Agents on a Riot

Decentralized agents for crowd simulation

Jan-Willem Boon (1691155)
8/1/2012

vrije Universiteit amsterdam

TNO

Abstract
Demonstrations recently happened around the world, ranging from a few people protesting for a better lifestyle large masses as seen in Egypt, Libya, and Syria. Usually control forces try to prevent escalation by demonstrators. How control forces respond is important; if they use excessive force demonstrators will likely revert to violence as well, and the local population will lose trust in the government and peacekeeping forces. The interaction between control forces and a crowd is important because a crowd will behave differently given the actions of control forces. This interaction and overall crowd behaviour can be modelled using techniques from the field of Artificial Intelligence. The resulting virtual crowd can be used for either training control forces or creating mission planning.

The aim for this thesis is to develop a simulated crowd that uses intelligent agents to model demonstrators. The hypothesis is that a decentralized approach is better suited for crowd simulation than a centralized approach. In a decentralized approach each agent determines his actions by himself while in a centralized approach an agent is told what to do. Important factors are that a simulated crowd should be visualized using Virtual Battlespace 2, respond to control forces, and interact with non-lethal weapons.

Three research questions regarding a decentralized crowd have been formed: “How can a crowd of demonstrators be modelled using decentralized techniques?”, ”Is a decentralized crowd more realistic than a centralized crowd?”, and “How does the decentralized crowd perform in Virtual Battlespace 2?”.

Three literature studies have been conducted to find various interactions and aspects that are important in crowds, what actions and non-lethal weapons control forces use, and what existing artificial intelligence paradigms and cognitive models can be used to model a virtual demonstrator. To determine whether a decentralized crowd is better than a centralized crowd an experiment by a 2 by 2 design was set up. This experiment consists of a decentralized crowd and centralized crowd, which are the first factor, and two distinct scenarios, which are the second factor. Each type of crowd is scored on three attributes, number of agents in the vicinity, number of times an agent shouts, and the number of items thrown, during each scenario and these attributes are used to determine if a decentralized crowd is significantly different from a centralized one. The same attributes are also used to determine which crowd scores better, a higher score would mean that this crowd would be better suited. A normalized sum was constructed of these attributes. Normalization was applied to compare the observed values without bias. The difference between the two crowds is determined by an ANOVA and the means of the attributes are used to determine which crowd is better suited. Usability of the designed crowd is determined by measuring the frames per second in Virtual Battlespace 2 in comparison to the performance of the agents provided by Virtual Battlespace 2.

The results of literature studies, design, implementation, experiment, and measurements is a decentralized crowd model based on the “behaviour of individuals confronted with non-lethal weapons” model and uses emotions and behaviour trees for internal decision making and reasoning. In two out of three attributes the decentralized crowd was significantly different and had a higher mean. Using the weighted sum also determined that the decentralized crowd was significantly different and scored higher. The measurements of the frames per second showed that around 35 to 50 agents could be used at the same time to model a crowd.

The main conclusion of this thesis is that a decentralized approach to crowd simulation seems to be a good way to simulate a crowd. Overall scores were higher and the behaviour seems to be significantly different from a centralized approach. The designed crowd is also one of the first simulated crowds which uses behaviour trees to control the agents and it can be used in combination with Virtual Battlespace 2.

Table of Contents

Table of Figures

Table of Tables

Introduction

Imagine that you want to make a stand against the government. You gather some friends and go to the central marketplace which has the governmental palace as its neighbour. You start scanting slogans and throwing some rocks. Your friends are doing the same and someone in your neighbourhood is trying to rally the people around him. But then you see the government control forces. You are becoming slightly worried but keep going and then suddenly the control forces start deploying tear gas at your location. How would you act and what would you do?
TNO is currently working on the Asymmetric Threat Environment Analysis (ATHENA) program. This project aims to give an analysis of what asymmetric warfare entails and tries to describe this using models and scenarios. Asymmetric warfare is a situation where trained and well equipped forces are confronted with forces which are poorly trained and are less well equipped and compensate by using different tactics. An example is a demonstration on a city square as described in the first paragraph or the war in Vietnam between the United States Army and the Vietcong. Part of the ATHENA program is to create models of asymmetric threats in urban environments. The part of the program that is subject of this thesis is creating a model of a demonstration. Showcasing how the crowd would react is important because it gives insight in how people could react in real life situations. The resulting model can be used by control forces to train in handling demonstrations and what kind of impact deploying non-lethal weapons (NLWs) has. An important factor in conducting defensive and peacekeeping operations is winning the hearts and minds of the population, because peacekeeping operations are performed to increase stability in a region and to remain stable after the peacekeepers have left. Stabilizing a region is important because a peacekeeping force does not want to create a feeding bottom on which destabilizing factors can feed and take over when the peacekeepers leave.
One aspect of winning hearts and minds is how peacekeeping forces handle a demonstration. They do not want to lose cooperation of inhabitants by handling the demonstration in a wrong way, for example by using excessive force. To help train control forces to handle a crowd of demonstrators it is necessary to increase awareness of their actions and the influence on the demonstrators. A validated agent model can be used to replace role-players and provide control forces with more opportunity to train.
The main research question for this thesis will be: "**How can a crowd of demonstrators be modelled using decentralized techniques?**" Decentralized agents are agents which are responsible for their own actions. There is no central system telling agents what they should do; their behaviour is determined by their own beliefs, reasoning, actions and the observations of the environment in which they exist. The aim will be to research which techniques can be used to model a decentralized crowd and if the proposed solution is suited for use with Virtual Battlespace 2 (VBS2). The second research question is: "**Is a decentralized crowd more realistic than a centralized crowd?**" After choosing a suitable decentralized model a comparison is needed to measure the effectiveness of the decentralized crowd. This will be done by comparing the decentralized crowd with a centralized crowd using an experiment. The hypothesis is that demonstrators modelled with decentralized agents are better suited than a centralized approach, because a crowd's collective behaviour is an emergent phenomenon. The third research question is: "**How does the decentralized crowd perform in VBS2?**". VBS2 is a simulation environment, similar to a modern entertainment game, used for training contemporary military operations. To determine usability, the modelled decentralized crowd has to be combined with VBS2 in such a way that the agent can control entities in VBS2. This research question will also look at the scalability of the proposed agent model.

Research Methodology

To answer the questions "**How can a crowd of demonstrators be modelled using decentralized techniques?**", "**Is a decentralized crowd more realistic than a centralized crowd?**", and "**How does the decentralized crowd perform in VBS2?**" several steps have to be taken. These steps are combined in a roadmap which is described in the following paragraph.

Step 1 Research on how a crowd of demonstrators works. Determining the interactions between the demonstrators, with the control forces and NLWs. What is the influence of a leader on the crowd, what are the interactions between the various crowd members and how are subgroups formed? These aspects will be researched by using a literature study on crowds and crowd behaviour.

Step 2 What decentralized techniques exist to model individuals and how suited are they for crowd simulations? There exist several methods to model individual agents such as BDI, cellular automata and behaviour trees. These and others will be compared and reviewed. The comparison will be on the ability to react to various situations and if they can incorporate aspects like leadership, subgroups and emotional states.

Step 3 Design of the decentralized agent model. When a suitable artificial intelligence (AI) model has been found a design of the agent model will be made. In this model the various concepts and relations, found in step 1 and 2, will be integrated and described. In this step the design of the centralized crowd for comparison will also be described.

Step 4 Determining if a decentralized crowd is better than a centralized crowd is done in this step. To determine if there is a difference, each crowd is scored on three values: number of shouts, the number of thrown objects and number of agents in the vicinity. A normalized sum which combines these three values will be made. An experiment with a 2 by 2 design will be conducted, with the crowd type and the type of NLW as independent variables to determine if the decentralized crowd is significantly different from the centralized crowd and if it is better suited for crowd modelling. This will also be the hypothesis. To determine the difference, and if it is significant, between crowds an ANOVA test will be performed.

Step 5 The last step is determining if the designed agent model can be used in combination with VBS2. Most important aspects will be how fluent the simulation is and how many agents can be used at the same time. To establish this is, the frames per second within VBS2 will be recorded given the number of agents. A comparison will be made with the standard agents of VBS2 and determining the impact of the implemented agent model on the performance.

Scope

The scope of this graduation project is limited to a crowd of demonstrators. Designed agents will model the individual demonstrators in the demonstration. Relevant factors are: forming of the crowd, behaviour within the crowd, how crowd members respond to deployment of NLWs and stance of control forces. Outcome of this project will be an agent model for simulating demonstrators based of literature related to crowds, leadership and asymmetric warfare. The resulting model will be validated through a controlled experiment, its feasibility and usability for simulation will be determined by measurements and an experiment. Modelling control forces as agents which can respond to their environment will not be part of this thesis and the typical responds control forces will be to escalate by deploying NLWs.

Outline

In Chapter 1 the three literature studies regarding the NLWs, crowd behaviour and possible AI paradigms used in crowd simulation will be discussed. The design of the agent model based on the results of the literature studies will be described in Chapter 2. The implementation of the agent model which interacts with VBS2 will be discussed in Chapter 3. The validity and usability of the agent model will be described in Chapter 4. Following the experiment and results will be the conclusion and discussion, which are located in Chapter 5.

Chapter 1 Literature Study of Non-Lethal Weapons, Crowds and AI Paradigms

1.1 Crowd Behaviour

This Section is part of step 1 in the roadmap as described on page 2 and gives a partial answer to the research question ‘How can a crowd of demonstrators be modelled using decentralized techniques?’ This chapter entails the literature study performed to find the different kind of interactions between crowd members, leaders, control forces and other parties involved. Found aspects will be used in the design of the agent mode.

1.1.1 Interactions in Crowds

Crowds come in various forms. Some are large and have peaceful intentions while others are small and aggressive. According to Huis in 't Veld et al. (2011) the following types of crowds can be distinguished:

- General public,
- A mob. The crowd is active and often has an aggressive stance,
- A riot. A large mob with less cohesion,
- A crowd in panic. The crowd tries to escape the current situation,
- Social movement. A deliberately organized crowd that try to make a change happen.

The demonstration for this thesis is located in a Middle Eastern country and the aim of the demonstrators is to occupy a governmental building. Crowds in the Middle East are often quite large, as seen in recent demonstrations in Egypt, Bahrain, Libya and Syria, and consist of members from all ages and classes within society. Therefore the chosen definition for a crowd will be the riot. Dobias et al. (2006) states that a crowd is not a homogeneous whole. Every demonstrator is present for his own reasons but can be influenced by outside sources. He will sometimes enter a demonstration with friends or people that have the same kind reasons to demonstrate. The group of demonstrators tend to stick together throughout the demonstration and will most likely also leave together. The crowd as a whole will often have multiple subgroups of likeminded demonstrators, which have about 10 to 15 people as subgroup members as mentioned by (Dobias et al. 2006). Often it is the case that a subgroup is responsible for escalation of violence in a peaceful demonstration. Incorporating subgroups in the simulation will be important because of these reasons. In Figure 1 subgroups have been modelled by having different kinds of leaders and their interaction with subgroup members and the interaction between subgroup members. Subgroup members tend to stay physically close during a demonstration as stated by Dobias et al. (2006). This behaviour is described in Figure 1 by the ‘Trying to stay close’ line between a crowd member and a subgroup member. This behaviour will only occur if they belong to the same subgroup.
Besides subgroups it is also important to consider the individual in a crowd. For example Le Bon (1897) stated that people within a crowd feel anonymous and that an individual cannot decide for himself but just follows group decisions. Both cases have been proven not to be the case as stated by Couch (1968) and Kenny et al. (2001). They state that an individual is not a part of a larger entity which controls the individual’s behaviour. When an individual makes a decision this decision is based on his choices and is not dictated by the group. Meaning that following a group and behaving in the same way is a conscious choice. This explains why a crowd of people will only act in the same way in a very brief timeframe, such as cheering for a goal at a soccer match, and continues doing something different. For the agent model it means that agents will be modelled as individuals and letting them reason for themselves, i.e. decentralized. Nevertheless an individual can be influenced by a leader. According to Huis in 't Veld et al. (2011) leaders represent a big factor which determines if a crowd will turn aggressive or not. There are strictly spoken two kinds of leaders, calming or inciting leaders. These leaders also provide a source of trusted information for individuals who follow him. They would sooner adopt information provided by a leader than provided from other sources because of the trusted origin. This is called herding theory as stated by Hamilton (1970) where the survival of individuals

increases when they herd, in this case listening to trusted sources. This effect will be modelled by increasing the effect of leaders on individuals in comparison to the effect of crowds members on each other. Because leaders have such a major impact on a crowd they will be part of the agent model and are modelled in Figure 1 as the inciting and calming leader and their connection to crowd members.

When demonstrations tend to become aggressive the control forces are forced to respond. Behaviour of control forces and actions they perform also influence crowd members and their behaviour. Wetzer et al. (2010) states that a deescalating control force will also result in a deescalating crowd and escalation will result in an escalating crowd. However this will not be part of the agent model because control forces will only escalate. Control forces have however several responses they can use as stated by Kenny et al. (2001). They can hail demonstrators and notify that control forces are present; this usually makes the demonstrators aware of present control forces and makes them consider their action more carefully because of the potential response. Besides hailing control forces also adopt show of force, this shows demonstrators what they are capable to handle the demonstrators. Usually this makes the crowd more hesitant to escalate. Besides hailing and show of force, control forces can also warn demonstrators when they are going to deploy a NLW. Aim is to deter the crowd from escalating behaviour because they know that a reaction might be imminent. These different actions are described by hailing, show of force, and warning arrows in Figure 1.

When these actions do not work control forces will use NLWs to gain control of the demonstrators. Crowd members can either observe the deployment of a NLW or are directly influenced by it. These are modelled by the 'deployment of NLWs' and 'deployment' arrows in Figure 1. Besides the direct effect of a NLW, there is also an indirect effect; a crowd member can observe another crowd member getting hit by a NLW. This might cause the crowd member to become scared or maybe even angry. Besides deployment of NLWs and direct actions of control forces the effect of removing a leader can have an influence on the crowd. Removing a calming leader might escalate the crowd while removing an inciting leader might deescalate a crowd as stated by Kenny et al. (2001). Effectiveness of control forces also provides a big impact on the question if a crowd will become violent or not. If control forces are unable to handle a crowd, the demonstrators will see little risk and tend to perform more violent behaviour as stated by Frini et al. (2008). However if the effectiveness is high the demonstrators will normally not escalate. Effectiveness of the control forces will be part of the agent model and can found in Chapter 2.

Other aspects which could be relevant for behaviour of crowds are social economic, demographic, and political variables that contribute to the composition of a crowd. However these have been proven to be a bad indication of whether or not a demonstration will become aggressive and escalate as stated by Kenny et al. (2001). Therefore these will not be modelled in the agent model. Another factor is possible media influence. Members of a crowd can look at the impact of their actions and determine how effective they are. Early media coverage might even lead to a heightened risk of violence or contribute to more demonstrations but the direct influence on the crowd is rather limited. When media is present, the behaviour of the crowd is scarcely different compared to when media is not present as stated by Van Vliet and Fennis-Bregman (2004). Another factor which contributes to the behaviour is rumours and the spreading of them through the crowd. In uncertain environments a rumour might spread false information. Nervous people might accept this information as being true and behave accordingly Van Vliet and Fennis-Bregman (2004). This is will not part of the model because a demonstration will exist for a short time period and the effect of having rumours spreading around would probably be very small.

All interactions which have been deemed important are depicted in Figure 1. Spheres depict the various agents. Lines between spheres describe that there is a connection between agents and what kind of connection it is, for example observation or incitement. Leaders and their influence on the crowd member are described by Calming Leader / Inciting Leader sphere and arrows 'Calming' and 'Inciting to Violence'. Removing leaders is modelled by the 'Observe' arrow going from the crowd member to the 'Removal' arrow. Subgroups are depicted by leaders and subgroup member sphere. The control forces and their actions are depicted by the Control Forces sphere and 'Hailing', 'Show of Force' and 'Warning' lines between control forces and crowd members. Control forces can also use a NLW which interacts with a crowd member if he is being affected. A crowd member can also see a deployment even if he is not directly affected by it, as depicted by the 'Deployment' arrow between the NLW and the other Crowd Member and the 'Observing' arrow to this Deployment'.

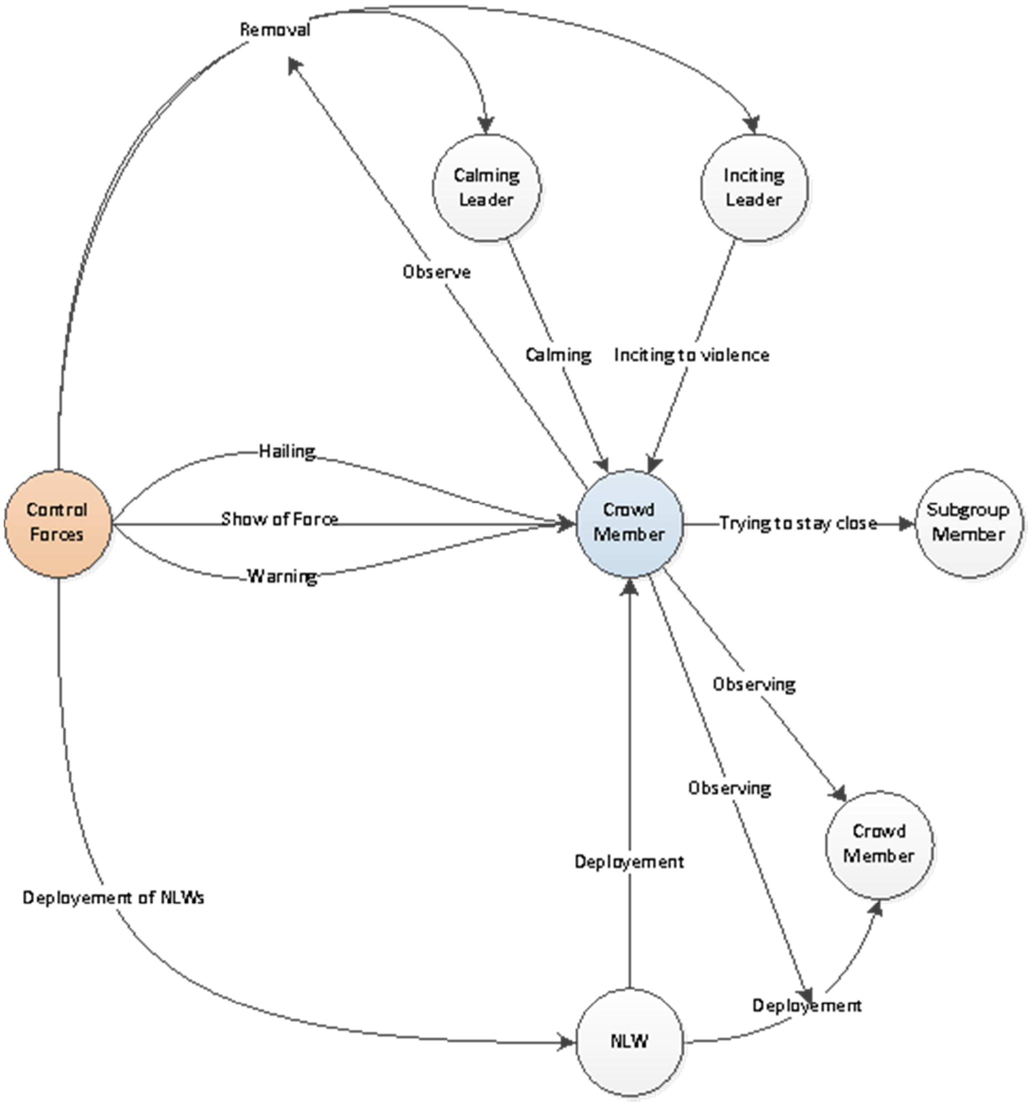

Figure 1. Overview of interactions in the crowd.

1.2 Non-Lethal Weapons

This Section is part of step 1 of the roadmap, introduced on page 2, and will answer a part of the research question 'How can a crowd of demonstrators be modelled using decentralized techniques?' using a literature study on control force operations and NLWs which can be used. The operations will be described in the following Section. Suitable NLWs are described in the Section Selected Non-Lethal Weapons.

1.2.1 Control Force Operations

During a demonstration control forces have to maintain a certain level of control over a crowd. Several operations have been defined by control forces that aim to achieve this. Linking these operations to usable NLWs is crucial to determine which NLWs have to be modelled in the agent model and used in the experiment. One of the operations that control forces perform is securing an object. Object in this case refers to a government building that has to be protected from demonstrators. Several important factors regarding this task are described by Wetzer and Horst (2011). Important factors for suitable NLWs are that they should be usable on land and in an urban environment. The NLW should be able to affect a large crowd when deployed and should have at least an effective radius of 60 meters, an immediate effect on the subject, and a high level of compellingness, meaning that the crowd cannot shrug away the effect. Several NLWs are suitable, such as teargas (chlorobenzylidenemalononitrile, known as CS), malodorants, calmatives, spoken word, irritating sounds, water cannon, high power microwave (HPM), slippery surface, sticky foam, net, baton, and a shield. These NLWs have in common that their effect is immediate on the subject; they work in an urban environment, and have a suitable range and radius when correctly used. Some NLWs are either too passive (in case of the sticky foam, net and slippery surface), or have questionable ethics such as calmatives. Others are deemed not effective enough, such as irritating sound or spoken word because they might lack the compellingness in comparison to others.

Another task that control forces perform is trying to control a crowd. This means that control forces have to be able to steer a crowd into a direction they want and prevent them from escalating. Important factors for NLWs for this operation are described by Wetzer and Horst (2011). Most important factors are that a NLW has to work in an urban environment, have an effective range of 200 meters, but the effect on the subject does not have to be immediate but should have a high level of compellingness when the effect is present. Suitable NLWs for this task are the calmatives, malodorants, spoken word, calming sound and pleasant light.

The third and final operation control forces perform during a demonstration is taking out an individual when the need arises, e.g. when a leader is enticing the crowd to become more aggressive. For this operation there are several factors that are important, as described by Wetzer and Horst (2011). Most important factors are that the NLWs should perform in an urban environment; the range is between 5 and 20 meters, have an immediate effect on the subject, and a high level of compellingness. Suitable NLWs for this task are the water cannon, HPM, baton, rubber bullets, or baton rounds. These can be aimed at a single person and deliver just enough force to only have an effect on this person but not on demonstrators standing around him.

Teargas and HPM were chosen to be implemented and form the basis of two scenarios because they are usable in two of the three operations. Teargas is also a common NLW known by riot police and peace keeping forces and the effect on people is known. The HPM was selected because it is useful in two operations and is a new technique which is interesting to model, however the effect on people is not yet known in detail so certain assumptions have to be made, such as the exact influence on behaviour. Other known lethal weapons were not selected because their effect is either too small, such as pleasant light and sounds, or are questionable in nature such as the calmatives, or are similar to one of the selected weapons such as the malodorants in comparison to teargas.

1.2.2 Selected Non-Lethal Weapons

In this Section the selected NLWs will be described briefly to give insight in what they are and what kind of effect they have on people.

Teargas (CS) is a gas which influences the tear ducts or the human body causing people to cry, couch and feel miserable. This form of teargas was developed in the 1920s by Ben Corson and Roger Stoughton. Teargas is frequently deployed by police units all over the world and is deemed to be non-lethal. In war the use of teargas by armed forces is illegal because there are international laws against the use of chemical agents in war because of the potential to escalate to lethal gases. In some cases teargas has proven fatal because of a lack of ventilation or because of overexposure, but in common situations the effects wear off after time and no serious harm would come to people.

Figure 2. Teargas being deployed in the United Kingdom.

A relatively new NLW is the high power microwave (HPM). This is a device that sends a microwave beam towards a crowd. The people who are affected will get a sense of burning skin and will have the immediate sensation to leave the beam because of the burning feeling. When the person leaves the beam the effect is immediately gone and no lingering effects are present. The NLW is currently under development and is not in active service, but it seems to pave the road for future NLWs with its immediate effect without short-term consequences. Because this kind of NLW is still in development the long-term effects on humans are not yet known and might result in the HPM not being used at all.

Figure 3. High Power Microwave mounted on a 8 wheeled truck.

1.3 Suitable AI Paradigms and Cognitive Models

In this Section existing AI paradigms used for crowd simulation will be researched for applicability. Besides the paradigms, cognitive models for individuals will be researched to see if they can be used for the agent model. In both cases a literature study was used to research this. This chapter is part of step 2 introduced in the roadmap on page 2 and answers a part of the research question 'How can a crowd of demonstrators be modelled using decentralized techniques?'.

1.3.1 AI Paradigms

This Section describes various techniques which can be used to model crowds or individuals in AI literature. Important aspects which models should have is that the world they represent is continuous, agents are capable to use emotions, possibility for leadership, and interaction with NLWs. Continuous meaning that there is no overlying structure which divides the world, such as a grid, which makes integration with VBS2 easier.

1.3.1.1 Particle Systems

Particle systems or social force models consist of individuals who are modelled as a particle Helbin and Molnár (1998). Particles are attracted or repulsed by other particles or objects in the world and do not have an internal state. This world in which they exist is a continuous world, meaning they can move in every direction they want. An example of a particle system is a simulated swarms of birds, such as those shown in Figure 4 which is based on the research by Reynolds (1987). Particle systems can also be used for simulation of evacuation plans and other planning problems where congestion might pose a problem. In case of crowd simulation each particle will depict an individual crowd member which is attracted or repulsed by other objects such as walls or other particles. If there is a large number of particles around him than there are more forces being applied and changes his behaviour. To incorporate emotions new types of particles have to be introduced where each emotion would correspond to a type of particle. When a particle updates its emotion, the particle should be replaced by a new one which corresponds to the emotion. Aspects like motivation, physical limitation, and stress cannot be modelled by particles because a particle cannot store an internal state and can only interact with external forces. Goals such as a government building could be modelled by using a location in the virtual world with an certain attraction for particles and based on the emotion a particle could either move towards it or avoid getting near.

Figure 4. Swarm of Simulated Birds using a Particle System.

1.3.1.2 Cellular Automata

Cellular automata are models where the world is divided into a grid like structure as stated by J. von Neumann (1966) and where each grid cell has a state. The state of a grid cell is determined by the states of neighbouring grid cells. A famous example of cellular automata is the game of life of Conway (1970) as shown in Figure 5. In the game of life each cell can either be dead or alive given the number of neighbours. If a cell is surrounded by two neighbours he will stay alive, if he is surrounded by three neighbours he will become alive and in the other cases a cell will change to the dead state. Cellular automata are used to model tree growth or development of urban areas on a high abstraction level but they can also be used for crowd simulation. Each grid cell would describe an individual where movement of an individual can be described by using rules. These rules can be in the form of that an agent wants to move in a certain direction and looks if a grid cell in that direction is free, if so the agent moves to this free space, if not than the agent looks in a different direction. Subgroups and leadership can be modelled by letting an agent look at its neighbours and see which belong to the same subgroup or is a leader of the subgroup. If this is the case he can try to move in that direction to stay close. Cellular automata can also be used to model influence of a leader. A leader would spread its influence, based on rules, among cells. An agent can be determined to be influenced when he falls inside on these influenced grid cells. In the same way the spreading of emotions can be modelled. For example when an agent becomes panicked the grid cell in which he currently is can change its state and the other grids around him can adopt this state based on internal rules. This way agents who are affected by a panic can be modelled using cellular automata. Aspects such as motivation, stress, and physical factors cannot be modelled because these are internal factors of an agent and cannot be modelled as a single state, which forms the basis of cellular automata. Another problem with cellular automata is that the world is represented as a grid. When a cellular automata has to be combined with simulation software a translation mechanism would be needed to translate grid cells to correct coordinates in the virtual world. This could prove to be time consuming and possibly reduced the accuracy of the agents.

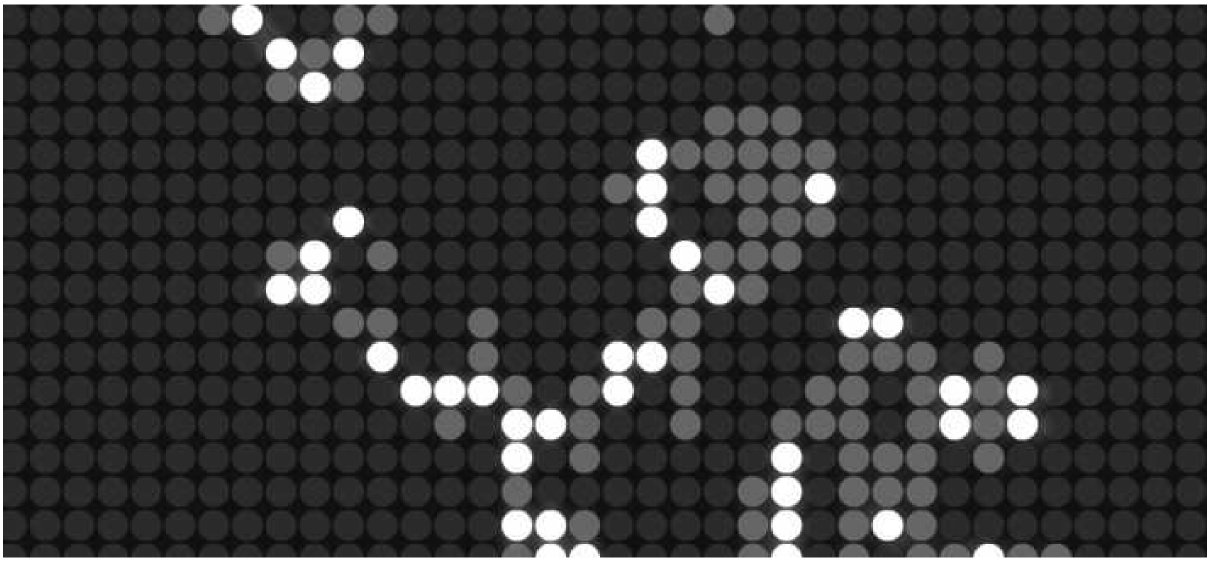

Figure 5. An Iteration in the Game of Life.

1.3.1.3 Game theoretic

Game theoretic models are models where decisions are made based on a game played by multiple parties von Neumann and Brown (1950). An analogy would be playing a game of chess. An agent will determine its decision and then reason about what the opponent would do in reaction. The same is done by the other party; assumption for both sides is that they want their own score. One could view a demonstration as a complex game where each agent wants to achieve its own goals, in this game case storm a government building and not getting hurt. The agent has to take other agents, such as control forces, into account and determine which course of action would lead him to his goal. These models can also be used to model leaders. A leader could have a certain goal and has to determine how he can influence other agents around him in such a fashion that they act in a way that the leader's goal is reached. This model however only works well in conjunction with multiple parties. When no other agents are within reach, the game theoretic model would be trivial and the agent has to adopt a different model such as beliefs desires or intentions to make its own decision. Another disadvantage is that this model is computationally expensive because of the multitude of other agents which the agent has to reason with, which increases the search space. Continuous space in which the crowd exists also provides a hampering factor because this increases the number of possible reactions.

1.3.1.4 Belief Desire Intention

Beliefs Desires Intentions (BDI) is a model described by Anand and Georgeff (1995) which works in the same way as a OODA loop as described by Boyd (1987). An agent constructs beliefs based on the environment. Desires are goals this agent wants to achieve and the agent will reason about these beliefs and desires to determine which plan is best to achieve these desires. BDI agents are able to construct plans based on the environment they see, so they can operate in an unknown environment and still achieve their internal goals. Emotions and other factors which are important can be modelled as beliefs of an agent about himself. Leadership is also a factor which can be modelled. An agent could see a leader and record this as an belief and reason about it. The same goes for subgroups. An agent can decide to follow and join a subgroup but can also decide to leave a subgroup when they want to perform some aggressive actions which the agent does not want to be part of. Adding weapons is also a possibility; the presence of certain tools can enable an agent to achieve goals he could not achieve before. A downside of BDI agents is that they are resource intensive in comparison to particle systems and cellular automata because of the reasoning aspect of an agent. Particle systems have simple rules while a BDI agent has to search for a correct plan given the situation. The expectation is that only a limited amount of BDI agents can be deployed in a crowd. Another downside is that the behaviour of a BDI agent can be difficult to explain. An agent might reason differently than expected because of its internal reasoning, goals, and actions the agent has. This poses a problem when someone wants to use a crowd with certain behaviour in mind but the agents decide to behave differently.

1.3.1.5 Behaviour Trees

Behaviour trees are a relatively new paradigm which is commonly used in videogames to describe behaviour of opponents Millington and Funge (2009). Example of games which incorporate behaviour trees are Crysis, Red Orchestra 2, Halo, and Spore. An example of a behaviour tree is given in Figure 6 could describe the basic behaviour of an enemy in a game. Behaviour trees describe behaviour as a hierarchical structure which consists of four kinds of nodes: conditions (a true/false question), actions, a sequence node, and a selector node. When a node has been called in the tree it has state, which can either be success or failure. A sequence node is successful when all its children, underlying nodes, are successful and a selector node is successful when one of his children is successful. When using these nodes complex behaviour can be described by combination and forming trees. Emotions, internal values and subgroups can be used by implementing them as condition nodes. For example a condition can be made to check if an agent is currently angry or if he sees a subgroup member around him. The same can be done for NLWs or if a leader is close by. Each kind of agent can have their own behaviour tree; a young male demonstrator will display different behaviour than an old woman, meaning that their behaviour trees would be different. When using behaviour trees it is possible to reuse predefined behaviour in new trees. This can be done by defining behaviour, for example fleeing, and use this in the behaviour trees of an old lady and a young man. Another advantage of behaviour trees is that determining behaviour means going through the tree and executing nodes. This is procedural and makes behaviour predictable, meaning that an agent will not suddenly decide to do something which was not described beforehand. This is an advantage when someone creates a tree; he knows what an agent would do in each situation. However all behaviour of an agent has to be defined in advance. Another advantage is that a tree is a simple structure and does not need a lot of computational time to traverse and execute each node. The assumption is that running a behaviour tree and determining the actions of an individual takes less time in comparison to a BDI agent who needs a lot of reasoning before decides on an action.

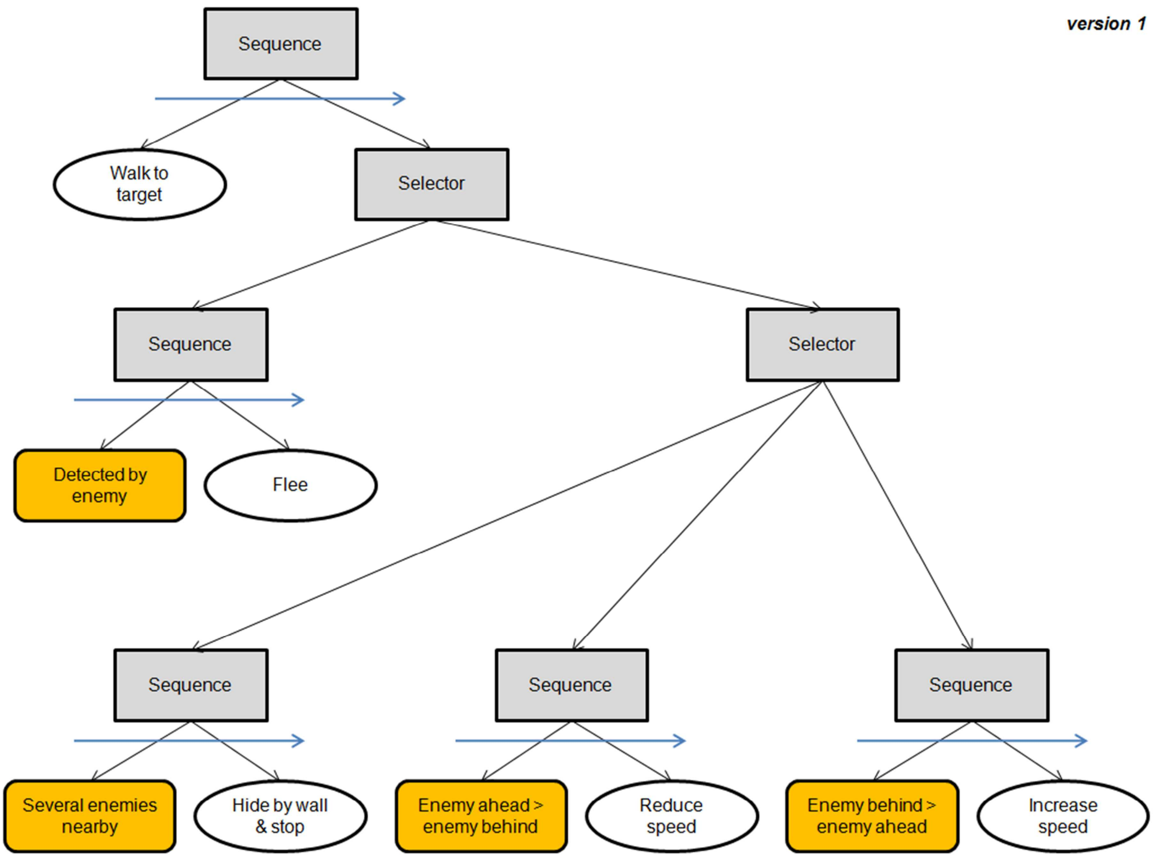

Figure 6. Example of a Behaviour Tree.

1.3.1.6 Discussion

The previously described models are used either in crowd modelling or modelling an individual which could be part of a crowd. Important note is that each model provides a trade-off between the numbers of agents versus the cognitive ability of each agent. Particle swarms and cellular automata provide the ability to use large amounts of agents but which lack cognitive capability because of simple rules which determine the behaviour. While BDI, game theoretic, and behaviour trees provide a larger cognitive ability for an agent but can support fewer agents during simulation. The choice was made to continue with behaviour trees. This model is a relatively simple and easy to adapt when required. This makes it easier for future users to change behaviour by adding or removing nodes within the tree. Behaviour trees can support aspects like NLWs, emotions and subgroups and are able to model the agents required for crowd simulation.

Use of behaviour trees for crowd simulation is not recorded as of yet. This would provide an opportunity to determine if behaviour trees would be suitable and if they are promising in this area.

Execution of a behaviour tree is done by moving through the tree and executing the nodes when needed, no explicit reasoning about plans or goals is required. This should lead to the ability to have more agents in the simulation compared to BDI agents. However the expectation is that using behaviour trees will have a lower performance when compared to particle systems or cellular automata which work with relatively simple rules.

The downside to behaviour trees is that they are inflexible by nature. All plans should be defined in advance and no creativity or flexibility can be expected from the agent when confronted with a new situation. In case of confrontation between a crowd and NLWs, the exhibited behaviour should be limited and defining behaviour trees in advance should not pose a problem. When using the crowd for training purposes having predefined behaviour can be an advantage because the user knows how a crowd would respond in a situation. Another positive characteristic of behaviour trees is reuse. Predefined behaviour can be shared amongst different agents and each agent's behaviour tree does not need to be written from the ground up. Existing pieces can be used and expanded on.

The ability of the behaviour tree to model emotions, NLWs and subgroups together with the reuse, and relatively high performance caused this paradigm to be chosen for the agent model.

1.3.2 Cognitive Agent Models

Huis in 't Veld et al. (2011) describes different cognitive models which describe how a person determines its behaviour. These five models are described in the following sections. Each subsection has a short description with the advantages, disadvantages, and the suitability for crowd simulation. The cognitive model will form the basis of the agent model, where the behaviour tree will perform the reasoning. Important aspects are emotions, other individuals and NLWs.

1.3.2.1 OODA Loop

The OODA loop described by Boyd (1987) and shown in Figure 7 is developed by John Boyd. As a pilot he came up with this model to describe his opponents behaviour and defeat him by breaking his opponents loop and trying to predict what they would do or causing them to make errors.

The model consists of four stages:

1. Observe, which means that a person is observing the world and retrieving relevant information.
2. Orient, which cause a person to filter this information based on his experience, cultural background and other aspects.
3. Decide where a person will make a decision based on the filtered information from the previous step. The decision is called the hypothesis and is the behaviour the person has determined to use.
4. Act step, when all information has been processed and the decision is made the decision has to be executed. This will influence the world which forms as the input for the Observe step and closes the loop.

This loop describes how an individual would make a logical choice based on all information in the world and enough time to filter and evaluate. In some cases it is possible for a person to go straight from observe to decide. This is done by using special heuristics in certain situations; it could be experience with tear gas which causes the person to run away.

In some cases the person might go straight from observe to act. As soon as a person sees something, like a house fire or is confronted with a gun, he will act without reasoning about the situation and coming to a conclusion. This can be seen as stimulus-response where a person would immediately respond to a certain situation while bypassing the cognitive decision making.

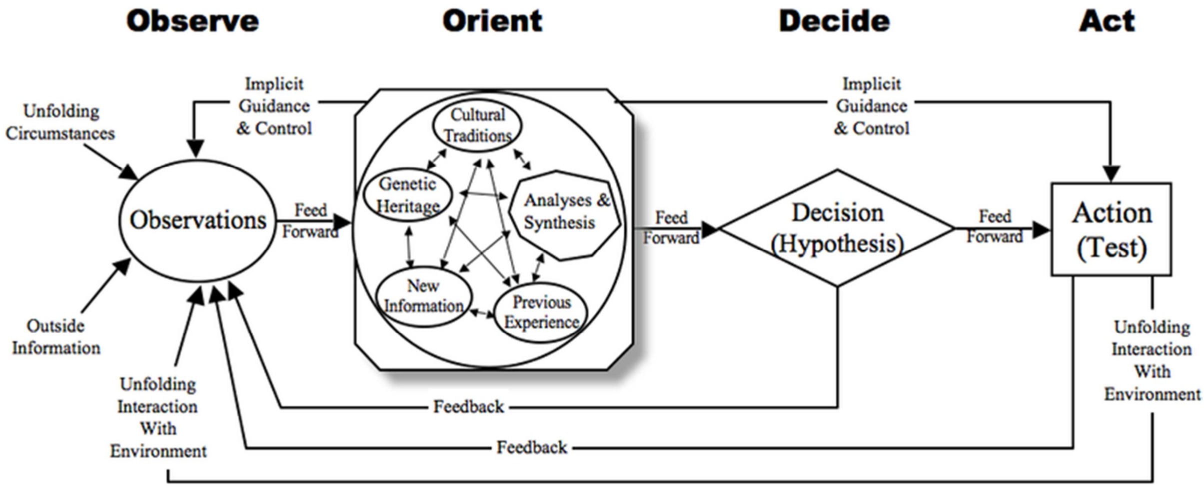

Figure 7. OODA Loop.

This model provides capability to observe the world and reason with this information. This is useful because external information such as NLWs and control forces can be treated as observations. An agent would then be abled reason about them and determine its own action. This model however does not have an explicit place for emotions or interactions between people. This could be part of the observations but is not stated as such. This model however provides the basics which are needed but was not chosen because of the lack of support for emotions and interactions with other individuals.

1.3.2.2 Situation Awareness

The situation awareness model described by Endsley (1995) is given in Figure 8. The same kind of steps as in the OODA loop can be found. Main difference between the two models is that the observe step and orient step from the OODA loop are combined into one situation awareness step. This situation awareness step has multiple levels which contribute to a more complex understanding of the world. At the first level only the world is perceived, such as water falling from the sky while at the second level the world is analysed and possible conclusions are made. These conclusions are based on the current situation the person is in, such as it is currently raining. The third level depicts the person reasoning about the possible future state of the world, for example if he would walk in the rain than he would become wet. These levels combined form input for the decision phase where a person will make a decision in this phase based on previous experience, heuristics, and carry out the decision. In the rain example the person might reason that he will get wet so he probably should use an umbrella. The impact of the actions is observed and becomes part of the environment.

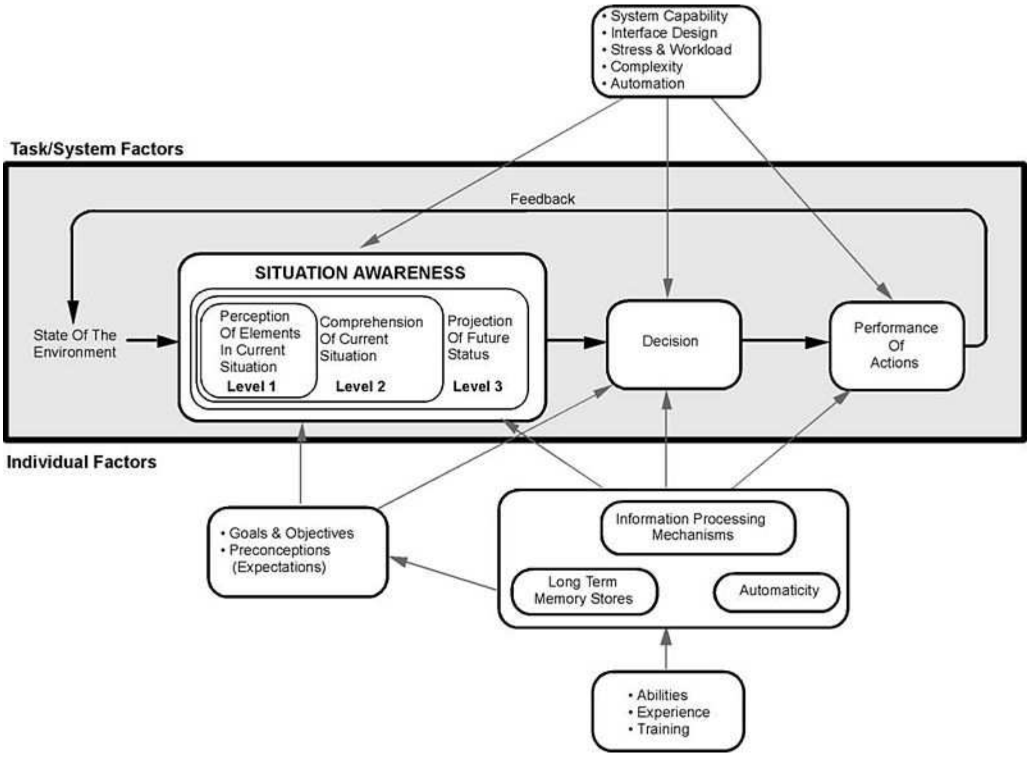

Figure 8. Situation Awareness model.

This model provides the capability for an agent to reason about the future world given the current state. This is a nice addition in comparison with the OODA loop because the agent could reason that a NLW which is deployed a couple of meters beside him might form a threat if he continues to stand there. However just like the OODA loop there seems to be no place for emotions and interaction with other people. However there is room for automaticity, meaning that stimulus-response situations can take place within this model which is commonly seen in panic situations, especially when NLWs are involved. However situation awareness was not chosen because of the lack of interaction with other individuals and the lack of support for emotions.

1.3.2.3 Risk-as-feelings

The risk-as-feelings model is described by Loewenstein et al. (2001) and shown in Figure 8. The basis for this model is that people will make different decisions in high risk situations in comparison to low risk situations. When there is a high risk for a person he does not have time to analyse the situation and reason about all possible responses, he would base his responses on his current emotion. For example intense fear during an evacuation causes a person to try and run while a better solution would be walking as described by Helbing et al. (2002). Risk-as-feelings model provides two ways which influence the outcome. One is through cognitive evaluation, where the world is evaluated without using emotions but only logic. The other way is based on emotions of an individual. These two ways influence each other and the final decision. In a high risk situation the individual would listen more to his emotions while in a calm situation the individual will make a more rational decision as described by the subjective probabilities.

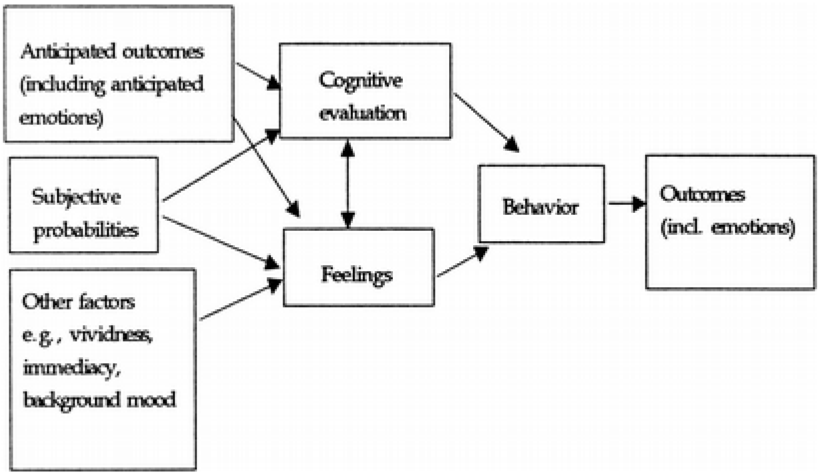

Figure 9. Risks-as-Feelings model.

The risks-as-feelings model provides an agent with the capability to reason with emotions and rationality and switch between them given the situation. However this model seems to be only using internal states of an agent and does not have a place for the external world. Having a NLW deployed in the vicinity cannot seem to be processed within this model. Also interactions with other agents or persons are not part of this model. Because the model lacks these aspects it was not selected.

1.3.2.4 Conceptual model of an individual

The conceptual model of an individual is described by Frini et al. (2008) and is shown in Figure 10. It describes how an individual during a demonstration makes decisions and has a similar kind of loop as the OODA loop. The world is observed in the perception stage which influences various internal states of a person, such as his emotional state or physical state. Difference with the previously mentioned models is that the decision is not for a specific plan of action but is a choice for a social identity. This social identity depict to which social group a person belongs and determines what behaviour he can exhibit. Because this model is described with demonstrators and control forces with NLWs in mind there are lines going from behaviour of other crowd members and control force actions to the perception phase. This way they influence the internal states of a person and influence his behaviour.

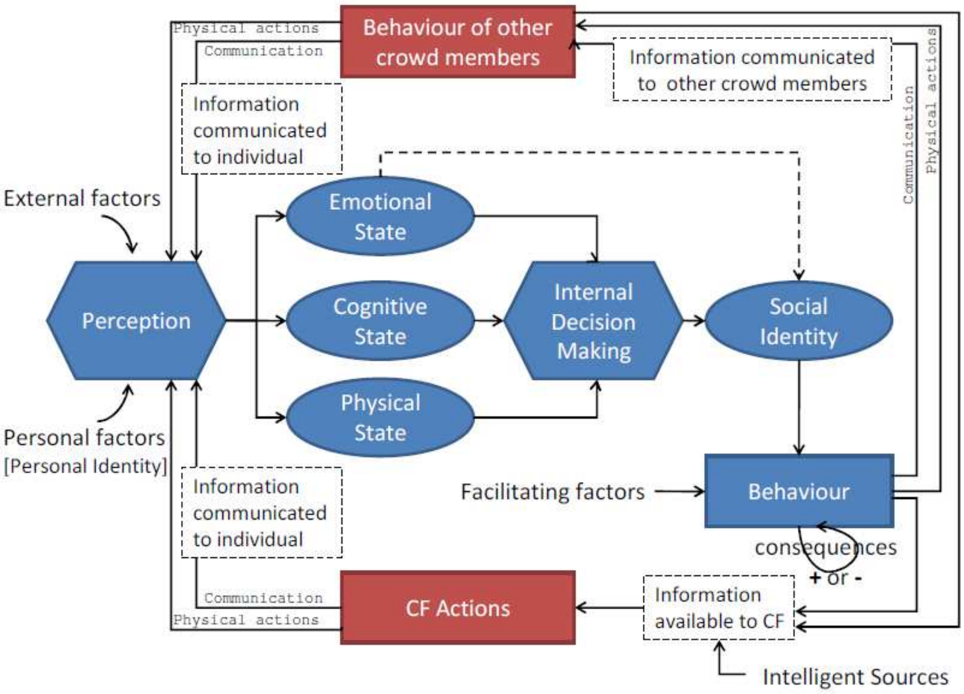

Figure 10. Conceptual Model of an Individual.

This model provides capability to reason with emotions described by an emotional state. This state is updated based on perception of both external factors and internal factors of a person. Emotions form a basis, besides the cognitive and physical state, for the person in the model to come to a decision for behaviour.
The down side of this model is that the internal decision making step leads to a social identity. This social identity determines which subset of behaviour the agent can choose from. This limits the choices a person might make and makes it seem like the group, to which this agent belongs, determines his actions, which is in contrast to Couch (1968) and Kenny et al. (2001). However this model does explicitly provide the ability to interact with control forces and other crowd members. Social identity which dictates the behaviour caused this model to be rejected although it provides aspects which the other models did not provide such as emotions and interaction with other agents.

1.3.2.5 BIN Model

BIN model stands for Behaviour of Individuals confronted with NLWs and is described by Huis in 't Veld et al. (2011). This model is influenced by the previously mentioned models. The world is perceived and appraised which updates the internal states of a person and forms the basis for internal decision making. When a decision is made the behaviour of a person is chosen and this decision is applied to the world by an action of the person. This changed world provides input for the perception and appraisal as in the other models.
Other crowd members and control force members are part of this model and either influence perception and appraisal or directly the states of an agent. NLWs are modelled within the BIN model as the control force's equipment, which influences the actions of the control forces and the perception of a person. There is no explicit mention of subgroups and leaders; the assumption is made that this falls under the observation of other crowd members. However there appear to be a lot loops inside this model. For example one can go from perception and appraisal to emotional state and back. Or there is an influences connection between the emotional and physical state which also receives input from the perception and appraisal step. One could say that a person can perceive his physical state in perception and appraisal which on its turn will influence the emotional state. This causes to all the information to enter the same place within the model but the effect on the final behaviour would become indirect.

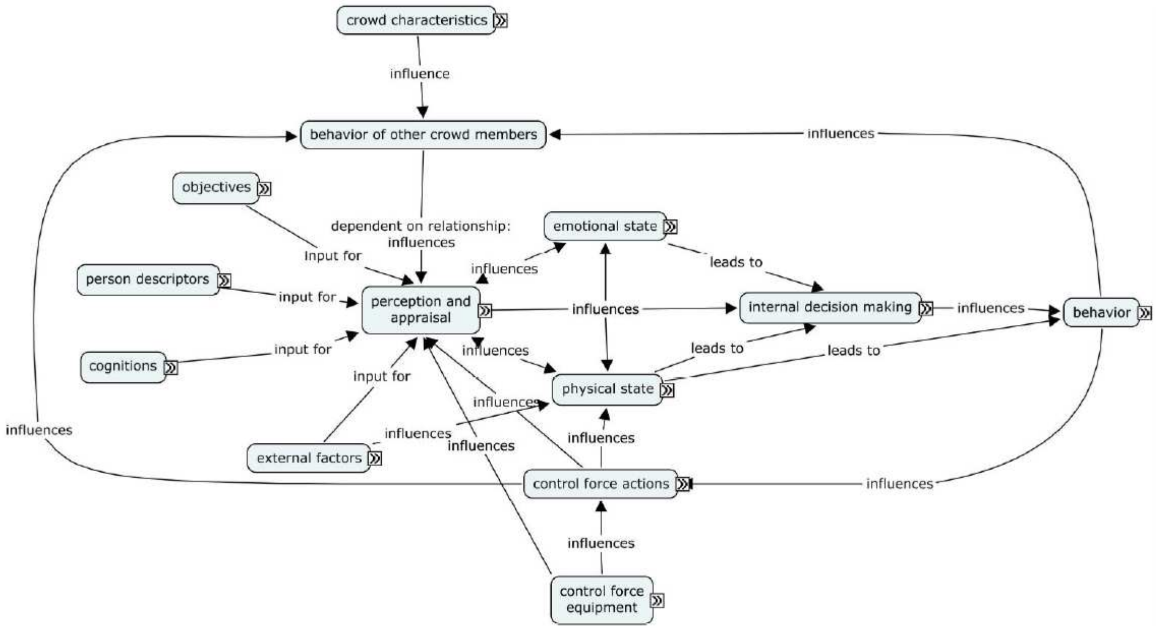

Figure 11. Behaviour of Individuals confronted with NLWs Model.

The BIN model is closely related to the conceptual model as mentioned in Section 1.3.2.4. This model provides both internal and external factors which influence the internal states of an agent and its internal decision making. This model also has an explicit place for the interaction and influence of control force members and other crowd members. However as stated before there seem to be redundant lines between certain components. Never the less this model will form the basis for the agent model with some redundant lines removed.

1.3.2.6 Discussion

The most important aspects which should be part of the final model will be the emotions of the agent and the interaction between crowd members, control forces, and NLWs. Most of the models above do not explicitly have these aspects or focus on one in particular. The choice was made to use the BIN model as a basis for the agent model because this models the behaviour during a demonstration with NLWs and control forces. However this model will be streamlined and some lines are move to through perception and appraisal instead of influencing states directly. The final version of the adapted BIN model can be seen in Figure 12.

Chapter 2 Design of the Proposed Agent Model

This chapter describes the design of the agent system which flows from the literature study of the behaviour of crowd members and the literature study about the suitable models and AI paradigms which exist.
This is chapter is part of the research question 'How can a crowd of demonstrators be modelled using decentralized techniques?' and is part of step 3 in the roadmap. Two crowds, decentralized and centralized, will be described in combination with how they reason about the world, how this world influences their states and what the resulting behaviour is.

2.1 Cognitive Model of the Agents

The chosen cognitive model in Figure 12 is similar to the BIN model in Figure 11 but has less influence lines going to different components. These lines now go through the world appraisal which translates the influence of external factors to an agent's internal state. The spheres on the left side of the figure form inputs for an agent and the sphere on the right side is the output or behaviour of an agent. An agent will perceive the world around him, which consists of crowd members, control forces, subgroups and possible NLWs and their respective internal states and positions in the world. Perceiving this information causes the agent to change its internal values. How this is done is described in Section 2.3. These values are changed with formulas for each of the external factors. An overview of these is given in Section 2.4. The world has an effect on internal states of an agent such as his resolve and arousal while other factors influence his personal risk, the exact internal variables can be found in Section 2.2.1. These internal states define the emotional state of an agent and are determined by the emotional updater. A complete overview of these emotional states is given in Section 2.2.2. Based on this emotional state a decision is made by using the behaviour tree of the agent, this behaviour is described in Section 2.5. For example if an agent is calm he will try to move towards visible subgroup members but avoid control forces members. The action an agent takes influences the world around him, which influences the world. Having crowd members and control forces in the world is similar to the conceptual and BIN model but there is no notion of a social identity such as in the conceptual model. The resulting cognitive model is used to represent each individual agent.

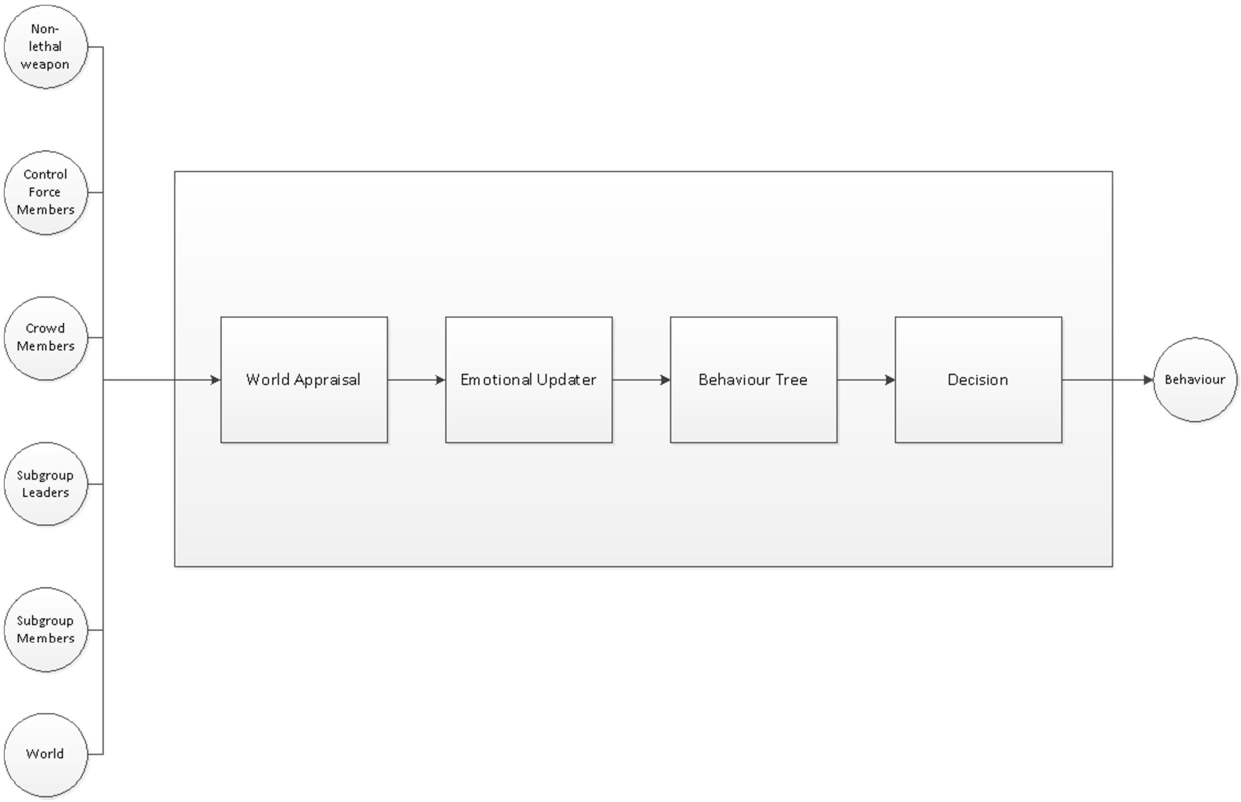

Figure 12. Overview of the agent's cognitive model.

2.2 Internal Composition of the Agents

This Section describes what is contained inside an agent for crowd simulation. The first Section describes the internal variables of an agent will be described followed by his emotional states. The relation between these two and what behaviour follows will be described by an internal model.

2.2.1 Internal Variables

Interval variables of an agent describe certain psychological and physical aspects which can change over time given the situation. There are several which are deemed important for a person during a demonstration. These are described by Huis in 't Veld et al. (2011) and form the basis for the chosen variables for an agent. One of these variables is resolve. Resolve is based on motivation which is described by Huis in 't Veld et al. (2011). However motivation was not used because this would describes how motivated somebody is to do a certain action. This could be partaking in the demonstration but also running away in fear. A better descriptor for motivation had to be found where a low value describes somebody who does not want to continue with the demonstration and a high value would mean a highly motivated individual. In the end resolve was chosen which describes how determined an agent is to continue his part in the demonstration. If the resolve is low the person would not want to continue with the demonstration, while having a high resolve would mean the opposite.

Pain is another important aspect, especially when an agent is confronted with NLWs.

Another variable which was chosen was arousal. Arousal means how much energy the person has and how agitated he is. If this value is low a person would be very calm and docile while having a high level would mean that a person is quite aggressive and wants to initiate action.

Other factors like ethnicity, social economic status and political preference are not taken as internal variables because their impact on the escalation of a demonstration is not verified as stated by Dobias et al. (2006) and they do not change over time. Another aspect called tiredness could have been modelled but a demonstration will only be held in a relatively short time period, so the expected influence of tiredness during a demonstration would be relatively small and was not considered to implement at this moment.

Initially the aspect stress was also selected but was later dropped because an agent also has a notion of personal risk. Personal risk describes how much risk the agent currently believes it faces, if it is high an agent will most likely get hurt, while a low risk means that the situation is completely safe. In the design phase the aspect of stress and personal risk were closely related meaning that in a violent situation personal risk would increase but so would stress. In a calm situation both would decline. The concept of stress is also vague because specific behaviour inside the crowd could not be explained by the stress level and it was difficult to tie specific emotions to it. Therefor stress was not chosen to be part of this agent model.

The chosen variables are depicted in Table 1. These are described using a decimal value and have a minimum value of 0 and a maximum value of 100.

Using levels for defining psychological factors of an individual is also used by Kramer and Rathmann.

Table 1. Chosen internal variables of an agent.

Name
Arousal
Personal Risk
Resolve
Pain

2.2.2 Emotional States

This Section describes the various emotional states an agent will have and what transitions there are between these emotions. Real persons have numerous emotions which influence their behaviour, especially during a demonstration. One could be angry, happy, or even sad and afraid. This is not only the case during normal everyday life but also during a demonstration. To find suitable emotions a link between observed behaviour by Frini et al. (2008) and Huis in 't Veld et al. (2011) to emotions was made. Recorded observed behaviour is described in Table 2.

Table 2. Found behaviour in crowds divided into categories.

Category	Behaviour
Calm	Neutral, Curious Non Aggressive Posture
Anger	Agressive posture, agressive non-lethal actions, violent lethal actions
Fear	Avoidance

Described behaviours are neutral posture, curious non aggressive posture, aggressive posture, aggressive non-lethal actions, violent lethal actions, and avoidance. Neutral and curious non aggressive posture are linked to calm because they pose no aggression to any party and are typically seen by calm crowd members when nothing violent is happening. Aggressive posture, aggressive non-lethal actions, and violent lethal actions have been linked to anger because they describe aggressive to very aggressive behaviour and are often initiated by crowd members who are angry. Often they attack control force members by throwing rocks at them or even Molotov cocktails and try to physically attack them. The final behaviour avoidance was linked to fear because this describes what fearful people in a demonstration typically do. They try to avoid other people to stay safe and avoid confrontation which might injure them.

There is another category added to this list which is called panic. This category describes a more advanced stage of fear where people are not able to reason and typically avoid everything and want to get as far away from the situation as possible because of the threat they experience. These categories have been mapped to emotions and the emotions anger and fear are also stated by Wetzer et al. (2010). Other emotional states which could have been chosen were: relief, happiness, hostility, excitement, and frustration. These emotions are contained in the previously mentioned emotions such as hostility, frustration and excitement which are combined into anger or happiness and relief which are described by the emotion calm. The resulting emotional states are described in Table 3.

Table 3. Selected emotional states.

Emotional States
Calm
Anger
Fear
Panic

Transitions between emotions are based on experiments by Wetzer et al. (2010). This report describes that demonstrators were initially calm but went to an angry state when they were confronted with control forces. When batons were used the crowd reeled back in fear but reassembled in a calm fashion and then tried to confront the present control forces again. This described behaviour lead to the decision to have an agent switch between emotions based on these internal states.

Transitions between emotions are based on personal risk, arousal, and resolve. If personal risk gets too large an agent might feel vulnerable and might go into fear and try to keep away from the source of the threat. If

personal risk is small an agent might try to stand up against the threat and transition to an anger state. The transition from anger to fear takes more perceived personal risk and a lower resolve in comparison to the transition between calm and fear. The reason being that an angry person is not easily scared in comparison to a calm person, therefore more extreme measures are needed to accomplish this. When an agent is in a fear state he can transition to a calm state. This happens when the threat to an agent is removed and he is sufficiently calmed down. The transition from fear to panic is a one way transition, focus of a panicking person is to leave the immediate area and get away as far as possible. This means that an panicked agent does not want to continue with the demonstration and will flee until safe. An agent might also go from an anger state to a calm state, which happens when an agent realizes that the personal risk is becoming too large and that it better to be calm than to be angry because of the increased risk.
Problem with these transitions is that all the agents have the same underlying system. In the exact same situation all the agents could switch to a new emotion on the same time. To prevent this thresholds were introduced. Each value which contributes to a transition is multiplied by a random threshold value set for each agent. This makes each agent unique while still following the same kind of transitions and emotional states. The chosen emotions and transitions are shown in Figure 13.

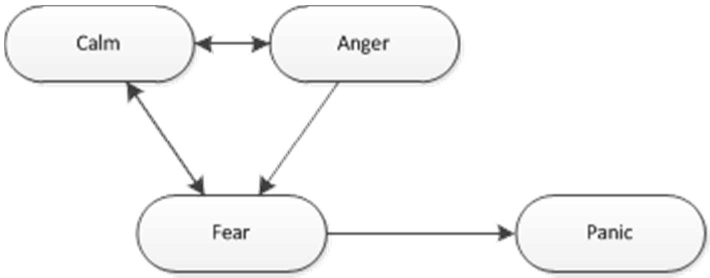

Figure 13. Transitions between emotional states.

As said before transitions between emotional states are based on the internal variables of an agent. The exact transitions with the corresponding values for internal variables are shown in Table 4. When an agent is calm but is confronted with a very risky situation, his personal risk will increase. If the personal risk of an agent is high and resolve is low than this agent will transition from calm to fear. So when a NLW is deployed, affected agents change their emotional state to fear and some will even panic. When an agent is angry the impact on him has to be larger to make him transition to fear, his personal risk has to be above 75 and his resolve has to be lower than 40. The higher values are because an angry person is focussed on his anger or rage and takes more effort to scare.
To make an calm agent angry his personal risk has to be lower than 50, because the agent has to see that there is little risk to being angry, and his arousal has to be above 50, arousal meaning how energetic or aggressive somebody feels. If both conditions are met the agent will transition from calm to anger. An agent will transition back to calm from anger when his personal risk is large, above 60, and his arousal has dropped below 60. Meaning that the agent is calming down and the risk is becoming too large. Finally an agent can be in fear but calm down in certain situations, when his personal risk has dropped below 50. This reduced risk means that an agent does not perceive as much threat as before and that he is safe. However the situation might also become risky again, when his resolve becomes low the agent will transition in to panic and will want to stay away from the demonstration. Values for these transitions are based on observing a simulated crowd and determining if the agents would transition to the correct emotions given the situation they were in. When this happened the values were recorded and used.

Table 4. Thresholds between emotions

Initial	Final	Description
Calm	Fear	personalRisk > (thresholdCalmFear * 50) and resolve < (thresholdCalmFear * 45)
Anger	Fear	personalRisk > (thresholdAngerFear * 75) and resolve < (thresholdAngerFear * 40)
Calm	Anger	personalRisk < (thresholdCalmAnger * 50) and arousal > (thresholdCalmAnger * 50).
Fear	Panic	personalRisk > (thresholdFearPanic * 80) and resolve < (thresholdFearPanic * 25)
Fear	Calm	personalRisk < (thresholdFearCalm * 50).
Anger	Calm	personalRisk > (thresholdAngerCalm * 60) and arousal < (thresholdAngerCalm * 60)

2.3 Internal Model of the Agents

This Section will describe the internal model of an agent and describes how the external, internal, emotional states and behaviour trees are linked to each other in an agent.

2.3.1 Internal Model of the Crowd Member

The internal model of the crowd member displays what kinds of inputs there are for the agent, how this influences the agent, and what behaviour follows. Figure 14 shows the model of a regular crowd member. On the right of the figure there are the various inputs from the outside world which an agent observes. These influence the internal variables of the agent such as arousal, personal risk and resolve. As stated in the Section about the emotional states, these variables determine what the emotional state of the agent will be, as shown by the arrows going from the variables to the emotional states. As described in Section 2.2.1 arousal determines if the agent will either be calm or angry and his resolve will determine if this agent will be either fearful or panic. Besides these two variables, personal risk is also a major contributor to the choice of emotional state. Connections between internal variables and emotions are described by the links in the figure. Precise transitions between the emotional states are described in Section 2.2.2. The left of model shows the actions an agent can perform. Lines between actions and emotional states describe what actions will occur in that emotional state. How the agent determines its actions is described in Section 2.5. Agents can observe certain attributes of other agents, such as agents themselves, their emotional state and their pain. In real demonstrations these attributes can be observed through non-verbal communication. For example when somebody is sweating and looking hastily around him, than he probably is in fear or panic. Another observable attribute of an agent is the subgroup to which he belongs, this way agents can recognize if another agent belongs to the same subgroup or not, which is important for the behaviour of an agent.

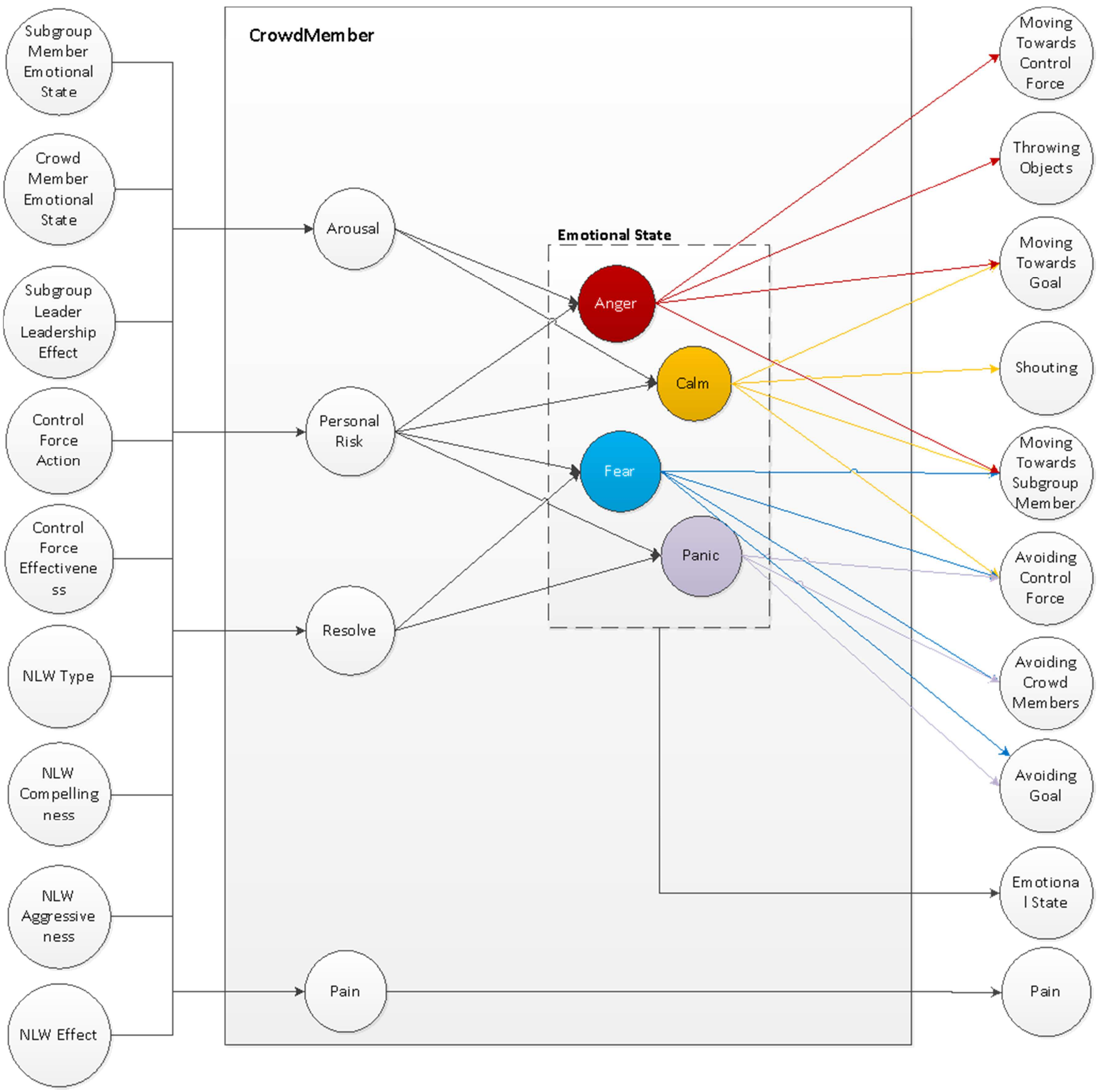

Figure 14. Internal model of a crowd member.

2.3.2 Internal Model of the Subgroup Leader

Besides regular crowd members a crowd also consists of leaders, which also has input from the world and observable attributes for other agents. The internal model for a leader is given in Figure 15. A leader shares the same internal variables of the crowd member because a leader can be seen as a specialized version of a crowd member. Internal variables which are shared between a crowd member and leader have not been described in the leader's internal mode. Difference between leaders and crowd members is that a leader has a leadership effect. This effect describes if a leader is calming or inciting, which can be seen as a state. These states also determine the general behaviour of a leader during the demonstration. If he is calming, he will try to avoid control force members and move towards subgroup members and the goal. But if he is inciting he will attack control force members if they are visible. Leadership effect also describes how effective a leader is on influencing crowd members around him. It ranges from -1 to 1 where 0 to -1 indicates inciting leaders and 0 to 1 calming leaders. The value between 0 and -1 or 0 to 1 depicts how effective a leader is. For example an inciting leader with a value of -1 is very effective, but if he had a value of 0 than he is ineffective and does not have any influence on other agents.

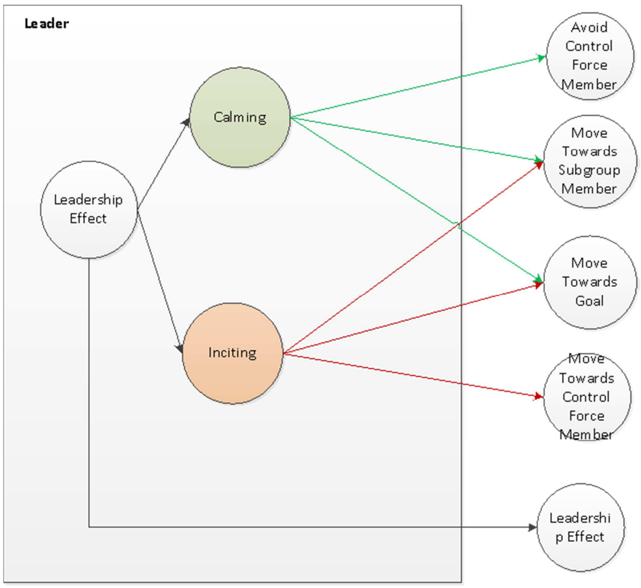

Figure 15. Internal model of a leader.

2.3.3 Internal Model of the Non-Lethal Weapon

Modelling a NLW as an agent might seem strange because it does not fit the description of an intelligent agent, it a static object in the world which does not exhibit reactive behaviour. However it is an object which influences other agents and reacts to changes in time, like releasing smoke after a second when deployed. The internal model of a NLW is given in Figure 16. It describes the internal variables, such as compellingness, aggressiveness, type, effective range, onset of the NLWs effect, and duration and show which of these are observable. Compellingness describes how compelled agents are when affected by a NLW. If this is high an agent will immediately respond, while a low compellingness means that he will ignore or partially ignore the effect. Another factor which is important for a NLW is aggressiveness. This describes how aggressive a NLW looks in the eyes of an agent. If a NLW looks very similar to a regular firearm the aggressiveness will be high, but if it looks like a toy water gun the aggressiveness will be low and agents would not perceive a lethal risk. As stated before a NLW reacts to changes in time. This can be seen in the onset of the effect and duration which relate to the time spend by a NLW in the world. Onset means how much time it takes after deployment for the effect to appear, for example forming an effective tear gas cloud. The duration describes how long the effect would last. These time-related attributes are important for the NLW itself but cannot be seen by other agents. However the other variables can be observed by other agents. The mentioned attributes are all described as being important for a NLW by Wetzer and Horst (2011).

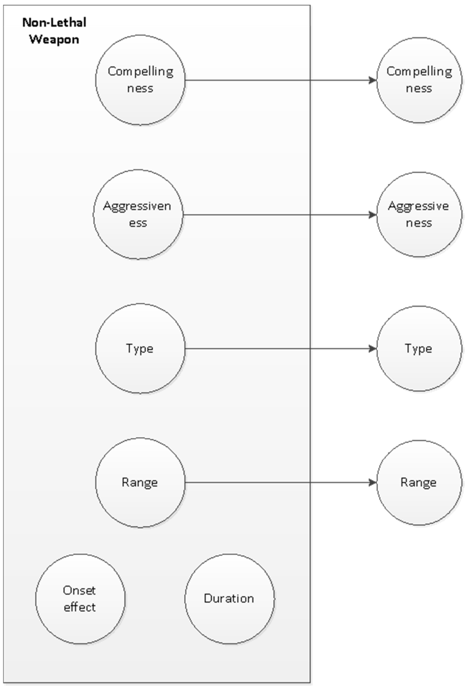

Figure 16. Internal model of a NLW.

2.3.4 Internal Model of the Control Force Member

In addition to crowd members, leaders, and NLWs there are also control forces in a demonstration. These will also be described as an agent and its internal model is shown in Figure 17. Control force members are agents which will not move in the world and will not adapt its behaviour in respond to the other agents in the world. This is not how the control forces in real life would interact with the demonstrators but modelling the control forces with movement and reasoning was not within the scope of this project. If the crowd would be used as a training tool, control forces would either be controlled by humans and in case of mission planning controlled by an agent model. A control force member can adopt two kinds of stances, warning and the show of force, as described in Section 1.1.1. These stances can be observed by other agents and will have an effect on their behaviour. Besides stances a control force member also has an internal variable called effectiveness. This describes how effective the control force member is in controlling the situation, and describes the capability of a control force member Kenny et al. (2001). This variable ranges from 0 to 1 and if this value is low a control force member is ineffective and will have little impact on present crowd members. However if this value is high the effect will be larger.

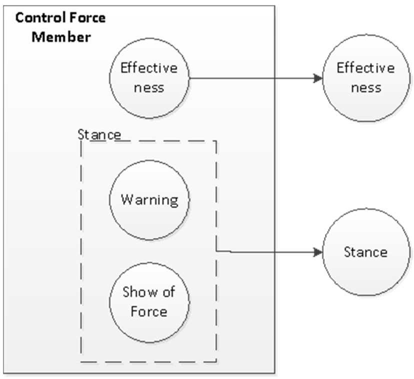

Figure 17. Internal model of a control force member.

2.4 Formulas Used To Change the Internal Variables

Interactions described in Figure 1 influence the internal variables of agents as seen in Section 2.3. The following formulas show what the exact impact is on the internal variables in different situations. Basis for all formulas is a linear equation where the previous value is changed with a certain factor, which can be any number either positive or negative. Most formulas have a normalizing factor present in the form of (1-distance/view range). This normalizing factor is meant to illustrate that the effect of an interaction diminishes when the distance increases, distance in this case described in meters. If an agent is close to the source, the effect is large but if he is on the edge the effect will be small. The following sections show the different formulas used in different interactions, a full overview with all factors filled in can be seen in Appendix A. Besides the interactions and their impact, there are also values which degrade over time. These are personal risk, which degrades with 0.05 per second, and arousal, which degrades with 0.01 per second. These are used because; a normal person cannot maintain an angry mind-set for a long time or be frightened indefinitely when something has occurred. After some time period they return to their normal state of being. The precise factors in Appendix A were chosen on how the crowd as a whole behaved. If the behaviour was off, factors were changed until the behaviour was suitable for the purpose of TNO.

2.4.1 Interaction between Crowd Member and Crowd Member

During a demonstration crowd members will interact with each other. These interactions, which are mostly through observation, have an impact on the internal variables of an agent. This impact is described by the following formulas:

$$personalRisk = oldPersonalRisk + FACTOR * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right),$$

$$resolve = oldResolve + FACTOR * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right),$$

$$arousal = oldArousal + FACTOR * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right).$$

There are three variables which are changed, personal risk, resolve, and arousal. These variables all range between 0 and 100. Meaning when an agent's personal risk is 0 he experiences no risk but if it is 100 he experiences a larger risk on his health and has a high chance of becoming injured. When the agent's resolve is 0 he is unable to continue because he lacks mental strength, if it is 100 he is very motivated and wants to continue with the demonstration. In case of arousal, 0 means that an agent is calm, docile, and lacks energy while an arousal of 100 means that is aroused, energetic, and ready or even eager for aggression.

In the formula there is a viewRange of an agent which can be set between 0 and 20 and represents how far an agent can see. DistanceToCrowdMember is a variable used for crowd members who the agent can see, and describes the distance between the current agent and the observed crowd memberThis means that the maximum value for distanceToCrowdMember will be the agent's viewRange.

The factor in each formula is determined by the situation, for example when a calm agent sees an angry agent different factors are used than the reversed situation. There are various situations which call for different factors. All the variations and their respective factors are given in in Appendix A.

2.4.2 Interaction between Crowd Member and Subgroup Member

Besides interactions between crowd members there is also interaction between subgroup members. Such as between subgroup members who are angry and calm. They influence the resolve and arousal as depicted in the following formulas. These formulas do not influence the personal risk of an agent because he is among friends or likeminded people and does do not feel threatened by it. This is described in the following formulas:

$$resolve = oldResolve + FACTOR * \left(1 - \frac{distanceToSubgroupMember}{viewRange}\right),$$

$$arousal = oldArousal + FACTOR * \left(1 - \frac{distanceToSubgroupMember}{viewRange}\right).$$

Again a normalized distance is used which means that the effect is smaller when the two members are far from each other. ViewRange is between 0 and 20 and distanceToSubgroupMember will be between 0 and the viewRange, if this value is 0 than it means that the subgroup member is at the same location as the agent, if this value is the same as the viewRange than the subgroup member is at the end of the agent's visible range.

2.4.3 Interaction between Crowd Member and Leader

Besides interactions between crowd members and their subgroup members there are also interactions with subgroup leaders. A leader can either have a calming influence or an inciting influence as stated in Section 2.3.2. Formulas for interactions with leaders are given below:

$$personalRisk = oldPersonalRisk + FACTOR * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right),$$

$$arousal = oldArousal + FACTOR * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right),$$

$$resolve = oldResolve + FACTOR * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right).$$

New values for personalRisk, arousal, and resolve are based on old value multiplied with a certain factor, which varies given the situation. Personal risk, arousal, and resolve have the same range, between 0 and 100. Leadership effect determines how effective the leader is and if he is inciting or calming. This value can be between -1 (inciting) and 1 (calming). In the formulas the absolute value is taken without the sign because the increase or decrease of the value is determined by the type of leader and corresponding factor. The leadership effect should depict how effective a leader is, so an range of 0 to 1 is suitable to describe the effectiveness. DistanceToLeader is similar to earlier mentioned distance variables and is between 0 and the viewRange of the current agent, 0 meaning that a leader is at the same location and if this value is the same as the viewRange that the leader is at the edge of the agent's visible range.

2.4.4 Interaction between Crowd Member and Non-Lethal Weapons

One of the most important interactions during a demonstration is where crowd members interact with a NLW. When a crowd member sees the deployment of a NLW for the first time without being affected the following formulas will be used:

$$personalRisk = oldPersonalRisk + FACTOR * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$arousal = oldArousal + FACTOR * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right).$$

The compellingness and aggressiveness factor are described by Wetzer and Horst (2011) and describe characteristics of the NLW as mentioned in Section 2.3.3. These variables have a minimum of 0 and a maximum of 1. Values for compellingness, aggressiveness, and effective range determined for each NLW are based on the report Wetzer and Horst (2011).
Besides viewing a NLW, an agent can also be affected by it. As long as a crowd member is in range of a NLW he will be affected on his personal risk, resolve, pain, and arousal. There is also a effect which triggers when a crowd member enters the NLW's effect for the first time. This can be viewed as shock of experiencing this NLW. In both cases the following set of formulas is used, the difference between experiencing the NLW for the

first time and continued exposure is the difference in factors, which can be viewed in Appendix A. This is also described in the following formulas:

$$personalRisk = oldPersonalRisk + FACTOR * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$resolve = oldResolve + FACTOR * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$pain = oldPain + FACTOR * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$arousal = oldArousal - FACTOR * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right).$$

As seen in previous formulas the old values are used with an applied factor to determine the new value. This factor is multiplied by the combined compellingness and aggressiveness and a normalized distance to the NLW. DistanceToNLW value has a minimum of 0 and a maximum value equal to nlwEffectRange. An NLW effect's range describes for example how far a teargas cloud would spread in a given environment. This can be a value between 0 and the maximum range of this type of NLW in the real world and is defined in meters.

2.4.5 Interaction between Crowd Member and Control Forces

Besides the interactions between crowd members, leaders, and NLWs there is also interaction with present control forces. Control forces have two stances they can employ as stated in Section 1.1. These are show of force and warning. Each stance of a control force member will have a different effect on a crowd member based on their emotional state. The effectiveness denotes how effective the control force member is; if this value is 1 the control force member is disciplined and capable, while a value of 0 means that he is incapable and probably acts chaotically. In some formulas a Section (1 – effectiveness) is used. This means that the effect of this formula is large if the control force is ineffective, but is small if they are effective. There is also a difference between seeing the action once or continuous. This is described in the following formulas:

$$personalRisk = oldPersonalRisk + FACTOR * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$arousal = oldArousal + FACTOR * (1 - effectiveness) * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve + FACTOR * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

As with previously mentioned formulas a normalized distance measure is used. DistanceToControlForce has a value between 0 and the viewRange of the agent.

2.4.6 Verification of Formulas

To verify that the chosen factors in the formulas are suitable a graph was constructed which shows how many agents are in a certain emotional state over time, time in this case means the number of steps executed with MASON. The resulting graph can be viewed in Figure 18. Emotions plotted are angry, calm, fear, and panic. Most of the agents in the crowd start in the calm state but a few agents start in the anger state because of random initialized internal variables. Besides normal crowd members there are also subgroup members present with their respective leaders. Their influence can be found in the decrease of calm agents and the slight increase of agents who become angry. Over time some agents go from calm to fear because of neighbours who are angry or control force members who are close. This makes the agents anxious and

increases their personal risk and therefore they transition to fear.
As expected when a NLW is deployed at time point 8750 as depicted by the dashed line, most of the agents go from calm to fear or even panic as expected. At around time point 16000 another NLW, depicted as a dashed line, hits a subgroup with an inciting leader. The number of angry agents drops and the number of calm agents increases because of their transition from anger to calm. After a short time this is counteracted by the inciting leader and agent's transition back into anger, as expected with an inciting leader.

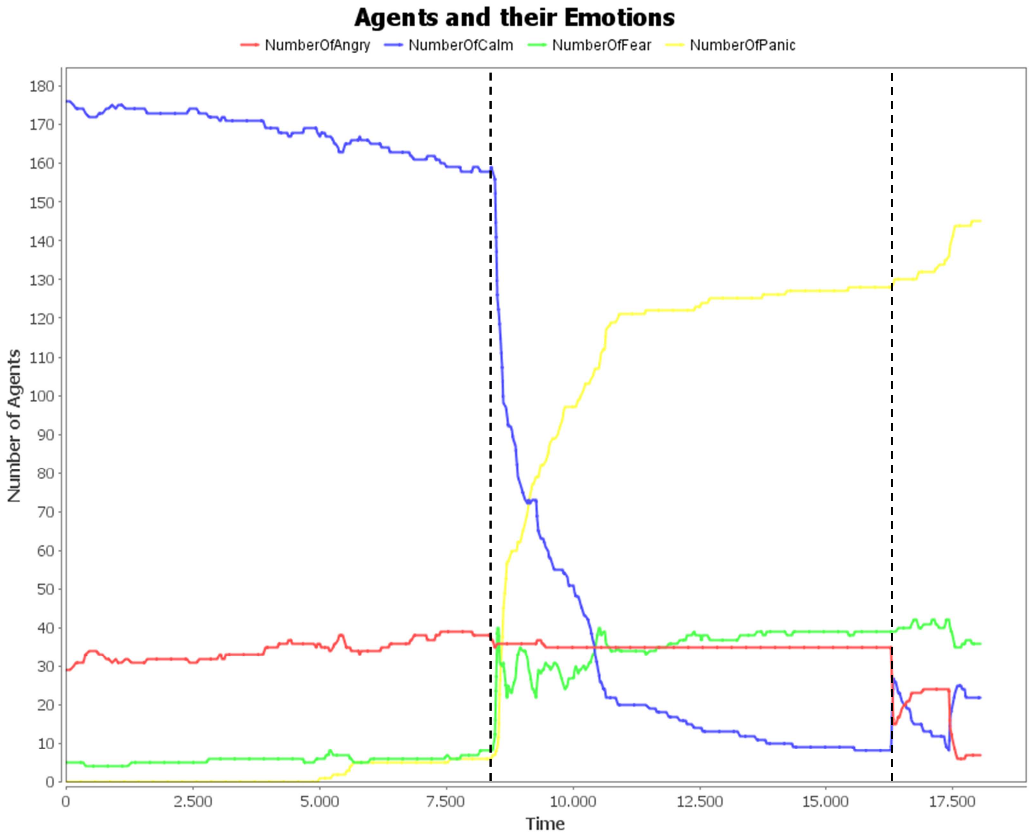

Figure 18. Agents and their emotional state over time.

Figure 18 also shows that there is a gradual transition between emotions when no major events happen. When a NLW hits the agents their emotional state changes according to the situation and they transition between calm, fear, anger and panic as expected. The effect of an inciting leader can also be seen in the graph, as indicated at around time point 16000, at which a second NLW hits the subgroup and agents transition from anger to calm, but because of the influence of the leader they switch back. There is also no major fluctuation between emotions. There is not a large group of agents who keeps transitioning between anger and calm or fear and calm. This indicates that the formulas and the emotional transitions defined in earlier sections seem to be suitable for modelling the influence of external factor on internal variables of agents and their emotional state.

2.5 Behaviour of the Agents

This Section describes the behaviour of each agent; in this case these are the crowd member and leader. Behaviour is described using a behaviour tree, which was chosen in Section 1.3.1. Each figure consists of various nodes in a tree structure. The squares with question marks denote selector nodes, squares with arrows depict the sequence nodes, red boxes describe the various actions the agent can perform, and yellow boxes depict the various conditions the agent can check for. All nodes return a true or false given their internal state after execution of the node as stated earlier in the Section about behaviour trees. In the end these designed behaviour trees will be implemented with the condition and action nodes functionality filled in.

2.5.1 Behaviour of the Crowd Member

Crowd members will behave differently given their surroundings and their current emotional state in a demonstration. A possible scenario for an agent is when a NLW has been deployed in his vicinity. When he sees a NLW the agent's first reaction will be to get away from the NLW as soon as possible. This is to prevent that he will be affected by the NLW and become hurt. If he is already affected, he will try to get away from his current position in a reverse direction in relation to the NLW. Again this is top priority for an agent and therefore positioned high in the behaviour tree.

When no visible NLWs are present there might be control force members who want to stop the agent from getting in the neighbourhood of a government building. A calm agent will try to stay away from them if this is possible. However if he is angry he will instead try to attack them by moving closer and start throwing rocks or even Molotov cocktails if the agent is armed. This kind of behaviour can also be observed in video footage concerning riots and interaction with control forces and corresponds to fight-or-flee behaviour.

Main goal of an agent is reaching his goal, in this case a government building. In normal situation when the agent is calm or angry, he will move towards this goal if nothing is interrupting him. However if he is in fear or panic he will try to avoid this goal because it poses a risk for the agent because of gathering crowd members. When this gathering becomes too large, the control forces will shift their attention to this gathering and thus the risk to individual crowd members would become larger. Therefore they tend to stay away and keep their distance. Besides avoiding his goal an agent, when in fear or panic, will try to avoid other crowd members. This seems to be common behaviour in people in fearful environments because they do not want to become trapped in a possible unpleasant environment. They try to stay away from each other and keep their distance. In case of fear an agent will not avoid his subgroup members because they provide a safe place, however when an agent is in panic he will also avoid his subgroup members because this also increases the risk.

Normal behaviour for calm or angry agents, when no special situation occurs, is to move towards subgroup members if they see them. This way they can form their group in the world and move together towards the goal. This is important because these subgroups have more influence in to comparison to individual crowd members. Behaviour described above is translated to the behaviour tree shown in Figure 19. The various sequence nodes depict behaviour for a certain situation. The order, from left to right, depicts the priority. High priority sequence nodes are executed first, if one fails the behaviour tree goes to the next sequence node until all have been executed.

2.5.2 Behaviour of the Leader

A leader has slightly different behaviour in comparison to a crowd member. At the moment he does not use emotions because he simulates a kind of fearless leader who wants to lead his troops. Emotions can be added, in the same fashion as the crowd member, by using the appropriate condition nodes. The behaviour tree of a leader is shown in Figure 20. Just like crowd members, if a leader sees a NLW he will try to avoid it and when he is being affected by one he will try to avoid it as well.

When a leader sees a subgroup member he will try to move towards this agent to form a physical group, just like crowd members moving towards subgroup members. This will result in a leader showing its followers to other subgroup members and forming larger groups. If a leader is trying to incite other crowd members and he sees that there is a control force member close by, he will move towards this control force member. The aim of this action is that following subgroup members will have become and will attack the control force member because they are led close to a control force member. If the leader is calming he will try to avoid control force member and try to keep his followers safe.

If a goal is present and the leader is either calming or inciting but not seeing a control force member, he will move towards the goal and lead his subgroup members towards it.

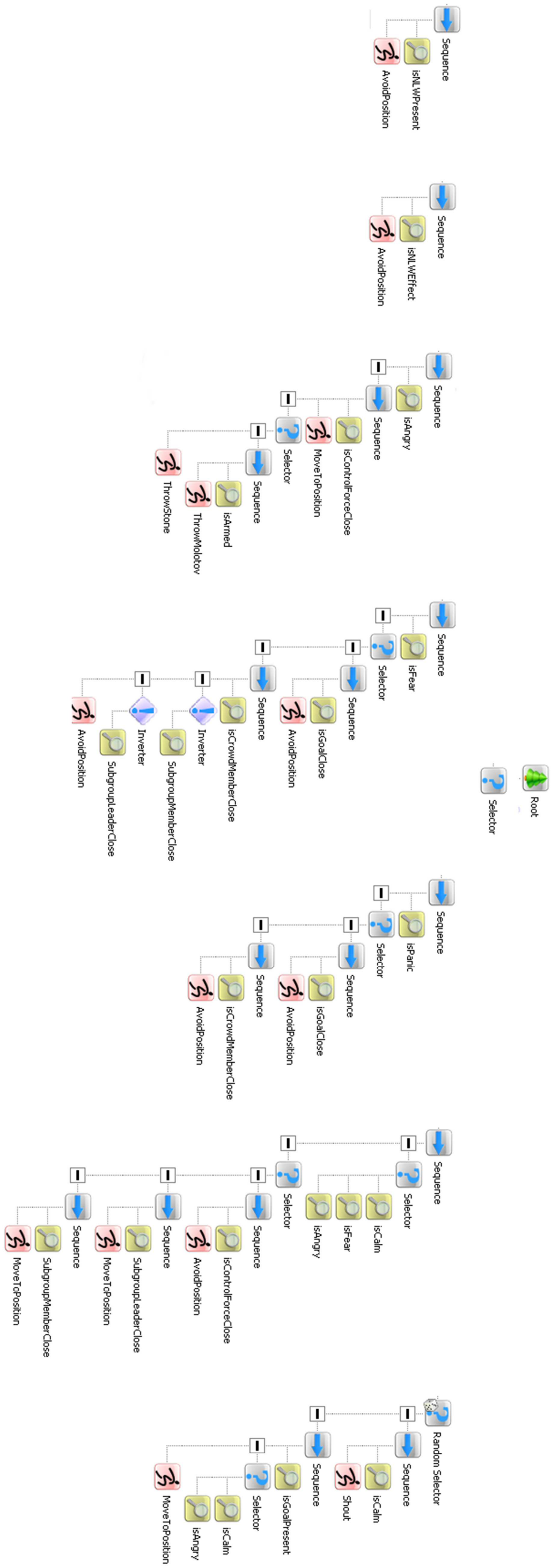

Figure 19. Behaviour tree of a crowd member describing the behaviour during a demonstration.

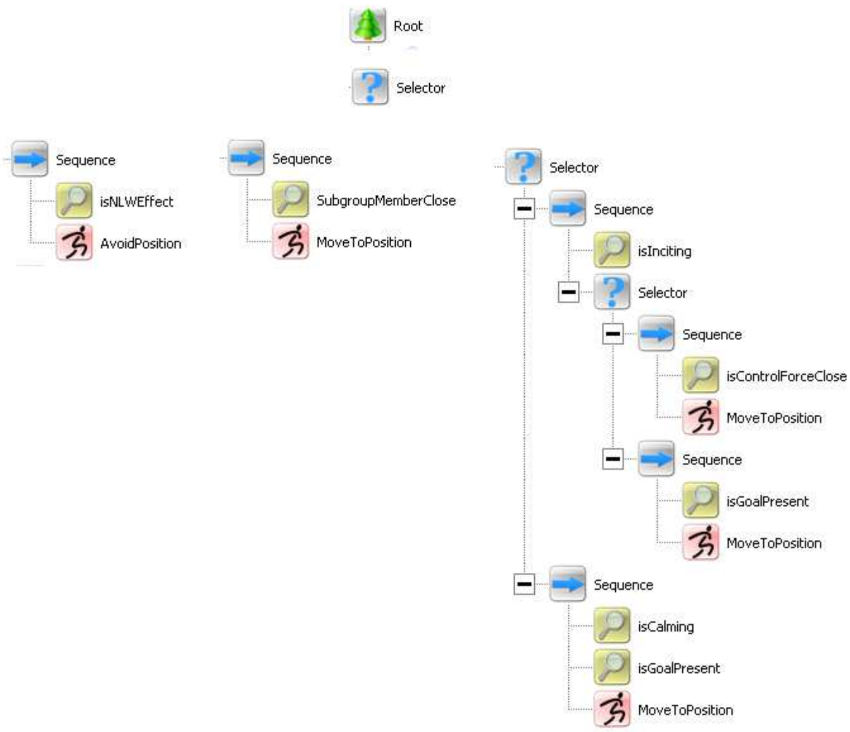

Figure 20. Behaviour tree of a leader describing the behaviour during a demonstration.

2.6 Centralized Agents

Besides the decentralized agents as described in the previous sections, there is also a model of a centralized crowd. The centralized crowd is made to compare the two kinds of crowds with each other and determine which is better suited for crowd simulation. The results of this comparison can be seen in Chapter 4. To accurately compare the two crowds and point any differences on centralized or decentralized approach, both crowd use the same cognitive model with the same internal models as described in Chapter 2. Same formulas are used to change the internal variables of the agent and to transition between emotional states.

Key difference between the types of crowds is the behaviour. The centralized crowd revolves around leaders and giving orders to other members of the crowd instead of influencing the crowd members. If a crowd member belongs to a subgroup he will listen to orders of his leader. This leader can order crowd members to move to a goal, avoid a certain location if there is a NLW present, or issue an attack order if the leader is inciting and a control force member is close. Crowd members will only react to these orders if the situation allows it. However leaders are not given total control over crowd members, if a crowd member encounters a NLW he will try get away from it without waiting for instructions of a leader.

During simulation there might be agents who do not have a leader or are not part of a subgroup. To maintain a centralized structure, agents who are leaderless will find the closest agent with the lowest ID and appoint him as a leader. The appointed crowd member will be notified and start sending out commands to everybody in range just like a subgroup leader would.

The behaviour tree for the crowd member is given in Figure 21. The two rows are on the same level and depict the various sequence nodes in priority order. Not greyed out areas depict additions or changes in comparison to the behaviour tree of a decentralized crowd member. Extra nodes are used for giving orders to other agents, listening to orders, appointing leaders, and checking if an agent is appointed as a leader. Orders which a leader can issue are issueMoveToOrder (telling agents to move), issueAttackOrder (telling agents to attack), and issueAvoidOrder (telling agents to avoid). A crowd member has also received several condition nodes which relate to these orders, these are: hasMoveToOrder, hasAttackOrder, and hasAvoidOrder. These nodes are used to determine if an agent has received and order and if he will execute them given the situation. There is also a condition node added which checks if the appointed leader is still visible for this agent. If this is not

the case a new leader will be appointed by the agent based on which agents he can see. When an agent itself is a leader he will issue orders to all the agents who are visible and have him as an appointed leader.
The behaviour tree for the leader is also changed in comparison to the decentralized version.. Extra nodes were added for giving orders to following subgroup members just as appointed crowd members. However the difference is that there is no need to check if this leader is also an appointed leader or if he himself has received orders. The adapted behaviour tree can be seen in Figure 22.

Figure 21. Behavior tree for a centralized crowd member, greyed out nodes depict nodes which were also present in the decentralized crowd member's behavior tree.

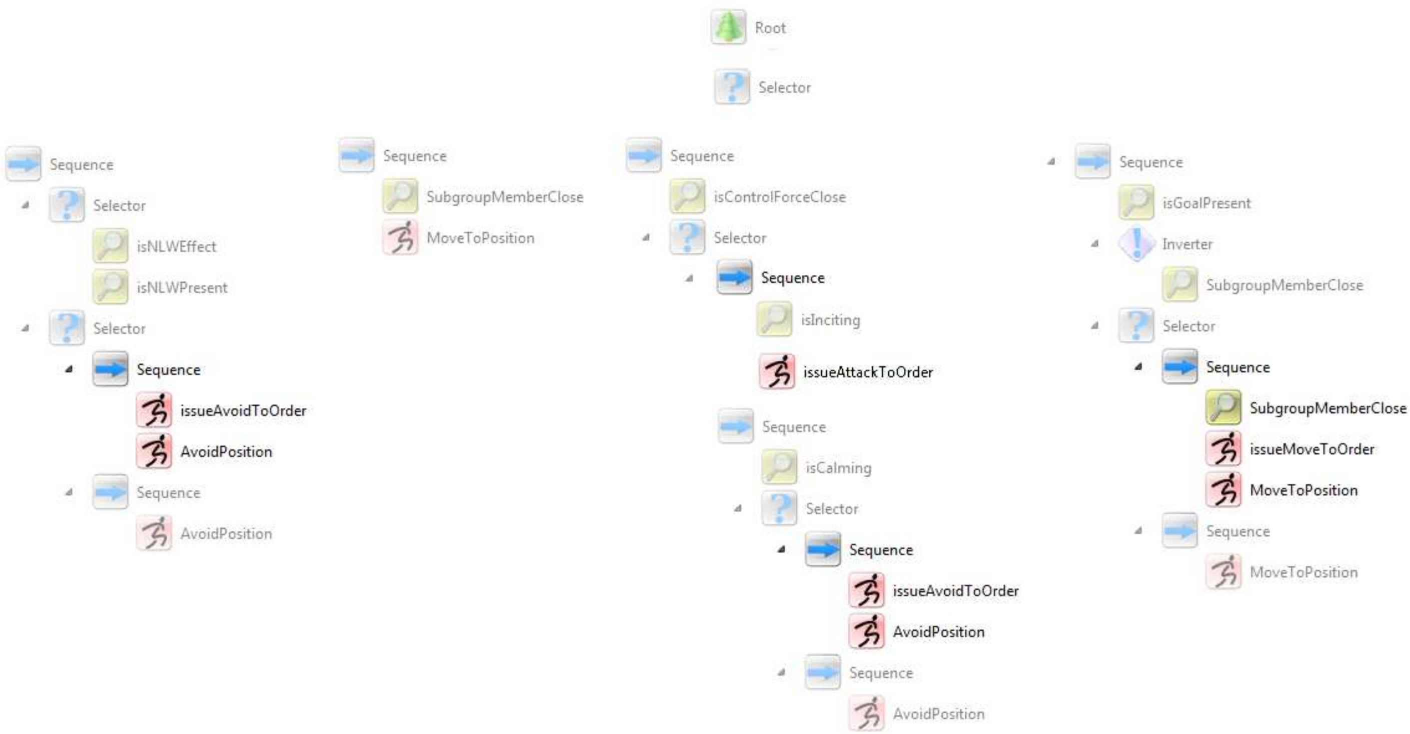

Figure 22. Behaviour tree for a centralized leader, greyed out nodes are nodes which are also present in the decentralized leader's behaviour tree.

Chapter 3 Implementation of the Proposed Crowd Model

This chapter describes how the decentralized agents described in Chapter 2 are implemented. It describes how MASON and VBS2 were used to create a crowd simulation and what their respective responsibility is. This chapter also describes how the two separate applications are connected to each other so they can exchange information. This chapter is part of the research question "How can a crowd of demonstrators be modelled using decentralized techniques?" and is part of step 3 of the roadmap.

3.1 Used Applications

To implement the agent model described in Chapter 2 two applications have been used: MASON (version 16) and VBS2 (1.6). MASON is a multi-agent simulation environment; build in Java, used to run various kinds of multi agent simulations such as cellular automata or particle systems. To use behaviour trees, a java library called Java Behaviour Trees (version of 2011-10-03) is used which can be combined with the agent simulation in MASON. MASON also hosts a couple of interesting features such as scheduling, modelling of a continuous virtual world, and the ability to start, stop, and pause a simulation. It also supports viewing individual agent's states during the simulation but also viewing global states of the simulation such as the total number of agents present and how many agents are in a certain emotion. When running the simulation MASON can provide real-time graphs of these internal and external states as seen in Figure 13. Each agent is modelled using a normal Java class with private and public variables and functionality in the form of functions making it easy to use when one has experience in programming Java applications. MASON also contains a scheduler which provides the possibility to simulate large amounts of agents by spreading the burden. To handle crowd simulation there are two components which were added, these are world appraisal and emotional updater. World appraisal component checks the current world in which an agent resides and checks which rules, as described in Appendix A, have to be used on the internal variables and changes these according to the formulas. Factors for these formulas are described in a separate XML file making it possible to externally change these without changing the source code. When these internal variables have been updated the emotional updater is called. This component decides in which emotional state an agent will be according to the rules as described in Section 2.2.2. This state is stored inside an agent as a separate attribute. Values used for transitions between emotional states are described in a separate XML file just like the factors for the world appraisal.
As mentioned earlier Java behaviour Trees are used to implement the agent's reasoning. This is an open source library and offers the ability to use behaviour trees and provides an editor to create and edit them. This editor makes it easier for users to create and adapt the behaviour trees without needing knowledge of Java programming. The downside of using this library is that it comes with a GPLv3 license, meaning that programs using this library need to adopt this GPLv3 and are required to give source code if asked for. This might pose a problem when the crowd simulation becomes part of a commercial product.
Each time a step is executed in MASON the behaviour tree for each agent gets activated. In this activation step the behaviour of an agent is determined by going through all nodes and executing them in order. These nodes are described in Section 2.5. To have behaviour tree use information about the world from MASON a blackboard, called context, is used. An agent can store information from MASON, such as the number of agents it sees or its movement speed, on this blackboard, which the behaviour tree can read. This is needed because the behaviour tree itself cannot gain access to information stored in MASON and would make reasoning and executing condition nodes about this world difficult. To conclude, MASON's responsibility is to handle agents, update their internal variables and emotional states, and determine their behaviour given the world they exist in.
To visualize the crowd, controlled by MASON, VBS2 was used. VBS2 is a serious game developed by Bohemia interactive and can be described a simulator for military operations. It is mainly used by armed forces to train soldiers or other personnel with tactics in field or how to communicate effectively between certain roles. In this case VBS2 is used to simulate a demonstration where VBS2 shows the agents in a virtual world and determine which agents an agent can see and if he is affected by a NLW or not. This is the responsibility of VBS2 and not of MASON because VBS2 contains information about the virtual world such as buildings, slopes

and other objects which influence visibility and effectiveness of NLWs. VBS2 also handles path finding for agents, for example around buildings and collision detection between virtual entities.
Information about visibility and if an agent is affected has to be shared with MASON. To connect the MASON simulation with VBS2 a messaging mechanism was needed which could pass messages between C++ and Java at a high speed. To do this IMB, publish-subscribe service developed by TNO, was used. IMB is based on HLA and provides applications the ability to send messages to each other regardless of the programming language in which they have been written. Because VBS2 is a C++ application and MASON a Java application, a translation mechanism of the messages between the two applications because of underlying differences in data types. For this translation Google Protobuf was used, which can translate predefined messages to the correct data types used by both Java and C++, making it possible to exchange messages without corrupting their contained data.

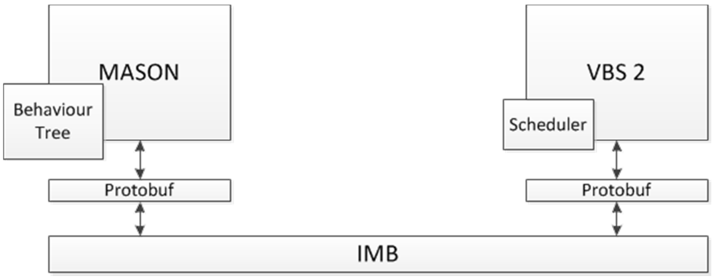

Figure 23. Overview of the implementation of MASON and VBS2 and the connection.

MASON sends messages to VBS2 about the initial position of agents, where agents should move to, what their emotional state is, and if they are influenced by a leader or not. These messages are received by VBS2 and handled accordingly by calling internal scripts. Some information, such as the visibility of the agents and the current position in the world has to be returned to MASON. To prevent that VBS2 periodically has to process this information in one go, a scheduler was introduced. VBS2 will try to schedule these subscriptions in such a fashion that the load is spread evenly and that the performance is steady. MASON can subscribe itself to certain information of agents, such as the position, and state how many times this information should be received. An overview of the communication between MASON and VBS2 can be seen in **Error! Reference source not found.**. All the protobuf messages used by MASON and VBS2 are given in Appendix D. The messages used for the scheduler can be found in Appendix E.
An overview of the connection between MASON and VBS2 is given in Figure 23. Using this setup also enables a user to switch out VBS2 in favour for other visualisation systems as long as this new system uses IMB and the same messages. Currently the crowd simulations runs on one machine but through this setup the simulation can be used on a network. One pc can handle the agents with MASON, while another does the visualisation with VBS2. Using this message based approach also means that the messages could be recorded during simulation and played back at a later time point. This makes it possible to replay a simulation.

3.2 Appearance of the Crowd Simulation

Both applications, MASON and VBS2, have a distinct graphical representation of the implemented crowd. The appearance of the simulated crowd in MASON is shown in Figure 24. It depicts various crowd members, which are located at the centre of the screenshot. The MASON representation of an agent consists of two parts, one of these is the circle around an agent with a specific colour, which shows to which subgroup an agent belongs. Shared colours mean that agents belong to the same subgroup, however if the colour is black than this agent does not belong to a subgroup. The other part is an internal dot, which in the screenshot is for most agents yellow, shows the emotional state of an agent. Yellow if an agent is calm, red if he is angry, blue if he is in fear, and purple if he is in panic.

The blue dots on the left side of the screenshot are control force members and the white large dot on the left of the control force members is a government building the agents wants to occupy. At the moment of taking the screenshot no NLWs were used, but they are depicted by a white dot and large white circle, which describes the effective range.

Leaders are depicted by a closed dot which describes its subgroup in the same way as crowd members. A leader also has a small dot in the centre depicting if he is either inciting, red, or calming, green.

Using this representation made it possible to view all the important aspects of the crowd in one glance without look at the internal state of the agents. MASON's representation of the world does not include objects like buildings. As said before, these kinds of objects are located within VBS2 and are not part of the responsibility of MASON. This makes the representation of the world in MASON relatively empty.

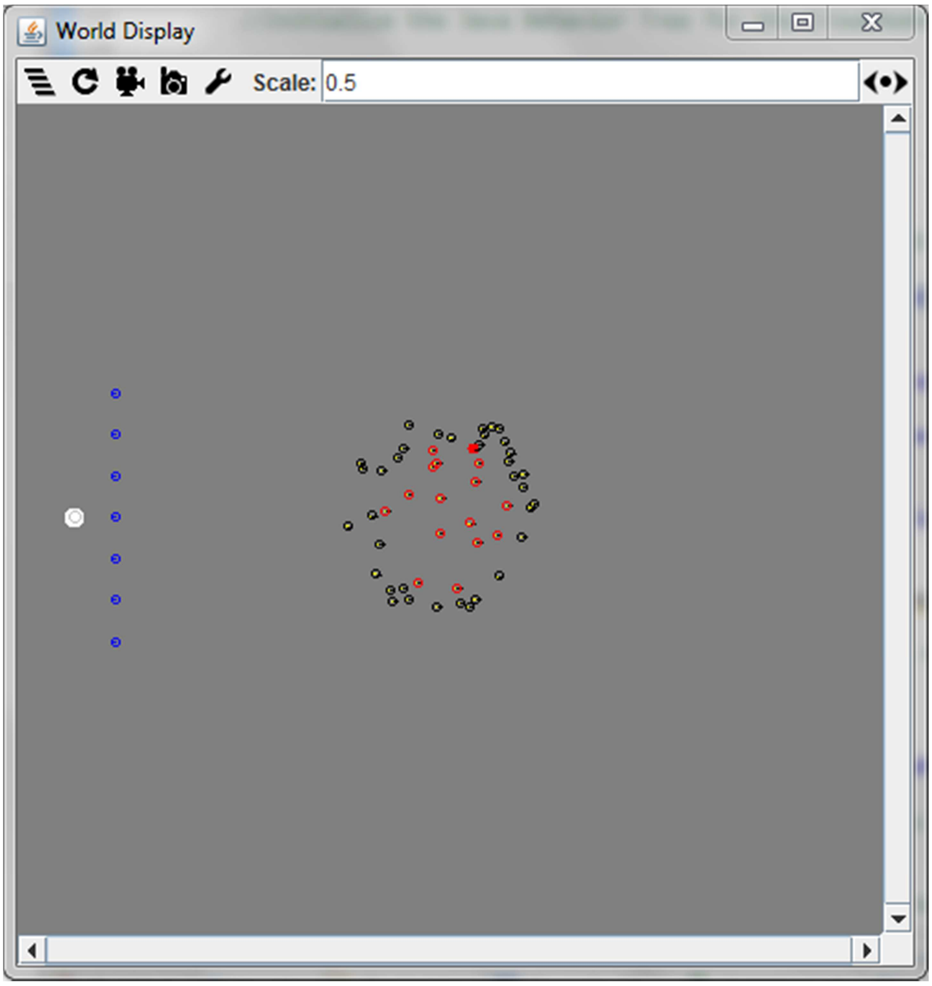

Figure 24. MASON representation of the crowd.

Representation of the simulated crowd in VBS2 is shown in Figure 25. Various entities are shown which are linked to the agents within MASON through IMB. Each character has a coloured text floating above it. This text describes the current emotional state of an agent. The text is red when the agent is angry and yellow if he is calm. When the agent is fearful the text will be blue but if he is panicking the text will be coloured purple. The colours correspond to the colours used in MASON. The screenshot shows that most of the calm and fearful agents have a reasonable distance between themselves and present control force members. Angry crowd members are attacking a control force members as seen by the number of angry crowd members around him. Not depicted on this screenshot but present in VBS2 are NLWs and their visualisation. When MASON starts a NLW, for example teargas, VBS2 gets notified and starts drawing a particle cloud at the location specified. This cloud will be as large as the effective radius of the used NLW as stated in Section 2.3.3. Other NLWs are also visualized. In the final version there are also indicators which show if a crowd member is being affected by a subgroup leader, who is an subgroup leader, and if an agent is being affected by a NLW. When an agent is being affected by a NLW this is shown by a white @, if he is influenced by a subgroup leader he has a white * above him. This subgroup leader has a larger yellow * floating above him. As with MASON these visual cue help to get an understanding about what is happening in one glance and makes it easier for users to understand what is going on.

Figure 25. VBS2 representation of the crowd.

Chapter 4 Validity and Usability of the Crowd Model

In this chapter the proposed model will be validated and tested for usability. This chapter is part of the research question "Is a decentralized crowd more realistic than a centralized crowd?" It contains the overall setup and describes which factors will be recorded for both crowds during the experiment. These factors will be used to score each crowd.

The results of the two-way ANOVA will be shown and discussed. Besides scoring the crowds, the usability in VBS2 has to be verified. This is done by recording the number of frames per second during runtime. The setup and the results will be described in this chapter. The setup will be explained and then results of the experiment will be given and analysed.

4.1 Validity of the Crowd

The assumption is that a decentralized crowd is better suited for crowd simulation than a centralized crowd. As mentioned in Section 1.1 demonstrators determine their own behaviour and actions are dictated by the group how to behave. To determine if a decentralized crowd is better suited it is compared with a centralized crowd, as described in Section 2.6. This can be determined by looking how well a crowd scores given the situation. But knowing which type of crowd scores higher is not enough. It might be that even though one scores higher that the difference between them is very low and that both types of crowds exhibit the same kind of behaviour. To determine the difference an ANOVA was performed and boxplots were generated. To determine which crowd is more suitable the means on the attributes were compared of the two crowd types. The exact experimental setup is described in Section 4.1.1.

4.1.1 Experimental Setup

To determine the validity both types of crowds have to be compared to each other. To prevent that one crowd type scores better because of a particular scenario, two scenarios have been used. First scenario has control forces using teargas, while in the second scenario they use a HPM. Both NLWs have been chosen in Section 1.2.2. The type of crowd will be factor 1 in the two by two design while the scenario will be factor 2. Each permutation of these factors will be a simulation which is run with MASON. VBS2 will not be used and some functionality is taken over such as observation of agents and checking if an agent is affected by an NLW. These will be replaced by looking at the distances between agents and determining if they are within range. This also means that there are no buildings present in the virtual world which might influence observation. This does not pose a problem because both crowds are put in the exact same scenarios. During each scenario three attributes for each agent will be recorded. These attributes are based on real-life demonstrations and are the number of times somebody is shouting, how many objects he throws, and the number of agents around him. At the end of each run the final value for each attribute is recorded. Each simulation runs for a total of 15000 MASON steps, making the observed attributes comparable between crowds. When the crowds have completed their scenarios an ANOVA test will determine if the two crowds are significantly different from each other on each attribute and on a sum of normalized attributes. An overview of the two-way ANOVA is given in Figure 26. To get the normalized sum each attribute was normalized to a value between 0 and 1 and then added to each other. Each attribute is normalized to prevent bias in the sum because of high numbers. The result is a value between 0 and 3 where 3 mean that an agent scores very high. Each of these scenarios is run for a total of 10 times, to prevent random factors possibly favouring a crowd which in other situations would not be the case. The expectation is that running each simulation for 10 times would result in an accurate description of the underlying distributions and would give enough evidence that the results found by performing the ANOVA are correct.

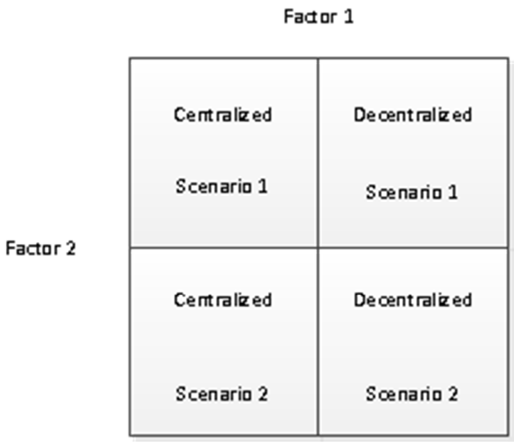

Figure 26. Two by two design used for the experiment.

4.1.2 Used Scenarios

In this Section the two scenarios will be described which will be used in the two by two research design. Both scenarios will be applied to the decentralized and the centralized crowds.

4.1.2.1 Commonly Shared Attributes between Scenarios

The demonstration takes place on a marketplace in a Middle Eastern country. A group of demonstrators have gathered and they want to storm a government building which is located on the marketplace. The control forces present need to keep the crowd members in check and make sure that they cannot reach the government building. To accomplish this they have NLWs at their disposal.

The composition of the crowd is given in Table 5. There are 35 crowd members present which belong to no subgroup. Besides these crowd members there is also a subgroup present. The subgroup is at the centre of the whole crowd and all the members are in each other's vicinity. This subgroup consists of 15 members and one leader which will try to incite the people in his subgroup. The leader is standing amongst his subgroup members.

The control force is present with 8 people, who are standing in a line formation in between the crowd and the governmental building. They will start with show of force which they maintain during the scenario. Before deploying a NLW they will give a warning. After the deployment of the NLW they will resume with the show of force stance.

Table 5 Crowd composition for both scenarios

Description	Number
Crowd Members	35
Crowd Members in Subgroup 1	15
Inciting Leader Subgroup 1	1
Control Force Members	8

The variables for the crowd members can be found in Table 6. All the crowd members are male and their age is between 20 and 25 years. They are calm and have a resolve of 45. They do not experience any pain of risk at the moment and are not armed. The thresholds which influence how the crowd member will transition between emotional states is set between 0.5 and 1.5 and varies from agent to agent.

Table 6. Variables for individual crowd members.

Variable	Value
isMale	True
Age	20 - 25
emotionalState	CALM
Resolve	45
Pain	0
Arousal	40
personalRisk	0
isArmed	False
Thresholds	0.5 – 1.5

The leader's variables can be found in Table 7. His leadership effect is set to the value of 1. This means that he is inciting the other crowd members in his subgroup and is very influential.

Table 7. Variables for the leader.

Variable	Value
leadershipEffect	1

The variables of the control forces can be found in Table 8. Their effectiveness is set to 1.0. This means that they are very organized and reflect this to the crowd members.

Table 8. Variables for control force members.

Variable	Value
Effectiveness	1.0

An overview of the situation is given in Figure 27. The control force is represented by the blue dots. The government building they have to protect is given by the light green rectangle. The crowd consists of the grey circle in which the purple circle represents the subgroup. The purple square represents the leader of the subgroup.
The circles depict the area in which the crowd members will be put when the scenario starts. This does not

mean that the crowd will keep in this circle formation during the simulation.

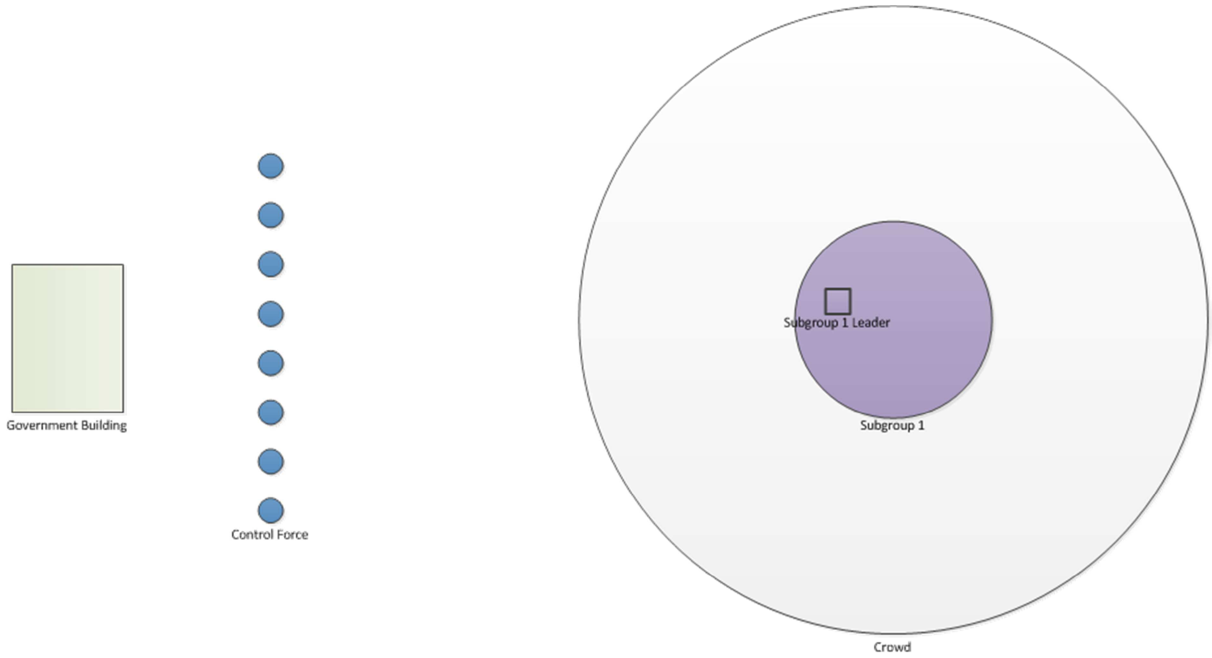

Figure 27. Overview of the shared information between the scenarios.

4.1.2.2 Scenario 1

The control forces deploy a single tear gas grenade in the centre of the crowd. Their aim is to disperse the crowd and thereby destroying the cohesion with the subgroup and the crowd. Subgroup 1 will get the most of the effect of the tear gas grenade. This situation is described in Figure 28. The impact point is in the middle of the crowd amongst the subgroup members. The variables of the tear gas are given in Table 9. The time point of deployment of the NLW is 30 seconds after starting the scenario. This gives each crowd the possibility to move towards the goal and form a crowd like form on their own.

Table 9. Variables for the teargas NLW in scenario 1.

Variable	Value
onsetEffect	0
durationEffect	>300 seconds
Aggressiveness	0.5
Compellingness	1.0
effectShapeWidth	20-30m
effectShapeHeight	20-30m
timepointNLW	30 seconds

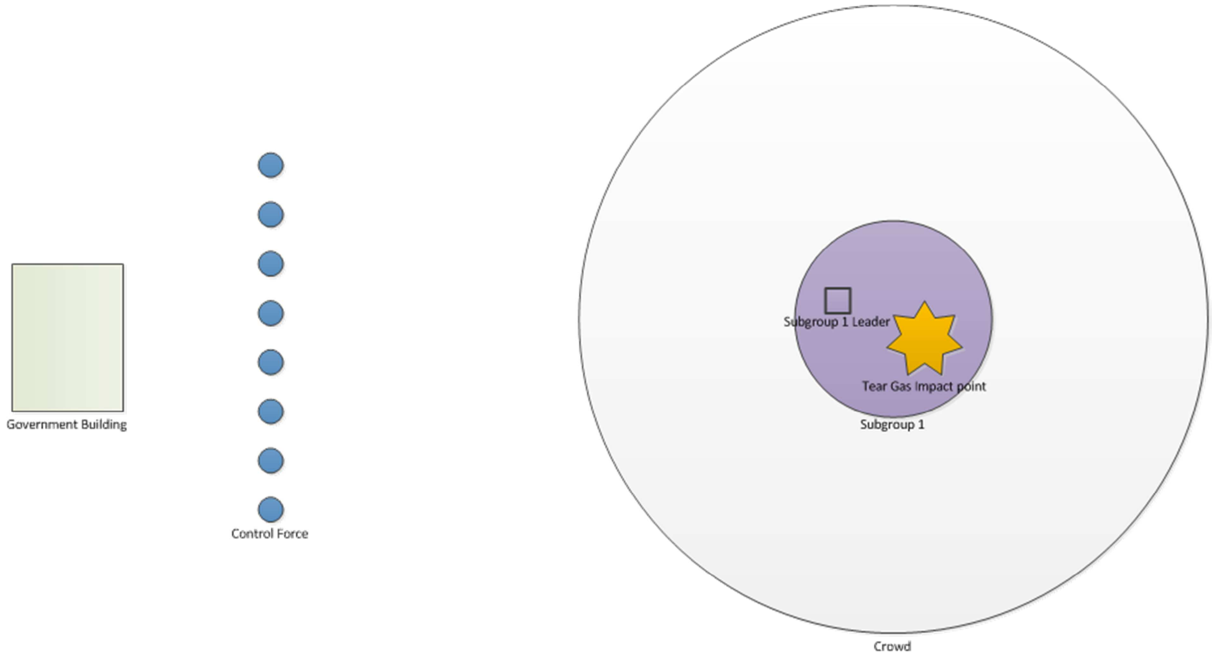

Figure 28. An overview of the first scenario showing where the tear gas grenade will hit the crowd.

4.1.2.3 Scenario 2

In this scenario the control forces will use a high power microwave on the crowd. They will again aim the NLW at the front of the crowd. Again the aim of the control force is to make sure that the crowd does not move into the direction of the government building. The situation is described in Figure 29. The high power microwave is fired at the first timeframe. The variables for the high power microwave can be found in Table 10.

Table 10 Variables for HPM in scenario 2

Variable	Value
onsetEffect	0
durationEffect	Continuous
Aggressiveness	0.0
Compellingness	1.0
effectShapeWidth	760m
effectShapeHeight	1m
timepointNLW	30 seconds

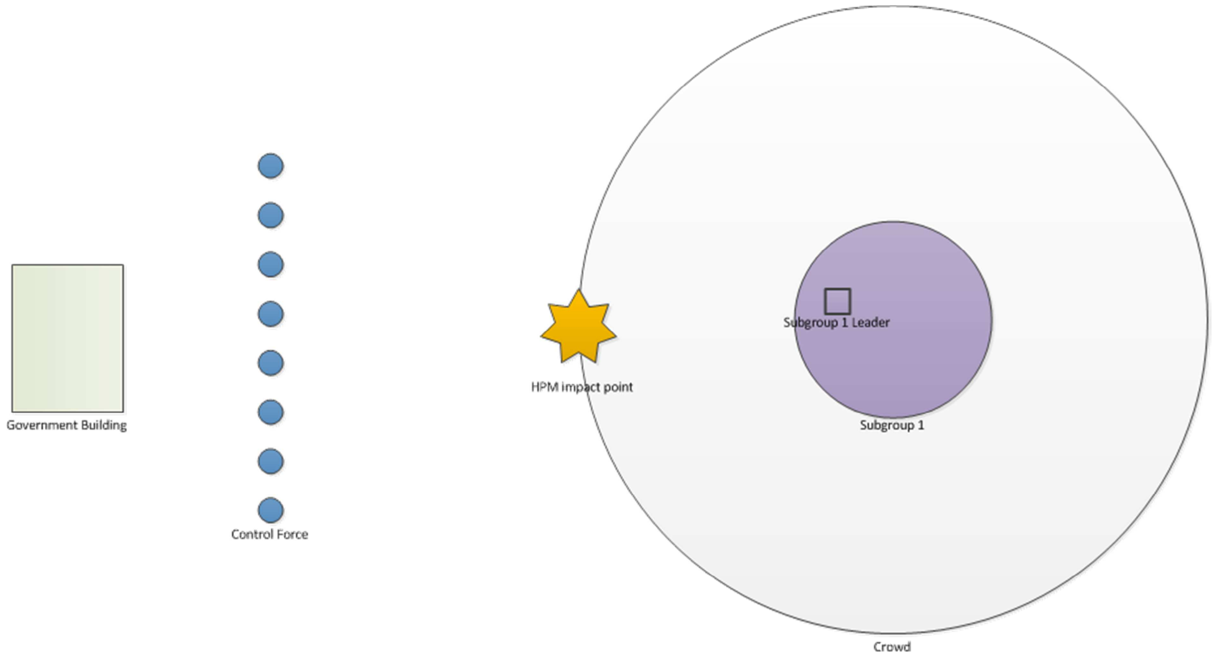

Figure 29. An overview of the second scenario, showing where the high power microwave will hit the crowd.

4.1.3 Results

This Section describes the results obtained from the experiment. Three different attributes for each agent were recorded: the number of times the agent has shouted, the number of objects he has thrown and the number of agents who are visible around the agent. Besides the attributes a normalized sum was constructed. Each type of crowd was run in each scenario for a total of 10 times for a maximum of 15000 steps within MASON.

4.1.3.1 Distribution of the Recorded Attribute Values

An ANOVA shows if there is a significant difference between the two crowds but does not show what the underlying distribution is. Boxplots were used to gain insight into the distributions of each crowd type. During each run the observed values for each attribute was recorded. As stated before each simulation was run 10 times. When these 10 runs were completed all observed values were added to each other to form a big list containing all runs. These values were used to generate the boxplots. Each figure in this Section shows two boxplots, one for each type of crowd, in the same scenario and on the same recorded attribute. In the following figures the median of the boxplot is represented by a red horizontal bar, maximum and minimum values are represented by short blue bars at the end of the blue box, and this box depicts the first and third quartiles. These five values together form the basis of a boxplot. Besides the boxplot there are also red circles and crosses. Circles depict values which lie further than 3 times the quartile away from the median while the crosses depict values which lie between 1.5 and 3 times the quartile. Both could be described as being outliers.

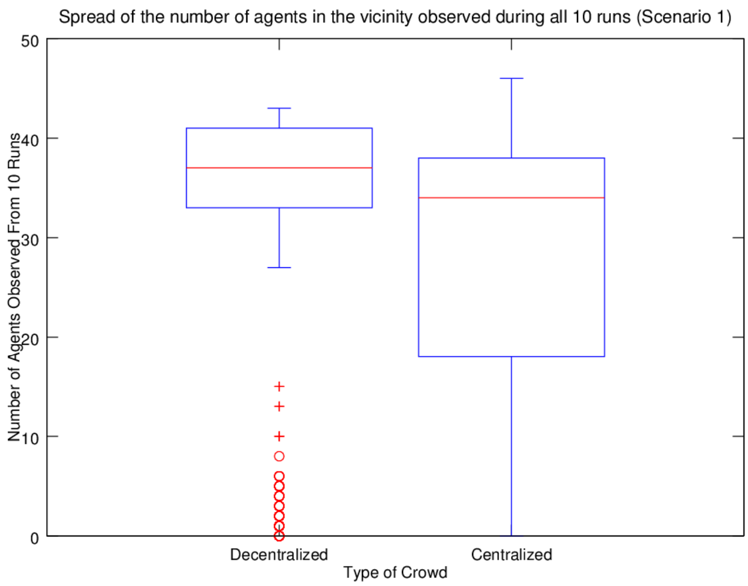

Figure 30. Boxplot for scenario 1 and the number of agents close to the agent.

The first boxplot shows the observed values for the number of agents in the vicinity for all 10 runs. This is done for both types of crowds during the first scenario, were the crowd is confronted with a teargas grenade. The boxplot shows that the decentralized crowd has a higher value for its lowest value, first quartile, median, and the third quartile, in comparison to the centralize crowd. However the centralized crowd has a higher maximum value and a different spread. This might be caused by centralized agents going towards a goal if a leader gives an order to do so. In the decentralized crowd an agent itself will decide to do this or not. If in the centralized crowd an agent does not have a leader and cannot appoint one he will stay put because he receives no order. This could explain why the density is lower in the centralized crowd because they do not gather around the same place, which would be the goal.

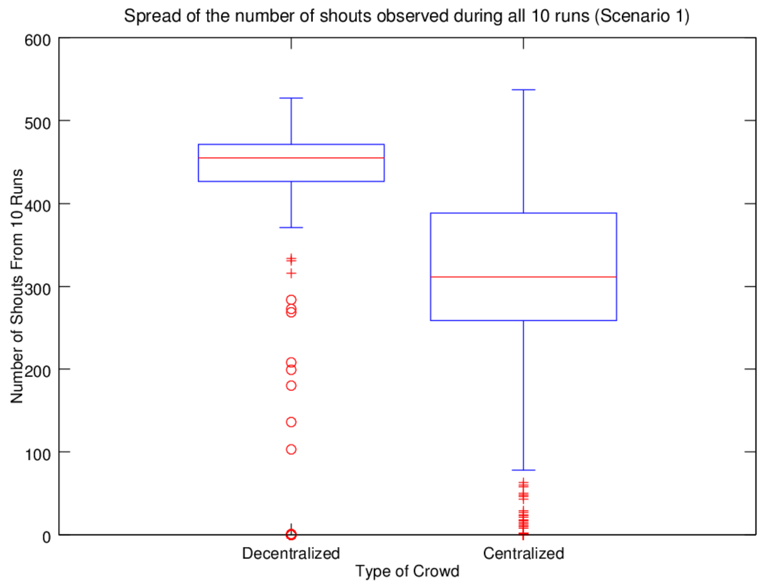

Figure 31. Boxplot for scenario 1 and the number of shouts.

The second boxplot describes the combined number of shouts in scenario 1. The decentralized crowd has a median of about 450 shouts while the median for the centralized crowd is lower and lies about 320. Maximum value for both kinds of crowds lie around the same value, between 520 and 560 but the overall spread of the boxplots is different. The decentralized crowd has more values between the median and third quartile in comparison to the centralized crowd. The cause might be the leaders in the centralized crowd telling other agents to attack control force members if they are close. This means that agents will be directed towards the control force members while the decentralized crowd have to decide to do this on their own and can only do this when they actually see a control force member. In the centralized crowd a leader could see a control force member and direct its followers, even if these cannot see the control force member themselves. This also relates to the lower number of shouts for the centralized crowd but higher number of thrown objects as seen in the next boxplot. In this crowd simulation an agent will not shout and thrown an object at the same time. The boxplot's five values are all around 0 because most agents in the scenario will not become angry and when they are not angry they will not throw objects.

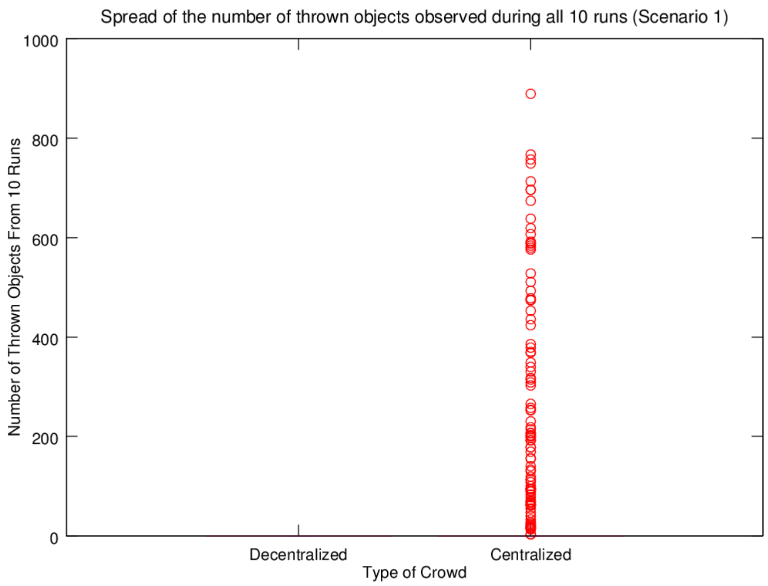

Figure 32. Boxplot for scenario 1 and the number of thrown objects.

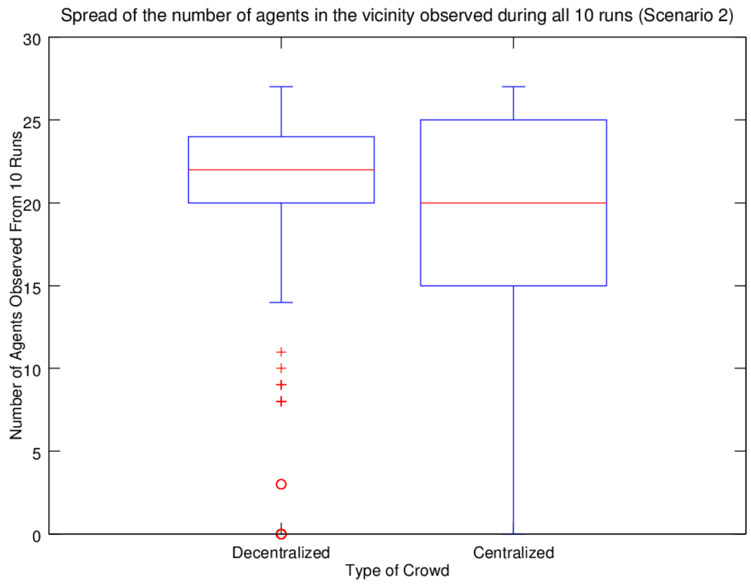

Figure 33. Boxplot for scenario 2 and the number of agents close.

The boxplot above depicts both crowds in scenario 2, where the HPM is used.
The median, third quartile and maximum value for both crowds are nearly identical in both cases. The difference between the crowds is the lowest value and first quartile. The centralized crowd has a larger spread, meaning that the decentralized crowd manages to get a compact spread of agents in the world. The reason for this might be agents who do not have a leader in the centralized crowd and have become isolated from the rest. In case of the decentralized agents the isolated agents will move towards the goal without explicit orders,

while the centralized agents need an order to move towards the goal. In the end the decentralized crowd will be mostly gathered at or near to the goal and therefore see more agents around them.

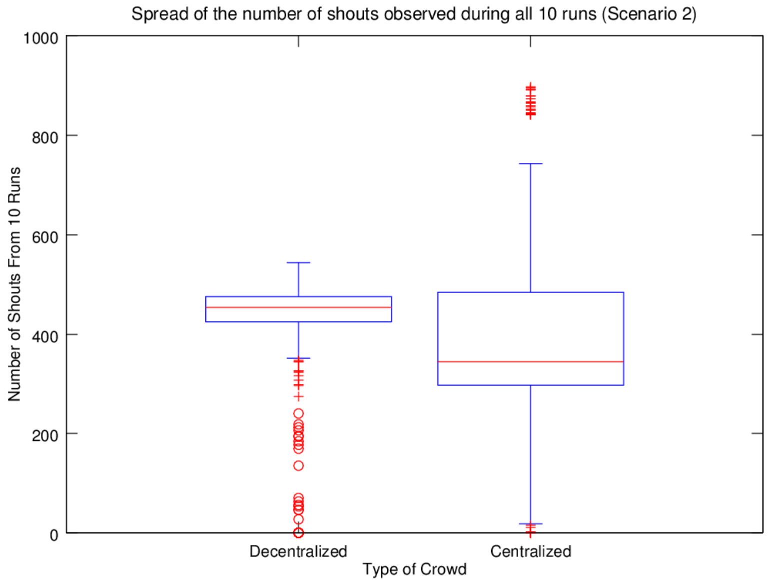

Figure 34. Boxplot for scenario 2 and the number of shouts.

Shown above is the boxplot for scenario 2 which displays the number of shouts for each crowd type. The decentralized crowd has a compact spread, as shown by the small boxplot, in comparison to the centralized crowd. The centralized crowd also has values which are far from the maximum value and third quartile, depicted by the red crosses at the top. The cause for this might be that the leaders tell the agents to stay away from the HPM when they see it. This causes agents who do not even see the HPM to move away from that location. When the agent is fleeing or moving away he will not shout, while a decentralized agent will move away when he sees a HPM by himself, which might be at a later point in time and gives the agent more time to keep on shouting.

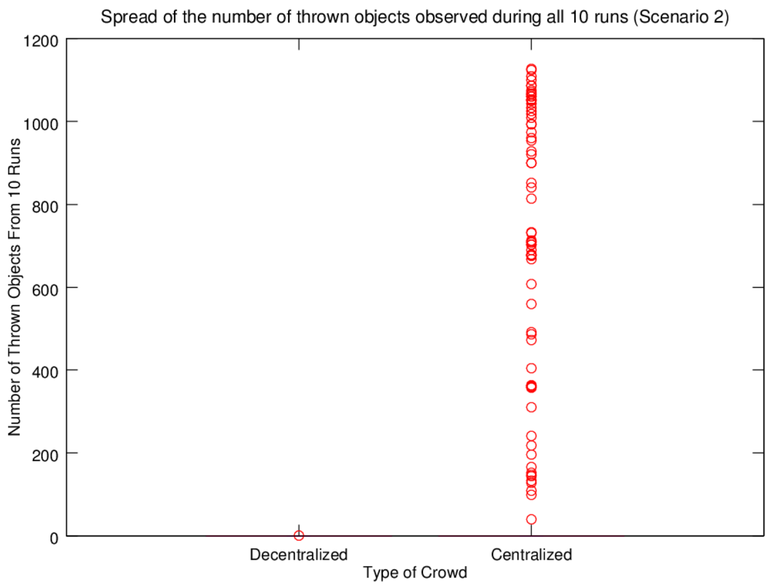

Figure 35. Boxplot for scenario 2 and the number of thrown objects.

Depicted above is the boxplot for the second scenario which shows the number of thrown objects by agents. In both crowd types the boxplot lies around 0. There are some outliers in the centralized crowd but overall the level of thrown object is low. Again there seems to be a relation between this and the number of shouts just as the first scenario.

To determine which of the crowds is scores better; the averages were calculated for each attribute. Results are shown in Figure 36, Figure 37, and Figure 38. On the number of agents close and the number of shouts, the decentralized crowd scores higher on both scenarios. However the centralized crowd has higher means on the number of thrown objects. This means that on two of the three recorded attributes the decentralized crowd scored better than the centralized crowd.

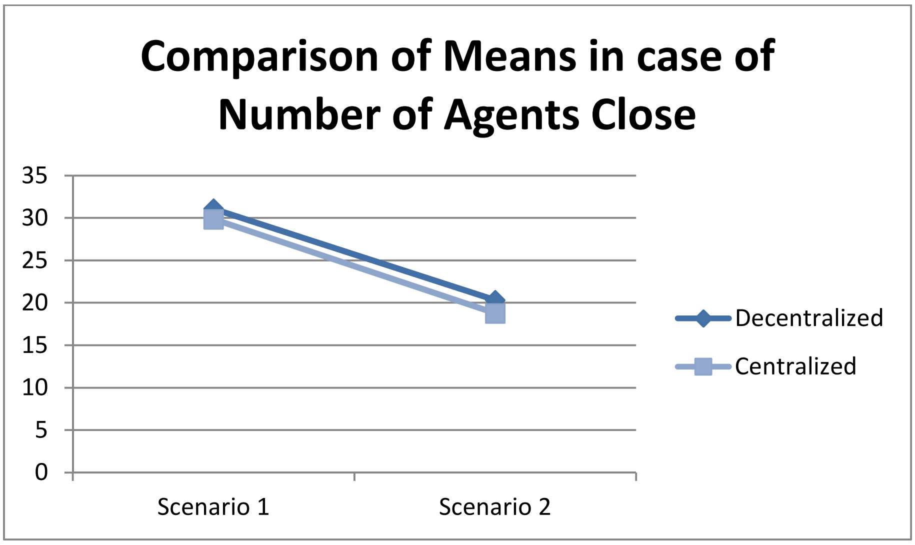

Figure 36. Comparison of means for the number of agents close.

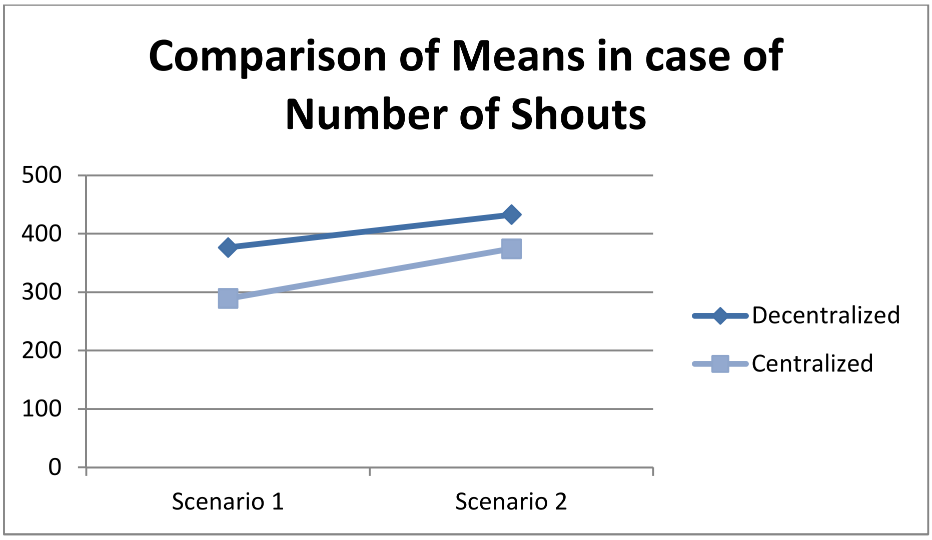

Figure 37. Comparison of means for the number of shouts.

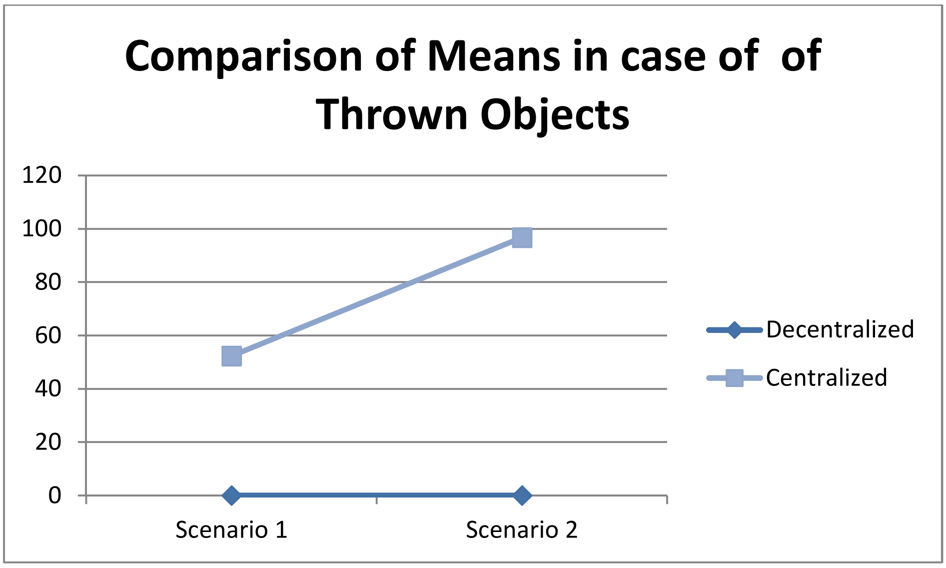

Figure 38. Comparison of means for the number of thrown objects.

These boxplots and histograms give an indication of the distributions of the crowds related to each other and the scenarios. The boxplots show that the decentralized crowd is different from the centralized crowd because of different distributions on recorded attribute values, however to determine if the two crowds are significantly different an ANOVA will be performed. Results of the ANOVA can be seen in Section 4.1.3.2.
As determined by comparing the means, the decentralized crowd scores higher in two of the three attributes which were recorded. The assumption was made a decentralized crowd would score higher, meaning that a decentralized crowd would be more useful for these scenarios than the centralized crowd.

4.1.3.2 ANOVA results

The decentralized crowd appears to be more suitable and seems to result in different behaviour given the boxplot. However it is not yet certain that the two crowds are significantly different from each other, it could be that they generally behave in the same way but with slight variations. An ANOVA can determine if the two types of crowds differ and if this difference is significant. The ANOVA was performed on each attribute and on the normalized sum.
Results of the ANOVA on the attributes and weighted sum can be found in Table 11 and in Appendix H. Independent variables are the scenario and type of crowd. The depended variables are the observed values for

the attributes: number of observed agents, number of shouts, and number of thrown objects. Besides the dependant variables there is also a normalized sum. Null hypothesis for all ANOVAs is that observed values in the sample come from the same distribution regardless of the type of crowd. The *p*-value describes how likely it is that all samples from both types of crowd are actually from the same population. If the *p*-value is high than this is probably the case, if it is below 0.05 than the crowds are probably from separate populations.

Table 11. Results of the ANOVA test.

Attribute	*p*-Value
Agents Close	$p = 0.0000$
Number of Shouts	$p = 0.0000$
Number of Thrown Objects	$p = 0.1070$
Weighted Sum of the Three Attributes	$p = 0.0000$

Results show that in three cases the *p*-value is lower than 0.05, the chosen significance level. Meaning that the null hypothesis, samples were taken from the same distribution, can be rejected. Because the null hypothesis is rejected there is a high possibility that the two crowds differ from each other. These results are based on 10 runs, to increase the assurance that the found results are correct. Results in Appendix H show that there is no significant interaction between the crowd type and the type of scenario on the three observed attributes, meaning that one crowd does not perform better in one particular scenario in comparison to the other. However there seems to be a significant linear interaction on the weighted sum. This might be an indication that one of the crowd types is better in one of the scenarios. However this was not observed in the other attributes.
The octave script used to generate the boxplots and calculate the ANOVA results can be found in Appendix J.

4.1.4 Discussion

As stated in Chapter 2 there is a difference in approach between the decentralized and centralized crowds. In a decentralized crowd agents decide for themselves what they are going to do, for example if they are going to attack a control force member or move towards the goal. A leader's task in the decentralized crowd is to motivate or influence agents around him to behave in the way he wants. In a centralized crowd the leader's and crowd member's behaviours are different. A leader can issue orders, telling an agent to attack or move towards a goal in the world. The expectation is that this difference in approach will cause the crowds to behave in different ways and that a decentralized crowd would be better for crowd simulation.

To validate this three attributes were recoded, the number of shouts, number of thrown objects, and the number of agents. These recorded attributes are aspects which distinguish a demonstration from a normal gathering of people.

Boxplots were generated to visually show the distributions of the recorded attribute for each crowd and scenario. These showed that there seems to be a difference between the two types of crowd. To verify this and to determine if this difference is significant an ANOVA was performed. The results showed that on two of the three attributes the crowds differed significantly with a *p*-value below 0.05, the exception being the number of thrown objects. A normalized sum was also constructed and again the two crowds seen to be significantly different. A possible answer to why the crowds are different is the change in movement of agents. In a centralized crowd leaders will tell where agents should move and if they need to attack control force members or not. Agents can also get into a situation where they will not have a leader around giving orders which stops an agent from moving. Difference in movement causes different local situations for agents and causes agents to flee earlier, attack control forces sooner, or get isolated and not get any orders. These local situations could cause the difference in attribute values as shown by the ANOVA and boxplots.

By looking at the results the conclusion can be made that the two crowds differ from each other on two attributes and on the normalized sum. The decentralized crowd also has higher values for two attributes as can be observed by comparing the means. Because the decentralized crowd scores higher, as expected, this means that it is probably better suited for crowd simulation.

4.2 Usability with Virtual Battlespace 2

This Section describes how usability of the decentralized crowd with VBS2 will be determined. First Section describes the setup with the used scenario. The following Section shows the results of running VBS2 with a decentralized controlled crowd, described in Chapter 2, compared to running VBS2 with its internal agents. The measurements were done by a Dell Latitude 6520 with an i7 Intel processor and an NVidia NVS 4200m with VBS2 1.6.

4.2.1 Setup

To test the usability within VBS2 the number of frames per second were recorded, a high number of frames per seconds means that the applications runs smoothly and the user does not experience stuttering or a slow depiction of the crowd. To measure these frames per seconds an internal function of VBS2 was used. This function records the lowest number of frames per second during simulation and the average number. To compare the two crowds the same kind of scenario is used, agents have gathered in an open area and are issued to move towards a certain point. In both cases VBS2 will provide the same kind of services, such as path finding and collision detection. The difference in frames per second will originate from the way the crowd is controlled. The scenario was run for 8000 steps in MASON and the number of agents starts at 16 and is incremented with 5 agents until the maximum of 86 is reached.

4.2.2 Results

Two kinds of frames per second were recorded, the lowest number and the average number of frames per second. The graph in Figure 39 shows the impact of increasing the number of agents on the lowest number of frames per second. For each type of controlling the agents a trend line was constructed to make the comparison easier and to give a way to predict what the frames per second would be given the number of agents.

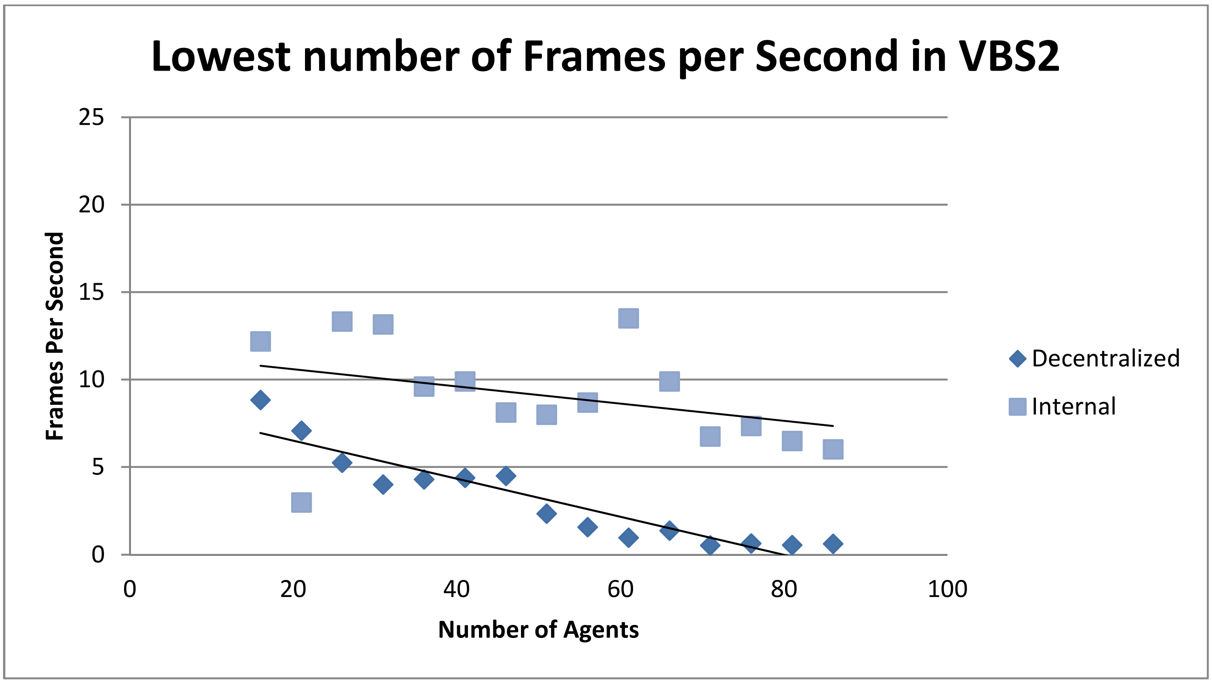

Figure 39 Comparison between internal agents and decentralized agents with lowest number of frames per second

In both cases these trend lines drop when the number of agents increases, as expected because adding agents increases effort VBS2 has to do to render the simulation and the increased difficulty in path finding and collision detection. However the trend line for the decentralized crowd drops fasters. The cause might be the extra functionality which VBS2 performs for the decentralized crowd, like determining the observations of an agent and if he is affected by a NLW. This functionality is not used by the internal agents of VBS2. When these operations are called for all agents at the same time this would result in a O(N^2) operation, where increasing the number of agents has an exponential effect on the performance. Because an scheduler is used the impact is reduced by spreading the calls. This decline in performance is also seen in the formulas which describe the trend lines, displayed below. The variable which denotes the slope of the line for the decentralized crowd is twice as large as for the internal agents. The added functionality is probably why VBS2 has a lower amount of frames per second in case of the decentralized crowd.

Formula lowest number of frames per seconds for the decentralized controlled crowd:

$$minimum\ frames\ per\ second = -0.1088 * number\ of\ agents + 8.6863.$$

Formula lowest number of frames per second for the internal agents:

$$lowest\ number\ of\ frames = -0.0493 * number\ of\ agents + 11.584.$$

The same kind of comparison can also be done for the average number of frames per second, of which the graph is shown in Figure 40.

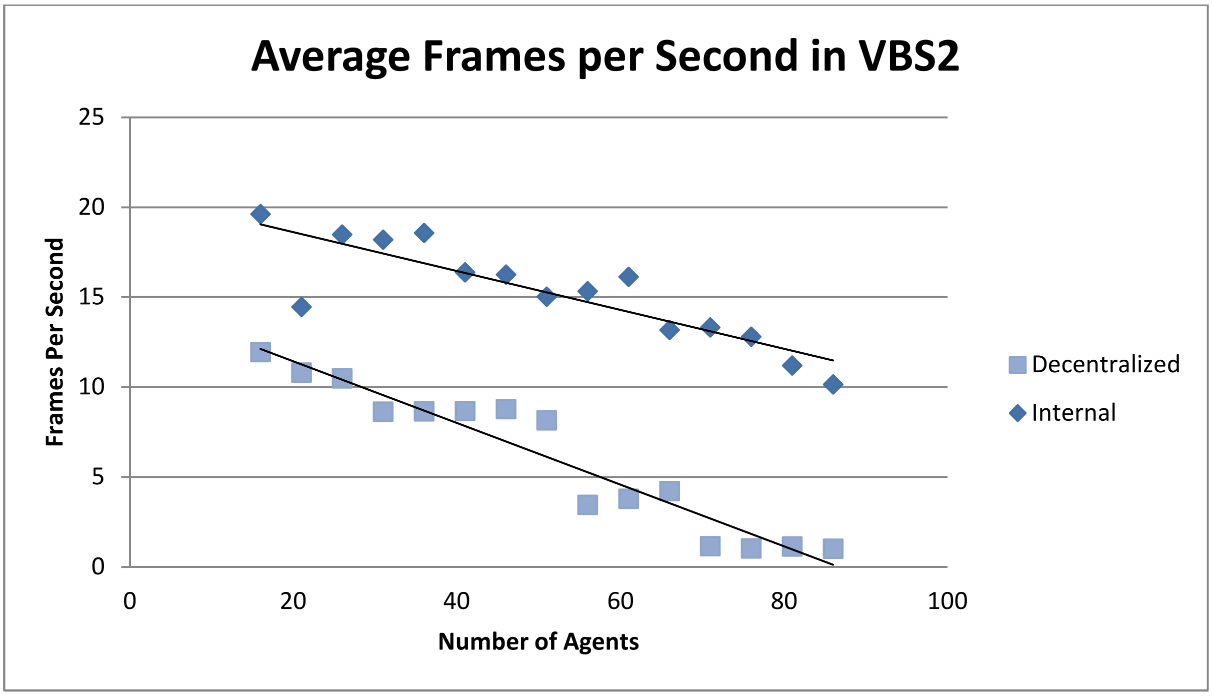

Figure 40 Comparison between internal agents and decentralized agents with average number of frames per second

Again in both types of controlling agents the number of frames per second declines when the number of agents increases. There is also a difference in the number of frames per second, which lies lower in the case of the decentralized crowd. The same explanation can be given as in the lowest number of frames per second, VBS2 has to use extra functionality which is only used in the decentralized crowd. This can also be seen in the formulas which describe the trend lines. The slope in the decentralized crowd is steeper than in the case of the internal agents. The starting position of the internal agents is also higher.

Formula decentralized crowd:

$$average\ frames\ per\ second = -0.1714 * number\ of\ agents + 14.861.$$

Formula internal agents:

$$average\ number\ of\ frames\ per\ second = -0.1081 * number\ of\ agents + 20.779.$$

4.2.3 Discussion

To determine the usability of the implemented decentralized crowd in VBS2 the frames per seconds were measured while running the simulation. Both the lowest and average frames per second were recorded. The choice was made to record these attributes because they indicate how fluent a simulation is. If the frames per seconds is low the user could experience stuttering and a delay between his input and the reaction of the crowd.

A decline in frames per second when the number of agents increases was expected. The trend lines confirm this. Because adding more agents adds complexity such as the increase in number of messages and time spend for agent observation and determining if an agent is affected by an NLW.

This is reflected by the resulting graphs and trend lines which describe what the influence is of the number of agents on the number of frames per second. As expected these trend lines decrease when the number of agents increase. When comparing the trend lines the difference is clear, the impact of increasing the number of agents running with MASON results in a higher impact on the frames per second in comparison to using internal agents. This also accounts for the lowest amount of frames per second.

The explanation is an increase of utility in VBS2 when using MASON to control the agents. VBS2 has a scheduler which periodically sends back messages about which agents an agent can see. This is an O(N) operation and takes time of VBS2 to complete, in worse case this would even be a O(N^2) operation when these calls are done at the same time. Time spend on this functionality is time which VBS2 cannot spend on generating new frames for displaying the simulation meaning that the frames per second will be lower. The impact of MASON and the functionality within VBS2 lowers the fluency of the simulation at about twice the rate in comparison to using internal agents.

When running the simulation 6 to 8 frames per second could still be called fluent. This would result in about 35 to 50 decentralized agents which can be used with VBS2. This might not be enough for a massive demonstration which fills an entire market square, such as in Egypt, but still would give a crowd of demonstrators. Increasing the number of frames per second might be possible by running the simulation on a machine with a better graphical card, the graphical card currently used was meant for netbooks with decent performance and low power usage. Another possibility is to optimize the function used for determining which agents an agent can observe and if an agent is being affected by a NLW. A possible solution could be to check some agent’s visibility from another agent’s point of view, and let MASON give the same visibility score to surrounding agents. Use of the expensive operation, checking visibility, is reduced but there is a high possibility that agents get a wrong visibility score and could be determined visible even if they are really behind a wall. The same technique can be used for determine affected agents. In both cases there is a tradeoff between accurate information and performance.

Trend lines which have been found can also be used to predict what the performance of the decentralized crowd would be, given the number of agents. This gives the opportunity to determine the performance beforehand in a simple way. However the outcome of the formula, which are frames per second, is only accurate on the type of machine on which the results, which form the trend line, is based.

Chapter 5 Conclusion and Discussion

5.1 Conclusion

The goal of this thesis was to research how a crowd can be simulated using decentralized techniques and if this is better suited than a centralized approach. A crowd in this case consists of people who have gathered to demonstrate and are confronted with non-lethal weapons (NLWs) and control forces. The result is a description of how a decentralized crowd can be modelled according to factors taken from literature and if the proposed model would be better suited then a centralized approach. The assumption is that a decentralized crowd is better suited for crowd simulation.

In the beginning of this thesis three research questions were formed. 'How can a crowd of demonstrators be modelled using decentralized techniques?', 'Is a decentralized crowd more realistic than a centralized crowd?' and 'How does the decentralized crowd perform in Virtual Battlespace 2 (VBS2)?'. The first research question was answered using three literature studies. The first was aimed at finding the interactions within a crowd and between the crowd and external sources. The various influences and factors that play a part in crowds and their behaviour formed the basis for mapping external factors, such as NLWs, control forces, subgroups, and other crowd members. These would influence an agent's internal factors as personal risk, resolve, and arousal and his emotional states: calm, fear, panic, and anger. The second literature study was used to determine the actions of control forces and which NLWs they use when confronted with a demonstration. This provided additional information about NLWs and control forces and the influence they have on agents. Such as compellingness and aggressiveness of NLWs and the different stances control forces use, such as warning and show of force, and their effectiveness.

The third literature study was used for finding suitable artificial intelligence paradigms that can be used to model a decentralized crowd. Five paradigms were considered: cellular automata, particle systems, game theoretic, beliefs desires and intentions, and behaviour trees. The choice was made to use behaviour trees because of the ease of use, performance, and it is a new technique which was not yet used for crowd simulation. To combine behaviour trees with the external and internal factors and emotions a suitable cognitive model had to be used. Again there were several different possibilities: the OODA-loop, situation awareness model, risks-as-feelings, conceptual model of an individual, and TNO's own BIN model. The BIN model was used as a basis because it incorporates emotions, control forces, and NLWs and has the ability to interact with other crowd members. The behaviour tree will perform the internal reasoning. The decentralized crowd was implemented using distributed simulation techniques, with MASON handling agents and VBS2 handling the virtual world, drawing of agents, determining visibility, and if an agent was affected by a NLW. Communication between the two applications was done through an IMB message bus and Google Protobuf messages, which enabled MASON to talk to VBS2 without losing or corrupting data. The result is a decentralized crowd model driven by behaviour trees that can interact with NLWs and be used in VBS2.

The second research question: 'Is a decentralized crowd more realistic than a centralized crowd?' was answered by performing an experiment. In this experiment the decentralized crowd was compared to the centralized crowd in two distinct scenarios. The two crowds were identical except for the behaviour tree which controls an agent. The centralized crowd had the addition of giving and receiving orders and choosing of leaders. An agent would select a leader if he is not present or visible and only accept and carry out orders given by this leader. Because of these additions the crowd had a more centralized structure where agents are told what to do and did not decide for themselves. In the decentralized crowd the leaders try to influence their followers to do what they want. The way centralized agents choose their emotion and the impact of the external world on internal variables were the same as with the decentralized crowd. During each scenario three attributes were recorded for each crowd's agent: the number of shouts, the number of thrown objects, and the number of other agents nearby. These attributes are based on real-life observation of demonstrations. With the observed values for each attribute and the 2 x 2 design a two way ANOVA could be performed to determine if the crowds are significantly different from each other. For two of the three attributes this was the case: the number of agents nearby and the amount of shouting. However the amount of thrown objects did

not prove that the crowds were significantly different. These attributes were also combined in a normalized sum. Performing an ANOVA on this sum showed that the two crowds are significantly different from each other. Boxplots were generated to visualize the distributions of these attributes and also showed that both crowds are different, which was supported by the ANOVA. A comparison between the means of the attributes showed that the decentralized crowd had a higher average on the number of agents nearby and the shouting. The expectation was that a decentralized crowd would score higher than a centralized crowd, which was supported by the averages. The higher averages and significant difference between the crowds show that a decentralized crowd is better suited for crowd simulation. However this does not immediately mean that this crowd is more realistic just better suited.
The last research question 'How does the decentralized crowd perform with VBS2?' was answered by measuring the frames per second during the simulation and comparing to VBS2 using its internal agents. When comparing the results the decentralized crowd performed worse than the internal agents of VBS2. This is probably because of the added functionality in VBS2 for determining the visibility of agents and if an agent is affected by a NLW. However by experience, an average of around 6 to 8 frames per second could still be called fluent. This means that the maximum amount of decentralized agents which can be used would be around 50, which is a decent result and shows that the proposed decentralized agent model with behavior trees can be used for crowd simulation.

5.2 Discussion and Future Work

The aim of this thesis was to research crowd behaviour in demonstrations and develop an agent model based on these findings. The resulting crowd should be able to interact with non-lethal weapons (NLWs) and control forces. To research the crowd behaviour and designing the agent model three research questions were defined: "How can a crowd of demonstrators be modelled using decentralized techniques?", "Is a decentralized crowd more realistic than a centralized crowd?", and "How does the decentralized crowd perform in VBS2?". Literature studies were used to find out what factors influence a crowd, what NLWs are commonly used in interaction with demonstrators and what existing AI paradigms could be used to model the crowd. With the resulting factors, NLWs, and suitable AI paradigms an agent model was created and later implemented in MASON and VBS2. The result was a decentralized crowd which could control 35 to 50 agents in VBS2 and where each agent would determine its behaviour on its own. This crowd could be used for training control force members in handling a demonstration or mission planning to see what kind of effect a NLW would have when deployed. At the moment only the teargas and high power microwave (HPM) were implemented and modelled, however there a more NLWs present and it would be recommended to model these to broaden the applicability of the crowd. Besides adding other NLWs it might be better if the control forces receive their own intelligent agents and making them capable to react to a crowd and move around. This would provide a more dynamic simulation and increase usability for mission planning.

As described in Chapter 2 there are a lot of external factors that influence the agent's internal state. The influence is described by using different factors for each situation. At the moment these factors were chosen because they resulted in believable crowd behaviour.

The performed ANOVA gave a good indication if there is a possible significant difference between the decentralized crowd and the centralized crowd based on three observed values (agents in the vicinity, number of shouts, and number of thrown objects) and a normalized sum. Each simulation was executed 10 times, to get a better view of the underlying distribution. However more simulation runs might be required. At the moment the sum is determined by combining normalized versions of the recorded attributes. It could be that the lifelikeness of a crowd is largely determined by the number of shouting. In that case weights can be added to individual attributes and a weighted sum can be constructed. It might be interesting to see if this normalized sum could be used as a fitness function and determine the factors, which describe the impact of external factors on internal variables, using an evolutionary algorithm. These factors are currently used in linear equations but other functions might also be used, such as exponents, logarithmic functions, or sigma functions. It might be worthwhile to compare these against each other and see if which is best for this application. The decentralized crowd scored higher on two out of the three attributes. However this does not mean that this crowd is more realistic. To get confirmation an expert can view the behaviour and determine if it is more realistic.

At the moment behaviour trees are used to model an agent's reasoning. It could also be interesting to determine if there is a difference between a BDI controlled crowd and the designed behaviour tree controlled crowd. The assumption was made that the performance of a behaviour tree controlled crowd would be better; however there is a possibility that this is not the case. A comparison between the two could determine this.

During this thesis it was interesting to see that there are many crowd simulations but very few are done with regard to demonstrations and not one was done using behaviour trees, therefore it is interesting to see that behaviour trees can be used to model crowd simulations and even handle aspects like emotions. At the moment there is only one behaviour tree which describes the behaviour for an agent but there could be a behaviour tree for each type of demonstrator, so one for a young aggressive male but also one for an old lady. Their behaviour is probably different and their factors with transitions between emotional states would also be different. Behaviour trees can also be created using an evolutionary algorithm and might also prove interesting research if a suitable fitness function is found because than an expert only has to validate the outcome and not define the exact behaviour for an agent.

At the moment the decentralized crowd is connected to VBS2 using a message service called IMB. However there might be alternatives which perform as good or even better, or have additional functionality which might be useful. Therefore it might be interesting to test multiple messaging services and compare them to each other and review their applicability this application. Another interesting thing to research is if there are aspects of the decentralized and centralized crowd that can be combined to form a new kind of crowd. It might be possible that over time or in certain situations a crowd would become more centralized.

Appendix A Complete overview formula's

This appendix describes all the formulas used within the crowd model with corresponding factors. These factors are mainly chosen because they provide a nice representation of a crowd when running the simulation. These values for the factors are not directly taken from research but are influenced by it. For example if the literature states that a leader has an high impact on a crowd member than these factors will be higher than the factors of an encounter with a normal crowd member. To validate these values an expert on crowds should look at the behaviour and determine if this is correct and which influences should be increased or decreased.

Interactions between Crowd Members

The following formulas are based on the observation of another crowd member. In this case the emotional state of this other crowd member is panic. Because of the panic state the personal risk of the agent will be increased with a factor 10, because there could be some threat which is not yet observed, and the resolve is lowered with a factor of -5 because the agent feels less safe and secure.
The factor 10 and 5 were chosen because this is an rule which triggers only when the agent sees this panicking agent for the first time. If the panicking agent would still be in view afterwards this rule would not trigger again so a relatively high value could be used which changes the values in a big way. However the values are not big when compared to the maximum value which could be set, i.e. 100. This is because the possible threat is only an assumption and not a fact. The reasoning is that the crewmember is not panicking without any reason. So the personal risk increases because of the unknown threat and the resolve of the agent will decrease. This counts for all the crowd members regardless of the current emotional state. These correspond to the following formulas:

$$personalRisk = oldPersonalRisk + 10 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right),$$

$$resolve = oldResolve - 5 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right).$$

When the crowd member is calm he will respond differently to the other crowd members. When the crowd member sees someone who is angry the personal risk will increase because of the more aggressive situation and his resolve will go down because of the agent feels less secure and safe. This is formula which is used each time the angry agent is in view and stays in view. Large values could not be used directly because the personalRisk and resolve would change rapidly within a few seconds. The impact of seeing an angry person is also deemed to be gradual and not something one would notice immediately. The values 0.2 and -0.2. The formulas are described below:

$$personalRisk = oldPersonalRisk + 0.20 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right),$$

$$resolve = oldResolve - 0.20 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right).$$

When the agent sees someone who is in fear the personal risk will increase because of the possible threat which the current agent might have missed and the resolve will drop again because of the less secure environment. Formula's which correspond to this situation are described below. As with previous formulas, this formula is repeated each time the agent is within viewing range. Again factors are chosen which do not cause a very large immediate impact on the resulting values. However the impact of seeing someone who is in fear is higher than seeing someone who is angry. The formulas are described below:

$$personalRisk = oldPersonalRisk + 0.5 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right),$$

$$resolve = oldResolve - 0.5 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right).$$

When the crowd member is in fear he will respond differently to the other agents. For example when he meets another crowd member who is angry his personal risk will increase by the formula given below. This is because of threatening situation around him. The factor of 1.0 is chosen because the impact of seeing someone when being fearful is larger than seeing the same person when being calm. This is because the agent is already afraid for something and seeing somebody who is angry increases the threat because the angry agent might be the cause. The formula is described below:

$$personalRisk = oldPersonalRisk + 1.0 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right).$$

But if the same fearful agent sees someone who is calm he might reason that he is being fearful for no good reason. So his personal risk will decrease by the following formula. The effect on personal risk is relatively small because it would be hard for someone who is fearful to shake off this fear. This explains the -0.05. The formula is described below:

$$personalRisk = oldPersonalRisk - 0.05 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right).$$

But if the same agent sees someone who is fearful this will increase his personal risk because the situation which caused the fear in the first place could still be on-going or there might be a new risk which this agent has not yet seen. The personal risk will change by the following formula. This also means having a larger factor to increase personal risk. The formula is described below:

$$personalRisk = oldPersonalRisk + 1 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right).$$

When a crowd member is angry and he sees another agent who is calm than his arousal will drop slightly. This is because the calm surroundings calm him down and make him feel less aroused. This factor is larger than the factor for the other way round. It is assumed that it is difficult for someone who is angry to stay angry when surrounded by calm people, while calm people will get influenced by the angry people. The formula is described below:

$$arousal = oldArousal - 5 * \left(1 - \frac{distanceToCrowdMember}{viewRange}\right).$$

Interactions between Subgroup Members

Crowd members can belong to a specific subgroup. If an agent encounters a subgroup member he will react differently to him in comparison to a crowd member who does not belong to it. When the crowd member is calm and he sees a subgroup member who is angry, his resolve increases because of the knowledge that he is among likeminded friends. He will become more aroused because of his angry friends who probably have good reason to be angry.

The value 0.25 for both formulas was chosen to make the agent more likely to get angry when confronted with angry subgroup members. These formulas also negate the effect of seeing a crowd member who is angry, because both formulas would trigger because a subgroup member is also a crowd member. The formulas are described below:

$$resolve = oldResolve + 0.25 * \left(1 - \frac{distanceToSubgroupMember}{viewRange}\right),$$

$$arousal = oldArousal + 0.25 * \left(1 - \frac{distanceToSubgroupMember}{viewRange}\right).$$

When an angry subgroup member sees a subgroup member who is calm he will become slightly less aroused because of the calming influence and his resolve will go up because he is among friends. The factors 0.15 and -0.15 were chosen to be smaller than the same situation reversed to simulate that subgroups tend to be the source of aggression. Therefore the impact is higher and counters the other influence. The formulas are described below:

$$arousal = oldArousal - 0.15 * \left(1 - \frac{distanceToSubgroupMember}{viewRange}\right),$$

$$resolve = oldResolve + 0.15 * \left(1 - \frac{distanceToSubgroupMember}{viewRange}\right).$$

These formulas together make it possible for a crowd member to adopt the emotional state of the other subgroup members around when they have a majority. For example if there are 4 calm subgroup members and only 1 subgroup member who is angry than the angry subgroup member will change its state to calm over time. The same goes the other way around.

Interactions between Crowd Members and Leaders

Besides interactions between crowd members and their subgroup members there are also interactions between crowd members and leaders when they encounter them in the world.
When a crowd member sees a calming leader who belongs to the same subgroup than the following formulas will be used. The absolute value of the leadership effect is used because the sign of the factor will describe what kind of an effect this will have on the agent. Each type of leader, calming or inciting, has their own factors, which is either positive or negative. This makes editing of the factors easier if a user wants to make the effect of a calming leader for example larger.
Because he feels that there is leader present he will feel more secure. This means that his personal risk will drop given the effectiveness and distance. Besides this reduction in personal risk he will also become less aroused because of the calming nature of the leader. As with the subgroup members his resolve will increase because of the presence of a leader who shares his views. The formula is applied as long as the leader is within viewing range. The factors are large in comparison with encountering crowd members repeatedly. This is because the leader is a trusted source of information and has a large impact on the crowd member. This is an effect of the herd theory (Hamilton 1970). The formulas are described below:

$$personalRisk = oldPersonalRisk - 5 * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right),$$

$$arousal = oldArousal - 10 * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right),$$

$$resolve = oldResolve + 5 * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right).$$

However if the leader is inciting the following formulas will be applied. His personal risk will decrease because of the presence of a leader, even if he is inciting, and the arousal will increase because of the inciting manner. As with the calming leader the resolve will increase because of the presence of subgroup members who think alike. These values are basically the same as the calming leader but the difference lies in the factor of the arousal. This is the effect of incitement which the leader does to the crowd member. He tries to get the crowd member to become angry. This is the exact opposite of the calming leader. The formulas are described below:

$$personalRisk = oldPersonalRisk - 5 * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right),$$

$$arousal = oldArousal + 10 * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right),$$

$$resolve = oldResolve + 5 * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right).$$

An interesting situation is when a crowd member who is fearful encounters a calming leader. Because of the calming presence the fearful crowd member will calm slightly down even when they do not belong to the same subgroup. The personal risk will decrease because of the calmer situation and the resolve will increase because of the calming effect on the crowd member who feels more secure given the situation. The factors are smaller than the normal interaction with leaders because this is basically a stranger. The formulas are described below:

$$personalRisk = oldPersonalRisk - 0.5 * |leadershipEffect| * \left(1 - \frac{distanceToLeader}{viewDistance}\right),$$

$$resolve = oldResolve + 0.5 * \left(1 - \frac{distanceToLeader}{viewDistance}\right).$$

Interactions between Crowd Members and NLWs

A major part of the agent simulation is the interaction between crowd members and NLWs which control forces may deploy. This also has effect on the internal variables of an agent. When a crowd member sees the deployment of a NLW for the first time the following functions will be used. The compellingness and aggressiveness factor are described by (Wetzer and Horst 2011). The compellingness factor of the NLW states how much the agent is forced to comply with the effect of the weapon. Having a rubber bullet fired causes a stronger urge to move away than having calming lights. The aggressiveness factor states how aggressive the weapon looks to the agent. When the NLW looks like a normal standard firearm the aggressiveness will be high. But if the NLW would be pink and fluffy the aggressiveness would be far lower. The personal risk increases because the NLW is a direct risk to this agent and threatens the safety of him. The arousal will drop slightly because the agent has difficulty trying to stay angry when he is confronted directly with NLWs. The formulas are described below:

$$personalRisk = oldPersonalRisk + 5.0 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$arousal = oldArousal - 5.0 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right).$$

There are also different categories of NLWs which may be deployed. These are the direct area clearing, indirect area clearing and barricades. Direct area clearing means that the target of a NLW is a person and the objective is to move that person to someplace else. The indirect area clearing is a larger effect with the intention to clear a specific place of persons. Finally the barricade is meant to deny people access to a location.

The interactions between the different NLWs are based on the initial viewing of the NLW and the repeated exposure to it. When a crowd member sees a direct area clearing NLW for the first time the following formulas will be used.

The direct area clearing NLWs, such as a rubber bullet, pose a severe threat to the agent because of the targeted nature of the NLW. So his personal risk will increase with a large amount, this explains the high factor of 50. His resolve will drop because the situation becomes more threatening and unsecure. So the resolve to stay at the demonstration and keep up will become smaller. Also the arousal will decrease because of the difficulty to maintain the aroused state when confronted with a NLW. The formulas are described below:

$$personalRisk = oldPersonalRisk + 50 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$resolve = oldResolve - 8 * (compellingness + aggressivess) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$pain = oldPain + 10 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$arousal = oldArousal - 4 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right).$$

The indirect area clearing NLWs have the following formulas when the crowd member sees it for the first time. As with the direct area clearing the personal risk will increase. But because the indirect area clearing NLW is less personal and more focused on a group the personal risk will increase in a slighter fashion, this explains the factor of 10. Again the resolve will drop with the same reason as the previous NLW because the confrontation with a NLW causes him to become less determined to continue. The agent also experiences pain or discomfort because of the effect and becomes less aroused. The formulas are described below:

$$personalRisk = oldPersonalRisk + 10 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$resolve = oldResolve - 4 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$pain = oldPain + 2 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$arousal = oldArousal - 2 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right).$$

When the agent stays in the effect of the indirect area clearing NLW the following formulas are applied. His personal risk will increase every second by a factor 10 because of the presence of the smoke or other material which is used and the resolve will keep decreasing because of the threatening situation. Because the effect keeps having influence on the agent the pain will also increase. The factor for the increase of pain is the same as with the direct area clearing NLWs because the effect on the person might be different in real life the experienced discomfort might be similar. The formulas are described below:

$$personalRisk = oldPersonalRisk + 10 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$resolve = oldResolve - 10 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$pain = oldPain + 10 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$arousal = oldArousal - 2.0 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right).$$

The influence of the barricades is significantly lower on a Crowd member in comparison to the other NLWs. When a crowd member is under influence of a barricade for the first time the following formulas are applied. The effect on personal risk is smaller for the agent in comparison to the other NLWs. This is because a barricade does increase the unsecure and risky feeling of the agent it does not have the same impact as seeing a cloud of tear gas move towards you or have rubber bullets fired at you. This is also reflected in the change in resolve and arousal. These effects on the agent are lower in comparison to the other NLWs. The formulas are described below:

$$personalRisk = oldPersonalRisk + 2.5 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$resolve = oldResolve - 0.5 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$arousal = oldArousal - 0.5 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right).$$

When the influence of the barricade continues the following formulas will be applied. The personal risk will increase because of the more threatening situation. The resolve will go down as will the arousal because of the less secure environment and the difficulty to stay aroused when affected by the barricade. The formulas are described below:

$$personalRisk = oldPersonalRisk + 15 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$resolve = oldResolve - 15 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right),$$

$$arousal = oldArousal - 4 * (compellingness + aggressiveness) * \left(1 - \frac{distanceToNLW}{nlwEffectRange}\right).$$

Interactions between Crowd Members and Control Forces

Besides interactions between crowd members, with leaders and NLWs; crowd members will also interact with control force members who are present.
Control forces have three stances they can employ. These are show of force, warning, and hailing. In this case the hailing is taken out of account because the situation in the demonstration is beyond this type of action. Control forces will only warn or use show of force.
Each stance of the control force will have a different effect on a crowd member based on their emotional state. The effectiveness denotes how effective the control forces are. If this value is 1 they are very disciplined and capable, while a value of 0 means that they are incapable and chaotic.
In some functions the Section (1 – effectiveness) is used. This increases the effect on the arousal when the control forces are very ineffective. There is also difference between seeing the action repeating or once.
First the formulas will be given for show of force and a crowd member who is calm and who sees this for the first time. Show of force has a relatively high impact on people because of the psychological effect of seeing trained control forces who are ready for handling escalating behaviour. This will calm people down because they know that they are outgunned. The show of force has a threatening influence on the agent. This means that the personal risk increases with a factor of 10 because the agent sees what the control forces are capable of and sees this as a risk to him. This also has an influence on the arousal with a factor of -5 because the agent will calm down under the presence of the control forces. The corresponding formulas are described below:

$$personalRisk = oldPersonalRisk + 10 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$arousal = oldArousal + 5 * (1 - effectiveness) * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve - 7.5 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

When a crowd member, who is calm, sees show of force in a continuous fashion than the following formulas will be applied. The factors are lower than the first time sighting because the agent is used to seeing the control forces like this and the effect of show of force diminishes greatly when seeing it in a continuous fashion. This explains the lower factors in comparison with the first time formula. The formulas are described below:

$$personalRisk = oldPersonalRisk + 0.5 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve - 0.05 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

If the crowd member is in fear the impact will be larger. The following functions will be used when he sees the show of force for the first time. Because of the fear and factors are larger than the calm case. The formulas are described below:

$$personalRisk = oldPersonalRisk + 15 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve - 10 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

When the effect is continuous while the agent is in fear than the following functions are applied, these factors are larger than the same situation where the agent is calm because the fear state increases the impact the show of force has on the agent. The agent is fearful and therefore sees more threat. The formulas are described below:

$$personalRisk = oldPersonalRisk + 1 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve - 1 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

When a crowd member is angry, the show of force will have a different effect on him. The functions which describe this are shown below. The factors are lower because the angry state of the agent causes him to overestimate himself and not see the same kind of risks when he would be calm or fearful. The formulas are described below:

$$arousal = oldArousal + 3 * (1 - effectiveness) * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$personalRisk = oldPersonalRisk + 7.5 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve - 5 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

When the show of force continues the following formulas are applied. Again these factors are lower when compared to the calm and fearful cases because of the same reason as above. The formulas are described below:

$$arousal = oldArousal - 0.05 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$personalRisk = oldPersonalRisk + 0.3 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

The other stance the control forces can take is the stance of warning. A warning is given to the crowd members and that they should stop their actions or face the deployment of an NLW. This effect is only one time and not continuous because a warning only works when it is given once and repeating it will not lead to different behaviour or sudden realization. The impact of the warning stance is larger than the show of force because this is usually followed by a NLW, which is a significant threat. The expectation is that control forces will not bluff about deploying NLWs because this would diminish the effect of warning over time.
Again, as with most formulas, the effect of this stance depends on the emotional state of a crowd member. Following functions can be applied to a crowd member who is confronted with a warning when he is calm. The formulas are described below:

$$personalRisk = oldPersonalRisk + 20 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$arousal = oldArousal + 10 * (1 - effectiveness) * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve - 30 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

When the crowd member is fearful the following formulas are applied. Again the factors are larger than in the calm case because the fearful state causes the agent to perceive a higher threat. These formulas are described below:

$$personalRisk = oldPersonalRisk + 30 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve - 30 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$arousal = oldArousal - 20 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

When a crowd member is angry a new set of functions is defined. The factors are the same as in the fear state because the angry agent senses that what he is doing might trigger a NLW which he does not want. Therefore he needs to be calm quickly to avoid this from happening. The formulas are described below:

$$personalRisk = oldPersonalRisk + 30 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$resolve = oldResolve - 30 * effectiveness * \left(1 - \frac{distanceToControlForce}{viewRange}\right),$$

$$arousal = oldArousal - 20 * (1 - effectiveness) * \left(1 - \frac{distanceToControlForce}{viewRange}\right).$$

Appendix B Nodes Decentralized Behaviour Tree

Name	Type	Description
AvoidPosition	Action	This node has an coordinate as argument which represents the place this agent should avoid. The agent will try to calculate his angle to this coordinate and invert it and move in this inverted angles direction. The action succeeds when the agent has moved away and is in a safe location.
MoveToPosition	Action	This node has a coordinate as argument which represents the position the agent should move towards. The action succeeds if the agent is with 1 unit of the desired position.
Shout	Action	The agent initiates a shout action.
ThrowStone	Action	The agent initiates the throwing of a stone.
ThrowMolotov	Action	The agent initiates the throwing of a Molotov cocktail.
SubgroupMemberClose	Condition	This condition checks if a subgroupmember of the current agent is close by. Close by means in this case that the other agent is visible by the agent. The position of the other agent is put on the context.
SubgroupLeaderClose	Condition	The same kind of condition as the SubgroupMemberClose but instead of looking at crowd members this condition will check at leaders in the subgroup. The position of the leader is put on the context.
isNLWPresent	Condition	This condition checks if the agent can see a NLW. The position of the NLW is put on the context.
isAngry	Condition	This condition checks if the current agent is in the Angry emotional state.
IsCalm	Condition	This condition checks if the current agent is in the Calm emotional state.
isFear	Condition	This condition checks if the current agent is in the Fear emotional state.
isPanic	Condition	This condition checks if the current agent is in the Panic emotional state.
isControlForceClose	Condition	This condition checks if a control force member is visible for the agent. The position of the control force member will be put on the context.
isCrowdMemberClose	Condition	This condition checks if another crowd member is within visible range for this agent. The position of the other crowd member is put on the context.
isArmed	Condition	This condition checks if the current agent is armed or not.
isGoalPresent	Condition	This condition checks if a goal is present in the world in which the agent is currently staying. The goal's position is put on the context.
isGoalClose	Condition	This condition is very similar to the isGoalPresent condition but in this case the condition check if the goal is within viewing distance of the agent.
isNLWEffect	Condition	This condition checks if the current agent is being affected by a NLW. This usually means that the agent is within the effective radius of the NLW. The position of the NLW who is affecting this agent is put on the context.
isInciting	Condition	This condition is specially meant for the leader who can have this state. This condition checks if the current agent wants to incite other crowd members or not.
isCalming	Condition	This condition is the same as the isInciting node but checks if the leader is trying to calm his subgroupmembers or not.

Appendix C Nodes Centralized Behaviour Trees

Name	Type	Description
issueMoveToOrder	Action	This action sends a moveTo order to all the agents within the currents agents within range and who have the current agent as an leader. The position which the other agents should move to is send with the order.
isueAvoidToOrder	Action	This action sends an avoid order to all agents within range who have the current agent as an leader.The position to avoid is send along with the order.
issueAttackToOrder	Action	The same as with the issueMoveToOrder and the issueAvoidToOrder. This order is send to all visible agents with this agent as a leader.
appointLeader	Action	This action looks at all the crowd members who are visible for the current agent. The agent who has the lowest ID is appointed by this agent as being his appointedLeader. The other agent is notified that he has become a appointedLeader.
hasMoveToOrder	Condition	This condition checks if the current agent has received a move to order. If this is the case the order is removed from the list the agent carries and the coordinate which was attached to the order is placed on the context.
hasAvoidOrder	Condition	The same as the move to order but used for avoid orders.
hasAttackOrder	Condition	The same as the move and avoid orders but used for attack orders.
hasAppointedLeader	Condition	This condition checks if the current agent has an appointed leader or not.
isAppointedLeader	Condition	This condition checks if the current agent has been appointed to be a leader by another agent.
isAppointedLeaderVisible	Condition	This condition checks if the appointed leader this agent has is still within visible range for this agent. Usually when this is not the case a new leader will be appointed.

Appendix D Google Protobuf Messages MASON – Virtual Battlespace 2

Name	Parameters	Description
AgentPosition	Id Computerid Type X Y	This message is returned by VBS2 telling where the agent given by the ID is located in the world.
InitialPosition	Id Computerid Type X Y Viewdistance	This message is send by MASON and tells where the agent should be put in VBS2. It also specifies the distance this agent can see.
MoveToPosition	Id Computerid X Y Speed	Tells VBS2 where an agent should move towards with a certain speed. This to let the agent either walk or run.
Shout	Id Computerid	Tells VBS2 that this agent is shouting.
ThrowRock	Id Computerid X Y	Tells VBS2 that this agent is throwing a rock and at which location this rock will be targeted.
ThrowMolotov	Id Computerid X Y	Tells VBS2 that this agent is throwing a Molotov and at which location this Molotov will be targeted.
NLWStart	Id Type Radius X Y Targetx Target	Tells VBS2 that a NLW is deployed. The type and the effective radius are given besides the position in the world. If the non-lethal has a target such as the high power microwave or watercannon than this target is also specified.
NLWStop	Id	Telling VBS2 that a NLW has stopped.
ChangeEmotionalState	Id Emotion	Tells VBS2 that this agent has changed its emotional state and which is his new emotional state.
LeaderBehaviour	Id Behaviour	Tells VBS2 what kind of leader this is. Is he inciting or calming.
AgentsVisible	Id Agents	Tells MASON how many agents and which agents the agent can see.
AgentsAffected	Id NLWs	Tells MASON which NLWs are affecting this agent.
UpdateNLWPosition	Id X Y	Tells MASON what the new target of the NLW is.
InfluencedByLeader	Id Leaderid	Tells VBS2 that this agent is currently being influenced by this leader.
Command	CommandValue	Tells VBS2 that MASON has started or stopped.

Appendix E Google Protobuf Messages Scheduler

Name	Parameters	Description
Schedule	Id Frequency Phase type	This message tells VBS2 that MASON is interested in certain aspects of an agent. This can either be the line of sight, if he is affected by a NLW and what his position in the world is. The frequency states the space between two messages. The phase gives the offset in relation to the other schedule messages.

Appendix F Experiment Results Boxplots

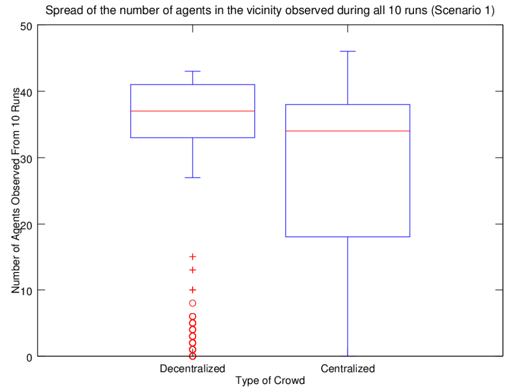

Figure 41. Boxplot for scenario 1 and the number of agents close.

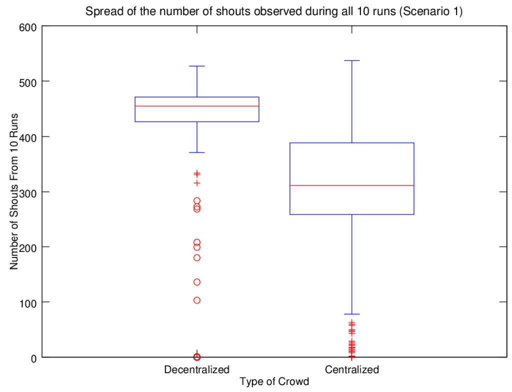

Figure 42. Boxplot for scenario 1 and the number of shouts.

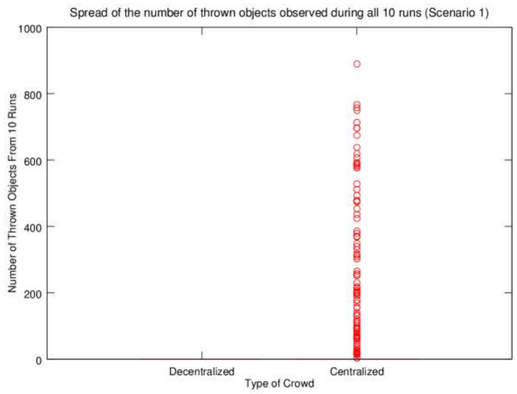

Figure 43. Boxplot for scenario 1 and the number of thrown objects.

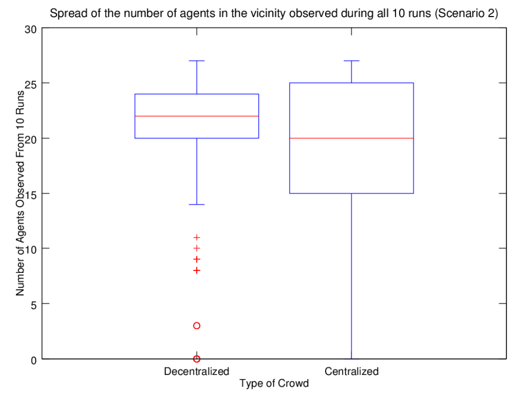

Figure 44. Boxplot for scenario 3 and the number of agents close.

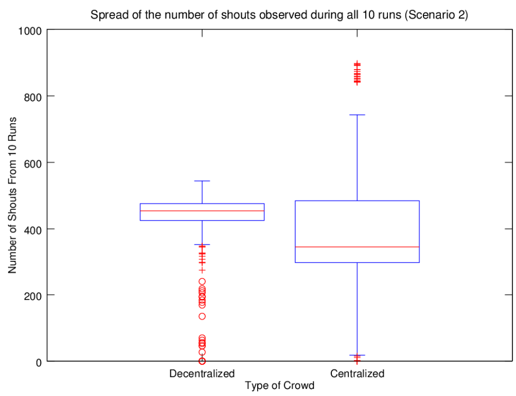

Figure 45. Boxplot for scenario 3 and the number of shouts.

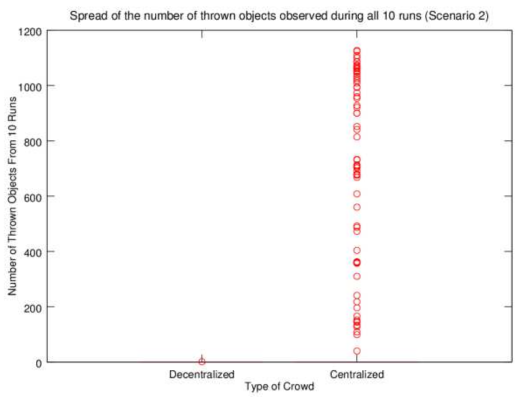

Figure 46. Boxplot for scenario 3 and the number of thrown objects.

Appendix G Experiment Results Comparison of Means

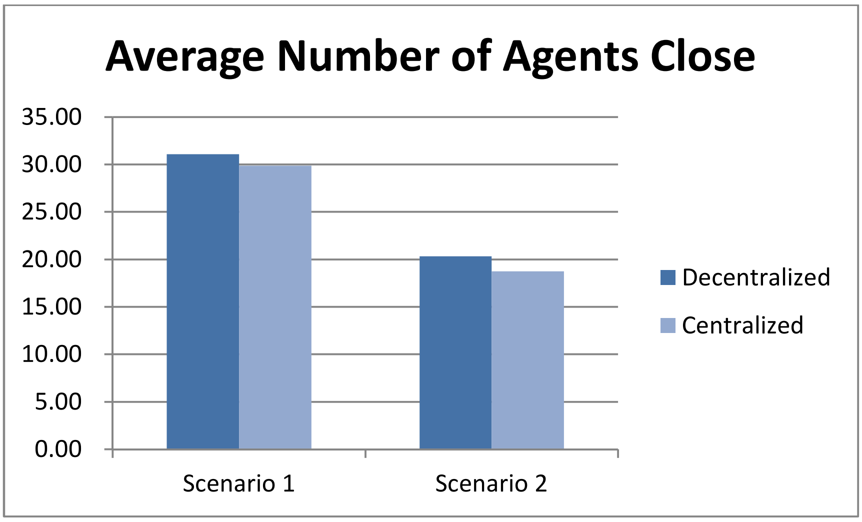

Figure 47. Comparison of means of number of agents close.

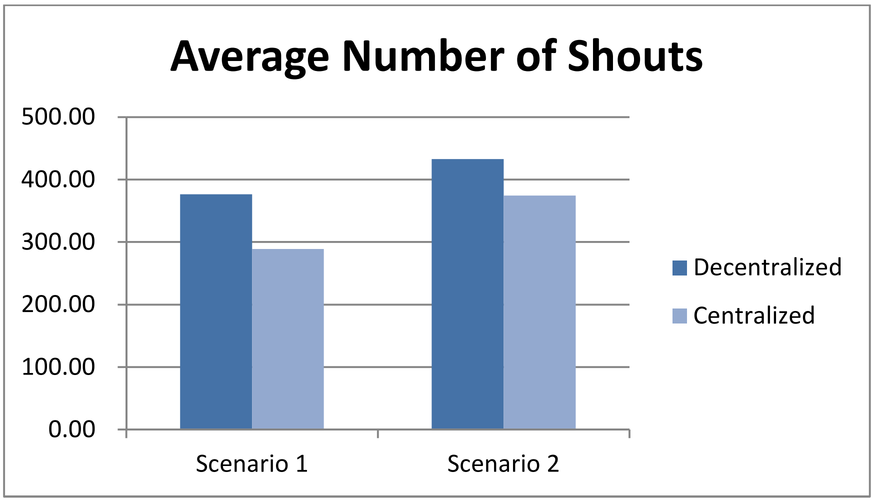

Figure 48. Comparison of means of number of shouts.

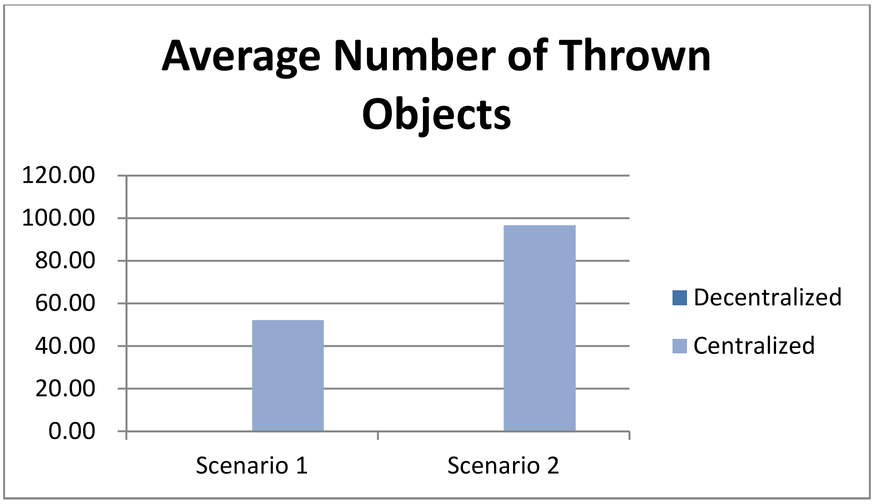

Figure 49. Comparison of means of number of thrown objects.

Appendix H Experiment Results ANOVA

Table 12. Two way ANOVA results for number of agents close.

Source of variation	Sum. Sqr.	MeanSS	Fval	P-Value
Crowd	10080.62	10080.62	124.802	0.000000
Scenario	877.97	877.97	10.870	0.001012
Crowd x Scenario	51.53	51.53	0.638	0.424645

Table 13. Two way ANOVA results for number of shouts.

Source of variation	Sum. Sqr.	MeanSS	Fval	P-Value
Crowd	8373897.08	8373897.08	234.830	0.000000
Scenario	1424232.12	1424232.12	39.940	0.000000
Crowd x Scenario	251.00	251.00	0.007	0.933155

Table 14. Two way ANOVA results for number of thrown objects.

Source of variation	Sum. Sqr.	MeanSS	Fval	P-Value
Crowd	152497.80	152497.80	2.604	0.106905
Scenario	2157881.21	2157881.21	36.849	0.000000
Crowd x Scenario	54037.20	54037.20	0.923	0.336986

Table 15. Two way ANOVA results for the normalized weighted sum of the three attributes.

Source of variation	Sum. Sqr.	MeanSS	Fval	P-Value
Crowd	9.66	9.66	77.265	0.000000
Scenario	1.04	1.04	8.318	0.003968
Crowd x Scenario	0.52	0.52	4.149	0.041796

Appendix I Experiment Results Frames per Seconds

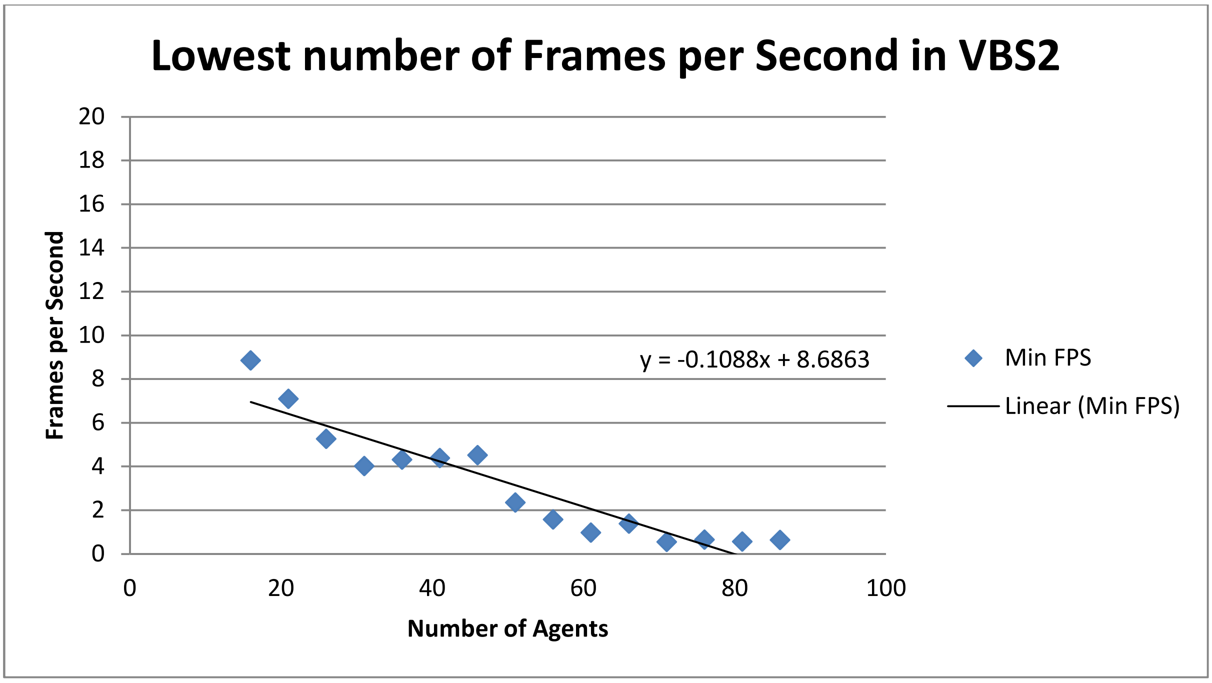

Figure 50. Lowest number of frames in VBS2 versus number of agents.

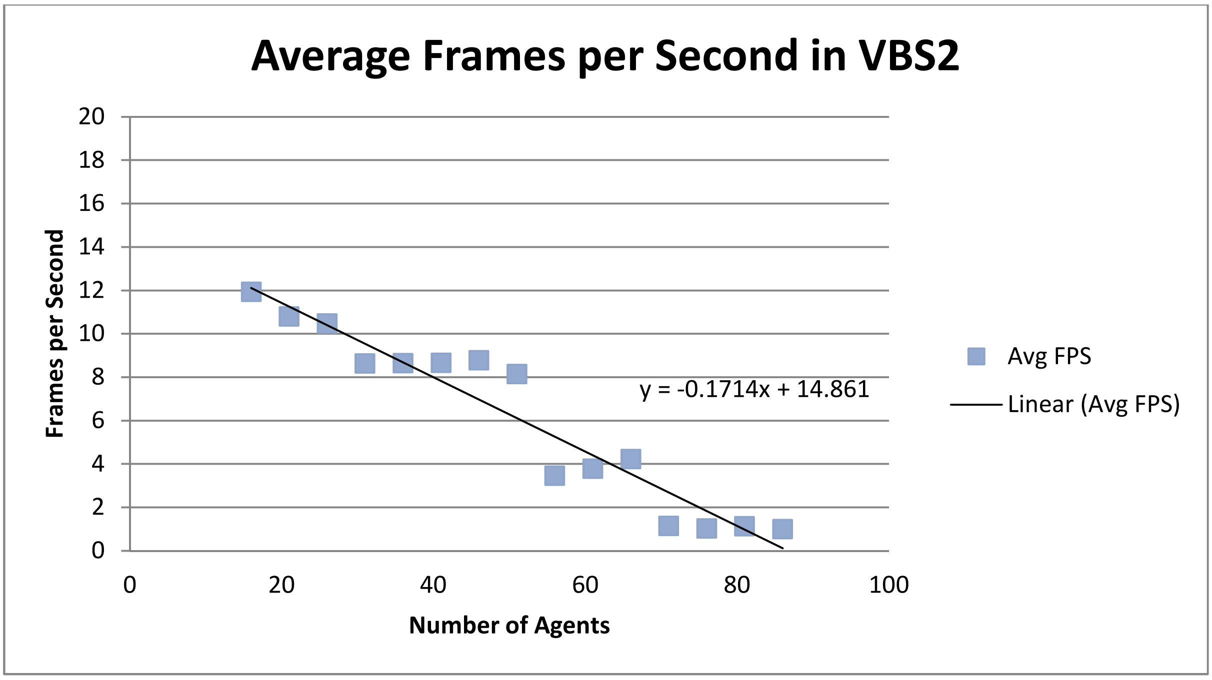

Figure 51. Average number of frames in VBS2 versus number of agents.

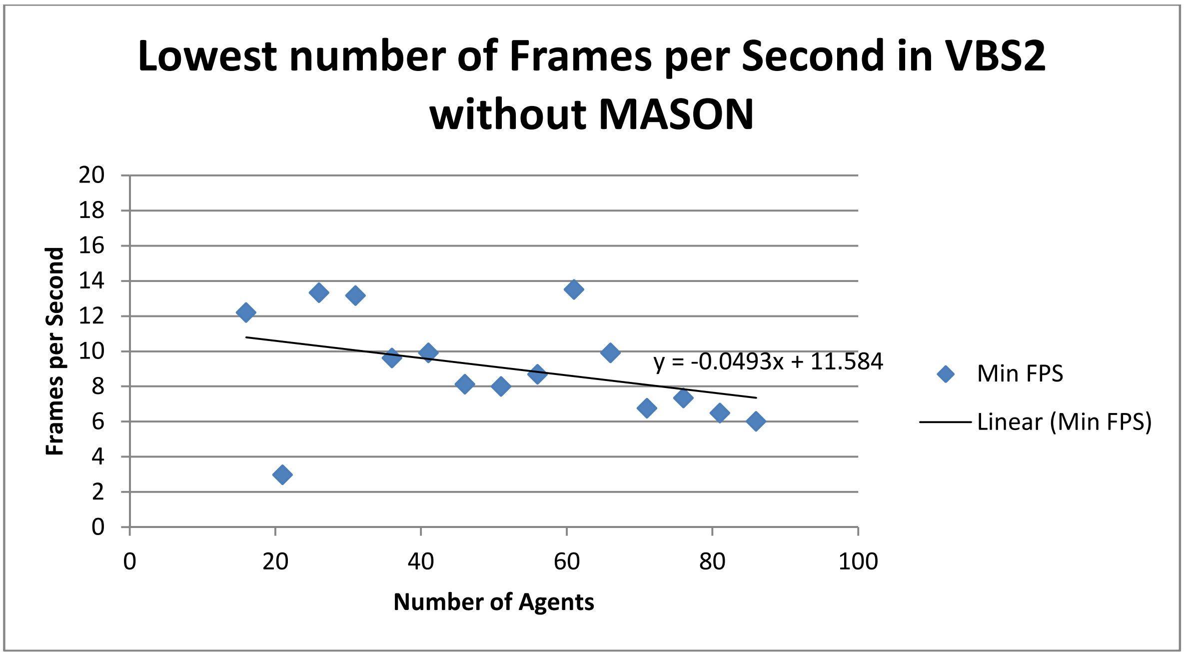

Figure 52. Lowest number of frames in VBS2 without MASON versus number of agents.

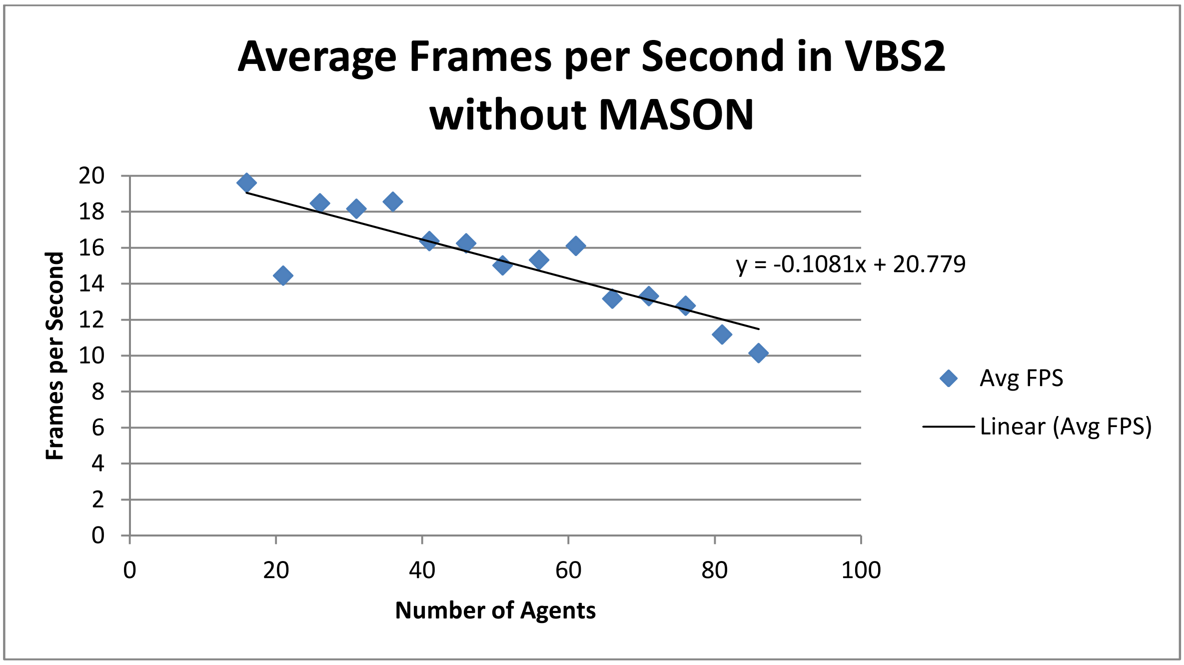

Figure 53. Average number of frames in VBS2 without MASON versus number of agents.

Appendix J Octave Script

```
% Octave script for generating the boxplots and running the ANOVA

% Go to the correct location where the results are located
% cd 'C:\JCBoon\SVN\afstuderen\TNO\Project\Experiment\Postprocessor\log'

% Read the .csv file and store it to a mat file and into the dataset_run1
dataset_run1 = csvread('.\combined-run1.csv');
dataset_run2 = csvread('.\combined-run2.csv');
dataset_run3 = csvread('.\combined-run3.csv');
dataset_run4 = csvread('.\combined-run4.csv');
dataset_run5 = csvread('.\combined-run5.csv');

%Get rid of the first line because Octave cannot handle this and these describe the columns by name.
dataset_run1 = dataset_run1(2:end,:);
dataset_run2 = dataset_run2(2:end,:);
dataset_run3 = dataset_run3(2:end,:);
dataset_run4 = dataset_run4(2:end,:);
dataset_run5 = dataset_run5(2:end,:);

% Load the dataset
% load ./octave_results_run1.mat

% Go to the correct folder to store the graphs and other interesting things
cd '.\';

%%%%%%%%%% RUN 1 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run1(:,1) == 1);
rowsDecentralized = find(dataset_run1(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run1(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run1(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run1(1:end,1),dataset_run1(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run1(1:end,3)];

%Construct the matrix for the number of shouts
```

```
anova_data_shouts = [dataset_run1(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run1(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run1 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run1 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run1 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 1");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run1.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 1");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run1.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 1");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run1.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 1");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run1.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 1");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run1.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 1");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run1.png;

%%%%%%%%%% RUN 2 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
```

```
rowsCentralized = find(dataset_run2(:,1) == 1);
rowsDecentralized = find(dataset_run2(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run2(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run2(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run2(1:end,1),dataset_run2(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run2(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run2(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run2(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run2 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run2 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run2 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the
centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 2");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run2.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 2");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run2.png;

%Thrown Objects
```

```
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 2");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run2.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 2");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run2.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 2");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run2.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 2");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run2.png;

%%%%%%%%%% RUN 3 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run3(:,1) == 1);
rowsDecentralized = find(dataset_run3(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run3(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run3(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run3(1:end,1),dataset_run3(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run3(1:end,3)];
```

```
%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run3(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run3(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run3 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run3 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run3 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 3");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run3.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 3");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run3.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 3");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run3.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 3");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run3.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 3");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run3.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 3");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run3.png;

%%%%%%%%%% RUN 4 %%%%%%%%%%
```

```
% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run4(:,1) == 1);
rowsDecentralized = find(dataset_run4(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run4(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run4(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run4(1:end,1),dataset_run4(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run4(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run4(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run4(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run4 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run4 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run4 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the
centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 4");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run4.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 4");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run4.png;
```

```
%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 4");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run4.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 4");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run4.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 4");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run4.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 4");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run4.png;

%%%%%%%%%% RUN 5 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run5(:,1) == 1);
rowsDecentralized = find(dataset_run5(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run5(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run5(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run5(1:end,1),dataset_run5(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].
```

```
%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run5(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run5(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run5(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run5 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run5 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run5 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 5");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run5.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 5");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run5.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 5");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run5.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 5");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run5.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 5");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run5.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 5");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run5.png;
```

```
%%%%%%%%%% RUN 6 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run6(:,1) == 1);
rowsDecentralized = find(dataset_run6(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run6(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run6(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run6(1:end,1),dataset_run6(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run6(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run6(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run6(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run6 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run6 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run6 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the
centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 6");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run6.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 6");
axis ([0,3]);
```

```
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run6.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 6");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run6.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 6");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run6.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 6");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run6.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 6");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run6.png;

%%%%%%%%%% RUN 7 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run7(:,1) == 1);
rowsDecentralized = find(dataset_run7(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run7(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run7(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run7(1:end,1),dataset_run7(1:end,2)];
```

```
%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run7(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run7(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run7(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run7 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run7 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run7 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 7");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run7.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 7");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run7.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 7");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run7.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 7");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run7.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 7");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run7.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 7");
axis ([0,3]);
```

```
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run7.png;

%%%%%%%%%% RUN 8 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run8(:,1) == 1);
rowsDecentralized = find(dataset_run8(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run8(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run8(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run8(1:end,1),dataset_run8(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run8(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run8(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run8(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run8 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run8 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run8 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the
centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 8");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run8.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
```

```
title ("Shouts - Scenario 1 - Run 8");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run8.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 8");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run8.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 8");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run8.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 8");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run8.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 8");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run8.png;

%%%%%%%%%% RUN 9 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run9(:,1) == 1);
rowsDecentralized = find(dataset_run9(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run9(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run9(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
```

```
groups = [dataset_run9(1:end,1),dataset_run9(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run9(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run9(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run9(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run9 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run9 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run9 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the
centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 9");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run9.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 9");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run9.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 9");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run9.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 9");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run9.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 9");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run9.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
```

```
title ("Thrown Objects - Scenario 3 - Run 9");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run9.png;

%%%%%%%%%% RUN 10 %%%%%%%%%%

% Find all entries related to the Decentralized Crowd, the indexing starts with 1
rowsCentralized = find(dataset_run10(:,1) == 1);
rowsDecentralized = find(dataset_run10(:,1) == 2);

%Retrieve the data and store it in the correct datasets
% x([1, 3, 4]) <-- selection by row index
% a(1, [1, 2])  # row 1, columns 1 and 2
dataset_centralized = dataset_run10(rowsCentralized, [2,3,4,5]);
dataset_decentralized = dataset_run10(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);

dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Define the groups for the anova
groups = [dataset_run10(1:end,1),dataset_run10(1:end,2)];

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [dataset_run10(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [dataset_run10(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [dataset_run10(1:end,5)];

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose_run10 = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts_run10 = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects_run10 = anovan(anova_data_thrownobjects, groups, 'model', 'full');

%Generate the boxplots for the first run where the boxplot is created for one scenario which compares the decentralized vs the
centralized

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Agents Close - Scenario 1 - Run 10");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_run10.png;
```

```
%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Shouts - Scenario 1 - Run 10");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_run10.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Thrown Objects - Scenario 1 - Run 10");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_run10.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Agents Close - Scenario 3 - Run 10");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_run10.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Shouts - Scenario 3 - Run 10");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_run10.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Thrown Objects - Scenario 3 - Run 10");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_run10.png;

%%%% All runs combined %%%%

%Combine the original datasets for each run together
combined_dataset = vertcat(dataset_run1, dataset_run2, dataset_run3, dataset_run4, dataset_run5, dataset_run6, dataset_run7,
dataset_run8, dataset_run9, dataset_run10);

groups = [combined_dataset(1:end,1),combined_dataset(1:end,2)];

rowsCentralized = find(combined_dataset(:,1) == 1);
rowsDecentralized = find(combined_dataset(:,1) == 2);

dataset_centralized = combined_dataset(rowsCentralized, [2,3,4,5]);
dataset_decentralized = combined_dataset(rowsDecentralized, [2,3,4,5]);

% Find the row indexes for the different scenario's for each kind of crowd
rowsCentralizedScenario1 = find(dataset_centralized(:,1) == 1);
rowsCentralizedScenario3 = find(dataset_centralized(:,1) == 3);

rowsDecentralizedScenario1 = find(dataset_decentralized(:,1) == 1);
rowsDecentralizedScenario3 = find(dataset_decentralized(:,1) == 3);

% Construct the datasets based on this information
dataset_centralized_scenario1 = dataset_centralized(rowsCentralizedScenario1, [2,3,4]);
dataset_centralized_scenario3 = dataset_centralized(rowsCentralizedScenario3, [2,3,4]);
```

```
dataset_decentralized_scenario1 = dataset_decentralized(rowsDecentralizedScenario1, [2,3,4]);
dataset_decentralized_scenario3 = dataset_decentralized(rowsDecentralizedScenario3, [2,3,4]);

%Alternatively, we can concatenate A and B along the second dimension the following way: [A, B].

%Construct the matrix for the number of agents close
anova_data_agentsclose = [combined_dataset(1:end,3)];

%Construct the matrix for the number of shouts
anova_data_shouts = [combined_dataset(1:end,4)];

%Construct the matrix for the number of thrown objects
anova_data_thrownobjects = [combined_dataset(1:end,5)];

%Construct the matrix for the weighted sum of all three attributes combined. Probably best to use normalized values in all cases.
%So find the maximum value each attribute and divide the others by this maximum value. Then combine them together by adding them together.

max_agentsclose = max(anova_data_agentsclose);
max_shouts = max(anova_data_shouts);
max_thrownobjects = max(anova_data_thrownobjects);

%Make all the attributes range from 0 to 1
anova_normalized_data_agentsclose = anova_data_agentsclose / max_agentsclose;
anova_normalized_data_shouts = anova_data_shouts / max_shouts;
anova_normalized_data_thrownobjects = anova_data_thrownobjects / max_thrownobjects;

%Weighted sum where all attributes have the same impact, ie. 1/3
anova_data_normalized = anova_normalized_data_agentsclose + anova_normalized_data_shouts +
anova_normalized_data_thrownobjects;

%Do the anova with N ways on the data from the agents close and the groups, it gets division by zeros
anova_results_agentsclose = anovan(anova_data_agentsclose, groups, 'model', 'full');
anova_results_shouts = anovan(anova_data_shouts, groups, 'model', 'full');
anova_results_thrownobjects = anovan(anova_data_thrownobjects, groups, 'model', 'full');
anova_results_weightedsum = anovan(anova_data_normalized, groups, 'model', 'full');

% Calculate the means
decentralized_scenario1_agentclose_mean = mean(dataset_decentralized_scenario1(1:end,1));
decentralized_scenario1_shouts_mean = mean(dataset_decentralized_scenario1(1:end,2));
decentralized_scenario1_thrownobjects_mean = mean(dataset_decentralized_scenario1(1:end,3));

centralized_scenario1_agentsclose_mean = mean(dataset_centralized_scenario1(1:end,1));
centralized_scenario1_shouts_mean = mean(dataset_centralized_scenario1(1:end,2));
centralized_scenario1_thrownobjects_mean = mean(dataset_centralized_scenario1(1:end,3));

decentralized_scenario3_agentclose_mean = mean(dataset_decentralized_scenario3(1:end,1));
decentralized_scenario3_shouts_mean = mean(dataset_decentralized_scenario3(1:end,2));
decentralized_scenario3_thrownobjects_mean = mean(dataset_decentralized_scenario3(1:end,3));

centralized_scenario3_agentsclose_mean = mean(dataset_centralized_scenario3(1:end,1));
centralized_scenario3_shouts_mean = mean(dataset_centralized_scenario3(1:end,2));
centralized_scenario3_thrownobjects_mean = mean(dataset_centralized_scenario3(1:end,3));

%% Scenario 1 %%

%Agents Close
boxplot ({dataset_decentralized_scenario1(:,1),dataset_centralized_scenario1(:,1)});
title ("Spread of the number of agents in the vicinity observed during all 10 runs (Scenario 1)","fontsize",11,'fontweight','bold');
xlabel("Type of Crowd");
ylabel("Number of Agents Observed From 10 Runs");
axis ([0,3]);
```

```
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario1_combined.png;

%Shouts
boxplot ({dataset_decentralized_scenario1(:,2),dataset_centralized_scenario1(:,2)});
title ("Spread of the number of shouts observed during all 10 runs (Scenario 1)","fontsize",11,'fontweight','bold');
xlabel("Type of Crowd");
ylabel("Number of Shouts From 10 Runs");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario1_combined.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario1(:,3),dataset_centralized_scenario1(:,3)});
title ("Spread of the number of thrown objects observed during all 10 runs (Scenario 1)","fontsize",11,'fontweight','bold');
xlabel("Type of Crowd");
ylabel("Number of Thrown Objects From 10 Runs");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario1_combined.png;

%% Scenario 3 %%

%Agents Close
boxplot ({dataset_decentralized_scenario3(:,1),dataset_centralized_scenario3(:,1)});
title ("Spread of the number of agents in the vicinity observed during all 10 runs (Scenario 2)","fontsize",11,'fontweight','bold');
xlabel("Type of Crowd");
ylabel("Number of Agents Observed From 10 Runs");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng agentsclose_scenario3_combined.png;

%Shouts
boxplot ({dataset_decentralized_scenario3(:,2),dataset_centralized_scenario3(:,2)});
title ("Spread of the number of shouts observed during all 10 runs (Scenario 2)","fontsize",11,'fontweight','bold');
xlabel("Type of Crowd");
ylabel("Number of Shouts From 10 Runs");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng shouts_scenario3_combined.png;

%Thrown Objects
boxplot ({dataset_decentralized_scenario3(:,3),dataset_centralized_scenario3(:,3)});
title ("Spread of the number of thrown objects observed during all 10 runs (Scenario 2)","fontsize",11,'fontweight','bold');
xlabel("Type of Crowd");
ylabel("Number of Thrown Objects From 10 Runs");
axis ([0,3]);
tics ("x", 1:2, {"Decentralized"; "Centralized"});
print -dpng thrownobjects_scenario3_combined.png;
```

Bibliography

Algemeen Douanebesluit. Amsterdam: Nederlandse Overheid, sd.

Ambstinstructie voor de politie, de Koninklijke marechaussee en andere opsporingsambtenaren. Amsterdam: Nederlandse Overheid, sd.

Anand, S, Rao, en P, Michael Georgeff. *BDI Agents: From Theory to Practise.* Melbourne: AAAI, 1995.

Bakhtiani, Afsaneh. „Social influences among drug users and mean field approximation of cellular automata." 2009.

Besluit geweldgebruik bij uitoefening buitengewone bevoegdheden. Amsterdam: Nederlandse Overheid, sd.

Besluit geweldgebruik defensiepersoneel in de uitoefening van de bewakings- en beveiligingstaak. Amsterdam: Nederlandse Overheid, sd.

Bewaffnung der Polizei: Information - Begriff - Definition im JuraForum.de. 16 January 2012. http://www.juraforum.de/lexikon/bewaffnung-der-polizei.

Binnendijk, Anika, Locke, en Ivan Marovic. *Power and Persuasion: Nonviolent strategies to influence state security forces in Serbia (2000) and Ukraine (2004).* Elsevier, 2006.

Boyd, J, R. „A discourse of winning and losing." 1987.

Convention on the Prohibiton of the development, production, stockpiling and use of checmical weapons and on their destruction. New York & Paris: Organisation for the Prohibition of Chemical Weapons, 1993.

Conway, John Horton. *The game of life.* Scientific American, 1970.

Couch, Carl J,. *Collective Behaviour: An Examination of Some Stereotypes.* University of California Press, 1968.

Crowd Confrontation Operations. National Defence, 2003.

Delmee, M.H.M., en A.W.G. van Oosterhout. *Verkenning van militaire randvoorwaarden bij het invoeren van niet-letale wapens.* Den Haag: TNO, 2009.

Die Polizeidienstvorschrift für Wasserwerfer (Dokumentation). 1985. http://www.nadir.org/nadir/initiativ/sanis/archiv/cs/kap_08.htm.

Dobias, Peter, Zakia Bouayed, Gerald Woodill, en Steve Bassindale. *Modeling Crowd Behaviour in Militairy Operational Research.* Toronto: Defence Research and Development Canada, 2006.

Driskell, James, E, en Eduardo Salas. „Group Decision Making under Stress." 1991.

Endsley, M.R. *Toward a theory of situation awareness in dynamic systems.* Human Factors: The Journal of the Human Factors, 1995.

Frini, A, L Stemate, S Larochelle, G Toussaint, en R Lecocq. *Crowd Confrontation and Non-Lethal Weapons.* DRDC, 2008.

Hamilton, W.D. *Geometry for the Selfish Herd.* London, 1970.

Helbin, Dirk, en Péter Molnár. *Social force model for pedestrian dynamics.* Stuttgart: Institute of Theoretical Physics, 1998.

Helbing, D, I Farkas, P Molnàr, en T Vicsek. *Simulation of pedestrian crowds in normal and evacuation situations.* 2002.

Hoogendoorn, Serge, en H.,L., Piet Bovy. *Simulation of pedestrian flows by optimal control and differential games.* New Jersey: John Wiley & Sons, 2003.

Hsu, P.L., en Herbert Robbins. *Complete Convergence and the Law of Large Numbers.* 1947.

Huis in 't Veld, M.A.A., I.M. Wetzer, D.A. van Hemert, J.H. Kerstholt, en T. Dotsch. *Behaviour under influence of non-lethal weapons: model development.* Soesterberg: TNO, 2011.

Java Behavior Trees. 2010. http://sourceforge.net/projects/jbt/.

Kenny, John M, et al. *Crowd Behavior, Crowd Control and the Use of Non-Lethal Weapons.* PennState University, 2001.

Klüpfel, Hubert Ludwig. *A Cellular Automaton Model for Crowd Movement and Egress Simulation.* Duisberg: Universiteit Duisburg-Essen, 2003.

Kota, Ramachandra, Vidit Bansal, en Kamalakar Karlapalem. *System Issues in Crowd Simulation using Massively Multi-Agent Systems.* Hyderabad: International Institute of Informatino Technology, 2006.

Kramer, Juergen, en Dieter Rathmann. *Modelling Human Behaviour in the Context of Non-Lethal Effectors.* Friedrichshafen: Cassidian, sd.

Le Bon, Gustav. *The Crowd.* 1897.

Lim, Chong-U. *An A.I. Player for DEFCON: An Evolutionary Approach Using Behavior Trees.* London: Department of Computing Imperial College London, 2009.

Loewenstein, G, F, E, U Weber, C, K Hsee, en N Welch. *Risk as Feelings.* Psychological Bulletin, 2001.

Militairy Police: Internment/Resettlement operations. Washington: Departement of the Army, 2001.

Millington, Ian, en John Funge. *Artificial Intelligence for Games: Second Edition.* Burlington: Morgan Kaufmann, 2009.

Narain, Rahul, Abhinav Golas, Sean Curtis, en C. Ming Lin. „Aggregate Dynamics for Dense Crowd Simulation." Chapel Hill, 2009.

Pillosu, Ricard. *Coordinating Agents with Behavior Trees.* Crytek, sd.

Policy for Non-Lethal Weapons. Washington: Departement of Defense, 1996.

Policy, Procedures and Rules: Use of Force. Brattleboro: Police Departement, 2008.

Regeling mobiele eenheid. Amsterdam: Nederlandse Overheid, 2007.

Reynolds, Craig W. *Flocks, herds and schools: A distributed behavioral model.* New York: ACM, 1987.

Richtlijnen voor gebruik CS-traangas. Amsterdam: Nederlandse Staatscourant, sd.

RTO. „SAS-040: Non-Lethal Weapons and Future Peace Enforcement Operations." 2001.

Shao, Wei, en Demetri Terzopoulos. *Autonomous Pedestrians.* Los Angeles: Computer Animation, 2005.

Uitrustingsregeling politie. Amsterdam: Nederlandse Overheid, 1994.

Use of Taser. Las Vegas: Las Vegas Metropolitan Police Departement, 2004.

Van Dyke Parunak, H., Paul Nielsen, Sven Brueckner, en Rafael Alonso. *Hybrid Multi-Agent Systems: Integrating Swarming and BDI Agents.* Farmington Hills: NewVectors LLC, 2006.

van Vliet, A.J., en M. Fennis-Bregman. *Onderzoek naar Crowd & Riot Control: een overzicht.* Soesterberg: TNO Technische Menskunde, 2004.

Veld, Huis in 't, Kamphuis, en Krabbendam. *Rekenomgeving voor niet-letale wapeneffectiviteit.* Rijswijk: TNO, 2011.

von Neumann, J, en G.W Brown. *Solutions Of Games By Differential Equations.* 1950.

von Neumann, John. *Theory of Self-Reproducing Automata.* Illinois: University of Illinois Press Champaign, 1966.

Wetzer, I.M., W. Kamphuis, D.A. van Hemert, M.A.A. Huis in 't Veld, en J.H. Kerstholt. *Gedrag onder invloed van niet-lethale middelen: twee veld experimenten.* Soesterberg: TNO, 2010.

Wetzer, Inge, en Marike van der Horst. *Niet-letale wapens bij bewaken & beveiligen en openbare ordehandhaving: Een verkenning.* Soesterberg: TNO, 2011.

Wragg, Tony. „Modelling the Effects of Information Campaigns Using Agent-Based Simulation.” Edinbrugh, 2006.

Xiaoping, Zheng, Zhong Tingkuan, en Liu Mengting. *Modeling crowd evacuation of a building based on seven methodological approaches.* Beijing: Elsevier, 2007.

Zhou, Suiping, et al. „Crowd Modeling and Simulation Technologies.” Singapore, 2010.