00	0	0	0	0	3	2	0	2	0	0	0	71	5	1	7	0
13:00 - 14:00	0	0	0	1	3	1	7	2	0	0	0	37	0	3	4	0
14:00 - 15:00	0	0	0	0	1	0	9	3	0	0	41	0	0	7	2	0
15:00 - 16:00	0	0	0	2	1	0	25	1	0	0	27	0	0	11	2	0
16:00 - 17:00	0	0	0	0	2	0	36	1	0	0	28	0	11	79	2	0
17:00 - 18:00	0	0	0	0	5	0	12	7	0	0	23	2	1	1	1	0
18:00 - 19:00	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0
19:00 - 20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20:00 - 21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21:00 - 22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22:00 - 23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23:00 - 0:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	0	0	3	29	6	90	59	0	0	126	211	17	107	21	0
	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph

	Sunday 2016-07-17	Monday 2016-07-18	Tuesday 2016-07-19	Wednesday 2016-07-20	Thursday 2016-07-21	Friday 2016-07-22	Saturday 2016-07-23	Sunday 2016-07-24	Monday 2016-07-25	Tuesday 2016-07-26	Wednesday 2016-07-27	Thursday 2016-07-28	Friday 2016-07-29	Saturday 2016-07-30	Sunday 2016-07-31
0:00 - 1:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1:00 - 2:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2:00 - 3:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3:00 - 4:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4:00 - 5:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5:00 - 6:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6:00 - 7:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7:00 - 8:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8:00 - 9:00	-	0	-	0	0	0	0	0	0	0	0	0	0	0	0
9:00 - 10:00	-	0	15	14	0	0	0	0	0	0	0	6	0	0	0
10:00 - 11:00	-	0	94	11	0	0	0	0	27	28	0	61	0	0	0
11:00 - 12:00	-	0	50	3	0	0	0	0	43	95	0	141	4	0	0
12:00 - 13:00	-	2	17	1	0	0	0	0	42	29	0	80	2	0	0
13:00 - 14:00	-	43	3	5	0	0	0	0	44	45	0	148	1	0	0
14:00 - 15:00	-	36	4	1	0	0	0	0	6	0	0	0	1	0	0
15:00 - 16:00	-	30	0	0	0	0	0	0	4	0	0	4	0	0	0
16:00 - 17:00	-	105	5	3	0	2	0	0	0	0	0	0	0	0	0
17:00 - 18:00	-	1	8	1	0	0	0	0	2	0	0	2	0	0	0
18:00 - 19:00	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19:00 - 20:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20:00 - 21:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21:00 - 22:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22:00 - 23:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23:00 - 0:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	0	217	196	39	0	2	0	0	168	197	0	442	8	0	0
	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph	Graph

A3.3 PIR BASED SYSTEM SIGNAL DISRUPTION WITH HEATER

measure-ment	total counted people per hour	measured peo-ple one mea-surement	Comment
1	12	8	
2	20	8	
3	28	8	
4	36	8	
5	44	8	
6	52	8	
7	60	8	
8	68	8	
9	76	8	
10	84	7	
11	91	8	Kay and Pim walking close together
12	99	8	
13	107	8	
14	115	8	
15	123	8	
16	131	7	
17	138	8	Roel and Kay walking close together
18	146	8	
19	154	8	
20	162	9	
21	9	8	One person to many duet o hour change
22	17	7	
23	24	8	kay loopt sneller
24	32	8	
25	40	8	
26	48	8	
27	56	8	
28	64	8	
29	72	8	
30	80	8	

measure-ment	total counted people per hour	measured people one measurement	Comment
31	88	8	
32	96	8	
33	104	8	
34	112	8	
35	120	8	
36	128	8	
37	136	10	Jan (diepens) doet een ronde mee
38	146	8	
39	154	8	
40	163	9	1 persoon teveel

Appendix B: Software code for the occupancy detection systems

B1 CODE THERMAL BASED SYSTEM

```
/ D6T-44L test program for Arduino
// D6T_test_Arduino.ino
// 2013/03/27 http://www.switch-science.com

#include <Wire.h>
#include <WireExt.h>

#define D6T_addr 0x0A
#define D6T_cmd 0x4C

int rbuf[35];
float tdata[16];
float tdataN1[16];
float tdataN2[16];
float t_PTAT;
int a = 0;
int b = 0;
int counter = 0;
boolean y = false;
boolean z = false;

void setup()
{
  Wire.begin();
  Serial.begin(115200);
  Serial.flush();

  pinMode(17,OUTPUT);    //
  digitalWrite(17,HIGH);
  pinMode(16,OUTPUT);    //
  digitalWrite(16,LOW);
  Serial.println("start the measurement");
}

void loop()
{
int i;

     Wire.beginTransmission(D6T_addr);
      Wire.write(D6T_cmd);
      Wire.endTransmission();

      if (WireExt.beginReception(D6T_addr) >= 0) {

        i = 0;
```

```
        for (i = 0; i < 35; i++) {
          rbuf[i] = WireExt.get_byte();

        }


        WireExt.endReception();

        }

        for (i = 0; i < 16; i++) {
          tdataN2[i] =  tdataN1[i];}

        for (i = 0; i < 16; i++) {
          tdataN1[i] =  tdata[i];}

        //tdataN2[i] = tdataN1[i];
        //tdataN1[i] =  tdata[i];


        t_PTAT = (rbuf[0]+(rbuf[1]<<8))*0.1;
        for (i = 0; i < 16; i++) {

          tdata[i]=(rbuf[(i*2+2)]+(rbuf[(i*2+3)]<<8))*0.1;}

      entry();
      leaving();
      delay(100);
         }

void leaving(){
  if(tdataN2[6] > 26){
    if((tdataN1[10] > 26) && (tdata[10]< 26) ){
  counter ++;
  Serial.print("#of people : ");
  Serial.println(counter);
delay(200);
  }
}
}

void entry(){
if(tdataN2[13] > 26 ){
    if( (tdataN1[9] >26) && (tdata[9]<26) ){
  counter --;
  Serial.print("#of people (outgoing) : ");
```

```
  Serial.println(counter);
  delay(30);
  }
}
}
```

CSV

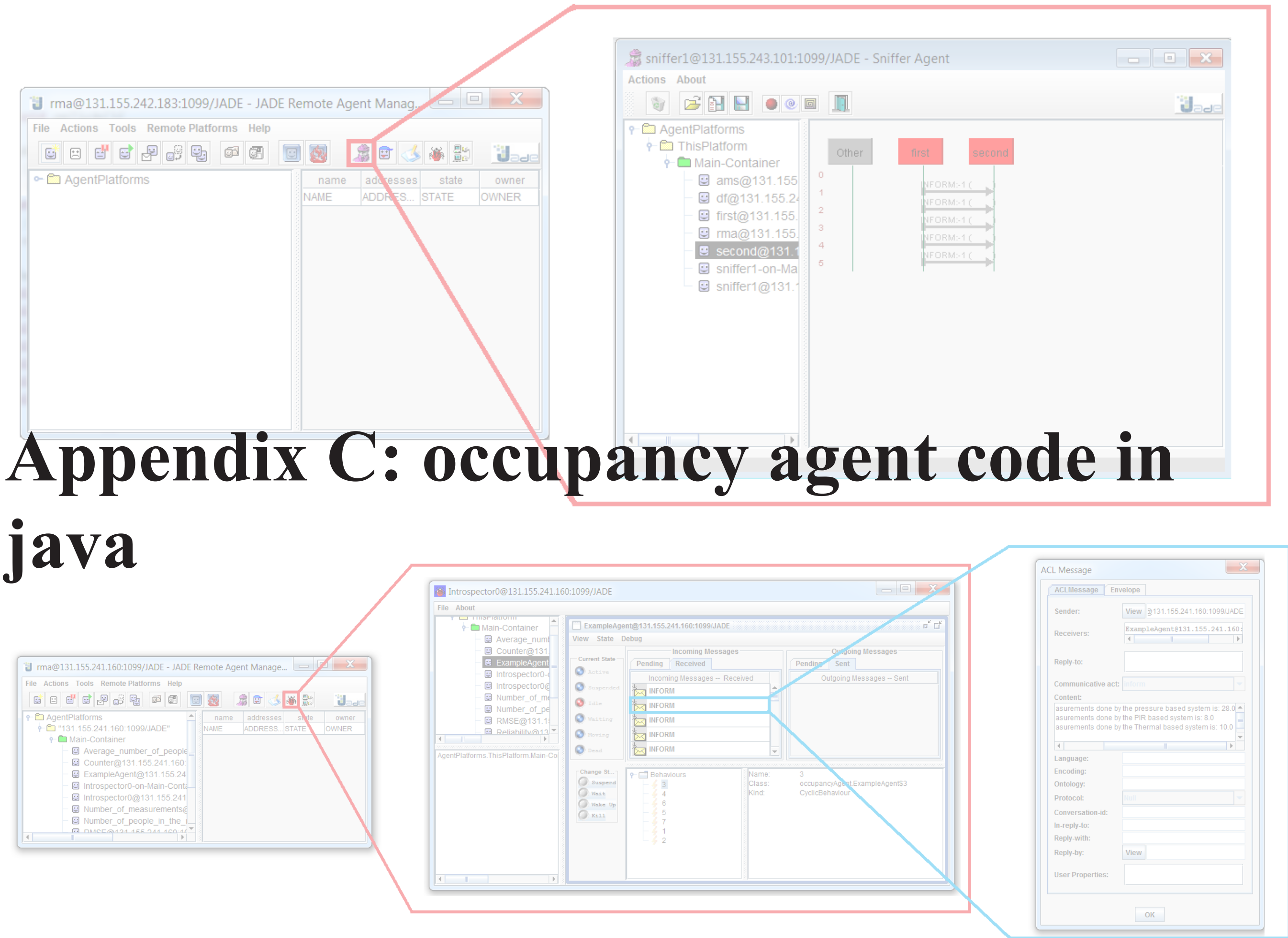
```
void output_csv() {
  for (int i = 0; i < 16; i++) {
    Serial.print(tdata[i]);
    if (i < 15) {
      Serial.print(“,”);
    }
    else {
      Serial.println();}}}
```

B2 : PRESSURE BASED SYSTEM

```
#include <SD.h>

File myFile;
float voltage = A0;
int val = 0;

unsigned long start, finished, elapsed;
unsigned long startt, finish, delapsed;
int counter = 0;

void setup() {
   pinMode(3, INPUT); // the start button
  pinMode(6, INPUT); // the stop button
   Serial.println("Press 1 for Start/reset, 2 for elapsed time");
  pinMode(voltage,INPUT);
  Serial.begin(9600);
  Serial.println("initializing the SD card...");

  if(!SD.begin(10)){
    Serial.println("initialization failed!!");
    return;
  }else{
    Serial.println("initialization complete!!");
  }
}

void displayResult()
{
  float h, m, s, ms;
  unsigned long over;
   elapsed = finished - start;
   delapsed = finish - startt;
  h    = int(elapsed / 3600000);
  over = elapsed % 360000;
    m    = int(over / 60000);
  over = over % 60000;
  s    = int(over / 1000);
  ms   = over % 1000;

  if(elapsed < 300){
    counter --;
    Serial.print("the number of people in the room is: ");
    Serial.println(counter);
    return loop();
```

```
    }else{
      //Serial.println("time has elapsed");
    }

    if(delapsed < 300){
    counter ++;
    Serial.print("a person is leaving the room: ");
    Serial.println(counter);
    return loop();
    }else{
      //Serial.println("time has elapsed (for a person leaving)");
    }
}

void loop() {
  if (digitalRead(3) == HIGH)
  {
    start = millis();
    finish = millis();
    delay(200); // for debounce
    displayResult();
  }

   if (digitalRead(6) == HIGH)
  {
    startt = millis();
    finished = millis();
    delay(200); // for debounce
    displayResult();
  }

  // SD card

  myFile = SD.open("DATA.csv", FILE_WRITE);
  if(myFile){
    myFile.print(counter);
    Serial.print(counter);
    myFile.print(",");
    Serial.print(",");
    myFile.println(millis());
    Serial.println(millis());
    myFile.close();
  }else{
    Serial.println("Error in opening file");
  }
 delay(20);}
```

Appendix C: occupancy agent code in java



C1: CASS NUMBER OF PEOPLE

```
package occupancyAgent;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.*;
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.OneShotBehaviour;
import jade.core.behaviours.TickerBehaviour;
import jade.lang.acl.ACLMessage;

public class NumberPeople extends Agent {
	protected void setup(){

// the Ticker behavior makes it possible to sent updates in a
predetermined time
	addBehaviour(new TickerBehaviour(this,2000){

	public void onTick(){
		FileReader pressure_Based = null;
		FileReader PIR_Based = null;
		FileReader thermal_Based = null;

// select files which the agent must read in order to obtain the
number of people in the room
	try {
		pressure_Based = new
FileReader("C:/Users/s119459/Desktop/test2.txt");
		PIR_Based= new
FileReader("C:/Users/s119459/Desktop/test.txt");
		thermal_Based =new
FileReader("C:/Users/s119459/Desktop/test3.txt");
					}
	catch (FileNotFoundException e) {
		e.printStackTrace();}

// Read the pressure based information file
		BufferedReader reader = new
	BufferedReader(pressure_Based);
		double current_Pressure_based_Measurement = 0;
		String Line = null;
```

```
// check the content of the pressure based information file for
line
        try {
                Line = reader.readLine();
        }catch (IOException e) {
                e.printStackTrace();}


// when the line is not null the "text number" is translated to a
double.
// when the line == null the loop stops and the last number is the
current amount of occupants
        while( Line != null){
                                double measurement =
Double.parseDouble(Line);
        try {
                        Line = reader.readLine();
        }catch (IOException e) {
                        e.printStackTrace();}
                        current_Pressure_based_Measurement =
measurement;}

                        BufferedReader read = new
BufferedReader(PIR_Based);
                        double current_PIR_based_Measurement = 0;

                                String Lines = null;
        try {
                        Lines = read.readLine();

        }catch (IOException e) {
                        e.printStackTrace();
                                }

        while( Lines != null){
                        double a = Double.parseDouble(Lines);

        try
                                Lines = read.readLine();
        }catch (IOException e) {
                                        e.printStackTrace();
                                }
                                 current_PIR_based_Measurement = a;}

//      Read the thermal based information file
```

```
                    BufferedReader reads3 = new
BufferedReader(thermal_Based);
                    double current_Thermal_based_Measurement = 0;

// check the line content of the thermal based information file
                          String Lines3 = null;
      try {
                    Lines3 = reads3.readLine();
      }catch (IOException e) {
                    e.printStackTrace();}

      while( Lines3 != null){
                    double measurement = Double.parseDouble(Lines3);

// when the line is not null the "text number" is translated to a
double.
 // when the line == null the loop stops and the last number is the
current amount of occupants
             try {
                    Lines3 = reads3.readLine();
             }catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();}
                    current_Thermal_based_Measurement =
measurement;}


// update message which sets the content of the message, type of
message and receiver
ACLMessage NumberPeople = new ACLMessage(ACLMessage.INFORM);

NumberPeople.setContent("The number of people in the room according
to the PIR based measurement is; " + current_PIR_based_Measurement
+ "\n" + "The number of people in the room according to the
Pressure based measurement is; " +
current_Pressure_based_Measurement + "\n" + "The number of people
in the room according to the Thermal based measurement is; " +
current_Thermal_based_Measurement + "\n" );

NumberPeople.addReceiver(new AID("ExampleAgent",AID.ISLOCALNAME));

send(NumberPeople);
```

```
            //request-inform-request-inform
            ACLMessage Counter = receive();
            if(Counter !=null){


      ACLMessage shoot = new ACLMessage(ACLMessage.INFORM);
      shoot.setContent("The number of; " +
current_PIR_based_Measurement );
      shoot.addReceiver(new AID("AgentRequest",AID.ISLOCALNAME));
      send(shoot);
                                    }


                  }
            });


      }}
```

C2: AGENT REQUEST

```
package occupancyAgent;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.*;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.OneShotBehaviour;
import jade.core.behaviours.TickerBehaviour;
import jade.core.behaviours.WakerBehaviour;
import jade.lang.acl.ACLMessage;
import javax.swing.JOptionPane;

public class AgentRequest extends Agent {


	protected void setup(){

// the Ticker waker makes it possible to sent a message ( in this
case a request) at a predetermined time
		addBehaviour(new WakerBehaviour(this, 15000) {
			protected void onWake() {

				// describe type of message ( in this
			case a request)
				ACLMessage Counter = new
ACLMessage(ACLMessage.REQUEST);
				// describe receiver
				Counter.addReceiver(new
AID("Number_of_people_in_the_room",AID.ISLOCALNAME));
				// describe request message
				Counter.setContent("try one");
				// describe content request
```

```
                        ACLMessage shoot = receive();
                        if(shoot !=null){
                              System.out.println("message 
received "+"\n" + shoot.getContent());

                        }else block();
                        send(Counter);
                  }
            });

      }}
```

C3: AVERAGE NUMBER OF PEOPLE IN THE ROOM

```
package occupancyAgent;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.TickerBehaviour;
import jade.lang.acl.ACLMessage;

public class AveragePeople extends Agent {

	private static final long serialVersionUID = 1L;

	protected void setup(){
				// the Ticker behavior makes it possible
			to sent updates in a predetermined time
				addBehaviour(new
TickerBehaviour(this,2000){

// select files which the agent must read in order to obtain the
number of people in the roo
					public void onTick(){
						FileReader pressure_Based =
null;
						FileReader PIR_Based = null;
						FileReader thermal_Based =
null;
	try {
		pressure_Based = new
	FileReader("C:/Users/s119459/Desktop/test2.txt");
		PIR_Based= new
FileReader("C:/Users/s119459/Desktop/test.txt");
		thermal_Based =new
FileReader("C:/Users/s119459/Desktop/test3.txt");
	} catch (FileNotFoundException e) {
		e.printStackTrace();}
// Read the pressure based information file
		BufferedReader reader = new
BufferedReader(pressure_Based);
		double total_Measurements_Pressure_Based = 0;
		double average_Number_Of_People_Pressure_Based = 0;
```

```
            double number_Of_Pressure_Based_Measurements = 0;

// check the content of the pressure based information file for
line
                          String Line = null;
      try {
            Line = reader.readLine();
      } catch (IOException e) {
            e.printStackTrace();
                          }

// when the line is not null the "text number" is translated to a
double.
      while( Line != null){
// the current total number of people the previous measurement +
the current number of people
      total_Measurements_Pressure_Based =
total_Measurements_Pressure_Based + Integer.parseInt(Line);

      try {
            Line = reader.readLine();
      } catch (IOException e) {
                                e.printStackTrace();}
// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
            number_Of_Pressure_Based_Measurements++;
// the average number of people can be calculated by dividing the
total number of people through the number of measurements
            average_Number_Of_People_Pressure_Based =
total_Measurements_Pressure_Based/number_Of_Pressure_Based_Measurem
ents;
                          }

// Read the PIR based information file
            BufferedReader read = new BufferedReader(PIR_Based);
            double total_Measurements_PIR_Based = 0;
            double number_Of_PIR_Based_Measurements = 0;
            double average_Number_Of_People_PIR_Based = 0;

// check the content of the PIR based information file for line
                          String Lines = null;
      try {
            Lines = read.readLine();
      } catch (IOException e) {
            e.printStackTrace();}
```

```
// when the line is not null the "text number" is translated to a
double.
       while( Lines != null){
// the current total number of people the previous measurement +
the current number of people
              total_Measurements_PIR_Based =
total_Measurements_PIR_Based + Integer.parseInt(Lines);;


       try {
              Lines = read.readLine();
       } catch (IOException e) {
              e.printStackTrace();}

// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
               number_Of_PIR_Based_Measurements++;
// the average number of people can be calculated by dividing the
total number of people through the number of measurements
 average_Number_Of_People_PIR_Based =
total_Measurements_PIR_Based/number_Of_PIR_Based_Measurements;


                          }

// Read the thermal based information file
              BufferedReader reads3 = new
BufferedReader(thermal_Based);
              double total_Measurements_Thermal_Based = 0;
              double number_Of_Thermal_Based_Measurements = 0;
              double average_Number_Of_People_Thermal_Based = 0;

// check the content of the thermal based information file for line

              String Lines3 = null;
       try {
              Lines3 = reads3.readLine();
       } catch (IOException e) {
              e.printStackTrace();}

// when the line is not null the "text number" is translated to a
double.
       while( Lines3 != null){
```

```
// the current total number of people the previous measurement +
the current number of people
 total_Measurements_Thermal_Based =
total_Measurements_Thermal_Based + Integer.parseInt(Lines3);

       try {
              Lines3 = reads3.readLine();
       }catch (IOException e) {
              e.printStackTrace();}
// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
              number_Of_Thermal_Based_Measurements++;
// the average number of people can be calculated by dividing the
total number of people through the number of measurements
              average_Number_Of_People_Thermal_Based =
total_Measurements_Thermal_Based/number_Of_Thermal_Based_Measuremen
ts;}



// update message which sets the content of the message, type of
message and receiver
              ACLMessage Average = new
       ACLMessage(ACLMessage.INFORM);

              Average.setContent("The average number of people in
              the room according to the pressure based system is
              is:" + average_Number_Of_People_Pressure_Based + "\n"
              + "The average number of people in the room according
              to the PIR based system is: " +
              average_Number_Of_People_PIR_Based + "\n" + "The
              average number of people in the room according to the
              Thermal based system is is: "+
              average_Number_Of_People_Thermal_Based);

                            Average.addReceiver(new
AID("ExampleAgent",AID.ISLOCALNAME));
                            send(Average);


                     }
              });

       }}
```

C4: COUNTER

```
package occupancyAgent;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.*;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.TickerBehaviour;
import jade.lang.acl.ACLMessage;
import javax.swing.JOptionPane;

public class Counter extends Agent {
	protected void setup(){

// the Ticker behavior makes it possible to send updates in a
predetermined time
		addBehaviour(new TickerBehaviour(this,2000){

// select files which the agent must read in order to obtain the
number of people in the room
	public void onTick(){
		FileReader pressure_Based = null;
		FileReader PIR_Based = null;
		FileReader thermal_Based = null;
	try {
		pressure_Based = new
FileReader("C:/Users/s119459/Desktop/test2.txt");
		PIR_Based= new
FileReader("C:/Users/s119459/Desktop/test.txt");
		thermal_Based =new
FileReader("C:/Users/s119459/Desktop/test3.txt");
	} catch (FileNotFoundException e) {
		e.printStackTrace();}
```

```
// Read the pressure based information file
                BufferedReader reader = new
BufferedReader(pressure_Based);
                double number_Of_Pressure_Based_Measurements = 0;

// check the content of the pressure based information file for
line
                String Line = null;
        try {
                Line = reader.readLine();
        } catch (IOException e) {
                e.printStackTrace();}

// when the line is not null the "text number" is translated to a
double.
// when the line == null the loop stops and the last number is the
current amount of occupants
        while( Line != null){
                double measurement = Double.parseDouble(Line);
        try {
                Line = reader.readLine();
        } catch (IOException e) {
                e.printStackTrace();}
// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
                number_Of_Pressure_Based_Measurements++;}

// Read the PIR based information file
                BufferedReader read = new BufferedReader(PIR_Based);
                double number_Of_PIR_Based_Measurements = 0;

// check the content of the PIR based information file for line
                String Lines = null;
        try {
                Lines = read.readLine();
        } catch (IOException e) {
                e.printStackTrace();}

// check the content of the PIR based information file for line
// when the line is not null the "text number" is translated to a
double.
        while( Lines != null){
                double a = Double.parseDouble(Lines);
```

```
        try {
              Lines = read.readLine();
        } catch (IOException e) {
              e.printStackTrace();}
// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
               number_Of_PIR_Based_Measurements++;}
// Read the PIR thermal information file
              BufferedReader reads3 = new
BufferedReader(thermal_Based);
              double number_Of_Thermal_Based_Measurements = 0;

// check the content of the thermal based information file for line
              String Lines3 = null;
        try {
              Lines3 = reads3.readLine();
        } catch (IOException e) {
              e.printStackTrace();
                          }

// check the content of the Thermal based information file for line
// when the line is not null the "text number" is translated to a
double.
        while( Lines3 != null){
              double measurement = Double.parseDouble(Lines3);

        try {
              Lines3 = reads3.readLine();
        } catch (IOException e) {
              e.printStackTrace();}
// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
              number_Of_Thermal_Based_Measurements++;}

// update message which sets the content of the message, type of
message and receiver
        ACLMessage Counter = new ACLMessage(ACLMessage.INFORM);

        Counter.setContent("The number of measurements done by the
        pressure based system is: " +
        number_Of_Pressure_Based_Measurements + "\n" +"The number of
        measurements done by the PIR based system is: " +
        number_Of_PIR_Based_Measurements + "\n" +"The number of
        measurements done by the Thermal based system is: " +
        number_Of_Thermal_Based_Measurements);
```

```
Counter.addReceiver(new AID("ExampleAgent",AID.ISLOCALNAME));
send(Counter);
             }
      });

}}
```

C5: EXAMPLE AGENT

```
package occupancyAgent;
import jade.core.Agent;
import javax.swing.JFrame;
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.OneShotBehaviour;
import jade.lang.acl.ACLMessage;

public class ExampleAgent extends Agent {
	protected void setup(){

// establish communication to retrieve reliability information
			addBehaviour( new CyclicBehaviour(){
		public void action(){
			ACLMessage reliability = receive();
	if(reliability !=null){
			System.out.println("message received "+
reliability.getContent());
	}else block();
			}});

// establish communication to retrieve number of people information

			addBehaviour( new CyclicBehaviour(){
	public void action(){
			ACLMessage NumberPeople = receive();
	if(NumberPeople !=null){
			System.out.println("message received "+
NumberPeople.getContent());
	}else block();
			}});

//establish communication to retrieve number of people information
			addBehaviour( new CyclicBehaviour(){
	public void action(){
			ACLMessage NumberPeople = receive();
	if(NumberPeople !=null){
			System.out.println("message received "+
NumberPeople.getContent());
		}else block();
	}});
```

```
//establish communication to retrieve average number of people
information
                    addBehaviour( new CyclicBehaviour(){
      public void action(){
                    ACLMessage Average = receive();
      if(Average !=null){
                    System.out.println("message received "+"\n" +
Average.getContent());
             }else block();
      }});

//establish communication to retrieve RMSE information
                    addBehaviour( new CyclicBehaviour(){
      public void action(){
                    ACLMessage RMSE = receive();
      if(RMSE !=null){
                    System.out.println("message received "+"\n" +
RMSE.getContent());
      }else block();
      }});

//establish communication to retrieve information on the number of
measurements
                    addBehaviour( new CyclicBehaviour(){
      public void action(){
                    ACLMessage Counter = receive();
      if(Counter !=null){
                    System.out.println("message received "+"\n" +
Counter.getContent());
             }else block();
      }});
//establish communication to retrieve information on the time of
each measuement
                    addBehaviour( new CyclicBehaviour(){
      public void action(){
                    ACLMessage Time = receive();
      if(Time !=null){
                    System.out.println("message received "+"\n" +
Time.getContent());
             }else block();}});
}}
```

C6: RELIABILITY

```
package occupancyAgent;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.*;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.TickerBehaviour;
import jade.lang.acl.ACLMessage;
import javax.swing.JOptionPane;

public class reliability extends Agent {
	protected void setup(){

// the Ticker behavior makes it possible to sent updates in a
predetermined time
		addBehaviour(new TickerBehaviour(this,2000){

// Select files which the agent must read in order to obtain the
number of people in the room
	public void onTick(){
		FileReader pressure_Based = null;
		FileReader PIR_Based = null;
		FileReader thermal_Based = null;
	try {
		pressure_Based = new
FileReader("C:/Users/s119459/Desktop/test2.txt");
		PIR_Based= new
FileReader("C:/Users/s119459/Desktop/test.txt");
		thermal_Based =new
FileReader("C:/Users/s119459/Desktop/test3.txt");
	} catch (FileNotFoundException e) {
		e.printStackTrace();}

// Read the pressure based information file
```

```
                BufferedReader reader = new
BufferedReader(pressure_Based);

// Initialization of the used values
                double total_Measurements_Pressure_Based = 0;
                double average_Number_Of_People_Pressure_Based = 0;
                double number_Of_Pressure_Based_Measurements = 0;
                double current_Pressure_based_Measurement = 0;
                double previous_Measurement_Total_PB = 0;
                double
previous_Number_Of_People_In_Room_Pressure_Based = 0;
                double previous_Measurement_PB=0;
                double reliancePressureSystem = 1;
                double rmse_Pressure_based_measurement =0;

// check the content of the pressure based information file for
line
                String Line = null;
        try {
                Line = reader.readLine();
        } catch (IOException e) {
                e.printStackTrace();}

// when the line is not null the "text number" is translated to a
double.
        while( Line != null){
// obtain previous measurement
                double measurement = Double.parseDouble(Line);
                previous_Measurement_PB =
previous_Measurement_Total_PB;
// obtain total previous measurement
                previous_Measurement_Total_PB =
total_Measurements_Pressure_Based;
// total measurements
        total_Measurements_Pressure_Based =
total_Measurements_Pressure_Based + Integer.parseInt(Line);
// total measurements
        double previous_Number_Of_People_In_Room =
previous_Measurement_Total_PB - previous_Measurement_PB; //new
previous number of people
```

```
// check the content of the pressure based information file for
line
        try {
                Line = reader.readLine();
        } catch (IOException e) {
                e.printStackTrace();}

// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
                number_Of_Pressure_Based_Measurements++;
// calculate the average number of people in the room
                average_Number_Of_People_Pressure_Based=total_Measurem
                ents_Pressure_Based /
                number_Of_Pressure_Based_Measurements;

                current_Pressure_based_Measurement = measurement;
                previous_Number_Of_People_In_Room_Pressure_Based =
previous_Number_Of_People_In_Room;}

// compare previous measurement with current measurement to
determine if a person is leaving or entering the room.
//based on the experiments of the occupancy detection system, a
reliability can be set on the measurements
        if(current_Pressure_based_Measurement -
previous_Number_Of_People_In_Room_Pressure_Based == 0){
                reliancePressureSystem = 0;
        }else if(current_Pressure_based_Measurement -
previous_Number_Of_People_In_Room_Pressure_Based >0){
                reliancePressureSystem = 0.73;
        }else{
                reliancePressureSystem = 0.73;}

// calculate root mean square error
        rmse_Pressure_based_measurement =
        Math.sqrt(Math.pow((average_Number_Of_People_Pressure_Based-
        current_Pressure_based_Measurement),2)/number_Of_Pressure_Bas
        ed_Measurements) *
        ((100*number_Of_Pressure_Based_Measurements)/total_Measuremen
        ts_Pressure_Based);

// Read the PIR based information file
        BufferedReader read = new BufferedReader(PIR_Based);

// Initialization of the used values
                double total_Measurements_PIR_Based = 0;
```

```
                double number_Of_PIR_Based_Measurements = 0;
                double average_Number_Of_People_PIR_Based = 0;
                double current_PIR_based_Measurement = 0;
                double previous_Measurement_PIR = 0;
                double previous_Measurement_Total_PIR = 0;
                double previous_Number_Of_People_In_Room_PIR_Based =
0;
                double reliancePIR = 0;
                double rmse_PIR_based_measurement = 0;

// check the content of the pressure based information file for
line
                String Lines = null;
        try {
                Lines = read.readLine();
        } catch (IOException e) {
                e.printStackTrace();}

// when the line is not null the "text number" is translated to a
double.
                                while( Lines != null){
                                double a = Double.parseDouble(Lines);
                                previous_Measurement_PIR =
previous_Measurement_Total_PIR; // obtain previous measurement
                                previous_Measurement_Total_PIR =
total_Measurements_PIR_Based; // obtain total previous measurement
                                 total_Measurements_PIR_Based =
total_Measurements_PIR_Based + Integer.parseInt(Lines);// total
measurements
                                double s2pre =
previous_Measurement_Total_PIR - previous_Measurement_PIR; //new
previous number of people

// check the content of the PIR based information file for line
                                try {
                                        Lines = read.readLine();
                                } catch (IOException e) {
                                        e.printStackTrace();
                                }

// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
                                 number_Of_PIR_Based_Measurements++;

// calculate the average number of people in the room
```

```
                                average_Number_Of_People_PIR_Based = 
total_Measurements_PIR_Based/number_Of_PIR_Based_Measurements;
                                current_PIR_based_Measurement = a;

previous_Number_Of_People_In_Room_PIR_Based = s2pre;

                              }

// compare previous measurement with current measurement to 
determine if a person is leaving or entering the room.
//based on the experiments of the occupancy detection system, a 
reliability can be set on the measurements
                              if(current_PIR_based_Measurement - 
previous_Number_Of_People_In_Room_PIR_Based == 0){
                                     reliancePIR = 0;
                              }else if(current_PIR_based_Measurement - 
previous_Number_Of_People_In_Room_PIR_Based  >0){
                                     reliancePIR = 0.14;
                              }else{
                                     reliancePIR= 0.14;
                              }

// calculate root mean square error
                              rmse_PIR_based_measurement = 
Math.sqrt(Math.pow((average_Number_Of_People_PIR_Based-
current_PIR_based_Measurement),2)/number_Of_PIR_Based_Measurements)
*
((100*number_Of_PIR_Based_Measurements)/total_Measurements_PIR_Base
d);

// Read the thermal based information file
                              BufferedReader reads3 = new 
BufferedReader(thermal_Based);

// Initialization of the used values
                              double total_Measurements_Thermal_Based = 
0;
                              double 
number_Of_Thermal_Based_Measurements = 0;
                              double 
average_Number_Of_People_Thermal_Based = 0;
                              double  measurements3 = 0;
                              double previous_Measurement_TB = 0;
                              double previous_Measurement_Total_TB = 0;
```

```
                              double
previous_Number_Of_People_In_Room_Thermal_Based = 0;
                              double reliance3 = 0;
                              double current_Thermal_based_Measurement
= 0;
                              double rmse_Thermal_based_measurement =
0;

// check the content of the pressure based information file for
line
                              String Lines3 = null;
                              try {
                                     Lines3 = reads3.readLine();
                              } catch (IOException e) {
                                     e.printStackTrace();
                              }

// when the line is not null the "text number" is translated to a
double.
       while( Lines3 != null){
              double measurement = Double.parseDouble(Lines3);
              previous_Measurement_TB =
previous_Measurement_Total_TB; // obtain previous measurement
// obtain total previous measurement previous number of people

              previous_Measurement_Total_TB =
              total_Measurements_Thermal_Based;
                     total_Measurements_Thermal_Based =
              total_Measurements_Thermal_Based +
              Integer.parseInt(Lines3); // total measurements

//new previous number of people
double previous_Number_Of_People_In_Room =
previous_Measurement_Total_TB - previous_Measurement_TB;

// check the content of the thermal based information file for line
       try {
              Lines3 = reads3.readLine();
       } catch (IOException e) {
              e.printStackTrace();}

// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
               number_Of_Thermal_Based_Measurements++;
// calculate the average number of people in the room
```

```
                average_Number_Of_People_Thermal_Based = 
total_Measurements_Thermal_Based/number_Of_Thermal_Based_Measuremen
ts;
               current_Thermal_based_Measurement = measurement;
               previous_Number_Of_People_In_Room_Thermal_Based = 
previous_Number_Of_People_In_Room;
                              }

// compare previous measurement with current measurement to 
determine if a person is leaving or entering the room.
                              //based on the experiments of the 
occupancy detection system, a reliability can be set on the 
measurements
                              if(current_Thermal_based_Measurement - 
previous_Number_Of_People_In_Room_Thermal_Based == 0){
                                     reliance3 = 0;
                              }else 
if(current_Thermal_based_Measurement - 
previous_Number_Of_People_In_Room_Thermal_Based  >0){
                                     reliance3 = 0.9;
                              }else{
                                     reliance3= 0.9;
                              }


//calculate the reliability of the combined occupancy detection 
methods
                              double Rtot = 0;
                              Rtot = 1 - ((1-
reliancePressureSystem)*(1-reliancePIR)*(1-reliance3));

                              rmse_Thermal_based_measurement = 
Math.sqrt(Math.pow((average_Number_Of_People_Thermal_Based-
current_Thermal_based_Measurement),2)/number_Of_Thermal_Based_Measu
rements) * 
((100*number_Of_Thermal_Based_Measurements)/total_Measurements_Ther
mal_Based);

// update message which sets the content of the message, type of 
message and receiver
                              ACLMessage reliability = new 
ACLMessage(ACLMessage.INFORM);
                              reliability.setContent("The reliability 
of the measurement is " + Rtot);
```

```
                              reliability.addReceiver(new
AID("ExampleAgent",AID.ISLOCALNAME));
                              send(reliability);
                       }
               });

       }}
```

C7: RMSE (ROOT MEAN SQUARE ERROR)

```
package occupancyAgent;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.*;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.TickerBehaviour;
import jade.lang.acl.ACLMessage;
import javax.swing.JOptionPane;

public class RMSE extends Agent {
	protected void setup(){

// the Ticker behavior makes it possible to sent updates in a
predetermined time
		addBehaviour(new TickerBehaviour(this,2000){

			public void onTick(){
				FileReader pressure_Based = null;
				FileReader PIR_Based = null;
				FileReader thermal_Based = null;

// select files which the agent must read in order to obtain the
number of people in the room
				try {
					pressure_Based = new
FileReader("C:/Users/s119459/Desktop/test2.txt");
					PIR_Based= new
FileReader("C:/Users/s119459/Desktop/test.txt");
					thermal_Based =new
FileReader("C:/Users/s119459/Desktop/test3.txt");
				} catch (FileNotFoundException e) {
					e.printStackTrace();
```

```
                              }

// Read the pressure based information file
                              BufferedReader reader = new
BufferedReader(pressure_Based);
                              double total_Measurements_Pressure_Based=
0;
                              double
average_Number_Of_People_Pressure_Based = 0;
                              double
number_Of_Pressure_Based_Measurements = 0;
                              double current_Pressure_based_Measurement
= 0;
                              double rmse_Pressure_based_measurement
=0;

// check the content of the pressure based information file for
line
                              String Line = null;
                              try {
                                     Line = reader.readLine();
                              } catch (IOException e) {
                                     e.printStackTrace();
                              }

// when the line is not null the "text number" is translated to a
double.
                              while( Line != null){
                              double measurement =
Double.parseDouble(Line);
// the current total number of people the previous measurement +
the current number of people
                              total_Measurements_Pressure_Based =
total_Measurements_Pressure_Based + Integer.parseInt(Line);

                              try {
                                     Line = reader.readLine();
                              } catch (IOException e) {
                                     e.printStackTrace();
                              }

// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
                              number_Of_Pressure_Based_Measurements++;
```

```
// the average number of people can be calculated by dividing the
total number of people through the number of measurements
                          average_Number_Of_People_Pressure_Based =
total_Measurements_Pressure_Based/number_Of_Pressure_Based_Measurem
ents;
                          current_Pressure_based_Measurement =
measurement;
                          }

// calculate the Root Mean Square Error of the pressure based
                          rmse_Pressure_based_measurement =
Math.sqrt(Math.pow((average_Number_Of_People_Pressure_Based-
current_Pressure_based_Measurement),2)/number_Of_Pressure_Based_Mea
surements) *
((100*number_Of_Pressure_Based_Measurements)/total_Measurements_Pre
ssure_Based);

// Read the PIR based information file
                          BufferedReader read = new
BufferedReader(PIR_Based);
                          double total_Measurements_PIR_Based = 0;
                          double number_Of_PIR_Based_Measurements =
0;
                          double average_Number_Of_People_PIR_Based
= 0;
                          double current_PIR_based_Measurement = 0;
                          double rmse_PIR_based_measurement = 0;

// check the content of the PIR based information file for line
                          String Lines = null;
                          try {
                                 Lines = read.readLine();
                          } catch (IOException e) {
                                 e.printStackTrace();
                          }

// when the line is not null the "text number" is translated to a
double.
                          while( Lines != null){
                          double a = Double.parseDouble(Lines);
// the current total number of people the previous measurement +
the current number of people
                           total_Measurements_PIR_Based =
total_Measurements_PIR_Based + Integer.parseInt(Lines);
```

```
                                try {
                                        Lines = read.readLine();
                                } catch (IOException e) {
                                        e.printStackTrace();
                                }

// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
                                 number_Of_PIR_Based_Measurements++;
// the average number of people can be calculated by dividing the
total number of people through the number of measurements
                                 average_Number_Of_People_PIR_Based =
total_Measurements_PIR_Based/number_Of_PIR_Based_Measurements;
                                 current_PIR_based_Measurement = a;
                                }

// calculate the Root Mean Square Error of the PIR based system
                                rmse_PIR_based_measurement =
Math.sqrt(Math.pow((average_Number_Of_People_PIR_Based-
current_PIR_based_Measurement),2)/number_Of_PIR_Based_Measurements)
*
((100*number_Of_PIR_Based_Measurements)/total_Measurements_PIR_Base
d);

// Read the thermal based information file
                                BufferedReader reads3 = new
BufferedReader(thermal_Based);
                                double total_Measurements_Thermal_Based =
0;
                                double
number_Of_Thermal_Based_Measurements = 0;
                                double
average_Number_Of_People_Thermal_Based = 0;
                                double  measurements3 = 0;;
                                double current_Thermal_based_Measurement
= 0;

                                double rmse_Thermal_based_measurement =
0;

// check the content of the thermal based information file for line
                                String Lines3 = null;
                                try {
                                        Lines3 = reads3.readLine();
                                } catch (IOException e) {
                                        e.printStackTrace();}
```

```
// when the line is not null the "text number" is translated to a
double.
                              while( Lines3 != null){
                              double measurement =
Double.parseDouble(Lines3);
// the current total number of people the previous measurement +
the current number of people
                              total_Measurements_Thermal_Based =
total_Measurements_Thermal_Based + Integer.parseInt(Lines3);


                              try {
                                     Lines3 = reads3.readLine();
                              } catch (IOException e) {
                                     e.printStackTrace();}
// Each time the loop runs the counter is raised with +1, which
corresponds with the number of measurements done
                               number_Of_Thermal_Based_Measurements++;
// the average number of people can be calculated by dividing the
total number of people through the number of measurements
                               average_Number_Of_People_Thermal_Based =
total_Measurements_Thermal_Based/number_Of_Thermal_Based_Measuremen
ts;
              current_Thermal_based_Measurement = measurement;
                              }

// calculate the Root Mean Square Error of the PIR based system
                              rmse_Thermal_based_measurement =
Math.sqrt(Math.pow((average_Number_Of_People_Thermal_Based-
current_Thermal_based_Measurement),2)/number_Of_Thermal_Based_Measu
rements) *
((100*number_Of_Thermal_Based_Measurements)/total_Measurements_Ther
mal_Based);


                              ACLMessage RMSE = new
ACLMessage(ACLMessage.INFORM);
                              RMSE.setContent("The RMSE acording to the
pressure based measurement is: " + rmse_Pressure_based_measurement
+"\n" +"The RMSE acording to the PIR based measurement is: " +
rmse_PIR_based_measurement +"\n" + "The RMSE acording to the
Thermal based measurement is: " + rmse_Thermal_based_measurement
+"\n");
```

```
                        RMSE.addReceiver(new
AID("ExampleAgent",AID.ISLOCALNAME));
                        send(RMSE);
                  }
            });

      }}
```

C7: TIME

```
package occupancyAgent;
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.*;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import jade.core.AID;
import jade.core.Agent;
import jade.core.behaviours.CyclicBehaviour;
import jade.core.behaviours.TickerBehaviour;
import jade.lang.acl.ACLMessage;
import javax.swing.JOptionPane;

public class Time extends Agent {
	protected void setup(){
		addBehaviour(new TickerBehaviour(this,2000){
			public void onTick(){

		int day, month, year, second, minute, hour;
 GregorianCalendar date = new GregorianCalendar();

// get the date + time from the opperating system
				day = date.get(Calendar.DAY_OF_MONTH);
				month = date.get(Calendar.MONTH);
				year = date.get(Calendar.YEAR);
				second = date.get(Calendar.SECOND);
				minute = date.get(Calendar.MINUTE);
				hour = date.get(Calendar.HOUR);

// update message which sets the content of the message, type of message and receiver
				ACLMessage Time = new
ACLMessage(ACLMessage.INFORM);
					Time.setContent("the time is "
+hour+" : "+minute+" : "+second);
					Time.addReceiver(new
AID("ExampleAgent",AID.ISLOCALNAME));
					send(Time);
			}
		}); }}
```