NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

AGENT-BASED SIMULATION OF GERMAN PEACEKEEPING OPERATIONS FOR UNITS UP TO PLATOON LEVEL

by

Thomas Erlenbruch

March 2002

Thesis Advisor:	Arnold H. Buss
Second Reader:	Gregory K. Mislick

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 2002	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE: Agent-based Simulation of Peacekeeping Operations for Units Up to Platoon level.	**5. FUNDING NUMBERS**
6. AUTHOR(S) Thomas Erlenbruch	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

A significant challenge to the German Military Forces or "Bundeswehr" today is the development of tactics that will enable soldiers, airmen and sailors to be successful in small-scale peacekeeping operations.

This thesis develops an agent-based simulation for modeling peacekeeping operations at the platoon level. The simulation methodology combines agent-based modeling with discrete event simulation in two software packages called *Peacekeeping* and *TryShoot*. These software packages are used to model one part of a map exercise, the Kurzlage PRIZREN. The platoon of peacekeeping soldiers utilizes different kinds of tactics against different kinds of approaches by the civilians to reach their goal. This simulation yields insight into the modeled scenario and demonstrates the usefulness of agent-based simulation for the exploration of tactical concepts in a peacekeeping operation.

14. SUBJECT TERMS Peacekeeping, agent-based modeling, complex adaptive systems, combat model	15. NUMBER OF PAGES 122
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

AGENT-BASED SIMULATION OF GERMAN PEACEKEEPING OPERATIONS FOR UNITS UP TO PLATOON LEVEL

Thomas Erlenbruch
Captain, German Army
Dipl.Ing., Universitaet der Bunderwehr, 1991

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN
OPERATIONS RESEARCH**

from the

**NAVAL POSTGRADUATE SCHOOL
March 2002**

Author: Thomas Erlenbruch

Approved by: Arnold H. Buss, Thesis Advisor

Greg Mislick, Second Reader

James Eagle, Chairman
Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A significant challenge to the German Military Forces or "Bundeswehr" today is the development of tactics that will enable soldiers, airmen and sailors to be successful in small-scale peacekeeping operations.

This thesis develops an agent-based simulation for modeling peacekeeping operations at the platoon level. The simulation methodology combines agent-based modeling with discrete event simulation in two software packages called *Peacekeeping* and *TryShoot*. These software packages are used to model one part of a map exercise, the Kurzlage PRIZREN. The platoon of peacekeeping soldiers utilizes different kinds of tactics against different kinds of approaches by the civilians to reach their goal. This simulation yields insight into the modeled scenario and demonstrates the usefulness of agent-based simulation for the exploration of tactical concepts in a peacekeeping operation.

The analyzed results show that independent of the number of bystanders or the protester's behavior the peacekeepers always get better results in achieving their multiple objectives when they use a defensive tactical approach.

THIS PAGE INTENTIONALLY LEFT BLANK

DISCLAIMER

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the U.S. Department of Defense, the U.S. Government, the German Department of Defense, or the German Government.

The reader is cautioned that computer programs developed in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logic errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
- A. AREA OF RESEARCH 1
- B. MILITARY OPERATIONS OTHER THAN WAR (MOOTW) 2
- C. OBJECTIVES AND RESEARCH QUESTIONS 5
- D. COMPLEX ADAPTIVE SYSTEMS 8
- E. SOFTWARE AGENTS 11
- F. AGENTS IN WARFARE SIMULATION 16
- G. SCOPE OF THE THESIS AND METHODOLOGY 19

II. AGENT MODEL AND IMPLEMENTATION 21
- A. THE AGENT MODEL 21
  - 1. Holland's Agent Model 21
  - 2. Weiss' Belief-Desire-Intention Architecture 23
  - 3. Hiles' Agent Model 28
- B. IMPLEMENTATION CHOICES 31
- C. PEACEKEEPING AND TRYSHOOT STRUCTURE 32
  - 1. Graphical User Interface and Environment Layer 33
  - 2. Simulation Entity Layer 36
  - 3. Agent Modeling Layer 45
  - 4. Agent Intention and Action Generating Layer 47
- D. THE AGENT MODELING PROCESS IN PEACEKEEPING 48
  - 1. Characteristics, Abilities, and Effectors 48
  - 2. Personality 52
  - 3. Detectors 53
  - 4. Goals and Tickets 54

III. THE MAP EXERCISE "PRIZREN" 57
- A. THE MAP EXERCISE 57
  - Task order for 3./ Einsatzbataillon 1 57
  - First situation development until March 01, 2000, 1000 a.m. 60
  - Second situation development until March 2, 2000, 1000 a.m. 60
  - Third situation development until March 2, 2000, 1230 a.m. 61
- B. A GENERALIZED SCENARIO 64
- C. GENERATING THE MODEL 65

IV. SIMULATION RESULTS 69
- A. MEASURE OF EFFECTIVENESS 69
  - 1. Utility Functions 71
  - 2. Deriving the Function for the Measure of Effectiveness 75
- B. DESIGN OF EXPERIMENT 79
- C. REGRESSION MODEL 81
- D. REGRESSION MODEL VALIDATION 83

E. REGRESSION MODEL INTERPRETATION 88
F. REGRESSION MODEL SUMMARY 90
G. IMPLICATIONS FOR PEACEKEEPING TACTICS 91

V. CONCLUSIONS AND RECOMMENDATIONS 93

APPENDIX - - SOFTWARE USED IN THIS THESIS 97

LIST OF REFERENCES 99

INITIAL DISTRIBUTION LIST 101

LIST OF FIGURES

Figure 1. PKO-Agent model depending on Holland's reactive agent 22
Figure 2. Schematic diagram of a generic belief - desire - intention architecture.....26
Figure 3. PKO-Agent model incorporating Holland's reactive agent and Weiss' Believe -Desire -Intention agent model. 27
Figure 4. Hiles' agent model.....29
Figure 5. PKO-Agent architecture.....30
Figure 6. UML graphical notation legend 33
Figure 7. First layer of the PKO simulation model 34
Figure 8. Second layer of the PKO simulation model (Part 1).....37
Figure 9. Agent Editor graphical user interface 38
Figure 10. PKO New Agent Editor graphical user interface.....39
Figure 11. PKO Simulation Editor graphical user interface 40
Figure 12. Peacekeeping Simulation Graphical User Interface.....41
Figure 13. Call Reinforcement Graphical User Interface.....42
Figure 14. German Armored Personnel Carrier "Fuchs" [From: Ref. 18] 42
Figure 15. German Infantry Fighting Vehicle "Wiesel" [From: Ref. 18] 43
Figure 16. Second layer of the PKO simulation model (Part 2).....44
Figure 17. Third layer of the PKO simulation model.....45
Figure 18. PKOAgent Brain Lid Window.....46
Figure 19. Fourth layer of the PKO simulation model.....47
Figure 20. Coordinate system and size of a PKOAgent 49
Figure 21. Coordinate system and size of an APCMover 50
Figure 22. Coordinate system and size of an IFVMover 50
Figure 23. Task organization of 3./ Einsatzbataillon 1.....59
Figure 24. Organization of 3./ Einsatzbataillon 1.....60
Figure 25. Situation of 3./ Einsatzbataillon 1 at 021230 Mar 63
Figure 26. Generalized PRIZREN environment 66
Figure 27. Legend for the generalized PRIZREN environment.....66
Figure 28. Utility function for the goal to minimize access to the objective 73
Figure 29. Utility function for the number peacekeepers that are killed.....73
Figure 30. Utility function for the number of peacekeepers that are wounded.....74
Figure 31. Utility function for the number of protesters that are killed 74
Figure 32. Utility function for the number of protesters that are wounded.....75
Figure 33. Graphical display of the results of the sensitivity analysis 78
Figure 34. Histogram with density line of regression residuals 84
Figure 35. Boxplot of regression residuals.....84
Figure 36. Quantile-normal plot of regression residuals.....85
Figure 37. Residuals versus fitted values plot.....86
Figure 38. Correlogram for the regression model 87
Figure 39. Cook's distance for the regression model.....88
Figure 40. Regression cube 89

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. Weapon parameter 51
Table 2. Parameters for a thrown stone 51
Table 3. Characteristics of the programmed agents 67
Table 4. Personalities of the programmed agents 67
Table 5. Simulation characteristics for all experiment types 68
Table 6. Inputs for weight value sensitivity analysis 77
Table 7. Chosen weights for the sensitivity analysis 78
Table 8. Experimental design 80
Table 9. Regression output 82

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to express his thanks to the many people who were very helpful to the completion of this thesis. To Dr. Arnold Buss for giving me the freedom I wanted and the support I needed to do this work. To Dr. John Hiles who interested me in the field of agent - based simulation and taught me everything I know about agents. To Dr. Susan Sanchez and Dr. Steve Pilnick for their helpful advice and inspiration, To LTCOL Greg Mislick, USMC, for his terrific support as a Second Reader. To Oberst i.G. Folkerts for the support and information I got from the United Nation Training Centre in Germany. Finally, to my wife Kathrin, for without her loving support and huge patience none of this would have been possible.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Peacekeeping operations (PKO) present a special challenge to the Bundeswehr. Of particular concern to this thesis is the defense of a small area of responsibility by a platoon equipped with rifles, personnel carriers, and infantry fighting vehicles. This thesis develops a model of this scenario, which is used as a training tool at the German UN Training Centre, located in Hammelburg Germany. The goal is to explore and develop tactical concepts. The chosen method of modeling is driven by the singularity of peacekeeping operations.

Peacekeeping operations are usually small-scale encounters, which depend very much on the actions of each individual soldier. Whether a PKO succeeds or fails can completely depend on the action of one member of the peacekeeping forces. Often failures are caused by a lack of adequate preparation, training, leadership, or simply by deficient tactics or procedures. In opposition to combat, the primary goal in PKOs is to minimize casualties, both to the peacekeeping force and to the local people. Therefore special tactics for these kinds of operations are an important step to maximize the probability of success. The simulation model the author developed reflects the effect of individual actions and different leadership decisions, captures the process by which casualties occur, and allows the implementation of different tactics and training.

This thesis develops an agent-based simulation in order to model a defensive peacekeeping scenario. The purpose of the simulation is to get a better understanding and insight into the process of small unit PKOs. As these simulations are still abstract models of the real world, the results cannot be used to predict the outcome of encounters but to compare the relative worth of one training and leadership concept to another. In testing different personalities in the given situation and in studying the outcomes of the model, the user can gain insight as to why a special training and leadership concept may work in a certain setting while another fails. This insight enables the origination of better training.

The simulation was generated with two software packages, called Peacekeeping and TryShoot, both written in the Java language. The software packages are an extension of Simkit, a discrete-event simulation library. The agent model used is a derivation of

John Holland's software agent extended by Weiss' agent architecture, and a model, which is taught by John Hiles at the Naval Postgraduate School.

The scenario is that of an armored infantry company guarding Bishop's cathedral and offices in PRIZREN BOSNIA. On one side of the site a mob consisting of about 20 persons including women and children demonstrates, throws stones, and fires guns, while on another side it is endangered by three men with incendiary bottles. On the third side the area is under fire from snipers hidden in the upper floor of a house, and finally a group consisting of about 50 persons wearing concealing outfits attack the site using stones and incendiary bottles.

The author will focus on the first group attacking the site. The peacekeeping troops are armed with rifles and machineguns as well as personnel carriers equipped with mounted machine guns and armored fighting vehicles equipped with a 20mm machine gun. They do not possess non-lethal weapons. Their task is to defend the site without taking or causing casualties, if possible. A function was developed that evaluates a measure of effectiveness (MOE) as a linear relation of five objectives: minimize access to the red objective, minimize the number of peacekeepers that are killed, minimize the number of peacekeepers that are injured, minimize the number of protesters that are killed, and minimize the number of protesters that are injured.

Two tactics of the peacekeepers are tested: An approach that tries to dissolve the crowd; and a defensive approach with reinforcements consisting of an Armored Personnel Carrier. Additionally two forms of crowds are tested, one aggressive and one moderate crowd. The crowd also varies in the number of armed protesters and bystanders.

A factorial design for two levels and three factors of different starting conditions was used to conduct the experiment. To find a relationship between the MOE and the starting conditions, the author developed and validated a multiple regression model.

The regression was used to estimate changes in the MOE value for different environments in which the peacekeepers act and different tactics applied by the peacekeeping forces of the peacekeeping simulation model.

They showed that the peacekeepers are more successful in achieving their multiple objectives when they use a defensive tactic, when the protesters behavior is moderate, and when there are few bystanders. The regression model results were shown graphically as a regression cube.

Furthermore, the peacekeeper's defensive tactical approach always leads to a higher value of the measure of effectiveness compared to the moderate tactical approach. Since the military cannot control the protester's aggressiveness and the number of bystanders, this leads to the conclusion that the peacekeeper's should always choose the defensive tactic to best achieve their multiple objectives.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. AREA OF RESEARCH

Since the reunification of Germany, the German armed forces have participated alongside allies and partners in international peacekeeping missions, mostly within the framework of NATO and WEU. [Ref. 1: p. 9] Peacekeeping operations (PKOs) demand a higher level of training and preparedness of the individual soldier than common combat missions. The behavior of each individual service member and his individual actions can have a huge impact on the success of the mission. A small unit of platoon size has to be able to succeed in a variety of missions, such as controlling a checkpoint, manning an observation post, patrolling by vehicle or by foot, and guarding sites of military or political importance. Guarding presents a special challenge as the guarded sites are usually of special interest to all parties in the conflict area.

Research by the Bundeswehr into PKOs has intensified in recent years. For PKO research purposes and to train and prepare service members for Peacekeeping Operations in October 1999 the UN Training Centre was founded at the Infantry School in Hammelburg, Germany. Although different simulation models for land and joint combat exist, there is no model that was especially developed for research in PKOs. Since combat models are very powerful research and training tools, a simulation that models PKOs is a tool that would improve the UN Training Centre's abilities to perform research and to train military personnel. A combat model based on software agents would best suit this type of operations.

Software agents are more and more used in simulation. They provide the capability to model complex adaptive systems, systems that show coherence in the face of change, from the bottom up. Many of these systems have an apparently ordered behavior of the individual components that form the system. Researchers have used agent-based simulations to study a variety of phenomena from inner city decay over the central nervous system to financial flows on Wall Street. [Ref. 2] The US Marine Corps uses the agent-based simulation program Archimedes for combat simulations. This thesis

develops an agent-based simulation methodology to model a map exercise. The methodology can serve as a tool for the study and development for forces involved in peacekeeping operations.

B. MILITARY OPERATIONS OTHER THAN WAR (MOOTW)

The Basic Law, the German Constitution, puts Germany under the obligation to serve world peace as an equal part of a united Europe. [Ref. 3] Since Germany achieved reunification and regained full sovereignty in 1990, she has been willing to assume greater responsibility, particularly where the United Nations efforts to preserve world peace are concerned. [Ref. 1: p. 6] Therefore the Bundeswehr has increasingly taken part in peacekeeping operations, mostly in Europe, but also in Africa and Asia. The participation in peace missions presents the Bundeswehr with partially new tasks. The task spectrum will now range from the provision of humanitarian aid in areas hit by disaster or conflict, through participation in peacekeeping operations, to involvement in international crisis management activities. This means that in the future the Bundeswehr will perform two principal defense functions. On the one hand it must be able to cooperate with allies and partners in order to contribute at short notice to managing the likely international crises and conflicts; on the other hand, it must have the capability to build up and employ defensive forces adequate to defend Germany and the Alliance. [Ref. 4: p. 85]

While the soldiers, airmen and sailors of the Bundeswehr are trained to defend Germany and her allies, the military personnel that take part in PKOs need broader training than that for conventional warfare. In these missions each individual peacekeeper has to be a soldier, a policeman and a diplomat simultaneously. Additional training for PKOs includes the following areas: [Ref. 5: p. 509]

- The UN system and the general principles of peacekeeping including how soldiers are expected to carry out peacekeeping tasks. This includes the very important topic of Peacekeeping Rules of Engagement (ROE), which are usually more restrictive than ROEs associated with military operations in war. [Ref. 6: p. I-13]

- Knowledge about the political, cultural, climate and topographical situation in the areas where the peacekeeping forces will be deployed.
- Training in the function of areas they have been assigned to in the stand-by force.

In conventional combat, the main goal of each unit is the destruction of the enemy. In PKOs, however, the peacekeeping forces need to be neutral and impartial, and leaders of every level are burdened with political considerations of their actions. [Ref. 6: p. IV-1] In these types of missions, the peacekeeping forces do not have an identifiable opponent on which they can focus. Instead the peacekeeping forces have to deal with a number of different groups like allied forces, national and international police forces, International Government agencies, Non-Governmental Organizations, local politicians, the local populace, criminal groups and mostly ethnic minorities. In Kosovo, for instance, the 40,000 KFor peacekeepers have to deal with Albanian refugees from Macedonia and Serbia, Serbian minorities within Kosovo, UCK terrorists, arson in former Serbian villages and suburbs, the UN Government for Kosovo, the rebuilding of destroyed villages, the international police for Kosovo, and finally with organized crimes. [Ref. 7]

Within the Bundeswehr, peacekeeping operations are having an increased impact on training, equipment and even force planning. During the last ten years a great effort has gone into enabling all German services to take part in PKOs and to equip them with the necessary materiel. Since then, German forces, together with their European and transatlantic allies, have been very successful in these kinds of operations. They provide a wide variety of services ranging from military observer missions in Georgia, over Field Hospitals in Cambodia and Croatia, embargo control in the Adriatic Sea and on the Danube River, to large scale PKOs in Bosnia and Kosovo. Through these missions the Bundeswehr participates in international crisis management and brings Germany's influence to bear in order to prevent, contain or defuse emerging crises and to help to bring about a peaceful solution. [Ref. 1: p. 34]

While peacekeeping operations may accomplish all of their objectives, failures or actions by only one member of the peacekeeping force can bring national or international attention upon them and may be the reason for the failure of the whole mission. While

German peacekeepers in Somalia brought peace to their area of responsibility and additionally were able to build a school and reliable water supply, the mission made the nightly news when a German soldier on guard killed a Somali native who tried to enter the camp. The large-scale peacekeeping operation in Bosnia nearly failed after a Dutch officer drank vodka with the Serbian military leader in Srebrenica who afterwards ordered the atrocities in the city.

In PKOs, platoon and company commanders have to make decisions that might influence the whole mission. In their decisions they have to include the political goal of the missions and have to consider the usually very strict ROEs. This decision making process is much harder than the one in conventional combat where the military leader of a unit usually only decides if an enemy is in weapon range and if the enemy should be engaged now or later. In PKOs, on the other hand, military personnel must decide whether a given incident deserves a response, and if so, which kind of response. Platoon sergeants have to know whether to apply international law, German law, or the given Rules of Engagement. They have to decide if they are only allowed to observe a violent or hostile situation, or if they should engage. If they engage, they have to decide in what manner: by showing force, by separating different groups without using their rifles, by use of fire, or by calling for heavily-armed reinforcement. In contrast, in a typical combat situation the response to a hostile situation is very easy. The troops call for artillery fire and engage with all the firepower the unit possesses.

Troops assigned to PKOs are equipped with the same equipment as for combat but their task is to provide and ensure peace. While peacekeeping forces want to supervise free territories, cease-fires, and demilitarization [Ref. 6: p. IV-15], ethnic groups or local warlords may want to re-ignite or continue the conflict. This happened one year after the peacekeeping operation in Kosovo began, in the Presovo Valley in southern Serbia in May 2001, where UCK terrorists attacked the Serbian minority. [Ref. 7]

The German public and media measure success or failure of a peacekeeping mission in which German troops take part. Their measures of effectiveness may be completely different to those of the United Nations or the parties who were involved in

the conflict. For public and media, the behavior of the troops and the number of casualties among them is usually much more important than a solution to a conflict in a far away country. The success of a peacekeeping mission therefore depends ultimately on individual soldiers, airmen and sailors, and their ability to make appropriate decisions when confronted with violent or hostile situations. Peacekeeping forces need appropriate tactics and decision-making models in combination with the right equipment to enable their success at the point of potential conflict.

In the last ten years the Bundeswehr has put much effort in research and development on PKOs. Much effort and money has been invested to develop, test and by adequate equipment. The UN Training Centre in Hammelburg has conducted several PKO trainings and tests in realistic environments. These trainings and tests always involve a large number of troops and equipment and despite their effectiveness these tests are also very expensive and require intense planning and coordination. For conventional combat a first step of training and tests is usually done in simulations, which are cheaper and faster, and where the initial state of the test is easier to change. The Bundeswehr could benefit from the ability to simulate PKOs on platoon level and use these simulations to develop new tactics and test new equipment before it is purchased, in addition to its ongoing development efforts.

C. OBJECTIVES AND RESEARCH QUESTIONS

This thesis examines a wide range of peacekeeping tasks and leaves the possibility to add new scenarios and equipment for exploratory analysis. It models parts of a map exercise developed at the German UN Training Centre, which is used to train company and platoon leaders for their mission as part of the Kosovo Forces (KFor). The desire is to be able to generate a model that will support the exploration of tactical concepts for a wide variety of possible threats. Simulation is the best approach for this kind of problem, since it is the modeling method that will most likely aid in the examination of different tactics and trainings to a variety of threats. Insight that is gained from a simulation may be used as a fundament for other modeling methods.

The simulation of tactics and trainings in peacekeeping operations can be used to validate concepts, examine new requirements, model future threats, develop and test equipment, and develop doctrine. The Bundeswehr conducts development of peacekeeping doctrine at the UN Training Centre using methods such as wargaming and field-testing. Extensive field tests of concepts and equipment oriented towards PKOs are conducted there. While PKOs have been an area of focus for the Bundeswehr since the German reunification, the UN Training Centre has little in the way of computer simulations that specifically address tactics and engagements in peacekeeping missions. Computer simulations available in the Bundeswehr like SIRA [simulationsgestuetzte Rahmenuebungen (simulation based command field exercises)] and GUPPIS [Gefechtssimulationssystem zur Unterstuetzung von Plan-/Stabsuebungen und Planuntersuchungen in Staeben und Grossverbaenden und an Schulen und Akademien mit Heeresaufgaben (Combat simulation system to assist staff exercises and concept analysis in staffs and division size units and higher and at schools and academies with army tasks)] were developed for conventional warfare and model high intensity armored warfare using stochastic Lanchester equations to compute the outcome of a battle. This approach of simulation cannot be used for PKOs, because attrition of the opponent is usually not a goal of peacekeeping forces. A software update for SIRA, SIRA Peace Support Operations (SIRA-PSO) is still in a development phase. This update will use the same modeling approach as the master program.

Considering the problem, two questions have to be answered:

- What are the simulation requirements to model peacekeeping operations?
- Are there current warfare models that can be adapted to this use?

Warfare simulations used in the German Army come in two primary varieties: large-scale and small-scale. Large-scale models such as GUPPIS, which is installed at the Uebungszentrum Gefechtssimulation (Training Center for Combat Simulation) in Wildflecken, are used to develop and advance combat systems of the German Army, and to plan, prepare, conduct, and analyze computer-based exercises of division size units and higher. Corps- and division staffs use GUPPIS for their national or multi-national exercises. [Ref. 8: p. 1] This is clearly not a simulation that can be used to model PKOs

on the platoon level. Small-scale simulations like SIRA model conventional combat of armored battalions, armored infantry battalions, and infantry battalions including artillery, air defense, anti-tank helicopters, logistics and health service support. [Ref. 9: p. 2] Warfare is modeled discretely at the entity or small unit level (usually individual tanks and vehicles or infantry squads are modeled). Attrition is computed using stochastic Lanchester equations. This is also clearly not a simulation that can be used for PKO. Therefore, it is obvious that a small-scale simulation model is needed to model peacekeeping operations on a platoon level because each individual impacts the outcome.

The problem with the simulation of PKOs is that the goal of the peacekeeping forces is not to wear down the opponent, nor are the adversaries combatants, and thus they usually do not have a doctrine that can be programmed. Therefore, stochastic Lanchester equations or related models to compute attrition as well as computer simulations using these mathematical models cannot be used to model PKOs. A simulation model is needed where individuals have choices. The crowd in the map exercise might attack the platoon or it might decide to leave because of the show of force. This crowd has leaders, agitators, fighters, supporters, and bystanders, including men, women, and children. The criminals with their incendiary bottles also have choices, and the same is true for the peacekeeping forces. In opposition to conventional combat, the peacekeeping forces' goal is not to defeat the crowd or the criminals using all available force. Their reactions depend on the behavior of their opponents. The choices that all of these entities make are functions of their internal state, which can vary widely from one situation to the next. The issue of the internal state is one that the existing simulation model SIRA fails to capture. The SIRA simulation model uses algorithms to compute if there exists a line of sight between a firer and a target. If line of sight exists, the firer in the SIRA simulation model automatically fires at the detected target. Kill probabilities are used to calculate if the target has been hit and to compute its damage. The SIRA model does not give the firing entity the chance to try a different approach to deal with the enemy other than to try to kill him.

An agent-based simulation that models human behavior instead of kill-probabilities and firepower scores provides one solution to this difficulty. These kinds of

simulations provide the entity-level representation that seems to be best suited to problems regarding peacekeeping operations. They also possess the level of autonomy that is needed to model the issue of choice. Therefore agent-based simulations grant the flexibility that is necessary to examine different tactical concepts in PKOs.

D. COMPLEX ADAPTIVE SYSTEMS

Complex adaptive systems (CAS) are described in detail because the author believes that peacekeeping operations can be modeled as CAS. PKOs are dynamic systems composed of many nonlinearly-interacting parts. Entities in PKOs can be aggregated to soldiers, commanders, demonstrating civilians, and fearful children for example. Tagging takes place in PKOs, a number of soldiers belong to a platoon commanded by a platoon leader, and demonstrators build a group with a common goal. The interacting groups are composed of a number of nonlinearly interacting parts; sources include feedback loops in command and control hierarchy, interpretation of opponent actions, adaptation to opponent actions, decision-making process, and elements of chance. [Ref. 10: p. 30] There are flows between the individuals in PKOs; these flows are mostly information. The individuals in these operations are also diverse; on both sides there are leaders and followers, heroes and individuals driven by fear. All participants in a PKO have their own internal model, which is created by their environment, and by the way they realize it. And finally the individuals use building blocks to represent their view of the surrounding environment. The building blocks for the soldiers depend on their training experiences and their orders; those of the civilians depend on their ideology, goals, and information. In conclusion: Peacekeeping operations possess all features of complex adaptive systems. Forces and groups are composed of a number of nonlinearly interacting parts and at least the forces are organized in a command and control hierarchy; local action, which often appears disordered, induces long-range order; groups, in order to fulfill their goals, must continually adapt to a changing environment. There is no master “voice” that dictates the actions of each and every entity; and so on. [Ref. 10: p. 31]

There exists a wide variety of complex adaptive systems (CAS) in the world. One example is the central nervous system, which depends on the interaction of hundreds of millions of neurons, each undergoing thousands of simultaneous interactions in thousandths of a second. [Ref. 2: p. 3] Another is the City of San Francisco with its perpetual flux of people and structures. These examples are very different, so what makes both of them a CAS? Both are coherent under change and general principles rule their behavior; they are made up of large numbers of active elements that are diverse in form and capability. [Ref. 2: p. 6] Holland defines Seven Basics that are common to all CAS: Aggregation, tagging, nonlinearity, flows, diversity, internal models, and building blocks.

Aggregation is used as a standard way to simplify complex systems. Similar things are aggregated into categories. For the San Francisco example these might be theaters, shops and subways, which are then treated similarly. Furthermore aggregation is concerned with the emergence of complex large-scale behaviors from the interactions of the CAS entities. For the San Francisco example this might be the behavior of traffic on Broadway.

Tagging is a mechanism that consistently makes the formation of aggregates possible. An example for tagging used by Holland is a flag that is used to rally members of an army. [Ref. 2: p. 13] Tags enable us to observe and act on properties that were previously hidden by symmetries, because tagging is a pervasive mechanism for aggregation. Tags facilitate selective interactions, and when they are well established, they provide a basis for filtering, specialization, and cooperation.

CAS can be thought of as a dynamical system composed of many nonlinearly interacting parts. Examples for complex adaptive systems are economic webs and fluid flow. [Ref. 10; p. 30] Considering these properties of CAS it obvious that they are always nonlinear. Nonlinearity makes the behavior of the aggregate more complicated. Summing or averaging the behavior of the entities cannot predict the interactions within a CAS.

In the context of CAS flows are those of a network with nodes and arcs, but neither the flows nor the networks are fixed in time. Both reflect changing adaptations as time elapses and experience accumulates. Therefore flows in this context are not flows of fluids. Flows have two properties: A multiplier effect, which occurs if one injects

additional resource at some node. Typically this resource is passed from node to node, possibly being transformed along the way, and produces a chain of changes. [Ref. 2: p. 24] The recycling effect is the second property. This is the effect of cycles in the network. These cycles usually increase the output of the network, and the overall effect on a network with many cycles sometimes is remarkable.

The entities of a CAS are diverse. In San Francisco, for example, there exists a huge variety of different kinds of shops, and each of them fills a niche in the city. This diversity depends on the context provided by the other entities. If one entity is removed from the system, for example coffee shops, this usually creates a cascade of adaptations inside the system. As result of these adaptations a new entity will fill the hole that was created by removing one entity. In our example a bagel bakery might start to sell coffee in addition to their bagels. CAS often have the capability of self repair.

Internal models are a mechanism for anticipation. There are two kinds of internal models: Tacit models, which simply prescribe a current action, under an implicit prediction of some desired future state, and overt internal models, which are used as a basis for explicit, explorations of alternatives. [Ref. 2: p. 33] An entity inside a CAS has an effective internal model, if its resulting actions on changes in its environment anticipate useful future consequences.

The last of the Seven Basics of CAS are building blocks. The internal model of an entity is usually based on limited samples of the environment. In our example we build San Francisco using streets, houses and cars. Using different kinds of cars (yellow, red and green ones), different kinds of streets (one way, two lane, and four lane streets), and different kinds of houses (apartment buildings, semi-detached houses, and skyscrapers), we can build a huge variety of different looking suburbs just using these few building blocks. Everywhere in our environment building blocks are used to impose regularity on a complex world; therefore the use of building blocks to generate internal models in CAS is a convincing feature.

E. SOFTWARE AGENTS

This thesis uses software agents for modeling. Agent-based modeling starts with a set of assumptions; these assumptions are the rules of the simulation world. The agent-based simulation then generates data that can be analyzed. This is different than typical scientific induction, as the data does not come from the real world, but from a rigorously specified set of rules of the modeler's creation. Therefore the developed agent-based simulation model cannot be used for prediction, but it can be used as a tool to explore high-level behavior arising from various low-level interaction rules. The simulation model shall be a tool that provides insight into, and aids the exploration of, the fundamental behavioral tradeoffs involved among a large number of notational variables. [Ref. 10: p. 31]

The study of software agents is not new. They have been used in distributed artificial intelligence, the study, construction, and application of systems in which several interacting entities pursue some set of goals or perform some set of tasks, since the late 1970s. Nowadays the broad array of agent usage includes electronic commerce and electronic markets, real-time monitoring and management of telecommunication networks, modeling and optimization of transportation systems, information handling in information environments like the internet, analysis of business processes within or between enterprises and many other applications. [Ref. 11: p. 6]

Despite the fact they are used widely; there is no universally accepted definition of agents. Since this thesis relies on Holland [Ref. 2] and Weiss [Ref. 11] as its two primary sources, their definitions will be used. Weiss' work provides a broad approach to software agents and the computer science involved. Holland is more interested in modeling complex adaptive systems (CAS). He sees software agents as the logical tool to study CAS.

Weiss describes agents as computational entities such as software programs or robots. An agent can be viewed as perceiving and acting upon its environment. Furthermore an agent is autonomous in that its behavior at least partially depends on its own experience. As an agent is an intelligent entity, it operates flexibly and rationally in a

variety of environmental circumstances using its perceptual and effectual equipment. On the basis of key processes like problem solving, planning, decision making, and learning, an agent achieves behavioral flexibility and rationality. As an interacting entity, an agent can be affected by other agents in its activities. [Ref. 11: p. 1]

Agents perceive their environment through sensors and act upon it through effectors. Normally, an agent will have a repertoire of actions available to it, which represents its ability to modify its environment. [Ref. 12: p. 30] These actions are called tickets. Agents physically exist in the form of programs that run on computing devices.

That agents are autonomous means that they to some extent control their behavior and that they can act without the interventions of humans and other systems. To meet their design objectives agents pursue goals or carry out tasks. In general, these goals and tasks can be supplementary as well as conflicting.

That an agent is an "intelligent" entity means that agents pursue their goals and execute their tasks in order to optimize some given performance measures. [Ref. 11: p. 2] The intelligent agent is capable of flexible autonomous action. Flexible in this context means three things:

- Reactivity: intelligent agents are able to perceive their environment, and respond in a timely fashion to changes that occur in it in order to satisfy their design objectives;
- Pro-activeness: intelligent agents are able to exhibit goal-directed behavior by taking the initiative in order to satisfy their design objectives;
- Social ability: intelligent agents are capable of interacting with other agents in order to satisfy their design objectives. [Ref. 12: p. 32]

"Interacting," indicates that an agent may be affected by other agents in pursuing their goals and executing their tasks. Interaction can take place in two ways, either indirectly through the environment in which the agents are embedded, or directly through a shared language. [Ref. 11: p. 3]

While to some programmers and researchers it seems as if there is no difference between agents and objects as used in object-oriented programming, agents are different from objects in at least three points:

- Agents embody stronger notion of autonomy than objects. They decide for themselves whether or not to perform an action on request from another agent.
- Agents are capable of flexible behavior, while the standard object model does not incorporate such types of behavior.
- A multi-agent system is inherently multi-threaded. Each agent is assumed to have at least one thread of control. [Ref. 12: p. 34]

Architectures to build agents can be subdivided into four classes: logic-based agents, reactive agents, belief-desire-intention agents, and layered architectures.

In logic-based approaches to building agents, decision-making is viewed as deduction. An agent's decision-making strategy is encoded as a logical theory, and the process of selecting an action reduces to a problem of proof. While logic-based approaches have clean, logical semantics, they have one major disadvantage. The inherent computational complexity of theorem proving makes it questionable whether agents as theorem provers can operate effectively in time-constrained environments. The reason is that decision making in these kinds of agents is predicated on the assumption that the world will not change in any significant way while the agent is deciding what to do, and that an action which is rational when decision-making begins will be rational when it concludes. [Ref. 12: p. 47] Since this thesis deals with military operations in which the environment can change very fast, this kind of approach to build agents cannot be used.

Reactive agents implement decision-making in some form of direct mapping from situation to action. Advantages of reactive approaches are simplicity, computational tractability, and robustness against failure. But this approach also has important disadvantages. Purely reactive agents make decisions based on information about the agent's current state. Such a decision cannot take the past into account. Furthermore,

purely reactive agents cannot be designed to learn from experience, and to improve their performance over time. The last two points are an important part of a model that shall be used to gain insight into a system's behavior. [Ref. 12: p. 53] Consequently this type of agent cannot be used in this thesis.

Belief-desire-intelligence (BDI) agent architectures are practical reasoning architectures, in which the process of deciding what to do resembles the kind of practical reasoning that we appear to use in our everyday lives. The basic components of a BDI architecture are data structures representing the beliefs, desires, and intentions of the agent and functions that represent what it intends to do and how to do it. In the BDI model intentions provide stability for decision making, and act to focus the agent's practical reasoning. The BDI model is intuitive, as the agent acts in a way that is familiar to humans. Additionally it gives a clear functional decomposition, which indicates what sort of subsystems might be required to build an agent. [Ref. 12: p. 60] This approach to building agents will be used in this thesis' simulation model, as the agents behavior is nearest to that of humans.

In layered agent architecture, decision-making is realized via various software layers. Each of these software layers is more or less explicitly reasoning about the environment at different levels of abstraction. Due to the different layers, this way of building agents represents a natural decomposition of functionality. Reactive, pro-active, and social behavior can be generated by the layered architecture; therefore this is a very pragmatic solution. But there are two important disadvantages of layered architectures. First, they lack the conceptual and semantic clarity of unlayered approaches, and second, each layer is an independent activity-producing process. That makes it necessary to consider all possible ways that the layers can interact with one another. Both disadvantages of layered architectures make it very hard to gain insight into the agent's behavior and into agent interactions. Consequently this approach to agent building cannot be used in the simulation model as it shall be used to gain insight into group behavior. [Ref. 12: p. 66]

Holland widely describes adaptation as a feature of agents. In biological usage, adaptation is the process whereby an organism fits itself to the environment. Experience

guides changes in the organism's structure so that throughout time it makes better use of its environment for its own end. [Ref. 2: p. 9] The context of agents learning and related processes is also included in adaptation. To go back to the San Francisco example, when the bagel bakery starts to sell coffee, it adapts to its environment. The adaptation takes place because the firm can now earn more money. In CAS, agents always adapt by changing their rules that describe how they interact with the environment as experience accumulates. A major part of the environment of any given adaptive agent consists of other adaptive agents. Therefore a portion of any agent's efforts at adaptation is spent adapting to other adaptive agents. The bagel bakery interacts with its customers. When they change their behavior - let's say most of them want sandwiches with less fat - then the bakery adapts to that by offering new kinds of sandwiches, which fulfill the needs of the customers.

A framework that consists of five major components - a performance system, a personality, a history, a goal structure, and a number of available tickets - portray the adaptive agents programmed for this simulation.

The performance system specifies the agent's capabilities. It describes what the agent is able to do without any further adaptation. The basic elements of the performance system are the agent's detectors and effectors and a set of rules, which represents its capabilities for processing the information the agent receives from its environment. [Ref. 2: p. 88] The bagel bakery detects that there are customers in San Francisco who want bagels and sandwiches. The rule set implies that IF there are customers who want bagels, THEN let us produce bagels and sell them. By selling the bagels the bakery effects all of its customers.

The personality uses a specified number of parameters to define the agent's character. For the bagel bakery example parameters could be how much does the bakery want to maximize revenue, how much diversity in its products does it want to offer, or how much does it want to serve breakfast or lunch in the shop.

The history keeps track of the agent's actions in the past. It is used to determine the current action depending on the past. If the bagel bakery has not sold any cinnamon

rolls since last week, it will stop baking cinnamon rolls. This knowledge of the past is used to change the current action.

The goal structure defines the agent's desires. The desires depend on the agent's personality and the environment, as the agent perceives it. Out of a given number of goals the agent always tries to achieve its most important goal. This is one of the agent's ways to adapt to the perceived environment. The bakery's goals might be to sell cinnamon rolls, poppy seed bagels and French rolls. If nobody has purchased cinnamon rolls for the past two weeks, and the personality parameter of maximizing revenue is very high, then the goal of selling these rolls will become less important but another goal (for example, to sell poppy seed bagels) will fill in instead.

Tickets are the expression of how the agent achieves its most important goal. To use different tickets in different situations is the agent's second way to adapt to the perceived environment. For the bagel bakery's cinnamon roll problem, one ticket might be to sell the rolls "buy one, get one free"; another one might be to reduce the price. Which ticket is chosen depends on the personality and the history. If "buy one, get one free" sales have not been successful in the past, the bagel bakery will reduce the price on the cinnamon rolls to achieve its goal of selling them.

In conclusion, adaptive agents are software programs that can interact with their environment including other agents, and are able to adapt to changes in the environment.

Peacekeeping operations depend heavily on interactions between the different groups that take part in the operation and between the members of each of the groups. Moreover PKO takes place in a sometimes rapidly changing environment and the entities acting in this environment need to be able to adapt to the changes. Hence adaptive agents are the perfect tools for a simulation model that deals with these kinds of military operations and they will be used in this thesis to model the acting entities.

F. AGENTS IN WARFARE SIMULATION

As shown before, peacekeeping operations are complex adaptive systems. The same is true for land combat. In the words of Clausewitz “War is ... not the action of a

living force upon lifeless mass … but always the collision of two living forces." [Ref. 13: p. 77] Furthermore, the US Marine Corps' vision of combat is that of "small, highly trained, well-armed autonomous teams working in concert, continually adapting to changing conditions in the environment." [Ref. 10: p. 29] Land combat and peacekeeping operations possess all of the characteristic features of complex adaptive systems. The forces are composed of a large number of nonlinearly interacting parts and they are organized in a command and control hierarchy. Combat is self-organized and military forces, in order to survive, continually adapt to a changing environment.

These are the reasons why agent-based simulation has been applied to study warfare in recent years. A software simulation called Irreducible Semi-Autonomous Adaptive Combat (ISAAC) uses agents to model warfare as a CAS. The main advantages of this approach to model land combat over older simulation models is that it incorporates the psychological and decision-making capability of the human combatant and that it accounts for spatial variation of forces. Simulations based on Lanchester Equations, on the other hand, lack the spatial degrees of freedom to realistically model modern combat. Lanchestrian-based simulations model combat as a deterministic process, which requires the knowledge of attrition rates. These attrition rates are assumed to represent the entire force. The values of attrition rates are very difficult to obtain and errors in different attrition rates will lead to drastically different simulation results. Furthermore these simulations lack the ability to account for suppressive effects of weapons and terrain. Both can only be included into the attrition rate values, which makes it even more difficult to obtain them. Even models using extensions to Lanchester Equations by formulating them as stochastic differential equations or partial differential equations still evaluate the combat outcome as the result of force-on-force attrition. The results generated in these simulation models are essentially the long-term average results of that specific battle; they represent the expected results. But history shows that battles often have unexpected results and that the outcome of a battle generally lies more in the tail of the distribution than at its mode. Therefore a warfare model that provides military leaders with some understanding of the processes that lead to a result, which lies in the tail of a distribution, is of obvious value. Furthermore, Lanchestrian-based models cannot be used to simulate peacekeeping operations. PKO soldiers and Marines shall not destroy the

opponent but rather reach their goal by taking and inflicting as few casualties as possible. An optimal outcome would have zero casualties. A simulation model that uses Lanchester Equations cannot lead to a zero-casualties outcome.

Agent based warfare simulations are not designed to predict the outcome of a combat engagement but they can be used to intensify understanding. They shall help the analyst and military leader to …

- Understand how all the different elements of combat fit together in an overall combat space,
- Assess the value of information,
- Explore tradeoffs between centralized and decentralized command-and control structures,
- Provide a natural arena in which to explore consequences of various qualitative characteristics of combat like unit cohesion, morale, and leadership,
- Explore emergent behaviors and properties arising from low-level rules, and
- Address questions such as "How do two sides of a conflict co-evolve with one another". [Ref. 10: p. 41]

A newly developed agent based warfare simulation model is EINSTein (Enhanced ISAAC Neural Simulation Tool), a follow-on to ISAAC. EINSTein serves as a conceptual laboratory for the general exploration of combat as a complex adaptive system. Analysts have the opportunity to play multiple "What if?" scenarios and to experiment with fundamental issues of the dynamics of war. [Ref. 10: p. 24]

And finally the United States Marine Corps Combat Development Command's Project Albert sponsors an agent based simulation model called Archimedes, which will be able to represent discipline, cohesion, morale and personality in combat and to capture the nonlinearity of battlefield situations. It will be flexible enough to represent new challenges the military currently faces like Noncombatant Evacuation Operations, Small

Scale Contingencies, and Military Operations in Urban Terrain (MOUT). [Ref. 14: p. 119]

The strength of agent-based warfare simulation models is that they provide a powerful general approach to computer simulation and that they can provoke researchers, analysts and military leaders to ask new questions about warfare.

G. SCOPE OF THE THESIS AND METHODOLOGY

The methodology used in this thesis represents an attempt at combining agent-based modeling with discrete-event simulation to address peacekeeping operations. It has been developed to address a common peacekeeping operation scenario and has the capability to also address a wide range of other scenarios with this type of military operation. The United Nation Training Centre of the Bundeswehr developed these scenarios for the map exercise PRIZREN. Through modeling these scenarios, this thesis develops a simulation methodology that will assist researchers, analysts, and military leaders to better understand behavior in peacekeeping operations.

This thesis will simulate a tactical problem of PKO and investigate the question of tactics and training development. Additionally it will cover new ground on the method of modeling, as there are very few accessible implementation methods using the methodology this thesis is going to use. This is the reason why this thesis must develop the modeling methodology, before the specific problems to be modeled can be addressed. The author will consider more general applications from a theoretical and implementation perspective. The theoretical aspect is addressed by Holland's agent model.

Chapter II of this thesis will address the PKO-focused implementations of Holland's model called Peacekeeping and TryShoot. Chapter III addresses the problems to be simulated, the tactical questions the different scenarios raise, and the methods used to generate the simulations. In Chapter IV the results of the simulations are analyzed and discussed. Chapter V contains the conclusion and recommendations.

THIS PAGE INTENTIONALLY LEFT BLANK

II. AGENT MODEL AND IMPLEMENTATION

Peacekeeping and TryShoot are software implementations that model agents based on a mixture of Holland's [Ref. 2] reactive agent model, Weiss [Ref. 11] Belief-Desire-Intention architecture and Prof Hiles' agent model. The implemented architecture constructs an agent especially suitable for a PKO simulation.

Before describing the software implementations it is important to first understand the agent model that is used in this thesis. Then Peacekeeping's and TryShoot's structure are discussed. The chapter concludes with a description of the process of generating an agent-based simulation with Peacekeeping and TryShoot.

A. THE AGENT MODEL

In Hidden Order [Ref. 2] Holland provides an extensive description of a rule based adaptive agent. The agent model used in this thesis is primarily based on Holland's theoretical model, but recent agent model developments and the purpose of the programmed simulation make it necessary to deviate from his model. The author developed an agent architecture that implements a belief-desire-intention (BDI) architecture [Ref. 11: p. 34], together with Holland's adaptive agent, and that also incorporates Hiles' idea of tickets. This agent model will be called a PKO-Agent. For a better understanding the agent components of all the aforementioned agent architectures, which are used in the PKO-Agent, will be explained.

1. Holland's Agent Model

Holland develops a rule-based adaptive agent. This agent is used to develop computer models of complex adaptive systems. The agents in this model evolve new rules via genetic algorithms.

The agent described by Holland contains four components: a set of detectors, a set of effectors, a set of stimulus response rules, and a performance system. The agent perceives its environment through its detectors. An event that occurs in the surrounding environment causes the detectors to generate a message, which is transmitted to the existing rules. As a result the rules may or may not generate a message. Rule-generated messages then lead to instructions to the effectors to take a specific action in response to the event in the environment. Since different rules may generate different, and possibly opposing, messages the performance system filters the generated messages and chooses the message from the "fittest" rule to send to the effectors. In this model fitness is determined by the relative success of the rules in past decisions.

The author uses the idea of detectors, effectors, and the performance system as a filter from Holland's agent model. The agent model used in this thesis, the PKO-Agent, has a detector to perceive the outer environment and to build a picture of the perceived outer environment to build an inner environment. This inner environment is the knowledge of the outer environment the agent has. All agent decisions are based on the inner environment. The PKO-Agent also has effectors, to react to the outer environment. It furthermore has a performance system that filters the "fittest" reaction out of a set of possible reactions, depending on the inner environment.

Figure 1 shows the PKO-Agent as it is developed so far, showing how detectors and effectors couple the outer environment to the agent.

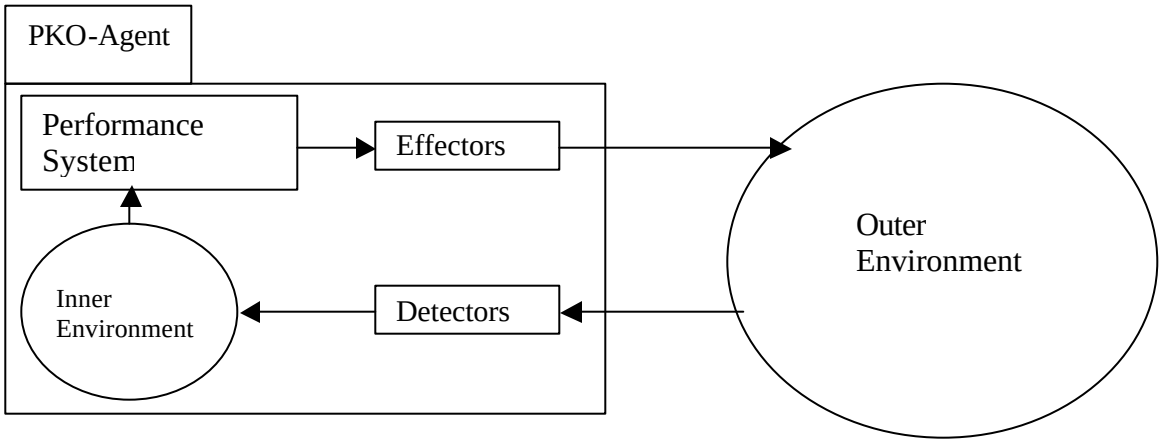

Figure 1. PKO-Agent model depending on Holland's reactive agent

The author does not use Holland's idea of rule generation by genetic algorithms, since this would lead to agents that change their behavior during the simulation of a peacekeeping operation. Since the goal of this simulation is to gain insight into PKOs; a changing agent behavior during a simulation run would make it impossible to distinguish if an event happens because of the actions and reactions in the complex adaptive system, or because the agents developed new rules, which changed their behavior.

2. Weiss' Belief-Desire-Intention Architecture

BDI architectures are based on practical reasoning, the process of deciding, moment-by-moment, which action to perform in the furtherance of existing goals. [Ref. 11: p. 54] Two important processes are involved in an agent's practical reasoning: First the agent has to decide what goals it wants to achieve and than it has to decide how it will achieve these goals. To gain an understanding of the BDI model, Weiss gives the following example [Ref. 11: p. 55]: When a student leaves a university with their first degree, he or she is faced with a decision about what to do with his or her life. This decision process typically begins by trying to understand what the available options are. If the student had a high GPA, one option is to become an academic. This option is not available when the student failed to obtain good grades. Another option is to enter industry. After generating this set of alternatives, the student must choose between them, and commit to some. The chosen options become intentions, which then determine the student's actions. Intentions then feed back into the student's future practical reasoning. For example, if the student decides to become an academic, then he or she should commit to this objective, and devote time and effort to achieving that end.

Intentions play a number of important roles in the agent's practical reasoning process; they usually lead to action. They drive "means-ends" reasoning. If a student has formed the intention to become an academic, then he or she will attempt to achieve the intention by deciding how to achieve it; for example, by applying for a PhD program. Intentions furthermore constrain future deliberation. The student, who decided to become an academic, will not entertain options that are inconsistent with this intention. He or she

will, for example, not apply for a job in industry. Additionally, intentions persist: the student will not give up the intention to become an academic without good reason. Finally, intentions influence beliefs upon which future practical reasoning is based. The student, who adopts the intention to become an academic, will plan for the future on the assumption that he or she will be an academic. [Ref. 11; p. 56]

A key problem in the design of practical reasoning agents is that of achieving a good balance between the different aforementioned roles of intentions. An agent should at times drop some intentions, because it comes to believe that the reason for having the intention is no longer present. Therefore an agent should reconsider its intentions from time to time. An agent that does not reconsider its intentions sufficiently will often continue to try to achieve goals even after there is no longer any reason for achieving them. On the other hand, an agent that constantly reconsiders its intentions will spend insufficient time actually working to achieve them and therefore runs the risk of never actually achieving any of them. Thus, goal directed and reactive behavior has to be balanced. For an agent that acts in a dynamic environment, the ability to react to changes by modifying intentions is very important, and therefore it should be able to use computational resources to reevaluate its intentions.

As a result of the above discussion, the agent that Weiss describes has seven main components: [Ref. 11; p. 57]

- A set of current beliefs, representing information the agent has about its current environment.
- A belief revision function, which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs.
- An option generating function, which determines the options available to the agent (its desires), on the basis of its current beliefs about its environment and its current intentions.
- A set of current options, representing possible courses of action available to the agent.

- A filter function, which represents the agent's deliberation process, and which determines the agent's intentions on the basis of its current beliefs, desires, and intentions.
- A set of current intentions, representing the agent's current focus - those states of affairs that it is committed to trying to bring about.
- An action selection function, which determines an action to perform on the basis of current intentions.

The agent starts out with a set of possible beliefs, a set of possible desires, and set of possible intentions.

The option generating function then maps a set of beliefs and a set of intentions to a set of desires. This function is responsible for the agent's process of deciding how to achieve intentions. Once an agent has formed an intention, it must consider options how to achieve this intention. Additionally the option generating function satisfies two constraints. First it is consistent; any option generated must be consistent with both the agent's current beliefs and current intentions. Second, it is opportunistic; it recognizes when environmental circumstances change advantageously, to offer the agent new ways of achieving intentions, or the possibility of achieving intentions that were otherwise unachievable.

The filter function represents the agent's process of deciding what to do. It updates the agent's intentions on the basis of its previously held intentions and current beliefs and desires. This function fulfills two roles. It drops any intentions that are no longer achievable and it retains intentions that are not achieved, and that are still expected to have a positive overall benefit.

The execute function simply returns an executable intention that corresponds to a directly executable action.

Figure 2 displays a diagram of the described BDI agent model. [Ref. 11; p. 58]

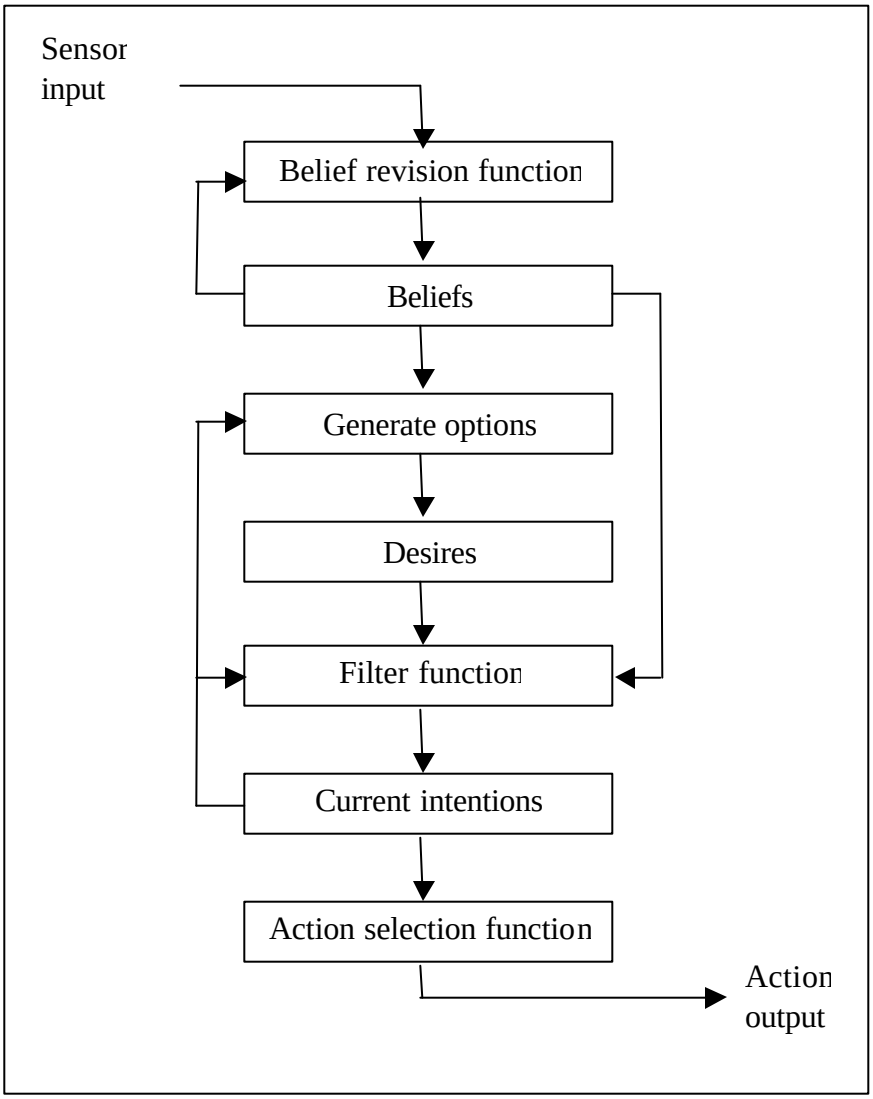

Figure 2. Schematic diagram of a generic belief - desire - intention architecture

The author uses the ideas of current beliefs the agent has about its current environment, an option generating function, a set of current options, a filter function, a set of current intentions, and an action selection function from Weiss' BDI agent model.

The PKO-Agent's inner environment that it gets through its detectors is the agent's current belief about its current environment. The PKO-Agent's set of current options is modeled by seven different goals. At all times the agent tries to achieve one of them. The

set of current intentions is the goal the PKO-Agent currently tries to achieve. The action selection function assigns a weight to each of the agent's goals. The weight depends on the inner environment and the history. The filter function checks the weight assigned to each of the seven goals. It then simply selects the goal with the highest weight as the agent's new intention. The option generating function uses the inner environment and the history of agent actions as input to evaluate an option of how to achieve the current goal.

Figure 3 shows the PKO-Agent as it is developed so far, showing how Holland's agent model is expanded and Weiss' BDI agent features are incorporated into the model.

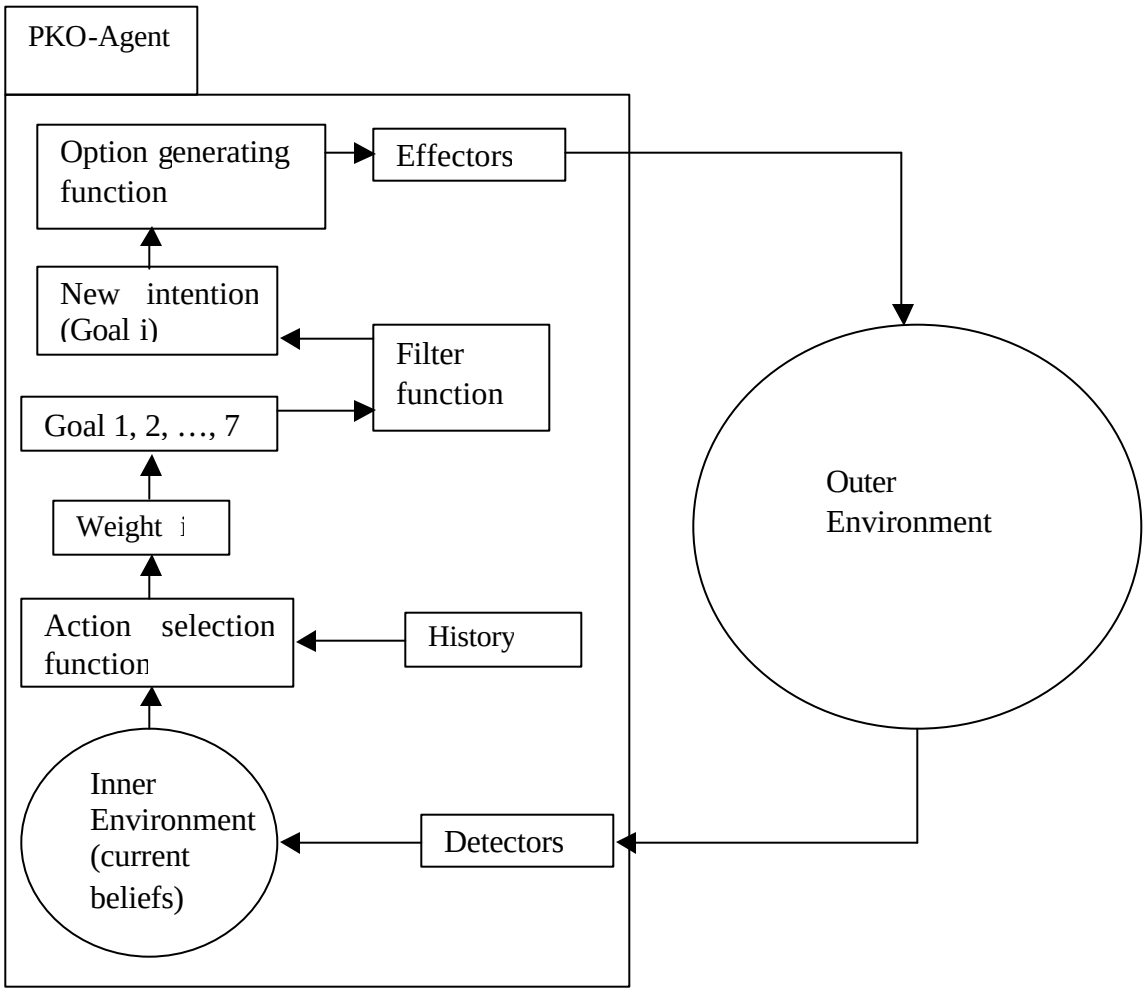

Figure 3. PKO-Agent model incorporating Holland's reactive agent and Weiss' Believe - Desire -Intention agent model.

The author does not use Weiss' idea of a belief revision function, which determines a new set of beliefs. As with Holland's rule generation by genetic algorithms, this function would lead to agents, which change their behavior by introducing new goals during the simulation of a peacekeeping operation. Therefore the belief revision function would make it very hard to display the agent's current goal and to evaluate how the agent's goals change as a result of the agent's picture of its surrounding complex adaptive outer environment.

3. Hiles' Agent Model

As with Holland's and Weiss' agent model, Hiles' agent architecture also uses sensors to form an inner environment of the perceived outer environment and effectors to react towards the outer environment. The agent in Hiles' model is similar to the agent in Weiss' model in having a certain number of goals and a weight associated with each goal. These goal weights change, depending on the inner environment and some measurement function. For example an agent that models a bird, a boid, has the goals: stay in flock, avoid obstacles, and avoid collision. The sensors measure the distance to all other perceived boids and to perceived obstacles. The boid always has one active goal: for example; to stay in the flock. When the boid senses an obstacle on its flight path, the weight for the goal "avoid obstacles" increases. At the point where this goal weight is greater than the weight for the current goal, "stay in flock", the active goal changes. The goal with the highest weight always is the active goal.

Each goal has a certain number of tickets, which generate the action the agent will take to respond towards the outer environment. In the case of the boid, actions might be turn right, turn left, hold speed, accelerate, and slow down. As the goals, these tickets also have associated weights and a measurement function to evaluate the weight value. The boid that flies in a flock might have hold speed and direction as its active ticket. Approaching the obstacle, the ticket weight for turn right might increase, so that the boid will change its active ticket and will turn right before it hits the obstacle.

Figure 4 shows a model of Hiles' agent architecture.

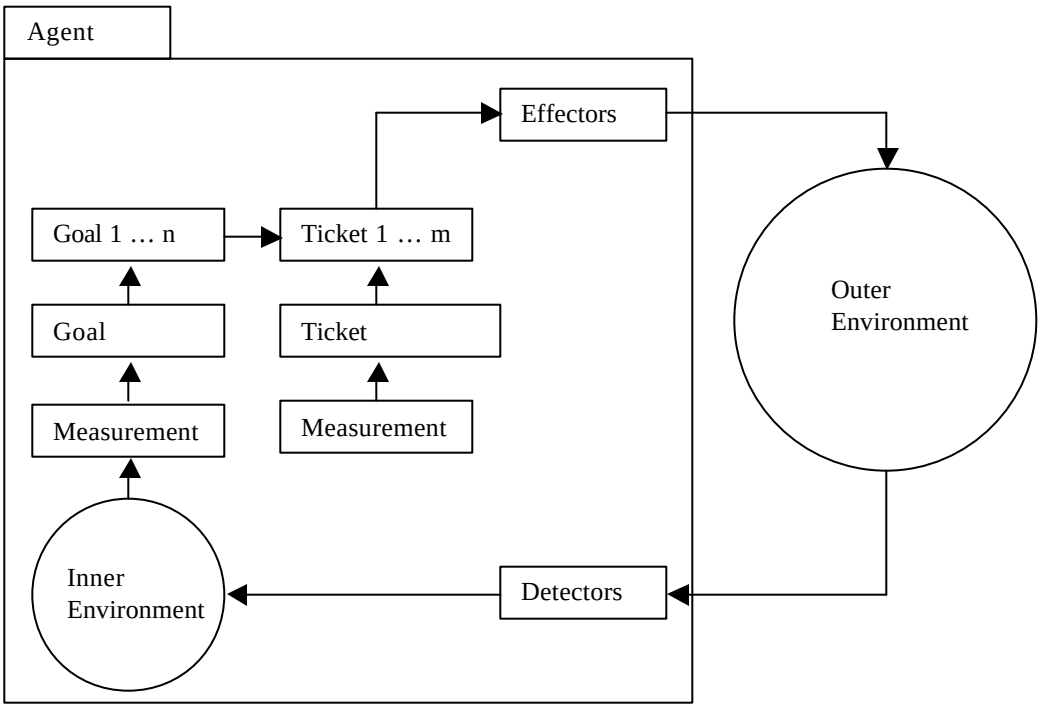

Figure 4. Hiles' agent model

The author incorporates Hiles' idea of tickets as a way to determine action into his agent model. Each PKO-Agent determines one ticket for each goal. This ticket consists of a speed and an action. The speed determines how fast the agent moves: it is bounded by zero as the lower limit, and has an upper limit depending on the agent type. The upper limit for an Armored Personnel Carrier, for example, is 30 km/h. The action is a response towards the outer environment Fire in the air is one example for a ticket action, movement another. Speed and ticket actions are connected to an active ticket, which then determines the agent's overall action. The measurement for the goal depends on the sensed environment and the agent's personality; the measurement for the ticket depends on the sensed environment, the agent's personality, and the agent's history of actions. Weiss' action selection function becomes the goal function and his filter function together with the option generating function becomes the ticket function.

The agent architecture that is used in this simulation is shown in Figure 5.

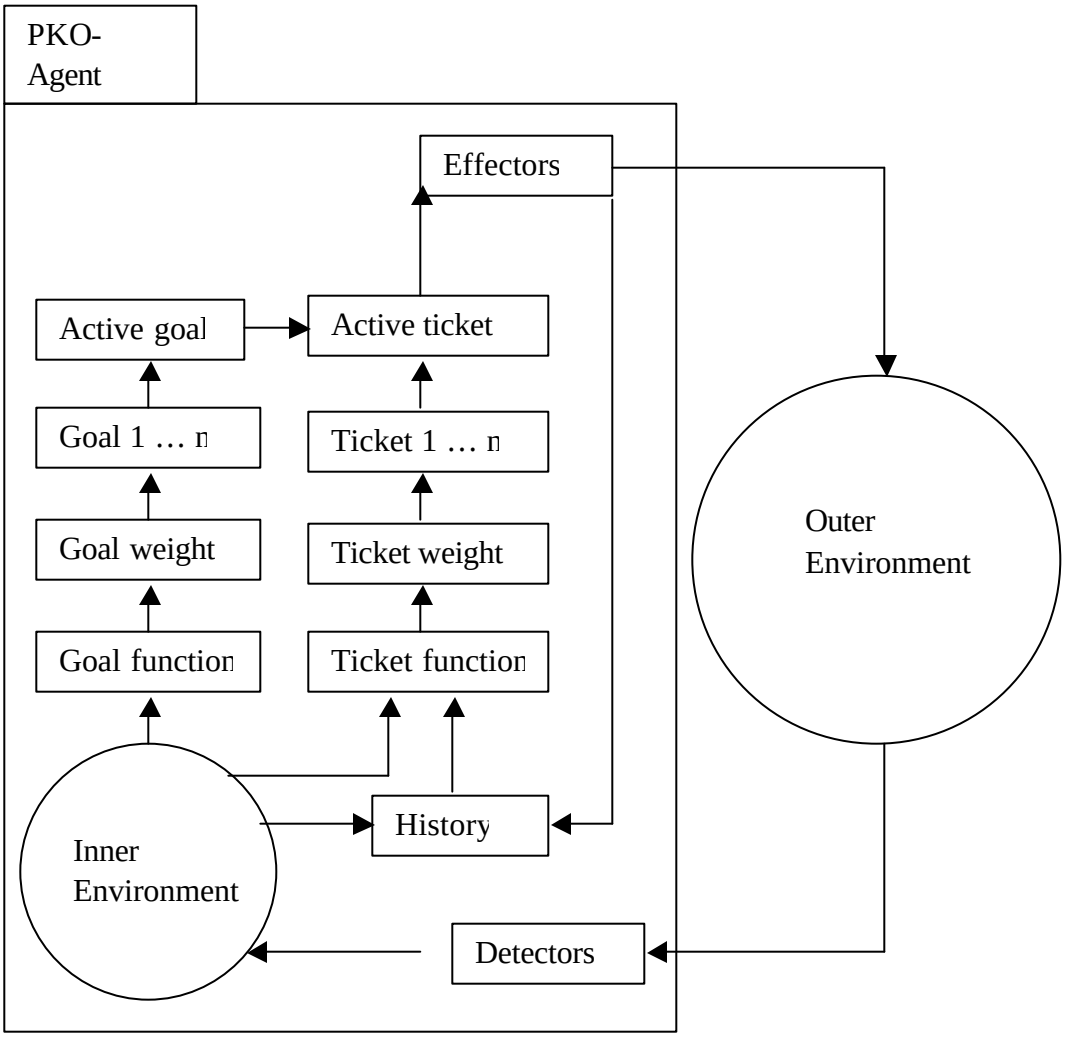

Figure 5. PKO-Agent architecture

B. IMPLEMENTATION CHOICES

The first step to implementing the PKO-Agent model was choosing a simulation methodology. There are two simulation methodologies to choose from: discrete event and time step.

Discrete event simulation concerns the modeling of a system as it evolves over time by a representation in which the state variables change instantaneously at separate points in time. These points in time are the ones at which an event occurs, where an event is defined as an instantaneous occurrence that may change the state of the system. [Ref. 15; p. 6] A time-step simulation, on the other hand, updates all states simultaneously at each processed time step and processes resulting events in a random manner.

The author chose discrete event simulation, DES, because a pure DES worldview provides more flexibility and modeling power than a pure process - oriented worldview. [Ref. 16; p. 1] In an evaluative model as the one provided in this thesis, it is furthermore very important that the state variables change instantaneous, and not at some time step that is chosen for programming purposes. The model must evaluate who acts first and how other entities react to the action. These issues cannot be observed when all state variables are updated simultaneously at a given point in time and the order of action is generated at random.

The second implementation choice was to choose a programming language. Agents are modeled in component objects; therefore an object-oriented language would be best suitable to implement agents in a simulation. Furthermore the model should be platform independent and run on most operating systems. A final prerequisite of the programming language is that it has powerful graphics to show setup and show the simulation for a user who is not a computer specialist. Java, with the additional package Java Swing, fulfills all criteria and consequently is the programming language used in this simulation.

Both simulation type and programming language lead to Simkit, a software package for implementing Discrete Event Simulation methodology, which is written in

Java and runs on any operating system with Java 2™ installed. [Ref. 16; p. 1] Simkit provides the components and organization necessary for a DES.

- A simulation clock: a variable giving the current value of simulated time.
- An event list: a list containing the next time when each type of event will occur
- Statistical counters: variables used for storing statistical information about system performance.
- An initializing routine: a class or method to initialize the simulation model at time 0.
- A timing routine: a class or method that determines the next event from the event list and then advances the simulation clock to the time when that event is to occur.
- Library routines: a set of classes or methods to generate random observations from probability distributions that were determined as part of the simulation model.

  [Ref. 15; p. 9]

The PKO simulation model uses these components, as well as interfaces and superclasses provided by Simkit. The interfaces are implemented and the superclasses extended to fulfill the needs of an agent based simulation.

C. PEACEKEEPING AND TRYSHOOT STRUCTURE

The Unified Modeling Language, UML, will be used to explain the structure of the PKO simulation model implementing the packages Simkit, Peacekeeping and TryShoot. [Ref. 17] Figure 6 shows the basic graphical notation.

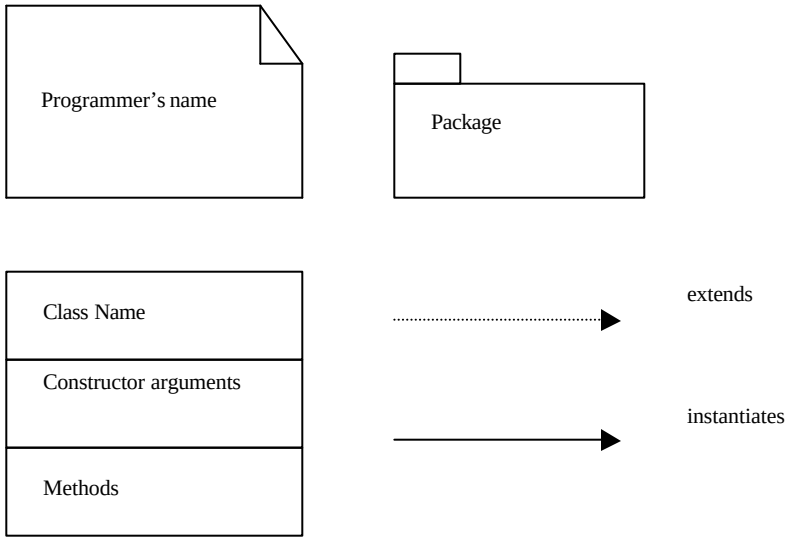

Figure 6. UML graphical notation legend

A class name without a package name as a prescript will always mean that the class is in the Peacekeeping package.

1. Graphical User Interface and Environment Layer

The simulation is organized in layers. The first layer builds all graphical user interfaces, GUIs, and the simulation environment, which contains the terrain features of the simulated area. This layer also contains the executable class, PKOAnimation. Figure 7 shows the structure of this first layer.

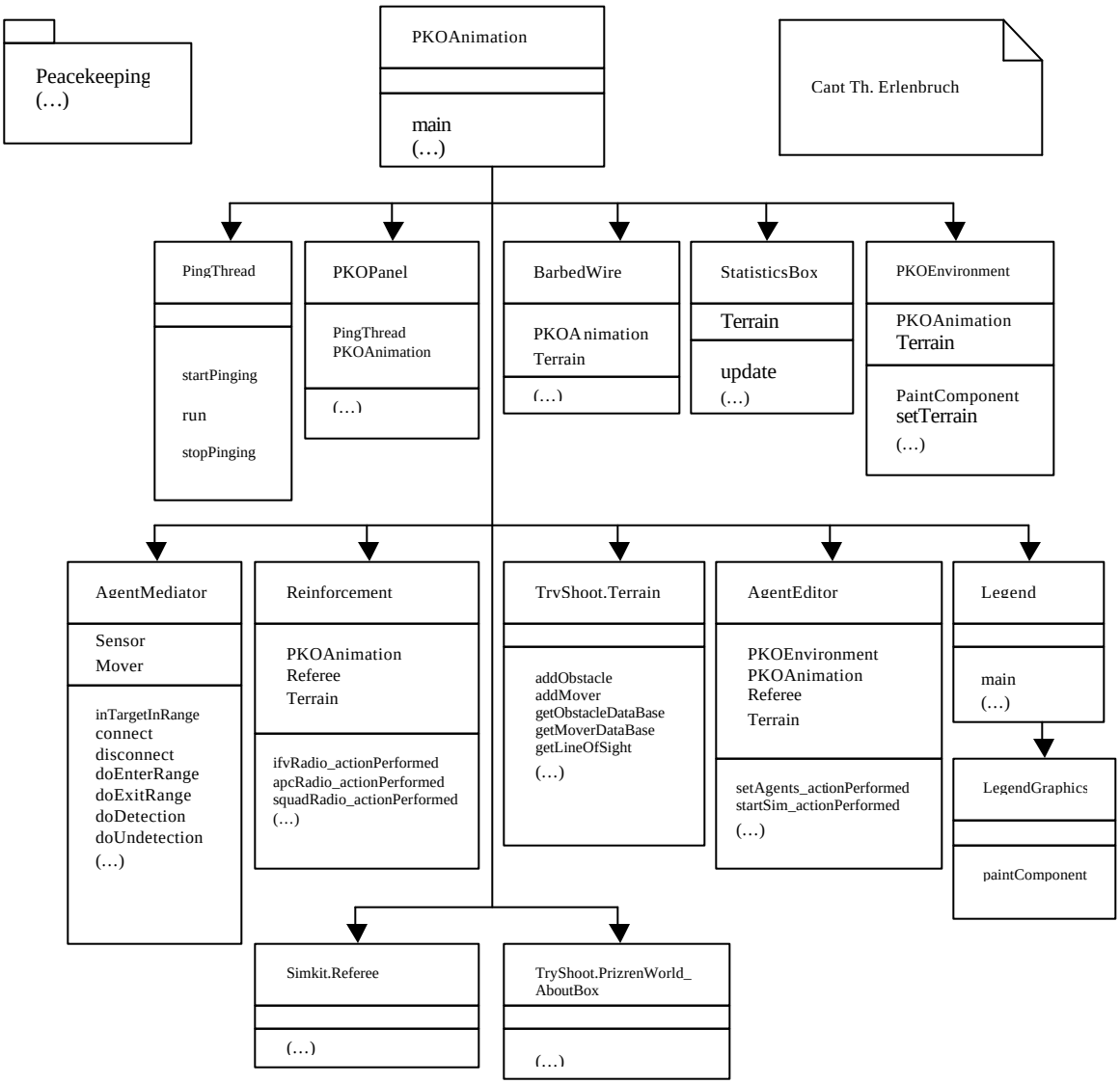

Figure 7. First layer of the PKO simulation model

PKOAnimation is the frame that holds the simulation. It contains the simulation panel and instantiates all the classes that are needed for the simulation setup and to run the simulation.

PKOPanel contains all the button and display functionality and is the panel on which every simulation entity and all terrain features are drawn.

PingThread is a class to animate Simkit programs. It is developed by Prof. Buss and upgraded by the author. A Ping event occurs after a given period of simulated time. The PKOPanel listens to the Ping event and draws all known entities at their updated positions on the screen whenever a Ping event occurs.

The PKOEnvironment is a class that is needed to draw the map and the agents. It paints the coordinate systems, all movers, and the obstacles into the panel.

The class Terrain in the TryShoot package stores all obstacles that lie in the terrain in a vector and stores all agents in another vector. It also checks if a detection occurs. When a sensor would see a target but there is a building in the line of sight, no detection occurs. The PKOEnvironment gets it obstacle and agent information from the Terrain.

The PKOAnimation additionally instantiates different GUIs:

- Legend: A class that draws a legend for the peacekeeping simulation. Legend itself instantiates LegendGraphics, a class that creates a panel for the legend.
- TryShoot.PrizrenWorld_AboutBox: A class that creates pop-up window that shows up, when "About" in the "Help" menu is chosen. The window contains some basic information about the simulation model.
- Reinforcement: A class where the user can call different types of reinforcement. Possible reinforcements are Infantry Fighting Vehicles and Armored Personnel Carriers.
- BarbedWire: A class that allows the user to build a new barbed wire fence in the terrain.

- StatisticsBox: A class to compute and display the simulation statistics.
- AgentEditor: The dialog where the user chooses an existing personality of an Agent or decides to create a new type of agent. AgentEditor is the source for the next layer of the simulation and will be discussed in detail later.

Finally the PKOAnimation class instantiates the Simkit.Referee and the AgentMediator objects. Though these objects are neither GUIs nor part of the simulation environment, they need to be instantiated in this layer, because they are needed in some of this layers' classes as well as classes of underlying layers.

In the PKO simulation model interactions between separate entities are handled by a third party. First, the Simkit.Referee tracks all entities that act in the simulation. When it determines by their trajectories that there will be an interaction between two entities, it assigns the AgentMediator to handle the specific interaction. The AgentMediator, an extension of Simkit's mediator class, handles detection by informing the sensor when it has detected a mover. The detection depends upon the sensors detection algorithm.

2. Simulation Entity Layer

The second layer is used to define the simulation entities, which will act in the model and the simulation specifics. The Reinforcement class and the AgentEditor class are the two links between the first and the second layer. Figure 8 shows how the AgentEditor class links the two layers.

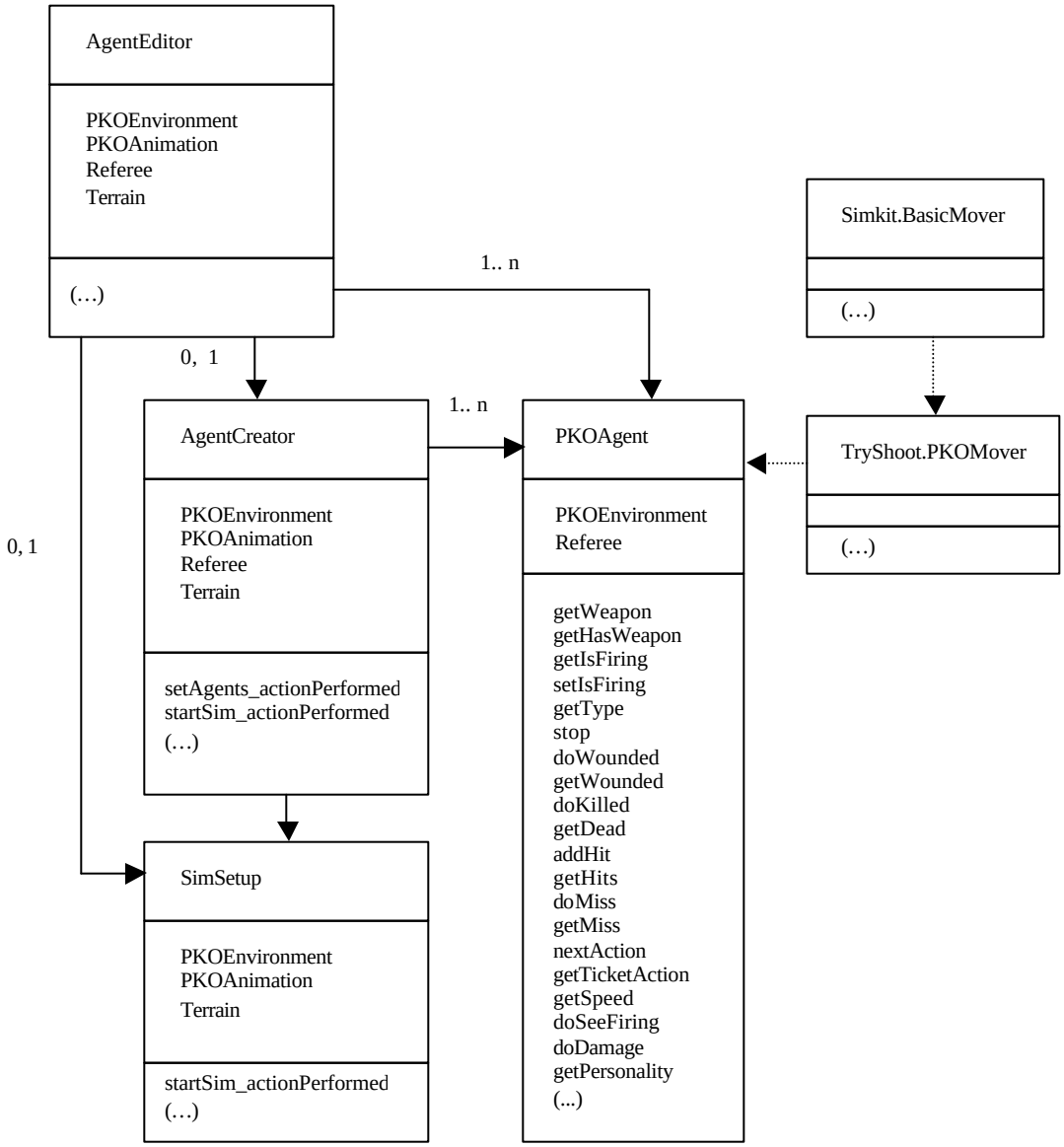

Figure 8. Second layer of the PKO simulation model (Part 1)

When the program starts, it first displays a GUI that is produced by the AgentEditor. (see Figure 9)

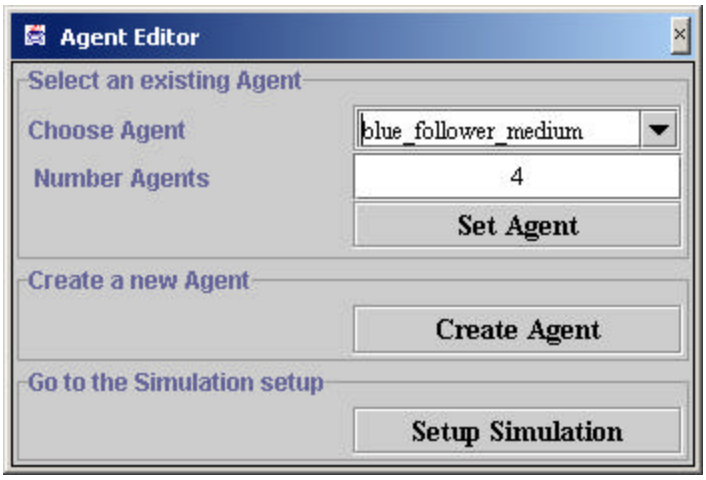

Figure 9. Agent Editor graphical user interface

The user has the possibility to select predefined agents for the simulation. These agents with all their characteristics are stored in property files. A knowledgeable user can define additional agents and they will automatically be displayed in the combo box.

The user adds a predefined agent type to the simulation by selecting it in the combo box, entering the number of agents into the text box, and clicking the "Set Agent" button. He or she can then select another agent type in the same way.

If the user has set all the predefined agents that are desired and does not want to define a new agent type, he or she clicks the "Setup Simulation" button.

To create an agent that is not predefined, the user clicks the "Create Agent" button.

The "Create Agent" button causes the instantiation of an AgentCreator object, which displays the "PKO New Agent Editor" GUI.

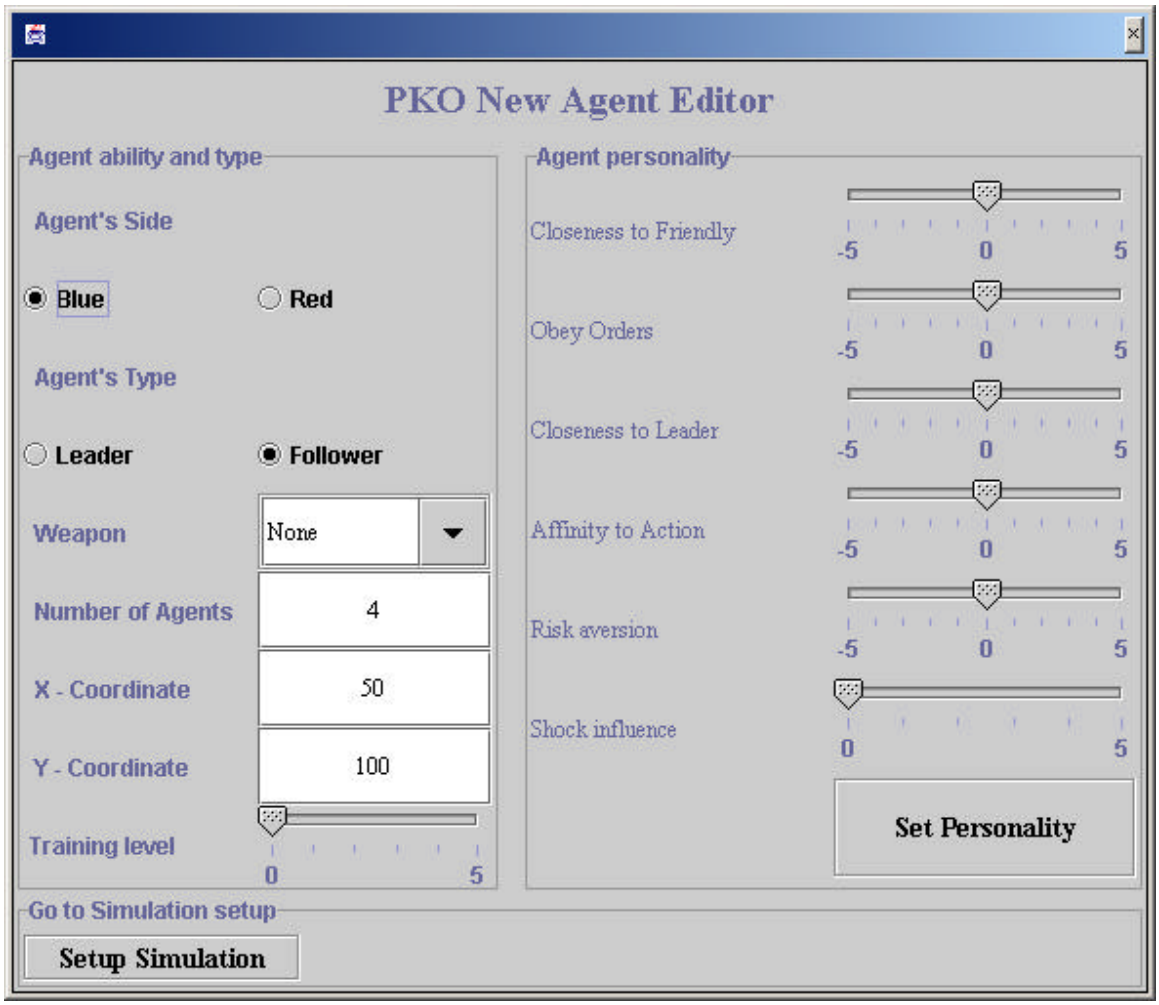

Figure 10. PKO New Agent Editor graphical user interface

Figure 10 shows how with this GUI the user can define an agent's ability and type as well as its personality. The user can define as many different agents as he or she wants. The description of the abilities and the personality features will follow in the next subchapter. If the user has set all the agents he or she clicks the "Setup Simulation" button.

The "Setup Simulation" button in either the "Agent Editor" GUI or the "PKO New Agent Editor" GUI causes the instantiation of a SimSetup object, which displays the "PKO Simulation Editor" GUI.

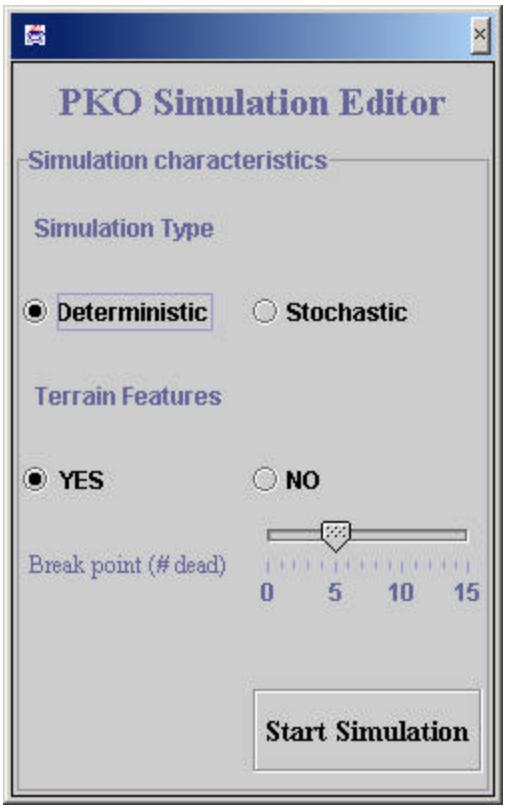

Figure 11. PKO Simulation Editor graphical user interface

With the "PKO Simulation Editor" GUI (Figure 11) the user defines the simulation type, deterministic or stochastic, and if the simulation includes terrain features or not. Additionally the user sets the simulation break point.

In a stochastic simulation run, hit probabilities are generated randomly, while the parameters are depending simply on the weapon type in a deterministic simulation run.

If the user chooses to include the terrain features, all obstacles that are in the database will be displayed. The database contains a highly generated map of the PRIZREN town center. If the terrain features are not included, the simulation will be run on an empty, plain terrain.

By default the simulation ends after 6 hours of simulation time or when one red agent has reached the red objective. The user can set the third possible condition to terminate the simulation by setting the break point. The simulation will end when the set number of agents is dead. The simulation will terminate when one of the three conditions is fulfilled.

To start the simulation run, the user clicks the "Start Simulation" button. This will cause the PKOAnimation class to display the "Peacekeeping Simulation" GUI.

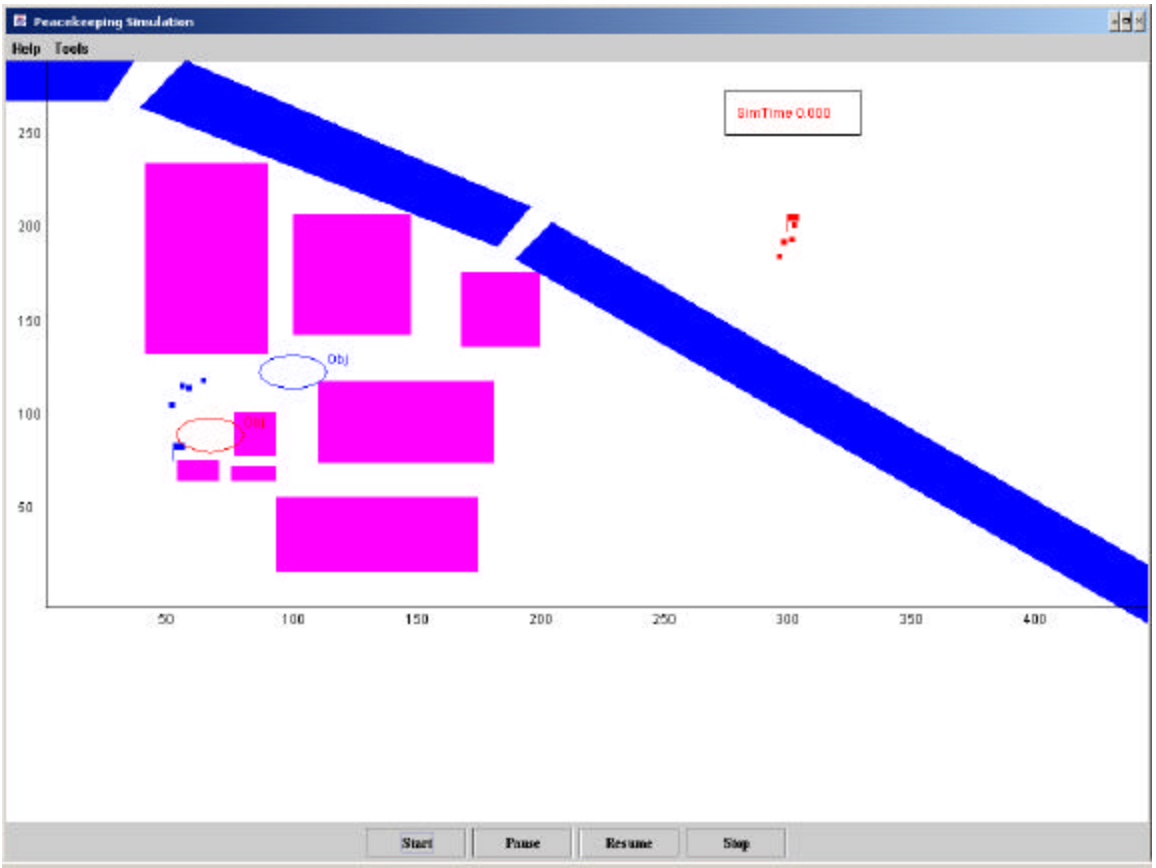

Figure 12. Peacekeeping Simulation Graphical User Interface

The "Peacekeeping Simulation" GUI (Figure 12) shows the simulation time, the terrain and its features, the blue and red objective, the blue and red secure position, and all agents together with a local coordinate system. Additionally it contains all buttons and menus to interact with the simulation.

The user can call for reinforcements during a simulation run using the "Tools" menu. This will instantiate a Reinforcement object, which displays the "Call Reinforcement" GUI. (Figure 13)

Figure 13. Call Reinforcement Graphical User Interface

The GUI gives the user the possibility to select between an Infantry Fighting Vehicle (IFV) or an Armored Personnel Carrier (APC) as reinforcement for the blue agents. Clicking the "OK" button will instantiate either a TryShoot.IFVMover or a TryShoot.APCMover object. A TryShoot.IFVMover object models the German IFV "Wiesel" and a TryShoot.APCMover models the German APC "Fuchs". (see Figures 14 and 15)

Figure 14. German Armored Personnel Carrier "Fuchs"
[From: Ref. 18]

Figure 15. German Infantry Fighting Vehicle "Wiesel"
[From: Ref. 18]

Both objects are agents with predefined capabilities and a predefined personality. Both are of type "Blue". The APC is armed with a 7.62 mm machinegun and the IFV with a 20 mm machinegun. Additionally both vehicles can fire smoke grenades. They will be shown at the blue secure place and advance towards the blue objective.

Figure 16 shows how the Reinforcement class links the first with the second layer and how the classes are build.

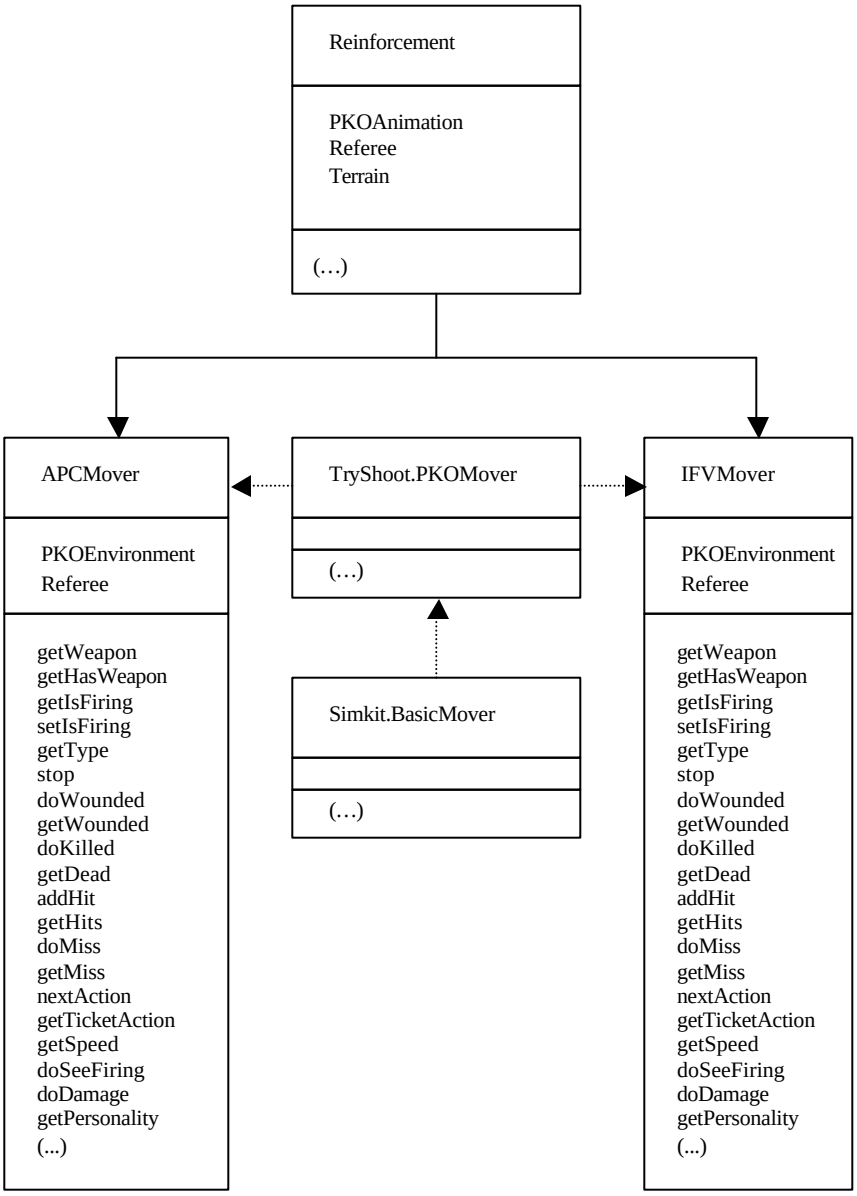

Figure 16. Second layer of the PKO simulation model (Part 2)

3. Agent Modeling Layer

The third layer consists of the objects each agent owns and which define its capabilities and intentions. (Figure 17)

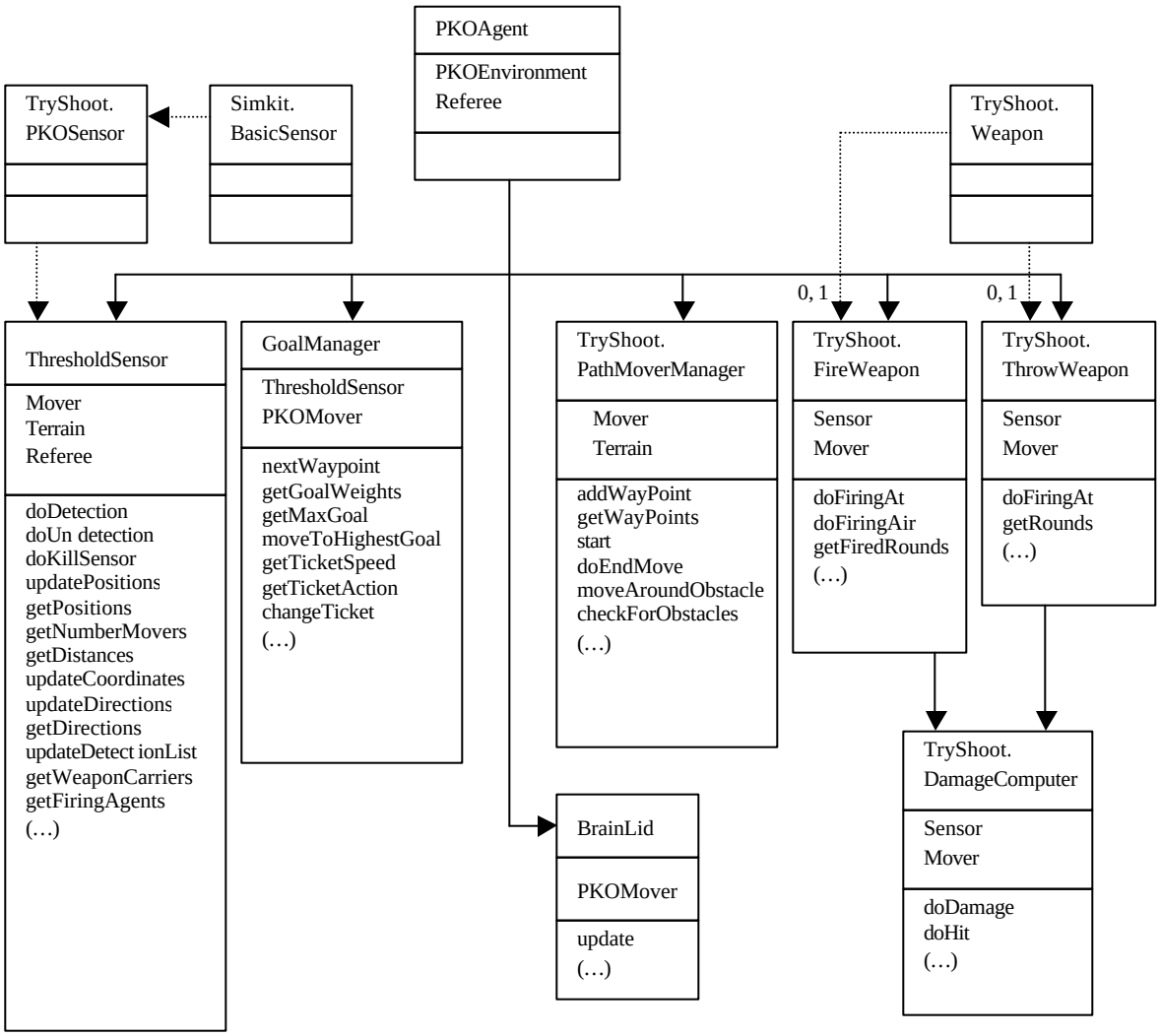

Figure 17. Third layer of the PKO simulation model

The BrainLid class instantiates an object that allows the user to get information about the agent's personality, inner environment, and goals by simply clicking at an agent in the simulation. The selection of an agent causes the BrianLid object to display a window with the desired information. (Figure 18)

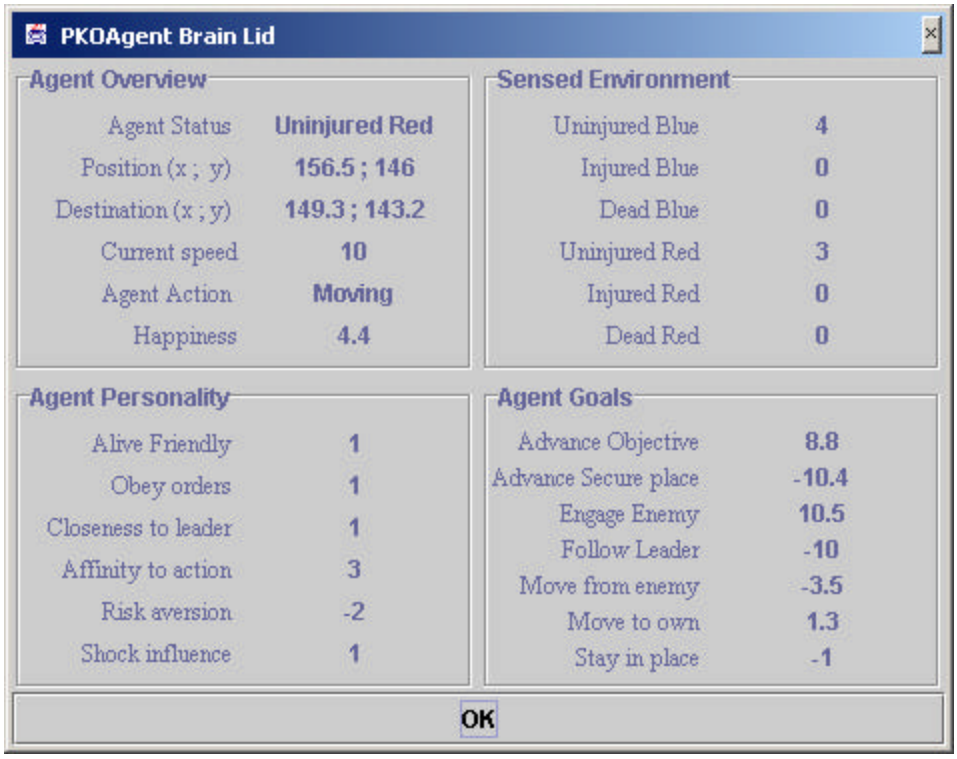

Figure 18. PKOAgent Brain Lid Window

The description of how the classes define the capabilities and the personality of an agent will follow in the next subchapter.

4. Agent Intention and Action Generating Layer

The fourth and final layer of the simulation uses the input of the agent-modeling layer and generates the agent's intentions and actions. How this is done will also be explained in the next section. (Figure 19)

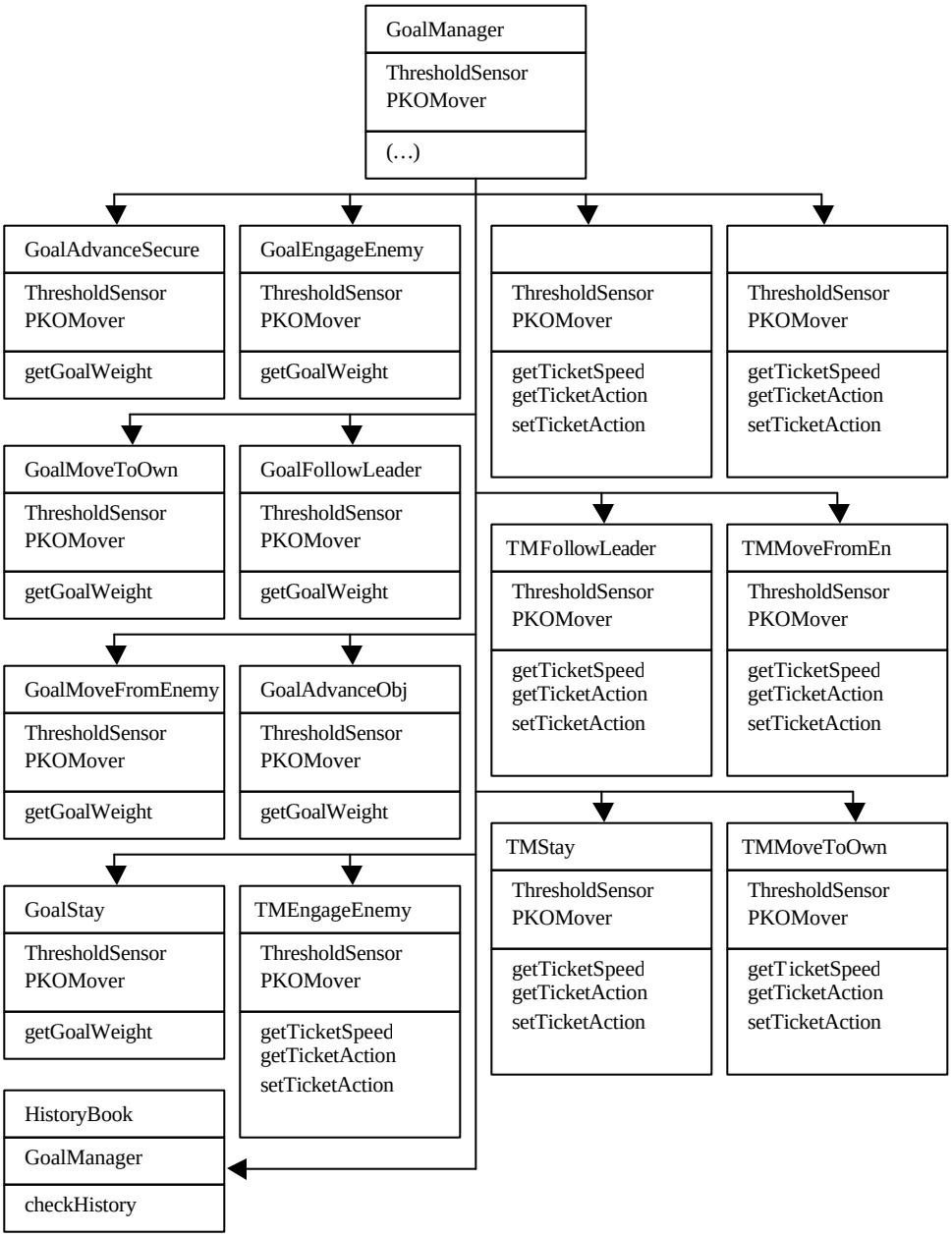

Figure 19. Fourth layer of the PKO simulation model

D. THE AGENT MODELING PROCESS IN PEACEKEEPING

According to the agent model the author developed in chapter 2.A, the agent needs detectors to perceive the outer environment, goals and tickets to generate actions, effectors to react in and towards the outer environment and to affect the complex adaptive system the agent is part of, and a history of model knowledge.

1. Characteristics, Abilities, and Effectors

Each agent has certain abilities and effectors to act in and affect the environment.

There are two kinds of abilities: those that are predefined and those that are set by the user. One way of interacting with the environment is to move. Therefore the PkoStandards class sets the agent's maximum speed (10 km/h for persons and 30 km/h for APCs and IFVs). Furthermore it defines the red and blue secure place, and the red and blue objective, possible places to move to.

Two other important agent characteristics are its starting position and its type. The user defines these characteristics by either choosing one of the predefined agents or when he or she creates a new agent. There are six different agent types in the simulation model: blue leaders, blue followers, red leaders, red followers, armored personnel carriers, and infantry fighting vehicles. A leader is instantiated at the exact starting position the user chooses. A follower is instantiated at a random location inside a square of 20-meter width around the position the user chooses. This is done to avoid displaying agents that sit on top of each other. APCs and IFVs are instantiated at the blue secure place, because this is the place where reinforcement would tactically be positioned.

Another way of interacting is to shoot or to throw stones at other agents. Therefore the agent's size and its local coordinate system for damage assessment is defined in the TryShoot.DamageComputer class. Figures 20, 21 and 22 show the

predefined sizes and coordinate systems for the modeled agent types. The origin of the coordinate is the aim point an opponent chooses, when it fires at an agent.

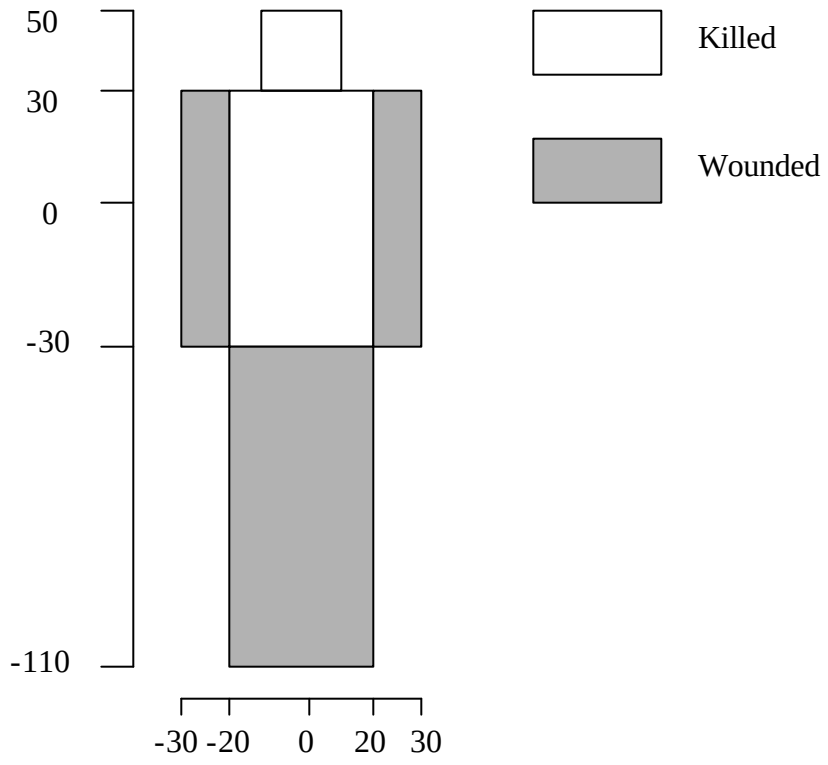

Figure 20. Coordinate system and size of a PKOAgent

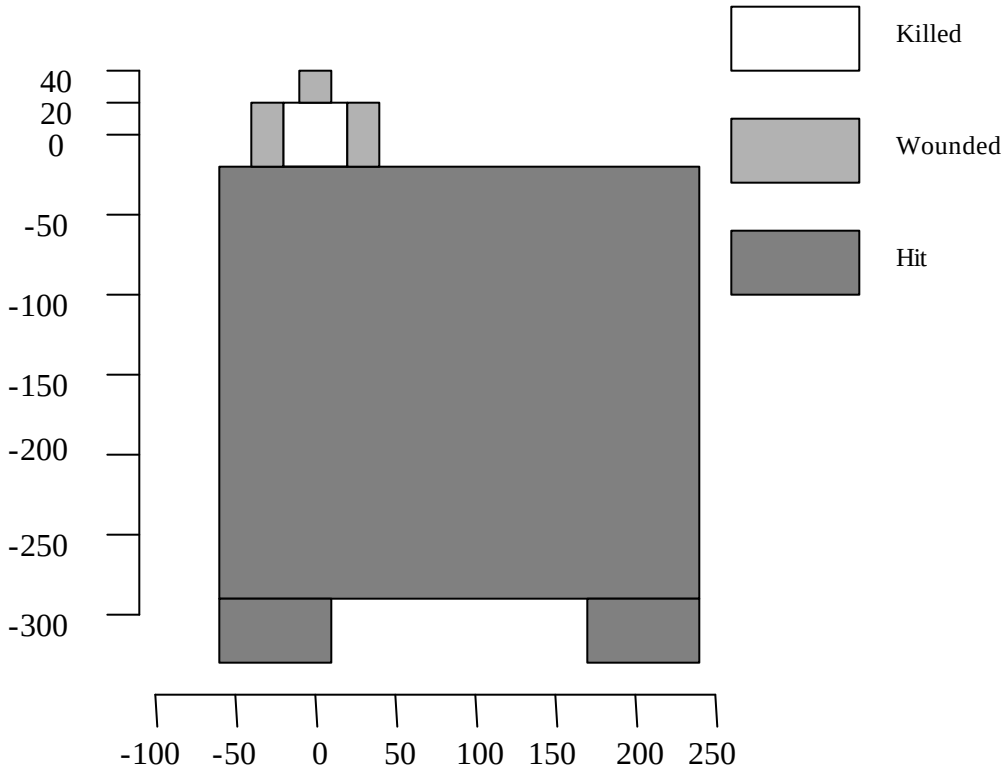

Figure 21. Coordinate system and size of an APCMover

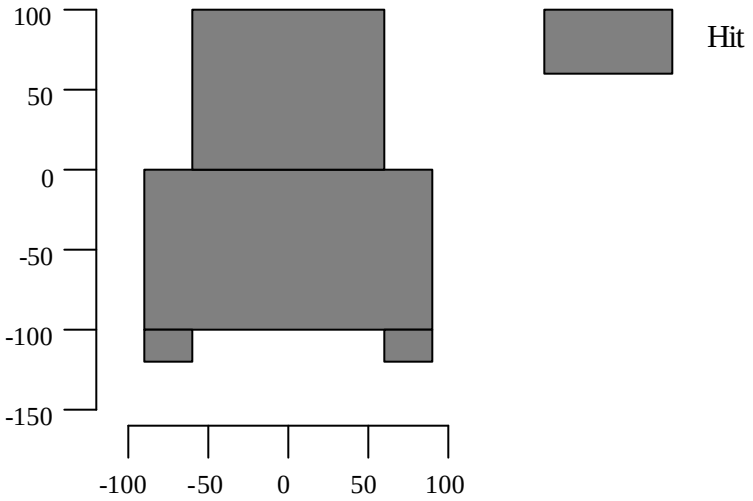

Figure 22. Coordinate system and size of an IFVMover

When the user defines the agent's abilities he or she can select a weapon for the agent. The possible weapons to choose from are typical weapons used in the German military and in civil war environments. The table shows the available weapons and their parameters. Sigma is the standard deviation for the impact point of a round caused by the weapon and the shooter:

sigma = circle * 100.0 / maxRange.

Weapon	max Range [m]	capacity	circle [m]	rpm	sigma [m]
G3	300	20	0.25	7	0.08
MP2	100	32	0.30	7	0.30
MG3	600	100	0.50	6	0.08
P8	25	15	0.15	7	0.60
G22	1100	5	0.15	4	0.01
G36	500	30	0.30	15	0.06
BMK	750	50	1.20	30	0.16

Table 1. Weapon parameter

If no weapon is selected, the agent has the ability to pick up stones and to throw them at opponents. The modeling parameters for a thrown stone are displayed in the following table.

Weapon	max Range [m]	capacity	rpm	sigma [m]
Stone	30	100	5	20

Table 2. Parameters for a thrown stone

Agents modeling armored personnel carriers or infantry fighting vehicles can also fire smoke grenades. A smoke grenade is fired at the center of the opposing group and creates a 10 m by 10 m obstacle. Sensors cannot detect through the obstacle but movers

have the possibility to move through. The obstacle disappears 10 minutes after it has been created.

The final agent ability is its training level, which like the agent type is either predefined or set by the user. The training level is used to generate the agent's reaction time, which is a normal random variable with mean ((10 - training level) / 100) minutes and standard deviation 0.01 minutes. The reaction determines the delay of the agent's action to fire its weapon or to throw a stone after it received an input from the environment that causes it to take this action. Additionally the training level is used to determine the standard deviation for the bivariate normal distribution, which generates the hit point of a bullet or stone in the TryShoot.DamageComputer class. This bivariate normal distribution has a mean of zero centimeters. Its standard deviation is sigma + (50 - training * 10) centimeters, with sigma being the weapons standard deviation.

2. Personality

Each agent possesses six personality characteristics. These are inputs for its beliefs, intentions and actions. The characteristics are:

- Closeness to friendly: it defines how important it is for the agent to stay close to friendly agents.
- Obey orders: it classifies how carefully the agent follows the given order to advance the objective.
- Closeness to leader: it characterizes how important it is for the agent to stay close to its leader.
- Affinity to action: it defines the agent' affinity towards action.
- Risk aversion: it describes how hard the agent tries to avoid risks.
- Shock influence: it characterizes how high the influence of an observed casualty is for the agent.

The first five characteristics are represented by integer values between ± 5 and the last characteristic is represented by an integer value between zero and five. A value of five for the personality characteristic obey orders, for example, would mean that the agent follows the order to advance toward the objective very closely, while a value of negative five for the same characteristic would mean that the agent does not care at all about the given order.

By setting the personality of an agent, the user defines its character, which will be used in the agent's goal and ticket function to determine its beliefs and intentions.

3. Detectors

The agent's main detector is a sensor. This sensor is modeled in the ThresholdSensor class, which extends Simkit.BasicSensor. The sensor is a cookie cutter sensor with a predefined range. This range is set to 200 meters in the PkoStandards class. The sensor models the agent's eyes; it detects other agents and discriminates their type. Furthermore, the sensor computes the distances toward the center of the group the agent belongs to, the distance towards the center of the opposing group, the distance towards the agent's leader, and the distances towards their own and opposing secure place and objective. The sensor also computes the direction towards the center of the opposing group and towards the own secure place. Finally the sensor stores the number of each type of agent and the number of opposing agents that carry a weapon and that shoot within the sensor range.

The sensor is modeled with no error and without time delay. Whenever an agent enters the sensor range it is immediately detected, its correct type is known and its correct position is used to update the computed distances, numbers, and directions.

The sensor checks if there is a building or smoke in the line of sight. If there is one, no detection occurs.

Each PKOAgent object has a method called "doSeeFiring". When an agent in the simulation fires it uses this method to inform all agents in its sensor range that it has

fired. For modeling purposes, this was a cleaner way to let other agents "see" that someone else in their sensor ranges is firing, rather than doing it through the sensor.

The TryShoot.PathMoverManager has methods that let an agent identify and move around obstacles. The sensing of obstacles is not modeled naturally by detecting them using a sensor, but by checking a database, which contains all obstacles. The reason for this implementation of obstacle detection is that the computational effort is smaller than an implementation of the natural way of "seeing" obstacles.

The "doSeeFiring", "checkForObstacles", and "moveAroundObstacle" methods are used as detectors in addition to the sensor. This is all the agent possesses to perceive its outer environment and to generate its inner environment.

4. Goals and Tickets

Each agent instantiates an object called "GoalManager" [Figure 17].

The GoalManager itself instantiates the agent's seven possible goals and their associated tickets [Figure 19]. The goals are:

- Advance towards the objective
- Advance towards the secure place
- Engage the enemy
- Follow the leader
- Move away from the enemy
- Move towards own agents
- Stay at the same place

Each goal has an associated goal function to compute its goal weight. The goal function uses the sensed environment and the agent's personality as input. The goal function for "Advance towards the objective" for example is computed as follows:

Force = (number sensed own leaders + number sensed own agents + 0.5 * number sensed own wounded agents) / (number sensed opposed leaders + number sensed opposed agents + 0.5 * number sensed opposed wounded agents) - risk aversion + affinity to action

Leader = 0.06 * distance to own leader * number sensed own leaders + closeness to friendly + closeness to leader

Goal weight = Force + Leader + obey orders.

Each time the agent reconsiders its intentions and actions all goal weights are recomputed. The goal with the highest weight becomes the active goal. The goal weight can be thought of as the agents desire to achieve the goal. The higher the goal weight, the higher is its desire to fulfill the goal. The ticket manager object that is associated with the active goal automatically becomes the active ticket manager and computes the agent's way to affect its environment. Therefore it generates two parameters, a speed and an action. The active ticket manager class uses the inner environment, the agent's personality, and the agent's history as inputs.

The history is modeled as an integer state variable in the HistoryBook class. The state variable can be thought of as stress under which the agent is. The agent's stress increases when it is unsuccessful and decreases otherwise. The agent's success is determined by comparing its current active goal, the associated goal weight, and its ticket with its most recent active goal, its associated goal weight and the most recent ticket. The agent is successful whenever the most recent goal and ticket are the same as the current ones, and the current goal weight is smaller than the most recent one. Whenever an agent changes either its goal or its ticket, the stress remains unchanged.

Each ticket manager has a "getTicketSpeed" method that computes the agent's speed and a "setTicketAction" with a ticket function to generate its action. Both methods use if-statements with the history, personality and inner environment as inputs for the computation and generation. Typical actions are move with half of the maximum speed towards your active goal, fire in the air while you do not move, or fire at an opponent

while you do not move. Other actions are to defend the objective or the secure place, to throw a stone, or to fire a smoke grenade.

III. THE MAP EXERCISE "PRIZREN"

This chapter presents a map exercise, which is used as a training tool at the German UN Training Centre, located in Hammelburg Germany. A simplified scenario is constructed that generalizes the map exercise. The scenario is then simulated using the Peacekeeping and TryShoot packages.

A. THE MAP EXERCISE

The map exercise takes place in PRIZREN, KOSOVO. German peacekeeping forces have been deployed to PRIZREN as part of the UN KFOR forces. The following extract of an order and three situation developments describe the current situation and the task for one company, 3./ Einsatzbataillon 1 of the TASK FORCE PRIZREN, which has been deployed to PRIZREN. The KPC and PBP, which are mentioned in the order, are groups that fought in the war and that are now illegal terror organizations.

Task order for 3./ Einsatzbataillon 1

Situation enforced 3./ Einsatzbataillon 1 / TASK FORCE PRIZREN 01 mar 2000

3rd Company has been deployed with TASK FORCE PRIZREN for 1 month.

The situation in the populace is tense but mainly stable and calm.

Intent of the battalion, tasks for the companies and task organization is attached.

3. Execution

a. Intent

(1) TF PRIZREN enforces the rules of the undertakings and of the military-technical agreement (MTA) in the whole area of responsibility (AOR). It ensures security and order by day and night through employment of most of

the battalion in an infantry mission – schwerpunkt PRIZREN – ALTSTADT. And it holds freedom of action with the battalion reserve (3 platoons).

(2) Employment of forces:

(a) one enforced company in PRIZREN, south of the BISTRICA

(b) one company in PRIZREN, north of the BISTRICA

(c) the air-mobile mechanized company (NL) south-east of the town on both sides of the line of communication (LOC) LION

(d) cavalry platoon in an area north of the town, along the LOC DUCK and east of it

b. Task

(1) 3./ EinsBtl 1

(a) - Monitors ALTSTADT

- Shows high military presence in the town including conversation reconnaissance.
- Enforces public security and order including the curfew.
- Stops illegal KPC actions.
- Controls abandoned houses.
- Detects arson and immediately fights fire.
- Prevents plundering and infringements on persons.
- Controls suspicious persons.

(b) Occupies observation point BURGRUINE, observes so ALTSTADT and so simultaneously prevents its use by other forces.

(c) Mans post at ALTSTADTBRUECKE.

(d) Protects EPISCOPAL SEE, KLOSTERSCHULE and BERGKAPELLE and prevents plundering and arson.

(e) Guards communications station on CVILIEN- Mountain.

(f) Reconnoiters and operates depending on the situation temporary checkpoints (TPC).

(g) Detaches in weekly change with 4./ EinsBtl 1 (Fridays 0800) the company headquarters squad for the brigade reserve, including medical

squad and maintenance squad, movement readiness within 120 minutes.

(h) Detaches from available forces (in change) one personnel carrier Fuchs with crew to cavalry platoon for the task battalion reserve, movement readiness within 30 minutes.

To fulfill its order the company has the task organization that is shown in Figure 23 and the organization, which is shown in Figure 24.

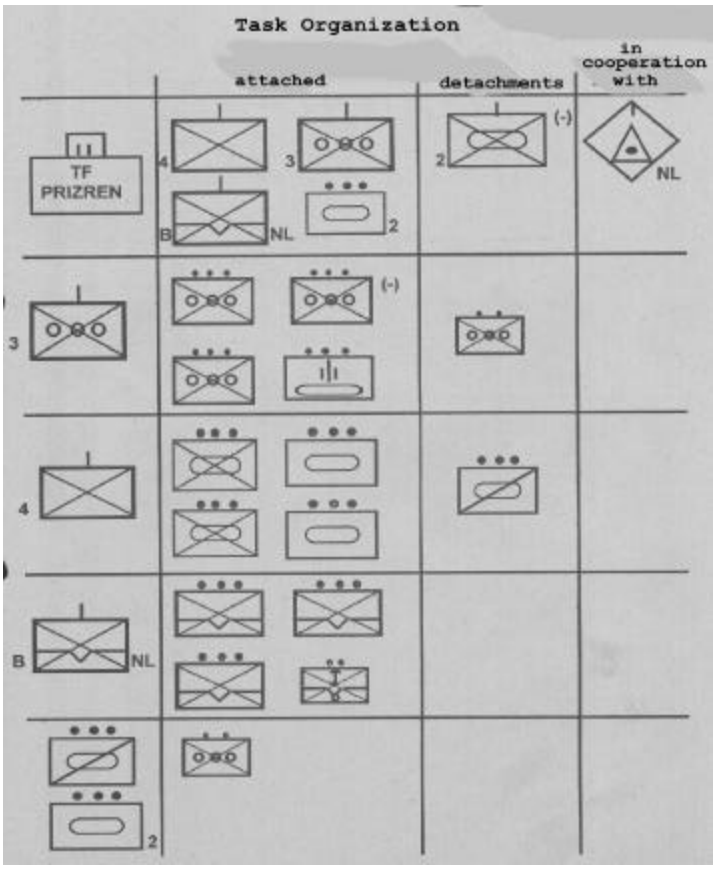

Figure 23. Task organization of 3./ Einsatzbataillon 1

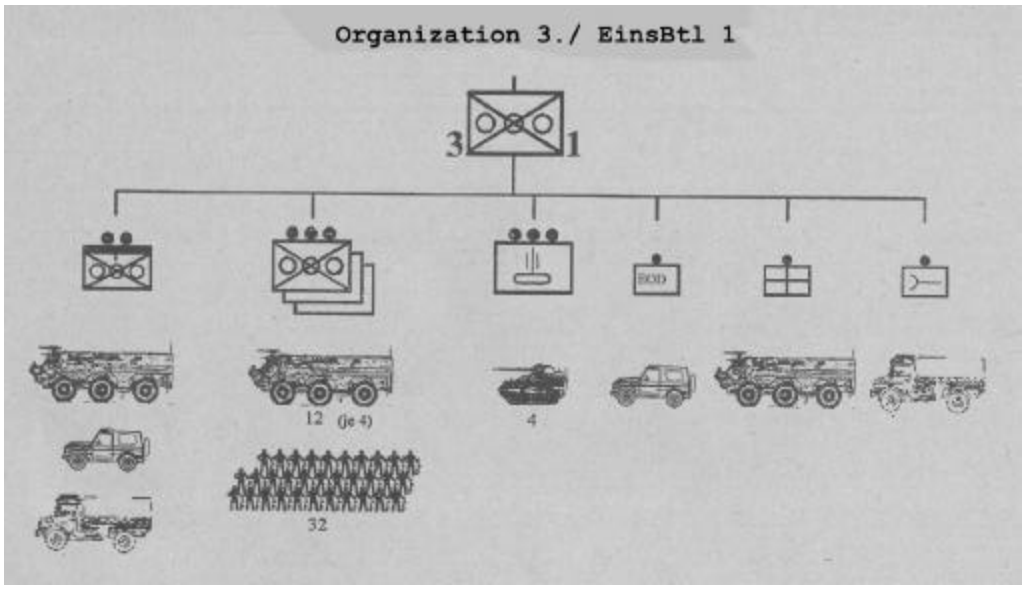

Figure 24. Organization of 3./ Einsatzbataillon 1

First situation development until March 01, 2000, 1000 a.m.

As a result of the patrol reports and the conversation reconnaissance form yesterday, a demonstration (about 3000 persons) organized by the GTK (the former PBP) is expected for March 2 (before noon). During this demonstration the BISCHOFSSITZ shall be stormed.

Therefore 3rd company got the following order at 011000 mar: "New schwerpunkt 3rd company – part (d). All other tasks are cancelled. Presentation scheme of maneuver (d) until 011600 to be approved by the battalion!"

Second situation development until March 2, 2000, 1000 a.m.

The company commander's scheme of maneuver was approved. 3rd company has been in the planned positions for 90 minutes.

One movable speaker squad is in cooperation with the company. The squad includes an interpreter.

Until 020950 Mar the company commander received much information about the protesters who are moving towards the EPISCOPAL SEE from the commander of 4th company. At 1000 the rally started at the intersection and in front of the UNMIK (United Nations Interim Administration Mission in Kosovo) building, which was protected by the UNMIK police.

The rally (about 3000 participants) has been organized and escorted by TMK personnel. Participants are men and women in civilian clothes and a few children. No weapons could be seen until now. A few protesters carry signs in Albanian language with the words “SERBS OUT!” – “Long live the Albanian Kosovo”. The main speaker is the former field commander Braci, who in the past has occasionally been seen as the head of a moderate group of the TMK. At 1015, at the end of Braci’s speech, a group of about 100 men, who dissolve from the audience, march over the eastern BISTRICA Bridge. They stop around 10 meters in front of a hastily installed barbed wire concertina fence and are observed by an APC Fuchs (1st Squad) of the 1st platoon, that has the task to block the entrance to the Altstadt. The men shout: "NATO out! Serbs out!" The leader and 12 soldiers of the 1st platoon are standing between the APC and the barbed wire. In all weapons a round is chambered. After initial protest shouts and united shouts of "NATO out! Serbs out!", the protesters try to breach the fence using boards and planks. The crowd’s behavior – now there are more than 200 men – gets more and more hostile. They throw the first stones; plastic bags filled with paint hit the APC and the soldiers, and the first soldiers get hit by stones. Master Sergeant Steiner, leader of the 1st platoon, is now with 3 soldiers of his platoon squad at the APC. The first hand-to-hand fights start…

Third situation development until March 2, 2000, 1230 a.m.

The hand-to-hand fighting and the attacks at the 1st platoon were curtailed temporarily by Master Sergeant Steiner’s decision to fire warning shots in the air. A short time later, another hostile crowd develops in the area of the 2nd platoon, which is located at the two western streets that lead to the church / EPISCOPAL SEE. There hand-to-hand

fighting also started, but these demonstrators retreated about 50 meters after warning shots were fired and the soldiers threw smoke pots. The crowd here continued to shout, to throw stones and bags filled with paint, and to threaten with planks.

At 1145 the leader of the 1st platoon reports that the demonstrators on the BISTRICA Bridge are storming his position again and that they are throwing stones at his soldiers. At that time Captain Traube is with his company headquarters with the 3rd platoon, which is positioned at the church / EPISCOPAL SEE, and immediately sets up the inner defense ring around the church. He orders the 1st and 2nd platoon to deplete the outer ring and to withdraw to prepared positions in the inner ring. This lasted until 021210 Mar.

After a short calm period, approximately one hundred demonstrators surround the area around the church. At 1230 the company commander has the following situation picture [Figure 25]:

a. In front of the church and on the playground a crowd has gathered (about 200 persons, around 100 meters away). Out of this crowd stones are thrown. Until now these stones have only hit the barbed wire fences and the sand barricades. Some of the people in this crowd are children and juveniles.

b. Simultaneously the company commander observes two pick-up trucks that stop at the eastern access road. Two or three adults unload them. The company commander identifies fuel canisters, observes bottles being filled, and other objects. (Distance around 200 m)

c. Some shots are fired from the multi-story building. (Distance around 250m / upper floor, west of the church) A little later the leader of the 3rd platoon reports: “Someone from the direction of the multi-story building fired at us. Two soldiers are seriously injured. Request permission for my two snipers to fire.”

d. At 1230 the leader of the 2nd platoon, who is positioned on the east side of the church, reports that a group of about 50 muffled up demonstrators are storming the barricades. (Distance 75 m in front of his position) They are preparing to storm the

church area throwing incendiary bottles and stones. At this time there are no casualties to report.

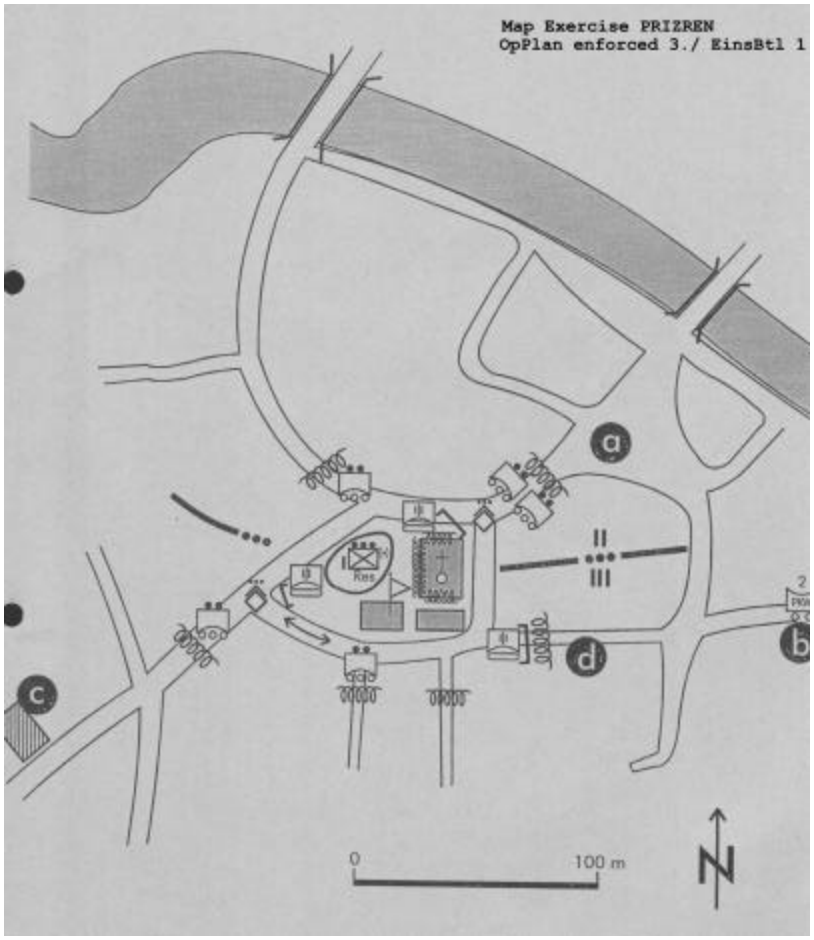

Figure 25. Situation of 3./ Einsatzbataillon 1 at 021230 Mar

B. A GENERALIZED SCENARIO

The idea of tactics and training development via agent-based simulation will be tested on an abstraction of the map exercise PRIZREN. The author will generalize that part of the map exercise, where a crowd consisting of violent protesters, juveniles, and children gather in front of the church. A small peacekeeping force armed with G3 rifles will guard the church. In one of the scenarios they can call for an APC FUCHS as reinforcement. Their implied mission is to minimize violence. Facing them is a group consisting of persons armed with G3 rifles and unarmed bystanders. The armed persons in this group seek to exert violence and seize the church. The bystanders will either follow the aggressors or avoid violence. By using these two types of groups on the side opposing the peacekeepers, the author hopes to generate behavior typical of peacekeeping missions. The urban environments in which peacekeeping missions take place are usually crowded [Ref. 19; p. 13] and the violent protesters are usually intermingled with unarmed persons, including women and children [Ref. 19; p.20]. Those unarmed protesters do not always flee a violent scene, but some of them get engaged [Ref. 19; p. 20]. Assume that the peacekeepers had several different trainings and tactics. The different trainings, which will be represented by different personalities, will be evaluated using Peacekeeping and TryShoot. For completeness, the different trainings and tactics will be tested against crowds of different personalities and of varying composition. Combinations of tactics and training against different crowd compositions can be studied using experimental design with the Peacekeeping and TryShoot model.

The mission of the peacekeeping force is to ensure and keep order in an urban zone torn by a civil war as described in the map exercise. Parts of the local population are driven by profound discontent with the current situation. Another part of the population is easily convinced to take part in protests.

The peacekeepers, having limited assets, deploy squads or platoons armed with rifles to guard assigned sites; in the simulated case, this is the Bishop's cathedral. Either armored personnel carriers or infantry fighting vehicles can reinforce the troops. The reinforcements are positioned at the peacekeepers command post. The policy for the

peacekeepers is to call for reinforcement when they think that they can no longer hold the position themselves. An early call for reinforcement will make it inaccessible for other troops.

The peacekeepers mission is to guard the cathedral, but in this, they have two other large concerns: to ensure their own survival, and to use the minimal force necessary to fulfill their mission. There are three measures of effectiveness for success: The primary measure is the survival of peacekeepers. The second measure is to avoid access to the site. Despite guarding the site in one position, it is still possible that engaging persons pass the peacekeepers and gain access to the defended site. If this happens, the peacekeeper's mission failed. The third measure is the number of persons killed or wounded in the crowd. To ensure the overall success of the peacekeeping mission inflicted deaths and injuries must be minimized. On the other hand, minimizing casualties on the side of the crowd by getting killed or wounded does not fulfill the mission either. A balance must be found between the two possibilities.

This scenario will be modeled as a simulation using Peacekeeping and TryShoot. The simulation will be used to conduct a two-factor, two-level experiment. The two factors are training and tactics, and each has two levels.

C. GENERATING THE MODEL

The author will apply the techniques of modeling CAS with agents to formulate the above-described scenario using Peacekeeping and TryShoot. The individual participants are modeled as agents with different kinds of personality and capabilities. All agents are instances of the PKOAgent class. The agents are then placed into the simulated reality and allowed to interact.

The environment the agents are supposed to interact in is a highly generalized PRIZREN with predefined objectives and secure places for both parties. [See Figures 26 and 27]

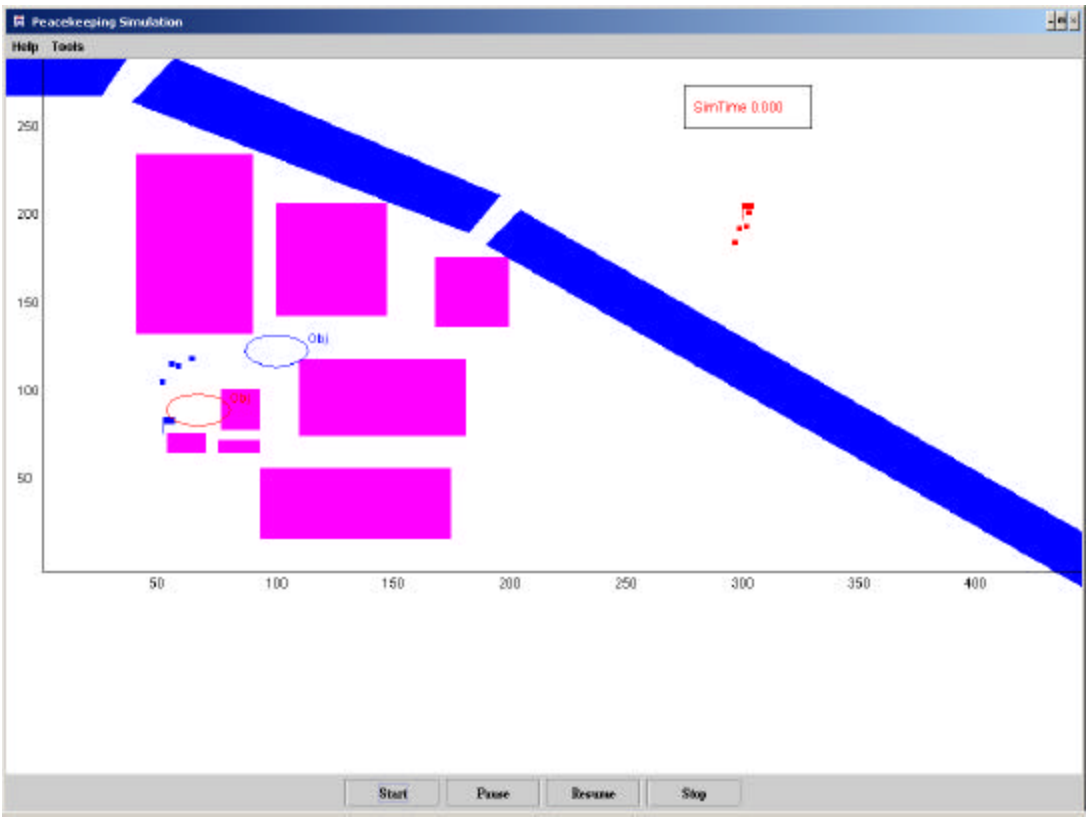

Figure 26. Generalized PRIZREN environment

Figure 27. Legend for the generalized PRIZREN environment

The simulation places a squad of peacekeepers at their command post and a group of protesters at their secure place. Both groups start to move towards their objectives. The group members always consist of a given number predefined agents. The parameters of those agents are contained in the following tables.

File Name	Description	Color	Type	Weapon	Number	x Coord	y Coord
Agent1	blue_follower_medium	Blue	Follower	G3	4	50	100
Agent2	red_follower_aggressive	Red	Follower	G3	5	300	200
Agent3	red_follower_bystander	Red	Follower	none	2	300	200
Agent4	blue_leader_medium	Blue	Leader	G3	1	50	100
Agent5	red_leader_aggressive	Red	Leader	G3	1	300	200
Agent6	blue_follower_defensive	Blue	Follower	G3	4	50	100
Agent7	blue_leader_defensive	Blue	Leader	G3	1	50	100
Agent8	red_follower_moderate	Red	Follower	G3	3	300	200
Agent9	red_leader_moderate	Red	Leader	G3	1	300	200

Table 3. Characteristics of the programmed agents

File Name	Training Level	Alive Friendly	Obey Orders	Closeness to Leader	Affinity to Action	Risk Aversion	Shock Influence
Agent1	4	2	3	2	0	-2	0
Agent2	2	1	1	1	3	-2	1
Agent3	0	4	0	0	-1	2	3
Agent4	5	2	4	0	0	-2	0
Agent5	2	1	3	0	3	-2	1
Agent6	4	2	4	2	-1	-1	0
Agent7	5	2	5	0	-1	-1	0
Agent8	1	2	1	1	1	1	1
Agent9	1	1	3	0	1	1	1

Table 4. Personalities of the programmed agents

The first peacekeeper setup models an aggressive approach by the peacekeeping forces, which try to dissolve the crowd.

The second peacekeeper setup models a more defensive approach by the peacekeeping forces, which mainly try to protect the cathedral. In this model the peacekeepers call for reinforcement as soon as the first casualty either on their or their opponents side occurs.

Different crowds will approach the red objective and interact with the peacekeeping forces. All crowd setups will use some of the programmed agents.

The simulation characteristics will be the same for all types of experiments. They are shown in the following table.

Deterministic	TRUE
Terrain	TRUE
Break Point	5

Table 5. Simulation characteristics for all experiment types

IV. SIMULATION RESULTS

One application of the simulation model developed in this thesis is, as mentioned before, to observe the performance of the agents during a single simulation run and to exploit their behavior. This model application is not very useful because the behavior in, and the results of, a single simulation run depend highly on randomness. Therefore the observation of a single simulation run cannot be used to obtain the desired insights from the model.

A better use of the model is to estimate measures of effectiveness based on analyzing multiple runs. A difficulty arises due to the lack of input data. Consequently the sensitivity of the model to various parameters has to be explored. For this thesis the author will focus on the influence different starting conditions have on the simulation outcome.

A mathematical model will be developed that allows qualitative predictions for known starting positions.

Note that the obtained results can only be used for a comparative analysis that evaluates which of the chosen starting positions is favorable for the peacekeeping forces.

A. MEASURE OF EFFECTIVENESS

To compare the outcomes of the different simulation runs, a numerical value consistent with the decision-maker's objective must be assigned to each outcome [Ref. 20; p. 11]. Therefore a measure of effectiveness (MOE) that satisfies the following properties will be assigned to each simulation outcome.

a) It will be quantitative.

b) It will be estimable from the output data.

c) A significant increase in the MOE value will correspond to a significant improvement in achieving the decision-maker's objective.

d) It will reflect both the benefits and the penalties of the simulation outcome.

[Ref. 20; p. 12]

There are no explicit procedures for choosing an appropriate MOE; every decision-maker has personal MOEs. This thesis uses utility theory to develop an MOE for a simulation outcome. Utility functions are frequently used to evaluate the overall performance of a system, where the performance of the system consists of the achievement of multiple objectives. A decision maker assigns a weight, which indicates the importance to each objective. Additionally a utility function will be constructed for each objective. The utility function will assign a value of zero to the least favorable reward and a value of 100 to the most favorable reward. [Ref. 20; p. 40] Finally the weighted sum of the utility values of the different objectives will be computed. This weighted sum is a single number, which is easily comparable to other results achieved by different inputs for the utility function. In conclusion one can say that utility functions are a way to compare apples and oranges by combining them to fruit salad.

In this thesis utility values will be assigned to each considered measure of a simulation result. Then the utility of the different achievable objectives will be evaluated using a nominal utility function based on the author's experience and training.

The different objectives of the peacekeepers are:

a) Minimize access to the red objective.

b) Minimize the number of peacekeepers that are killed.

c) Minimize the number of peacekeepers that are injured.

d) Minimize the number of protesters that are killed.

e) Minimize the number of protesters that are injured.

Finally the overall MOE will be a function of the utilities for the different objectives. This will be an extension of Slutzky's utility definition into military purposes. Slutzky states: "The utility of a combination of goods is a quantity possessing the property of assuming greater or less value according to the degree of preference for the combination expressed by the individual concerned." [Ref. 21; p. 184] In this thesis the

goods relate to the different goals and the individual concerned relates to the decision maker. This implies that the overall MOE is a function of the different utilities.

$$MOE = f(U_{Obj}, U_{kp}, U_{wp}, U_{kpr}, U_{wpr}) \qquad \text{with } U = \text{utility}$$

1. Utility Functions

In this part of the thesis the utility functions for the five different goals described above will be developed. The particular utility functions depend on the decision-maker. In this thesis the author acts as the decision maker and therefore the utility functions represent his opinion.

The peacekeeper squad always consists of five soldiers; therefore the utility functions for the peacekeepers evaluate the utility for at most five peacekeepers. Similarly, there are at most 12 protesters in the different scenarios. As a result the utility for up to 12 protesters is evaluated in each of the utility functions that concerns the protesters.

The mission in which the peacekeepers are involved fails as soon as at least one opponent reaches his objective. When some protester reaches the EPISCOPAL SEE it does not matter any more how many peacekeepers were wounded or how many protester were killed before. That the peacekeepers could not defend the EPISCOPAL SEE makes the citizens inside the peacekeepers area of responsibility believe that the peacekeeping forces are not able to protect them. A good example for such a result was the failure of the Dutch battalion, which was stationed in Srebrenica, Bosnia. Its task was to protect the Muslim population of Srebrenica against advancing Serbian troops. The rules of engagement made it impossible for the Dutch peacekeeping forces to fulfill this mission. Srebrenica was captured by the advancing Serbian troops, which then committed atrocities in the Dutch forces area of responsibility. These events demonstrated to the Muslim population of Bosnia that the peacekeeping forces were not able to protect them. Despite the fact that the Dutch commander had saved the lives of all his soldiers, the mission was a total failure and led to rules of engagement changes for the international peacekeeping forces in Bosnia. Consequently the goal to minimize access to the red

objective is either achieved or fails; there is no solution in between. This leads to the utility function shown in Figure 28.

The number of peacekeepers that are killed obviously affects the opponents' possibility of getting access to their objective. As more peacekeepers get killed the squad's combat power decreases. Since equal training is assumed for all peacekeepers and there are very few synergetic effects in defense operations the way they are conducted in peacekeeping operations, one can assume that two killed peacekeepers are twice as bad one killed peacekeeper and so on. Consequently the utility function for killed peacekeepers is linear. Its shape is shown in Figure 29.

A wounded peacekeeper has less combat power than an uninjured one but is still able to fight. Therefore the utility is greater than zero even when all peacekeepers are wounded. Applying the same assumption about training and synergetic effects that was used for killed peacekeepers leads again to a linear utility function. It is shown in Figure 30.

The peacekeepers goal is to defend the EPISCOPAL SEE without inflicting casualties. Casualties might lead to the failure of the whole mission. Killing opponents is regarded as much worse than inflicting injuries. Killing the first protester has the highest influence on the overall peacekeeping mission, since the peacekeeping crosses the line of defending peace without acting heavily violent. Further kills do not influence the overall mission as much as the first kill. This consideration leads to a kink in the utility function, which is shown in Figure 31. Figure 32 shows the utility function for wounded protesters.

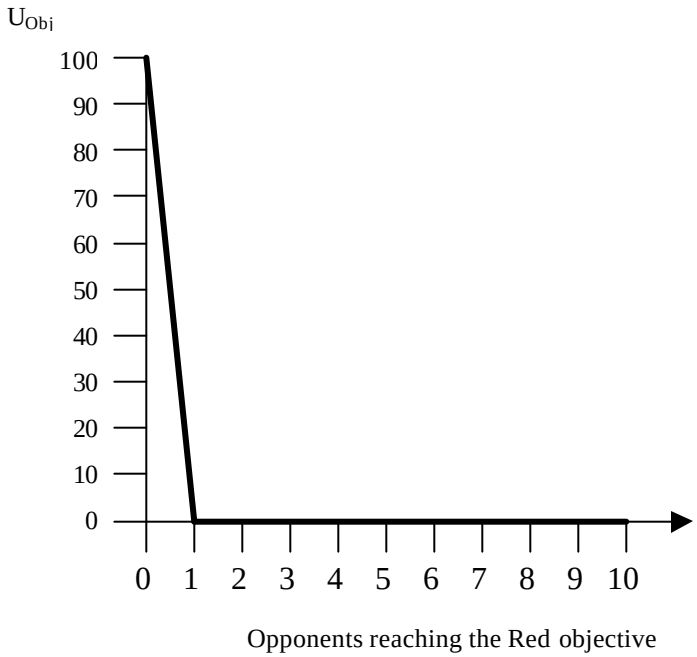

Figure 28. Utility function for the goal to minimize access to the objective

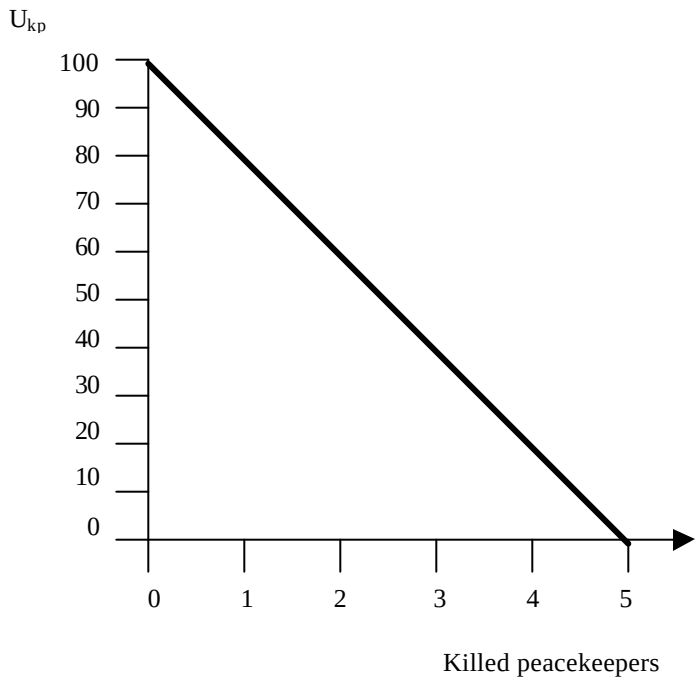

Figure 29. Utility function for the number peacekeepers that are killed

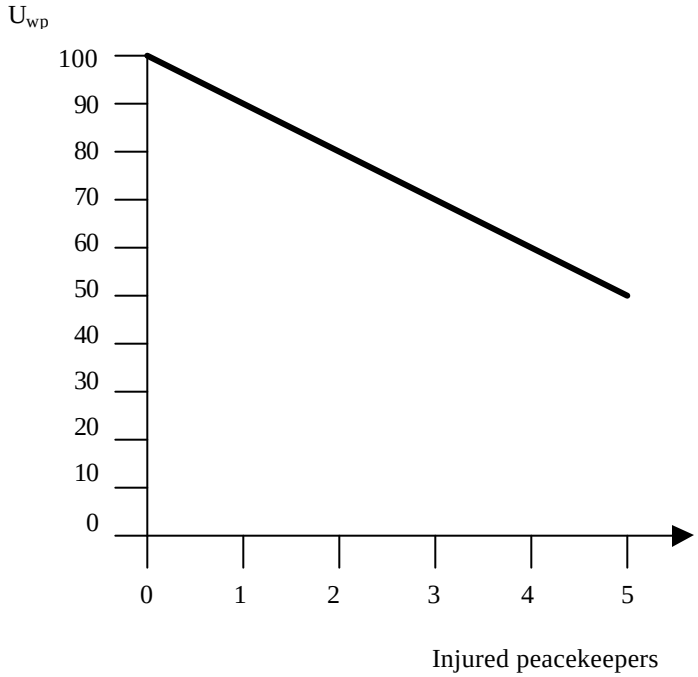

Figure 30. Utility function for the number of peacekeepers that are wounded

Figure 31. Utility function for the number of protesters that are killed

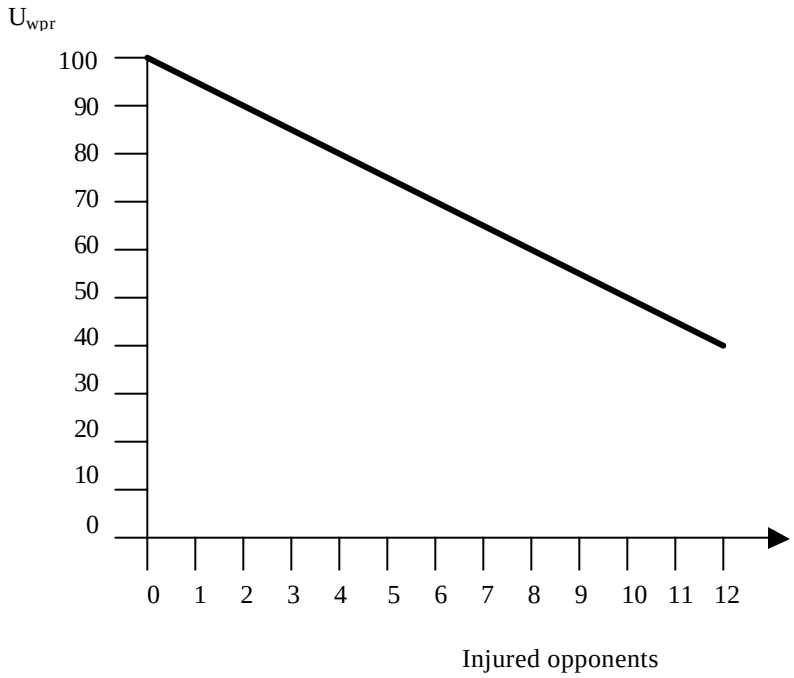

Figure 32. Utility function for the number of protesters that are wounded

2. Deriving the Function for the Measure of Effectiveness

Since the mission is a total failure whenever some protester reaches the EPISCOPAL SEE, the weighted utility for reaching the objective (U_{Obj}), which is either zero or 100, will be multiplied by the sum of the other weighted utilities. The overall MOE should be either zero, which indicates mission failure, or the result of the four remaining utilities. Therefore the weight for the utility parameter for reaching the objective (w_{Obj}) is a scale factor and chosen to be 0.01. Multiplying U_{Obj} by the scale factor leads to zero, whenever U_{Obj} itself is zero, and to one whenever U_{Obj} is 100. In conclusion 0.01 is the correct scaling factor to achieve the desired overall MOE result.

The relationship between the four remaining goals needs to be considered to establish the other weights. To set a limit on the value of the weights and to easily show the relationship between the different weights, the sum of the four remaining weights will be constrained to sum to one, with all individual weights greater than zero.

It is worse when a peacekeeper gets killed than it is when one gets injured, since a wounded peacekeeper still has some combat power and is able to partially fulfill his mission.

Additionally for the overall mission it is worse when a protester gets killed than when a peacekeeper gets killed. A dead peacekeeper has a high negative influence on the morale of the peacekeeping troops and on the support by the public of the nation that sent these troops. But when the peacekeeping forces kill a protester they endanger the complete mission, they can lose public support in the host nation, and may become part of one side of the conflict they are trying to solve.

A wounded peacekeeper is worse than a wounded protester because wounding a protester shows the peacekeepers' will to fulfill their mission and might cause the other protesters, especially bystanders, to reconsider their actions. Additionally the protesters lose combat power.

Finally, killing a protester is worse than wounding one. The same arguments that were used to explain why it is worse to kill a protester than a peacekeeper and why it is worse when a peacekeeper is wounded than when a protester is wounded apply here also.

Implementing the arguments from above, the author, as the decision-maker, establishes the following weights:

- Killed peacekeepers: $w_{kp} = 0.40$
- Wounded peacekeepers: $w_{wp} = 0.10$
- Killed opponents: $w_{kpr} = 0.45$
- Wounded opponents: $w_{wpr} = 0.05$

The arguments that led to the weights and the relationship between the different weights is subjective; another decision-maker might choose different weights.

Using the aforementioned relations, the function to evaluate the MOE (M) for each simulation run is:

$$M = w_{Obj} * U_{Obj} * (w_{kp} * U_{kp} + w_{wp} * U_{wp} + w_{kpr} * U_{kpr} + w_{wpr} * U_{wpr})$$

With the established weights the MOE value for each simulation run is the result of the following formula:

$$M = 0.01 * U_{Obj} * (0.40 * U_{kp} + 0.10 * U_{wp} + 0.45 * U_{kpr} + 0.05 * U_{wpr})$$

Sensitivity analysis was performed to assess what effect small changes in the weight values have on the calculated MOE value. To check this MOE, values were computed for different numbers of wounded and killed peacekeepers and protesters. The chosen numbers for wounded and killed agents of both sides are shown in Table 6.

sample	blue injured	blue killed	red injured	red killed
1	1	0	3	5
2	1	1	3	4
3	1	2	3	3
4	1	3	3	2
5	2	0	2	5
6	2	1	2	4
7	2	2	2	3
8	2	3	2	2
9	3	0	1	5
10	3	1	1	4
11	3	2	1	3
12	3	3	1	2

Table 6. Inputs for weight value sensitivity analysis

These inputs were applied to different weights, which satisfied the conditions developed in this chapter. In the first setup the suggested weights were applied (Series 1). In the second setup the weights for wounded agents were changed (Series 2); in the third setup the weights for killed agents were changed (Series 3); and in the fourth setup both changes were made simultaneously (Series 4). The setups are shown in the following table.

weight type	Series 1	Series 2	Series 3	Series 4
Objective	0.01	0.01	0.01	0.01
wounded peacekeepers	0.10	0.12	0.10	0.12
killed peacekeepers	0.40	0.40	0.42	0.42
wounded protesters	0.05	0.03	0.05	0.03
killed protesters	0.45	0.45	0.43	0.43
sum	*1.00*	*1.00*	*1.00*	*1.00*

Table 7. Chosen weights for the sensitivity analysis

The sensitivity analysis illustrates that the magnitude of the MOE value changes for different setups but the relative relationship between computed MOE values for the different casualty inputs is comparable for all chosen weights. The result is graphically displayed in the following figure.

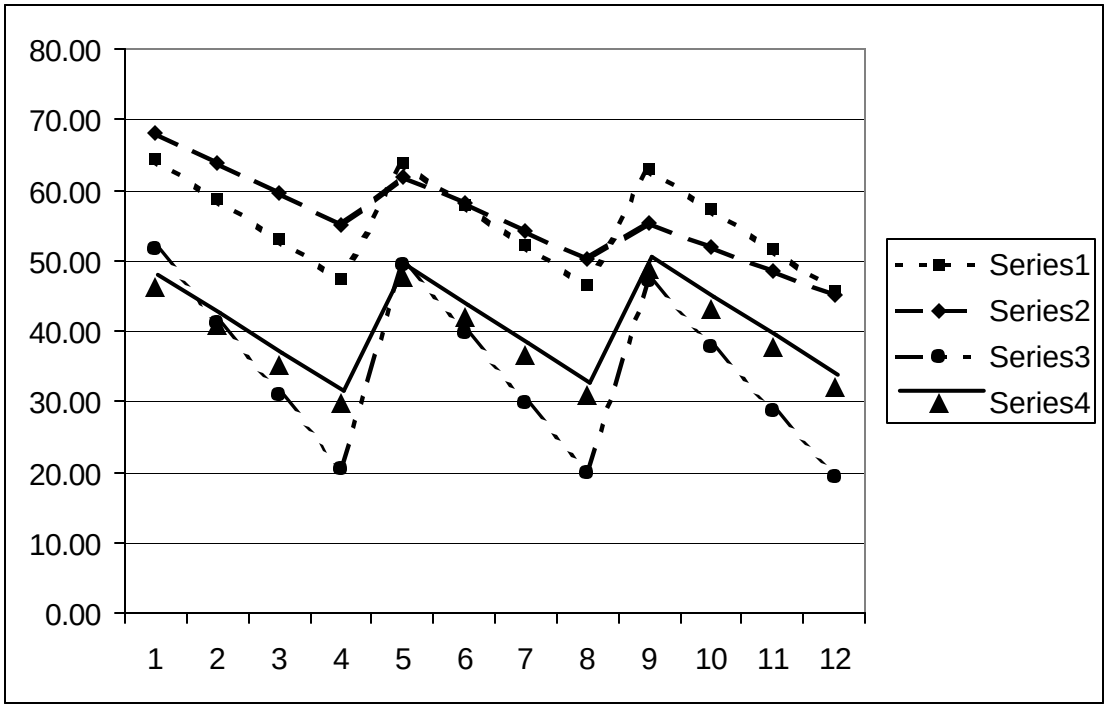

Figure 33. Graphical display of the results of the sensitivity analysis

Since the computed MOE value in itself has no meaning but is only used in comparison to other MOE values, the different computed magnitudes for different setups are no problem for the analysis conducted in this thesis. Small changes in the weight values do not influence the relative relationship between MOE values for different casualties, but this is exactly what the analysis in this thesis observes. In conclusion, the chosen weights can be applied, and small changes on the weights do not effect the relative relationship between computed MOE values.

B. DESIGN OF EXPERIMENT

As mentioned before, different kinds of crowds will advance towards the red objective and interact with the peacekeeping force. Three factors, the aggressiveness (aggr), the number of armed protesters (prot), and the number of bystanders (byst) define the crowd. Each of the factors has two possible levels; the aggressiveness of the armed protesters is either "aggressive" or "moderate", the number of armed protesters is six or eight, and the number of bystanders is two or four. The peacekeepers are defined by only one factor, their tactics. This factor has two levels; the peacekeepers' tactics, which is either "moderate" or "defensive".

A three factor two level experiment has 2^3 design points for each of the peacekeeper's levels. This leads to the following design of experiment for each of the peacekeeper's tactics.

Factorial Design for two Levels and three Factors		
Protester tactics	Number of armed protesters	Number of bystanders
aggressive	8	4
moderate	8	4
aggressive	6	4
moderate	6	4
aggressive	8	2
moderate	8	2
aggressive	6	2
moderate	6	2

Table 8. Experimental design

A factorial design has been chosen because it is possible to estimate main effects and interactions with maximum precision. [Ref. 22; p. 342] Furthermore it requires relatively few runs per factor studied. [Ref. 22; p.306] In the case studied in this thesis the factorial design requires $2 * 2^3$ runs of the simulation.

In each experiment the protesters' and peacekeepers' starting position and their objectives were the same. The engagement also always took place in the same terrain. This setup ensures that the possible differences in force or crowd size and tactics between the different engagements are comparative and it additionally eliminates known sources of discrepancy, for example different target acquisitions as a result of line of sight problems in urban terrain. [Ref. 22; p.105-106]

To have a higher number of degrees of freedom for the analysis and to produce an accurate measure of errors by replication, each design point is sampled ten times. The results of the samples are used to evaluate the measure of effectiveness for each design point using the function designed in part A of this chapter. These measures of effectiveness will be used to derive a regression model where the MOE is a function of the crowd's factors and levels and the peacekeepers' levels.

C. REGRESSION MODEL

A multiple regression model describes the relationship between two or more X (independent) variables and an expected Y (dependent) value, which is viewed as a combination of the different Xs. The X values are typically some measurements or observed data points. The user of the Peacekeeping simulation model might be interested to find a relationship between the MOE and the starting conditions of the simulation. Starting conditions can differ in the peacekeepers' tactics, the number of armed protesters, the number of bystanders, and the behavior of the crowd. These different starting conditions are the X values of the multiple regression model, while the estimated MOE is the Y value. Therefore a regression model will describe how different numbers or tactics affect the MOE. The regression model should consider starting conditions individually as well as interactions between different factors because interactions indicate synergetic effects. For example, does the combination of the crowd's behavior and the peacekeepers' tactics influence the MOE in addition to the crowds behavior and the peacekeepers' tactics by themselves.

The statistic software package S-PLUS was used to find the most suitable regression model with at most two term interactions. First a fitted model without interactions was defined. MOE was modeled using the blue tactics (blue), the red-behavior (tactics), the number of armed protesters (red), and the number of bystanders (bystanders).

This model was used as the required starting model for an automated process of stepwise selection of a regression model with two-term interactions. S-PLUS provides the stepAIC function in its Mass library for this process. The stepAIC function computes the AIC value for each possible combination of the considered terms. AIC is defined as:

$$AIC = n * \log (RSS/n) + 2p + const \qquad \text{[Ref. 23; p. 185]}$$

where: RSS = residual sum of squares, n = # observations, p = # parameters

The computed regression model shows a relation between the MOE and the blue tactics, the red behavior, the number of bystanders, the interaction between the red

behavior and the number of bystanders, and the interaction between the blue tactics and the red behavior. The regression results together with the coefficient values, their standard error, t-value and p-value are shown in Table 9.

Residuals				
Min	1 Quart	Median	3 Quart	Max
-15.96	-0.4554	0.0375	0.5639	12.33

Coefficients	Value	std. Error	t Value	Pr (>| t |)
Intercept	63.8668	1.1305	56.4932	0.0000
blue	0.9351	0.3739	2.5010	0.0134
tactics	-1.6832	1.1305	-1.4889	0.1386
bystanders	-0.7340	0.3591	-2.0440	0.0427
tactics:bystanders	-0.8902	0.3591	-2.4791	0.0142
blue:tactics	0.8351	0.3739	2.2336	0.0270

Residual standard error: 4.726 on 154 degrees of freedom
Multiple R-Squared: 0.4999
F-statistic: 30.78 on 5 and 154 degrees of freedom, the p-value is 0

Table 9. Regression output

The p-values of most of the coefficients of the regression model are smaller than 0.05, which shows that these coefficients are significant. The only coefficient that has a p-value greater than 0.05 is tactics, the coefficient for the protesters' behavior, with a value of 0.1386. Nevertheless there are reasons to use this coefficient in the model. Tactics is also present in a second degree term of two variables, tactics and bystanders. Therefore tactics as linear term is marginal to the second degree term. Furthermore the model uses an arbitrary origin. In cases where the origin is arbitrary one would normally consider regression models where for each term present all terms marginal to it are also present. Removing marginal factor terms from a fitted model is statistically meaningless. [Ref. 23; p. 184] Since the second term interaction between tactics and bystanders is statistically significant, tactics as a linear term has to remain in the model.

In conclusion all terms that are in the regression model are relevant and the expected MOE value E[MOE] is a function of the two-term interactions of the starting conditions.

D. REGRESSION MODEL VALIDATION

To validate the regression model and to check if the underlying conditions for a regression model are fulfilled, the residuals have to be checked for normality. Non-normal errors compound inefficiency and undermine the rationale for t- and F-Tests. This problem can cast doubt on the P-values computed in the regression output. [Ref. 24; p. 116]

A histogram with density line, a boxplot, and a quantile-normal plot of the residuals were computed and plotted to check this assumption. The histogram shows that the errors are distributed approximately normal with large tails on both ends of the distribution.

The boxplot shows the same result, the distribution of the residuals seems to be normal with a large number of outliers.

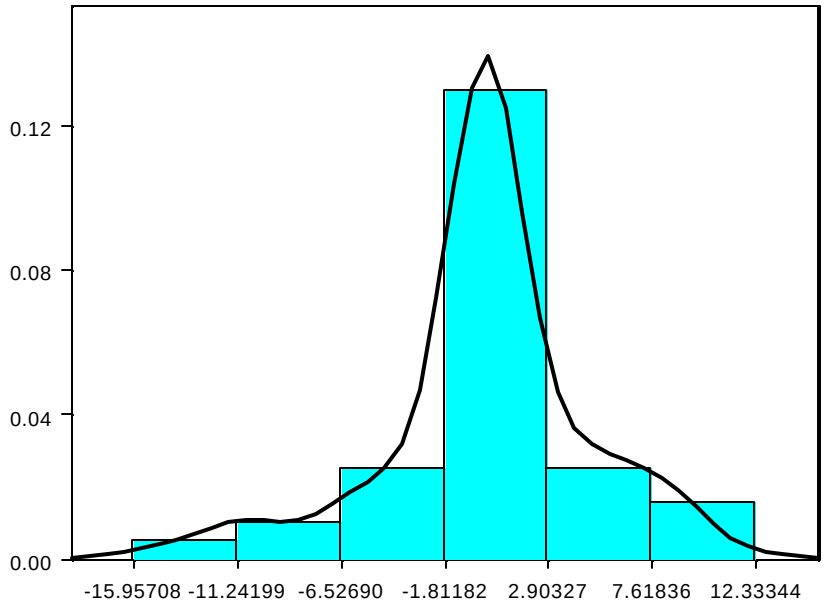

Figure 34. Histogram with density line of regression residuals

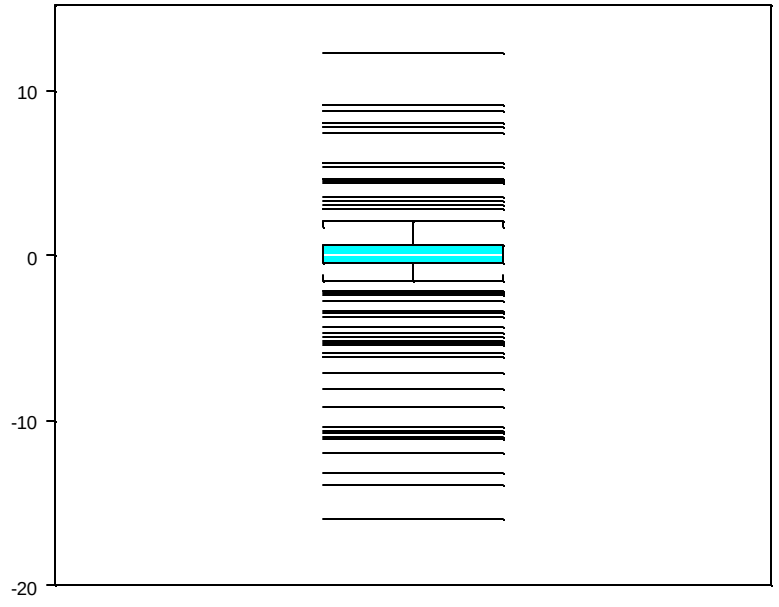

Figure 35. Boxplot of regression residuals

Finally the quantile-normal plot also shows the same result. The line shows the position of data if the distribution is normal. That the data lies on top of the line around an x value of zero indicates that the residuals are distributed normal. The large number of points, which lie below the line when the x value is less than zero and the large number of points that lie above the line when the x value is greater than zero indicate heavy tails. That the data points are spread out at both ends indicates a large number of outliers. [Ref. 24, p. 16]

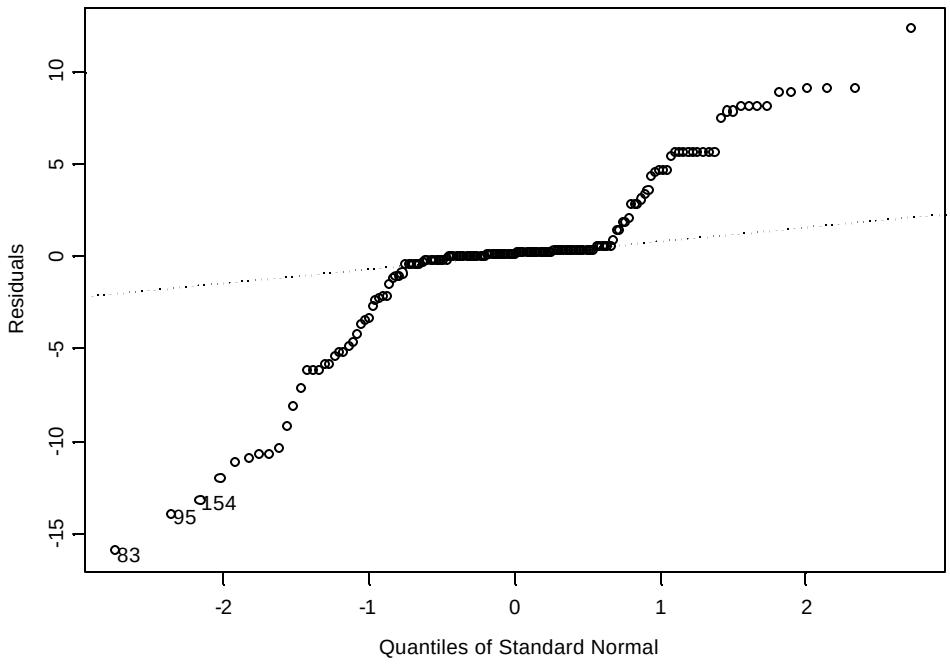

Figure 36. Quantile-normal plot of regression residuals

Another assumption, which must be fulfilled for a valid regression model is homoscedasticity of the residuals. Heteroscedasticity would lead to inefficiency and biased standard error estimates. These would, as nonnormality, cast doubt on the computed P-values. [Ref. 24; p.116]

A plot of the residuals verses the fitted values shows high variation for MOE values between 54 and 61 and small variation for MOE values around 66. Like the plots regarding normality this plot also shows a large number of outliers. The small variation

concerns only few data point and also only a small MOE value range. The vast majority of data points and MOE values are homoscedastic. The residuals around the MOE value of 66 will not have a large influence upon the models standard errors and t-tests because of the sample size of 180 data points.

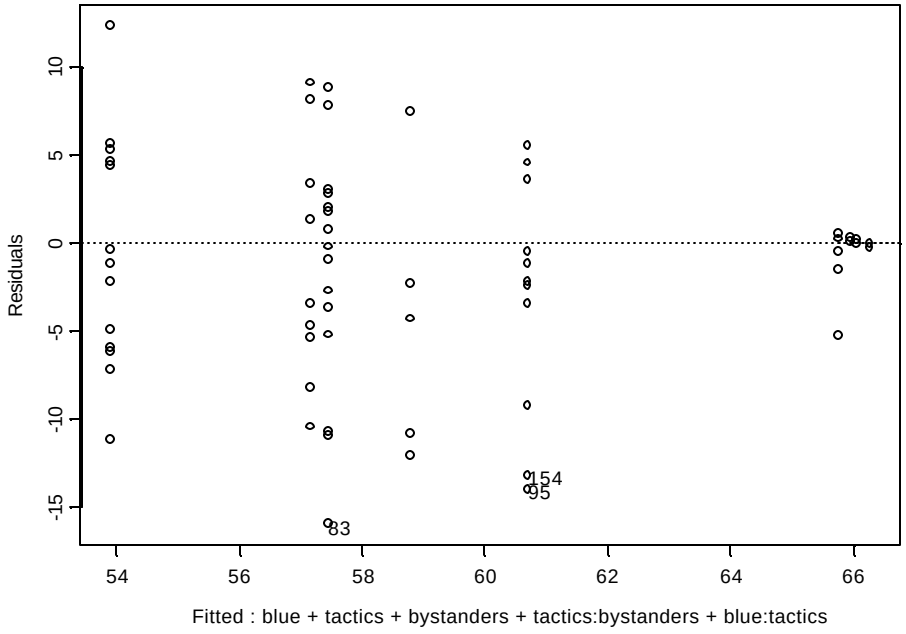

Figure 37. Residuals versus fitted values plot

Autocorrelation is another problem that might reduce the efficiency of ordinary least squares (OLS) and bias estimated standard errors. Autocorrelation refers to correlation between values of the same variable across different cases and reflects connections among cases. Autocorrelation often occurs with time series or geographically linked cases. [Ref. 24; p.118] It should not be a problem in the data sampled with the Peacekeeping simulation model. The following figure shows that this assumption is true. There is no significant autocorrelation between data points. The correlation is less than 0.2 for all lags up to 25. Lags greater than 25 are not considered because autocorrelation between higher lags is very exceptional when there is no correlation within lower lags.

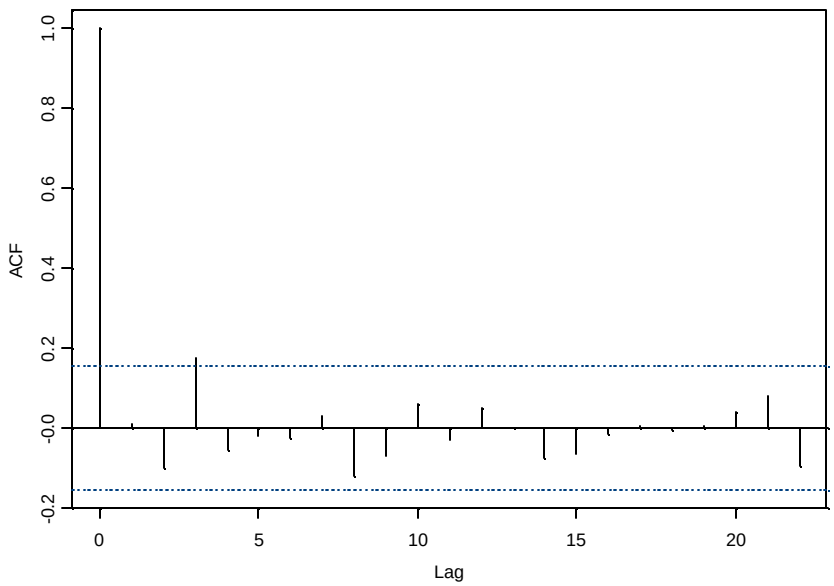

Figure 38. Correlogram for the regression model

To check for influence is the last common test to validate a regression model. Cook's distance is computed for each data point and graphically displayed. Cook's distance measures influence of a single data point on the model as a whole. That is, Cook's distance D_i reflects the influence of data point P_i on all estimated regression coefficients. [Ref. 24; p. 132] The graph shows that the highest individual influence has a value of 0.08. Furthermore many points have a relatively high influence on the model. But the absolute influence values a very small, in most cases less than 0.04. As a result all observed data points can stay in the model, because none of them is unusually influential.

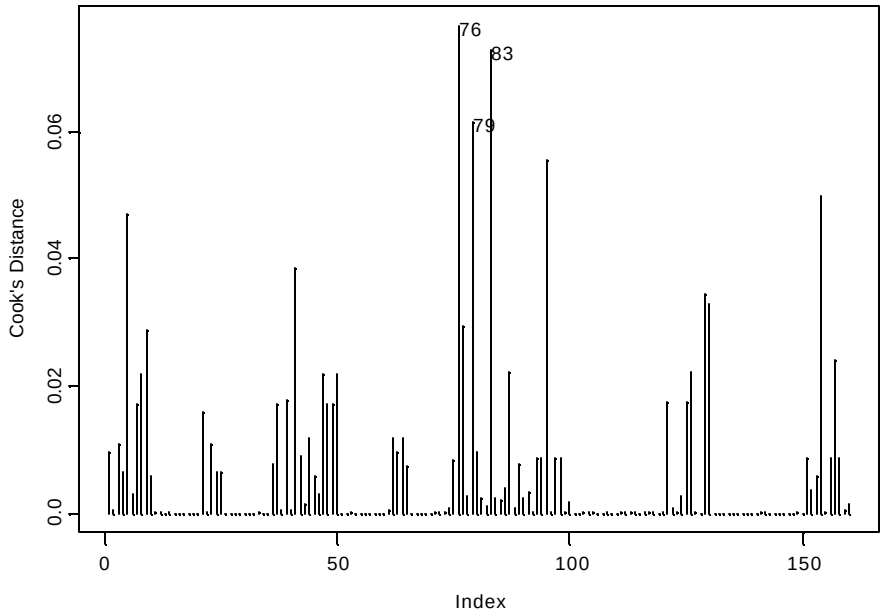

Figure 39. Cook's distance for the regression model

In conclusion, the conducted standard diagnostic checks on the residuals from the fitted model show no strong evidence of any failure of the assumptions. The residuals are distributed approximately normal, homoscedasticity can be assumed, there is no autocorrelation, and no data points have high influence on the model. Therefore the computed regression model is valid.

E. REGRESSION MODEL INTERPRETATION

The developed regression model considers the peacekeepers' defense tactics (blue), the protesters' advance tactics (tactics), the number of bystanders (bystanders), the interaction between the protesters' advance tactics (tactics * bystanders), and the interaction between the peacekeepers' defense tactics and the protesters' advance tactics (blue * tactics). The number of armed protesters does not influence the MOE value and neither do any second order interactions that contain the number of armed protesters.

The regression results can be shown graphically in a cube, where the direction of the arc indicates an increasing MOE value for one changing starting condition. The sides of the cube show the first order relationships and the diagonals the second order interactions. [Ref. 22; p. 310] For example the arc on the front bottom of the cube parallel to the "blue" axis specifies that the measure of effectiveness increases, when the peacekeeping forces change their tactics from moderate to defensive.

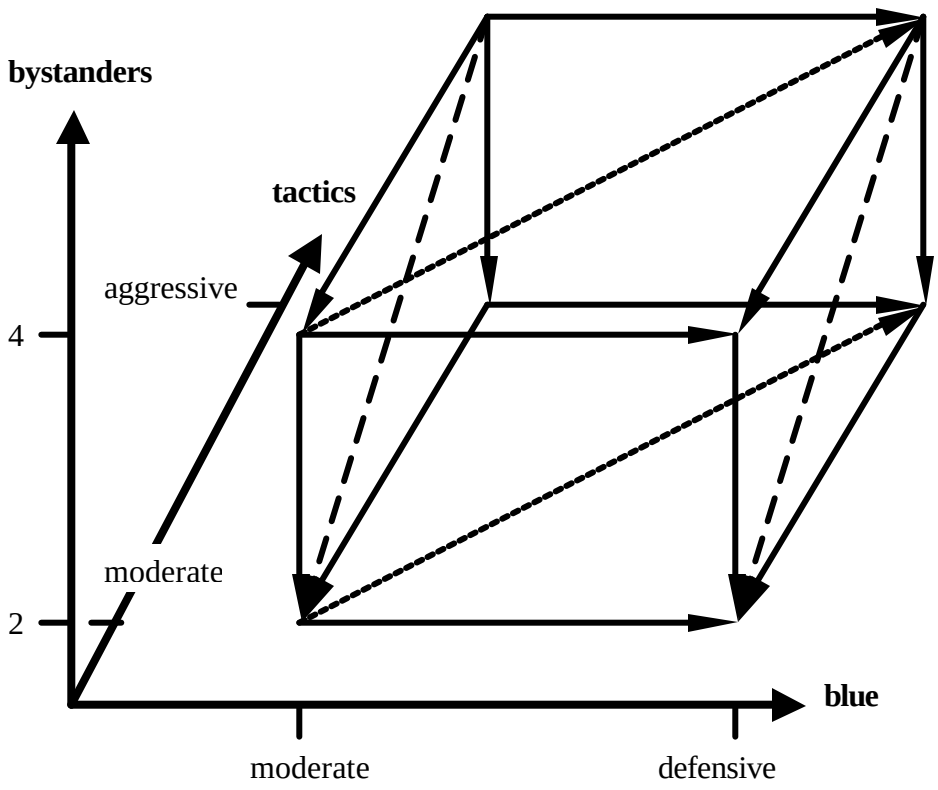

Figure 40. Regression cube

The coefficient for the peacekeepers' defense tactics is 0.935. This implies that when the peacekeepers' change their tactics from moderate to defensive the MOE value increases; the peacekeeping forces are more successful when they use defensive tactic.

The coefficient for the protesters' advance tactics is -1.6832, which indicates that when the protesters change their behavior from moderate to aggressive the MOE value

decreases. This means for the peacekeeping forces that they are more successful against moderate protesters.

The coefficient for the number of bystanders is -0.7340, which means that as the number of bystanders increases the MOE value decreases. For the peacekeepers follows that they can fulfill their mission better when there are fewer bystanders.

The coefficient for the interaction between the protesters' advance behavior and the number of bystanders is -0.8902. This means that when the protesters' behavior changes from moderate to aggressive and the number of bystanders increases simultaneously the MOE value decreases. From the peacekeepers' point of view this means that moderate protesters together with view bystanders makes it easier to achieve their goals.

Finally the coefficient for the interaction between the peacekeepers' defense tactics and the protesters' advance behavior is 0.8351. This suggests that when the peacekeepers' tactics changes to defensive and the protesters' behavior simultaneously changes to aggressive the MOE value increases. In other words, the peacekeepers' mission is more successful when they use a defensive tactic against aggressive protesters.

The absolute values of all regression coefficients are close to one, which indicates that there are no major changes in the measure of effectiveness when only one of the input values changes. Changes within the MOE add up as more changes in the simulation inputs occur.

F. REGRESSION MODEL SUMMARY

The regression model can be used to estimate changes in the MOE value for different starting conditions of the simulation model. These starting conditions illustrate the diverse environments in which the peacekeepers act and the different tactics applied by the peacekeeping forces.

The regression cube is a tool to explain the changes in the MOE value for changing conditions graphically.

The developed regression model can be used to make qualitative statements about differences in MOE values for various starting conditions. It should not be used to forecast MOE values since the value itself has no meaning. Additionally peacekeeping operations are, as mentioned in chapter I.3, complex adaptive systems, which are composed of a number of nonlinearly interacting parts. Consequently a model, which is a combination of states describing some of the starting conditions by using only addition as mathematical operation, cannot describe these nonlinear operations and their results precisely. But such a model can be applied to make a qualitative educated guess about a simulation outcome knowing the underlying environmental conditions, which are the simulation inputs. The user, who applies the developed regression model, will obtain an approximation of the simulation result. Additionally the regression model provides its user with meaningful relationships between different environmental starting conditions.

G. IMPLICATIONS FOR PEACEKEEPING TACTICS

Having to choose between a moderate and a defensive tactic for the protection of the EPISCOPAL SEE, the peacekeeping forces should always use the defensive approach because it improves the MOE value for all other starting conditions.

Additionally it is not important to the peacekeeping forces whether they protect their position against few or many armed protesters. This seems to be a result of the engagement taking place in urban terrain. It could be observed that the peacekeepers could never detect all armed protesters together. Due to the buildings in the mission area the protesters always appeared one after the other in the peacekeepers sensor range and the peacekeeping forces acted before all of them were sensed.

It is not surprising that the peacekeepers can expect to be more successful in achieving their mission with few casualties on both sides when the protesters approach with a moderate instead of an aggressive behavior.

They are also more successful in achieving their multiple objectives when there are fewer bystanders.

An additional observation was made during the simulation runs where the peacekeepers applied a moderate tactic. In this case they called for reinforcements immediately after the first casualty occurred on one of the opposing sides. The reinforcements could never influence the simulation outcome because the interaction between the two opponents always reached the stopping condition of five killed agents before the reinforcement had line of sight to any other agent. This shows that the peacekeepers need to call for reinforcement much earlier to influence the protesters behavior.

V. CONCLUSIONS AND RECOMMENDATIONS

This thesis developed and implemented an agent-based small-scale peacekeeping operation modeling methodology. Using this implementation, called Peacekeeping and TryShoot, this thesis generated a relationship between different environmental conditions in a peacekeeping scenario and the resulting effectiveness of the peacekeepers' actions. A model of peacekeeping operations in urban terrain was generated using agents defined by six personality characteristics, a type, a starting position, a weapon type, and a training level. The model served as a vehicle for experimentation to demonstrate the impact of tactics, protester behavior, and the number of bystanders in the modeled environment.

Agent-based simulation is a valuable method for the modeling of small-scale peacekeeping operations. A user can generate complex behaviors from simple entities with beliefs, desires, and intentions using this methodology. Additionally, this simulation model can be used to test ideas of how entities function in complex adaptive systems. Multiple simulation runs can be utilized to develop models describing the entities' behavior. The simplicity of the individual models, the few initial states that govern the interaction of the entities, and the information provided in different graphical user interfaces grant transparency of the model. Transparency is an important attribute in entity level simulations and especially in agent-based simulations. It assists both the analysis of the model and the communication of its results. Agent-based models are best used as tools to improve intuition rather than as means to generate predictions. If they are used in the latter case, only general outcomes should be predicted, not concrete and precise results. The agent-based model can be used for qualitative, but not for quantitative, predictions.

Combat operations can be modeled in large-scale aggregated simulations, as is done in most current warfare simulation models. In peacekeeping operations individual actions have much greater impact on the overall mission and its failure or success. Agent-based models show the greatest potential for modeling complex, small-scale scenarios of peacekeeping operations where the individual actions are important. In addition to the analysis done above, the developed model can be used to answer such questions as: what

is the effect of an increased number of bystanders; how does the outcome change when protesters and peacekeepers change their tactics simultaneously; and does a tactic exist at which the peacekeepers always achieve their objective of having a small number of casualties?

The benefits of the Peacekeeping simulation model, as of most agent-based models, are not so much in the answers it provides, but in the questions that its study and exploitation generates. Some of the results will be surprising to military leaders with experience in peacekeeping operations, which fulfills the objective of the simulation model. Its true intent is to increase insight into peacekeeping operations in urban terrain. By modeling at the entity level, the user is forced to study the scenario from the bottom up.

There are downsides in the use of agent-based simulation models. Users are tempted to increase the fidelity of the simulation by increasing the complexity of the agents and the richness and diversity of their possible actions. This can lead to a simulation that loses transparency. The aggregate behavior of agent-based simulation models is highly complex, despite the simplicity of the individual agents. The simplicity of the acting entities, their transparency, makes it possible to understand the complex system. When the agents get more complex, the understanding of the cause and effect relationship in the simulation model decreases.

Another problem of agent-based simulation models is finding the parameters that are needed to define an entity whose behavior is an approximation to human behavior. As human behavior is highly complex one might want to add more and more parameters to define the simulation entity. This will always lead to more complexity but not necessarily to more realism. An agent-based model must find the balance between the necessary complexity and the possible simplicity to model human behavior on the one hand and to make it possible to understand what is happening in the simulation model on the other.

Peacekeeping operations will be the environment requiring the most frequent application of the Bundeswehr. They are unlikely to get less challenging as can be seen in Afghanistan, where the peacekeeping forces deal with a new government, a large number of tribes, groups of outlaws that control parts of the country, hidden terrorists, and a

continuing war against terrorism. Agent-based modeling provides an important and inexpensive tool for experimentation and training in peacekeeping operations. As a result, agent-based simulation deserves continued study and exploration as a valuable method for the modeling of peacekeeping operations.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX - - SOFTWARE USED IN THIS THESIS

All original software used in this simulation was written by the author in the programming language Java.

Data generated by the simulation was analyzed using both Excel 2000 and S-PLUS 2000.

The code for the simulation is contained in two packages called PEACEKEEPING and TryShoot. As mentioned in the body of the thesis, both packages are an extension of Simkit.

All the source code can be obtained by contacting Prof. Buss via his web page at http://diana.or.nps.navy.mil/~ahbuss/.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1. Presse- und Informationsamt der Bundesregierung, German Security Policy and The Bundeswehr, Bonn 1997
2. Holland, Hidden Order, Massachusetts 1995
3. The Federal Minister of Defense, The Bundeswehr Advancing Steadily into the 21st Century, Berlin 2001
4. Federal Ministry of Defense, White Paper 1994
5. A.Eknes, Prepared for Peace-keeping: The Nordic Countries and Participation in U.N. Military Operations, in W.Kuehne, Blauhelme in einer turbulenten Welt, Baden-Baden 1993
6. Joint Pub 3-07, Joint Doctrine for Military Operations other than War, 1993
7. T. Sommer, Ausweg dringend gesucht, in Die Zeit, 15. Mai 2001
8. http://heer.bundewehr,de/trs/guppies/auftrag_und_organisation.htm
9. http://heer.bundeswehr.de/trs/infs/lehre/sira.html
10. A. Ilachinsky, Irreducible Semi-Autonomous Adaptive Combat (ISAAC), in Military Operations Research, V5 N3 2000
11. G. Weiss, Multiagent Systems, Massachusetts 1999
12. M. Wooldridge, in G. Weiss, Multiagent Systems, Massachusetts 1999
13. C. von Clausewitz, On War, Princeton, New Jersey 1989
14. W. N. Reynolds, D. S. Dixon, Archimedes: A Prototype Distillation, in Maneuver Warfare Science 2001
15. A. M. Law, W. D. Kelton, Simulation Modeling and Analysis, Boston 2000
16. A. Buss, Discrete Event Programming with Simkit, in Simulation News Europe, August 2001
17. UML Notation Guide, Version 1.0, 1997

18. J. Naumann, Bilder-CD Bundeswehr, Koeln 1999

19. M. Bowden, Black Hawk Down, New York 2000

20. D.H. Wagner, W.C. Mylander, T.J. Sanders, Naval Operations Analysis, Maryland 1999

21. A.N.Page, Utility Theory: A Book of Readings, New York 1968

22. E.P. Box, Statistics for Experimenters, New York 1978

23. W.N. Venables, B.D. Ripley, Modern Applied Statistics with S-PLUS, New York 2000

24. L.C. Hamilton, Regression with Graphics, California 1992

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Professor A.H. Buss (Code OR)
   Naval Postgraduate School
   Monterey, California

4. Professor J. Hiles (Code MV)
   Naval Postgraduate School
   Monterey, California

5. LTCOL G. Mislick, USMC (Code OR)
   Naval Postgraduate School
   Monterey, California

6. J. Wellbrink (Code MV)
   Naval Postgraduate School
   Monterey, California

7. Maj. R. Woodaman, Studies and Analysis Division
   MCCDC, Code C45
   300 Russel Road
   Quantico, VA 22134-5130

8. Director, Studies and Analysis Division
   MCCDC, Code C45
   300 Russel Road
   Quantico, VA 22134-5130

9. Heeresamt Abt. V 3
   Postfach 51 07 68
   50943 Koeln
   Germany

10. Amt fuer Studien und Uebungen der Bundeswehr
    Bereich Operations Research Modelle und Analysen
    Schaumburgweg 3
    51545 Waldbroehl
    Germany

11. Fachinformationszentrum der Bundeswehr
Manteuffelstraße 20
22587 Hamburg
Germany

12. VN Ausbildungszentrum der Bundeswehr
Rommelstrasse 31
97762 Hammelburg
Germany

13. Institut fuer Angewandte Systemforschung und Operations Research
Universitaet der Bundeswehr Muenchen
85577 Neubiberg
Germany

14. Science Applications International Corporation
1100 North Glebe Road, Suite 1100
Arlington, VA 22201