The Pennsylvania State University

The Graduate School

Department of Computer Science and Engineering

SIMULATING SENSORS FOR LIGHT ARMORED VEHICLES

A Thesis in

Computer Science and Engineering

by

Vivek Kaushal

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2014

The thesis of Vivek Kaushal was reviewed and approved* by the following:

Steven Haynes
Professor of Practice at College of Information Sciences and Technology
Thesis Advisor

Mahmut Taylan Kandemir
Professor of Computer Science and Engineering
Director of Graduate Affairs at Computer Science and Engineering
Thesis Advisor

Frank E. Ritter
Professor of Information Sciences and Technology and Psychology

Lee Coraor
Professor of Computer Science and Engineering
Director of Academic Affairs

*Signatures are on file in the Graduate School

Abstract

Requirements engineering is a multi-stage process of which requirement prioritization is a key element. Even though a significant amount of research is available for requirements engineering process, research about requirements prioritization is not yet exhaustive and has much room for improvement. Due to a lack of research and awareness, many organizations are unable to optimally prioritize their requirements [**1**]. This thesis describes an improvement to the requirements prioritization process for scenarios involving the use of the sensors. The effectiveness of the method is demonstrated by developing a tool using the Sense and Respond project as a case study. The key features of the tool are performance, reproducibility and low user interaction. The effectiveness of the tool was demonstrated by analyzing accuracy parameter for couple of sensors used in the Sense and Respond project, wherein a sharp change in outcome was observed by a two percent change in accuracy for one of the sensors.

Table of Contents

List of Figures vi

List of Tables vii

Abbreviations viii

Acknowledgements ix

Chapter 1 Introduction 1

1.1 Case Study 2
1.2 Research Questions 3
1.3 Acceptability Criteria 3
1.4 Work Organization 4

Chapter 2 Literature Review 6

2.1 Requirements Engineering 6
2.1.1 Elucidating requirements 7
2.1.2 Generating specifications 7
2.1.3 Prioritizing requirements 7
2.1.4 Requirement interdependencies 8
2.1.5 Analyzing impact of requirements 8
2.2 Decision-making 9
2.2.1 Analytic Hierarchy Process 10
2.2.2 Comparison of the S&R project to a business framework 12
2.2.3 Tools 13
2.2.4 The Sense and Respond Project 15
2.2.5 Comparison Metric 18
2.2.6 Scenario Utility Modeling and Measurement Integration Tool 21
2.3 Agent-Based Simulations 23
2.3.1 Multiple Agents 25
2.3.2 Agent-Based Simulations in Research 28
2.3.3 Agent Architectures 30
2.3.4 Reactive Architecture 31
2.3.5 Belief-Desire-Intention Architecture 31
2.3.6 Layered Architecture 32
2.3.7 Design Choice 33

Chapter 3 Design and Implementation 35

3.1 Scenario 35
3.2 Implementation 37
3.2.1 Discrete Event Driven 38
3.2.2 Point System 38
3.2.3 Events 38
3.2.4 Modular design 41

3.2.5 Sensors 46
3.2.6 Points Tracker 47
3.2.7 Input Variables 48
3.3 Method 60
3.3.1 Phase 1 61
3.3.2 Phase 2 61

Chapter 4 Results and Discussion 63

4.1 Interpreting the graphs 63
4.2 Sources of Error 64
4.2.1 Initial Value 64
4.2.2 Architecture Used 66
4.2.3 Implementation 66
4.3 Number of Simulation Runs 67
4.4 Shots Fired Sensor 67
4.5 Planetary Gear Temperature Sensor 69
4.6 Discussion 72

Chapter 5 Conclusion 74

Bibliography 76

List of Figures

Figure 1: Analytic Hierarchy Process [16] 10
Figure 2: Event Chain 18
Figure 3: Hierarchy of Measures [27] 18
Figure 4: SUMMIT Design (Haynes, 2009) 23
Figure 5: Overview of Agent Based Environment [36] 25
Figure 6: Architecture Used [36] 34
Figure 7: Simulation Environment 37
Figure 8: Modules 42
Figure 9: Classes Used 43
Figure 10: Impact of Defense Parameters on FLER 57
Figure 11: Impact of Attack Parameters on FLER 58
Figure 12: Impact of Sensor Parameters on FLER 59
Figure 13: Impact of Temperature Parameters on FLER 60
Figure 14: Sample Figure 64
Figure 15: Shots Fired - Reload Ratio 90% 68
Figure 16: Shots Fired – Reload Ratio 100% 69
Figure 17: Planetary Gear Temperature Sensor – No Breakdown Penalty 70
Figure 18: Planetary Gear Temperature Sensor Accuracy with Breakdown Penalty 72

List of Tables

Table 1: Variable Descriptions 48

Table 2: Parameter Presumed Relation to FLER 53

Abbreviations

AHP	Analytic Hierarchy Process
BDI	Belief Desire Intention
FLER	Force Loss Exchange Ratio
LAV	Light Armored Vehicle
RE	Requirements Engineering
R&D	Research and Development
S&R	Sense and Respond
SA	Situational Awareness
SHM	Structural Health Monitoring
SPEC	Standard Performance Evaluation Corporation
SUMMIT	Scenario Utility Modeling and Measurement Integration Tool
TLCM	Total Life Cycle Management
UML	Unified Modeling Language
USMC	United States Marine Corps

Acknowledgements

I would never have been able to finish my dissertation without the guidance of my committee members and help from my friends.

I would like to express my deepest gratitude to Dr. Mahmut T. Kandemir and Dr. Steven R. Haynes, who supported me throughout my thesis with their patience and knowledge, and provided an excellent atmosphere in which to complete this research. Dr. Ritter gave useful guidance during my thesis review to help me focus on key areas of research.

I would also like to thank Christina Patrick and Rama Murgan who, as good friends, gave me excellent suggestions during my research.

Finally, I thank my parents and my elder brother for supporting me and encouraging me with their best wishes.

Chapter 1

Introduction

Organizations have been trying to gain technological superiority for a long time, and in the last few decades, have seen a substantial increase in pressure to gain competitive advantage [**2**]. As software products form an important component of gaining strategic and competitive advantage, the importance of these products meeting customers' requirements is realized. Recent research provides empirical evidence that requirements-engineering process helps make higher quality products [**3**]. Organizations are becoming more aware of the fact that requirements engineering process can be the difference between the success and failure of a project [**3**]. In the words of Aurum [**2**], "Requirements Engineering (RE) is the process by which the requirements for software products are gathered, analyzed, documented and managed throughout the SE lifecycle. RE is concerned with interpreting and understanding stakeholders' goals, needs and beliefs".

Requirements engineering is a multi-stage process that includes: 1) capturing requirements, 2) modeling and specifying requirements, 3) prioritizing requirements, 4) charting dependencies between requirements, 5) analyzing the impact of requirements [**2**].

Among these steps, requirement prioritization is a key aspect of requirements engineering for the following reasons: 1) Constraints such as time, cost and resources, 2) risks of project failure, 3) stakeholder interests [**2**]. Carefully prioritized and selected requirements can ensure that projects are finished on time, are successful, stay within the estimated budget, have strategically placed advantages over competitors and are extendable during successive releases.

Prioritization of requirements is the hardest step in requirements engineering and is often incorrectly implemented because organizations lack familiarity with the best prioritization methods [**1**]. This thesis makes a modest contribution towards requirement prioritization in projects involving the use of sensors.

1.1 Case Study

The case study for this work involved a multi-billion dollar project that spanned over several years. The Sense and Respond (S&R) project undertaken by the United States Marine Corps (USMC) met these requirements. This project was started with the goal of enhancing the Situational Awareness (SA) of the Marines. Situational Awareness, in the current context, is defined as an increase in the knowledge of the Marines about the environment surrounding the Light Armored Vehicles (LAVs). LAVs are important equipment deployed by the USMC for various missions, including the missions in Iraq and Afghanistan; their purpose is to patrol and perform rescue operations and combat missions. This increase in SA was achieved by upgrading the LAVs' sensors, on-board computers and wireless communication devices with the latest technology. Since the increased SA gained by the use of sensors helps Marines to take actions that pertain to the mission [**4**], [**5**], sensors form an important component of this technological upgrade [**6**]. The scope of this technological upgrade was not limited to increasing the SA of Marines, but also assisted them in other purposes like diagnostics and prognostics of the LAVs [8]-[10].

In the S&R project, there were different types of sensors that could have been installed on an LAV, for example, a sensor to measure the temperature of planetary gears, a sensor to count the number of bullets fired, etc. Furthermore, each of these sensors could have had multiple options, for example, two temperature-measuring sensors with varying accuracy could have been available. In this work, sensors are considered requirements for the S&R project, and requirement prioritization is ordering the sensors in a priority order according to the utility derived from them.

1.2 Research Questions

Prioritizing the sensors used during the S&R project required finding answers to some fundamental questions related to the project:

1. What is an optimal resolution of the information collected from the sensors?
2. How accurate should the information provided by a sensor be?
3. What should be the refreshing rate for the information generated by the sensors?

All of these questions can be summarized as, how does one decide which sensor to select among the plethora of options? This research focused on answering the questions pertaining to the nature, resolution, accuracy and refresh rate of sensors. Answering these questions required the following steps:

- Find a metric to quantify effectiveness of missions involving LAV's.
- Determine a method to analyze impact of different sensors on mission effectiveness.
- Implement the technique and study the results.

1.3 Acceptability Criteria

For a requirement prioritization process to be useful, the process needs to possess following features:

- Is fast.
- Generates quantifiable results.
- Generates reproducible results.
- Has low user interaction.

The results of requirement prioritization need to be generated quickly so that users can try different models based on the results to obtain maximum utility out of the system. The results should be quantifiable, as qualitative results can at times be too vague to provide good insight into the system. Different runs of the comparison tool should produce similar priority order for users to trust the comparison tool. While most users of the tool need to be able to provide inputs and modify the requirements, excessive user involvement in each run of the tool can be a deterrent to adaptability of the comparison tool. The tool needs to be able to run with minimal input from the user and produce easily comprehensible results.

1.4 Work Organization

Chapter 2 provides a literature review of all the components used in this work. It starts with a discussion on requirements engineering, and then delves into the decision-making process. This is followed by careful evaluation of different techniques employed by the industry to solve similar problems, then provides different ways to quantify effectiveness of mission and ends with possible techniques to quickly evaluate each choice.

Chapter 3 describes the design and implementation of the simulation. This includes discussion about the use of agent-based simulation as a technique for evaluating the sensors and Force Loss Exchange Ration (FLER) as the metric to measure effectiveness of mission.

Chapter 4 describes and analyses the results obtained from running the simulation tool developed in this work. The analysis indicated that the accuracy of the shots fired sensor has a direct effect on the effectiveness of the mission (i.e., higher accuracy leads to more effective missions). For the planetary gear temperature sensor, the results indicate that effectiveness of mission declines sharply at some error percentage that is determined by the other surrounding parameters, but remains constant before or after that erroneous level. Chapter 5 concludes the work and discusses future work.

Chapter 2

Literature Review

A literature review was carried out to find methods for requirement prioritization in requirements engineering. The requirements engineering process and decision-making process were explored under two different categories: Requirements Engineering and Analytic Hierarchy Process. Based on this exploration, it was concluded that agent-based simulations would be a good method for requirement prioritization in the S&R project. Additionally, during the exploration of the decision-making process, a strong need for a quantifiable metric for effectiveness of the S&R missions emerged. It was concluded that force loss exchange ratio is the most appropriate metric for quantifying effectiveness of missions.

2.1 Requirements Engineering

A necessity of a user can be defined as requirement. Every project starts with the collection of requirements from stakeholders, where stakeholders are the people who influence or are influenced by the system [**2**]. Stakeholders include users, customers, suppliers, developers and businesses [**2**], [**7**]. The combination of what the stakeholders need from a system and how the system would achieve those needs defines the requirements of the system. The process of documenting, prioritizing, analyzing, and maintaining these requirements to ensure delivery of a system that fulfills the needs of stakeholders is known as requirements engineering.

2.1.1 Elucidating requirements

Elucidating requirements is the process of forming a comprehensive and accurate understanding of stakeholders' needs. Forming a clear understanding of the needs of all the stakeholders is a difficult task due to reasons including unclear needs, disagreements about requirements, unrealistic expectations, etc. [**2**] [**7**]. Having well elucidated requirements is key to establishing clear communication among the users and system developers. This step gives the project direction as well as a long-term goal.

2.1.2 Generating specifications

Using the requirements, a formal view of the system is designed by including all elements of the system. The view, usually based on the Unified Modeling Language (UML) diagram, provides system designers with a concise model from which they generate specifications. These specifications help bound the scope of the system in terms of what requirements will be satisfied by the system [**2**].

2.1.3 Prioritizing requirements

Prioritization is the process by which a requirements engineer considers all requirements and determines which requirements are worth pursuing based on their cost, benefit and risk analysis. Requirement prioritization is done in our day-to-day lives, for example, deciding which mobile phone to purchase. This prioritization can be based on budget constraints, intended use, etc. Even a small decision like which mobile phone to purchase can be complex, as the customer has to decide among the trade-offs. In a project, a user, manager or any other stakeholder based on utility analysis, can make prioritization and choices. It is important for this utility analysis to be highly accurate; yet, it is hard to get right, as it usually involves predicting the future. Analyzing how the system will behave once developed can be difficult even when the system environment remains nearly the same, but this becomes nearly impossible when the system's environment changes fast [**2**].

2.1.4 Requirement interdependencies

Requirements defined by the user are generally interdependent in a complex way [**8**]. This interdependency can arise from changes in project constraints or changes in customer requirements [**2**]. Usually this interdependency does not create issues, but it is important to be able to analyze the impact of the changes on the overall system quickly. Requirement interdependencies form one of the primary reasons for the requirement prioritization.

Systems with fast changing requirements require system design to be modular in nature. It is tricky to find a balance between modular design and the incorporation of requirement interdependencies. Stakeholders want to know the impact of change in requirements on the overall system as soon as possible, and as accurately as possible. On one hand, wiring in all possible interdependencies can speed up the analysis of the impact on the overall system; on the other hand, wiring in all interdependencies makes it hard to change the system under changing requirements. In short, changing one requirement should not require a change in the complete system, but the impact of this change should affect other requirements predictably.

2.1.5 Analyzing impact of requirements

A user can sometimes be flexible with requirements or vary their requirements over a period of time. Under both these conditions, an ability to analyze the impact of the requirements on the desired goal is very important. Having the ability to ascertain the impact of change in requirements on the utility achieved is useful in three cases: 1) in case of flexible requirements, it can help determine how the requirements can be modified to maximize utility; 2) during system design, it can be used to design a system that increases utility; 3) in case it needs change, it can help determine how the change affects the goal of the system [**2**].

2.2 Decision-making

In projects similar to the S&R project that have a multi year life span, lack clear scope, and involve a large group of people working towards a common goal; decision-making process cannot be based merely on intuition [**9**]. Decision making for such projects is studied under Multiple Criteria Decision-Making. Various theories for multiple criteria decision-making have been used in both academia and industry, namely, Multi- Attribute Utility Theory [**10**], the outranking approach [**11**], Analytic Hierarchy Process [**12**] [**9**], Qualitative Decision Theory [**13**], etc. Decision making process of the S&R project was studied using the requirement selection and the Analytic Hierarchy Process (AHP). This work also refers to Qualitative Decision Theory for reference purposes.

On a broader level, these decision-making processes involve decomposing the problem into smaller sub component, analyzing them and then combining the results of the analysis to form decisions. Analyzing those smaller components can be a challenging task and this work contributes towards prioritizing these smaller components based on their contribution towards final outcome.

2.2.1 Analytic Hierarchy Process

As applicable to most decision-making processes, the Analytic Hierarchy Process primarily involves decomposing the problem into smaller subcomponents, then modeling these into a hierarchical structure and finally, aggregating them to form a decision [**12**] [**14**]. As shown in Figure 1, Analytic Hierarchy Process organizes the problem into a structure, allowing for flow of information within the system. AHP is a guiding framework, which aids decision-making, instead of providing the exact decision.

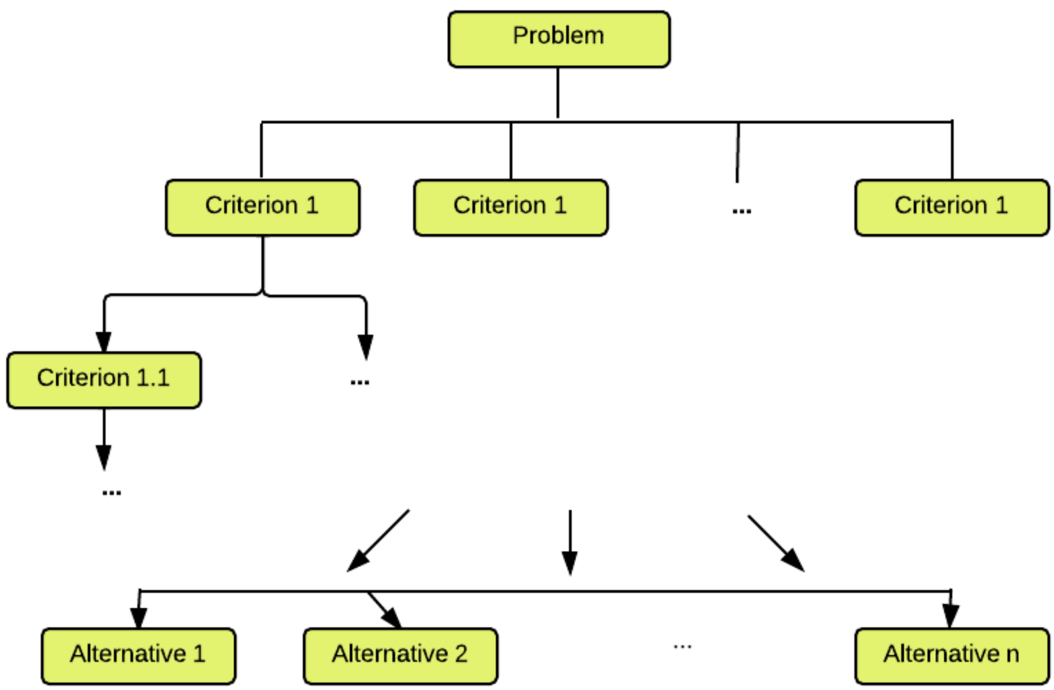

Figure 1: Analytic Hierarchy Process [16]

2.2.1.1 Decomposing the problem into a hierarchy

Depending on the kind of problem, a decision-problem can be split based on its functionality, available data, varied criterion, and available alternatives. A key factor in deciding this decomposition is that the sub-problems should be comparable, as this comparison helps to assign numerical values required for prioritizing the alternatives. For example, if somebody wants to buy a mobile phone, criterion can be cost, style, processing speed, etc. and alternatives can be iPhone and Android phone. These alternatives can have sub alternatives, like Android phone from Samsung or HTC. Similarly for the S&R project, mission effectiveness, cost, available resources can be the criterion, whereas available sensors, vehicles, etc. can be the alternatives. In short, the primary goal of this step is to describe the problem using its smallest independently analyzable and identifiable elements, and representing these into a hierarchical structure. [**14**] [**12**].

2.2.1.2 Quantifying and Prioritizing

Once the elements of the problem have been identified, the sub-problems are assigned some numerical value to facilitate the analytical hierarchy. This assignment of numerical values can incorporate both available data and human judgment, which also differentiates the AHP from other decision-making processes. For example, the numerical value for the processing speed of a mobile phone can be based on some benchmarking tool, whereas for the style it can be based on human judgment. A user may want to incorporate the combination of available data and human judgment while assigning numerical values, for example, to assign the numerical value for style of a mobile phone, data from a survey can be used, along with users personal judgment. Once numerical values are assigned, bottom-up approach is used to determine the numerical values of the elements higher up in the hierarchy [**15**]. This incorporation of both available data and human judgment helps making a justifiable decision.

2.2.1.3 Synthesis

Based on the numerical values assigned to the leaf elements in the hierarchical structure, the values of elements higher up in the tree are calculated. This synthesis is based on criterion. Several alternate decision-making processes can be used to perform other steps of the AHP, like analysis or quantification, but none of those techniques provides a synthesis mechanism applicable to such a large variety of systems [**14**].

2.2.2 Comparison of the S&R project to a business framework

Decision-making tools are most frequently used in business environments and thus understanding the similarities between the S&R project and business projects was helpful to finding the right set of tools for prioritizing the S&R project requirements. Kalloniatis [**16**] analyzed the application of a business process model to a military organization using a modeling and simulation framework and noted the need for business process modeling in military organizations.

To compare the business process model with the S&R project, a brief overview of product development projects given by Ulrich [**17**] was found useful. The process described by Ulrich [**17**] contains five steps for developing a project plan: identifying opportunities, evaluating and prioritizing projects, allocating resources and plan timing, completing pre-project planning, and reflecting on the results and the process. The Marines identify opportunities based on observations of the real environment and foresight of future requirements. Sensors provide information about the environment and choosing the right set of sensors forms the future requirements.

The project managers of the S&R project have budget constraints to meet the project goals, limiting the opportunities that can be pursued [**17**]. This makes the identification of opportunities that provide the highest returns an important task. The problem of identifying the highest returns is done by finding the expected utility of each project [**18**] and has been solved using several different techniques depending on the problem area, namely qualitative decision theory [**19**], [**20**], qualitative models [**21**], value of information [**22**] and qualitative data analysis.

2.2.3 Tools

Based on the study of Requirements Engineering and AHP, three tools were considered for logically ordering the options: surveys, interviews and simulations. The data collection techniques employed for each of these and are discussed in detail below.

2.2.3.1 Surveys and Interviews

Because data collected through both surveys and interviews are based the perception of system users and developers, they can be discussed together in the context of this research. Surveys compile the information collected from users who answer a set of questions to the best of their knowledge. The value of information generated during surveys is based on the questions chosen and the active participation of the users. In the case of interviews, each participant is asked a set of questions verbally by an interviewer. Hence, this data generally needs to be analyzed manually.

Data from surveys and interviews have two important benefits. First, they capture the user perception that directly affects the outcome of the next project-planning phase. Second, the method is able to evaluate every type of sensor, which is hard with other methods studied during this research.

Although surveys and interviews are great sources of user perception, they have a few issues. First, it is a slow and time-consuming process [**23**], so, to save time, participants who lack motivation [**24**] tend to fill surveys out quickly without giving adequate thought to the questions at hand. Second, results from surveys and interviews tend to differ during different runs of the collection. Finally, surveys and interviews tend to affect the user perception by the process itself, as participants tend to change their perspective or learn new information based on the questions asked.

Haynes [**25**] carried out a detailed evaluation of the S&R project base using the surveys and interviews methodology. The program managers of the S&R project considered the results provided by that evaluation very useful and thus further extension of the evaluation mechanisms was considered. The current research has been considered to be an extension of the work carried out during the previous evaluation. Since this methodology has already been used in the past to evaluate the S&R project [**25**], this research aimed to find an alternative method to justify the use of sensors.

2.2.3.2 Simulations

Simulation is the technique of imitating a real life environment or real-time behavior. A broader definition of simulations given by Barrett et al. [**26**] defines computer simulations as interactions and transactions between real world identifiable components, where these real world identifiable components are some algorithmic representations. The interactions and transactions between these algorithmic components can be calculated using computers, and the process is defined as simulations.

In the current context, we use computer simulations where the behavior is simulated in a computational environment. The developer programs simulations to cover all those aspects of an actual scenario that affect the final outcome of the study to be conducted. Other aspects that do not affect the final outcome may be ignored. The efficiency of the simulation is dependent on how well designed the simulations are, in other words, how closely can the simulations imitate the real world environment.

Because of the complexities involved in imitating the real world environment, the simulation methodology may not be able to simulate all the sensors installed on LAVs. Even for the sensors for which simulations can be used, this method may not be able to simulate all the parameters. This problem makes it inevitable to use alternate techniques to cover the evaluation of all the sensors. Another problem associated with simulations methodology is that improperly implemented simulations can produce results that do not correspond to the real scenario. Although simulations were not considered useful for all the sensors, simulations provide an ability to simulate future sensors.

Given these requirements of finding a quantifiable and easy-to-use solution for justifying the utility of sensors, it was determined that among the methodologies studied, simulations are the most suitable. Simulations can be designed to provide a quantifiable measure of utility and are easy to use once implemented.

2.2.4 The Sense and Respond Project

In order to simulate the S&R project, a clear understanding of the S&R project was required. Simulation design required splitting the S&R project into three major components: 1) the environment, 2) the inputs, that is, sensors, 3) the output, that is, effectiveness of mission.

2.2.4.1 Environment

The environment, in this context, refers to the two approximate spatial boundaries where the sensors operate and send data. The first spatial boundary is the *platform* level, which reflects the S&R activities at the LAV. The other spatial boundary is the *off platform* level, which refers to all S&R activities taking place off the vehicle [**25**].

At the *platform level*, on-board computers display the information gathered from the usage of sensors. This information assists Marines by providing them with improved visibility, vehicle health information, location and better communication with other vehicles. This information can be used to make informed decisions and are therefore helpful in achieving goals effectively. Alberts [**27**] describes some of the new sensors that are under development, like sensors that can provide the information while a vehicle is under attack. Alberts [**27**] also points out the importance of transmitting this intelligence to the next spatial level, the *off platform* level.

2.2.4.2 Inputs

In the given context, the inputs refer to the sensors and their parameters. Sensors read information from the environment surrounding the LAVs and this information is then processed by the on-board and *off platform* computers to be consumed by the Marines. These inputs increased the awareness of the Marines about the LAV and its environment and helped them to make the right decisions. The information from sensors can be delayed, missing or erroneous, thus leading to decreased awareness of the Marines or, in worst case, to wrong decisions. Thus, trustworthiness of sensors is an important component of the S&R project.

2.2.4.3 Outputs

In the context of this project, output can be described as the extent of achieving the goals set by the USMC. Examples of USMC goals include maintaining peace in a given region with minimum civilian casualties, popularly known as peacekeeping, or winning a battle with minimum Marine casualties. The success of these goals is dependent on the quality of the information; available combat power and the strategy employed by both the friendly and adversary forces. Quality of information is further dependent on the technological superiority among the forces. For the S&R project, technological superiority is achieved by the use of the sensors.

2.2.4.4 Inputs to outputs

Understanding the effect of sensors on other components formed a chain of events (i.e., the information obtained from the sensors), which increases the situational awareness of the operator assisting the decision-making process [**5**] and thus affects the mission effectiveness. This chain of events, represented in Figure 2, is an extension of the one given by Alberts [**27**] for his five measurement steps of network centric warfare as shown in Figure 3.

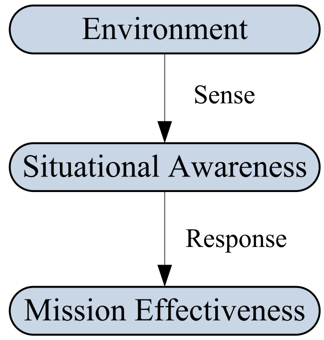

Figure 2: Event Chain

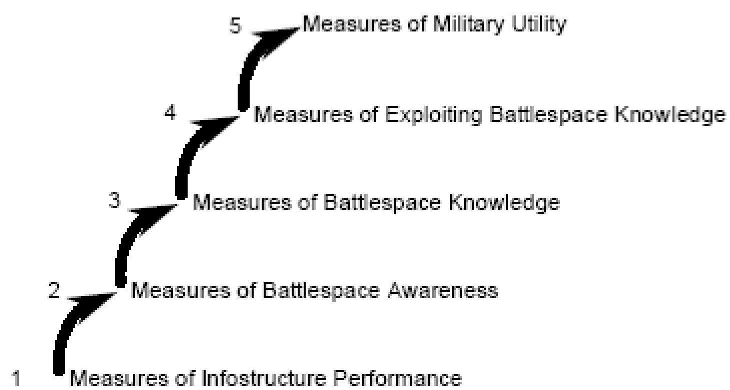

Figure 3: Hierarchy of Measures [27**]**

2.2.5 Comparison Metric

For the results of simulations to be quantifiable and easily distinguishable, the output of simulations had to be quantifiable. To find a quantifiable metric for effectiveness of mission, the effectiveness is decomposed into smaller problems [**22**]. The chain of events shown in Figure 2 was used to decompose the problem of measuring the effectiveness of the S&R project missions

[**27**]. Using the decomposition technique, it was able to find a quantifiable metric for either the situation effectiveness or the mission effectiveness.

2.2.5.1 Situational Awareness

Endsley [**28**] defines situational awareness (SA) as, "knowing what is going on around you". SA is generally used in operational terms, that is, it is not important to know everything and the information known may not be relevant to the mission. Thus, the context in which the information is used is important, that is, SA is a means to an end and not the end itself. SA holds meaning only in the defined context and relative to a goal. For example, knowledge of what an enemy had for breakfast is not important during a combat mission, but how far away he is and what ammunition he possesses is critical.

During this research, it was observed that the techniques known to measure SA could not be easily quantified. This claim is partially supported by Evertsz et al. [**29**] when they mention that the most widely accepted definition of SA (given by Endsley [**28**]) is qualitative in nature and knowledge provided by SA is, in general, subjective. To provide further support to this claim, a couple of techniques to measure SA – subjectively – are discussed.

The first technique employed a subject observation method in which simulations are run on real fighter planes while experienced pilots acting as observers mark down a predefined checklist of measures of SA [**30**]. Along with the problem of quantifying SA, this method was not applicable to the current problem since it leads to different measures of SA based on the observers' understanding of SA.

Situational Awareness Global Assessment Technology (SAGAT) was another technique used for measuring SA. Endsley [**28**] defined it as the "global assessment tool to access SA across all of its elements based on a comprehensible assessment of operator requirements". Operator's perception was observed at random time intervals by asking a set of questions which were directly related to SA, thus assessing the SA. In order to correctly assess the SA, the tool uses the concept of freezing the moment. Blanking the displays in front of the operator freezes the instantaneous knowledge of the operator, thus avoiding new information being provided to the operator. This tool, like the one given by Bell and Lyon [**30**], is based on subjective measurements and cannot be easily quantified. Another issue with using this tool in the current research was excessive user involvement. All of the measurements require at least one operator to be involved in the process.

2.2.5.2 Mission Effectiveness

In the context of this project, mission effectiveness can be described as the extent to which the goals set by the USMC are achieved. Few quantitative measures are known to measure this technological superiority, which directly affects mission effectiveness but has not been standardized.

The first method considered to measure mission effectiveness is the ratio of the knowledge of Blue forces to the knowledge of Red forces [**31**]. Knowledge in this context is similar to the SA defined in section 2.2.5.2. Although it's a good indicator of battle efforts, measuring knowledge requires analysis of qualitative data as described in section 2.2.5.2. This qualitative data fails to provide persuasive evidence in the case of minor gains [**30**].

Another method to analyze mission effectiveness is the force loss exchange ratio (FLER) [**31**]. This ratio provides an estimate of the approximate losses faced by each force in combat.

$$FLER = F = \frac{Fraction\ of\ Red\ Losses}{Fraction\ of\ Blue\ Losses}$$

One of the complexities associated with the use of this metric is that many unforeseen factors can also contribute to the losses in the real world environment [**32**]. For example, bad weather can impact one force more than the second force depending on the kind of vehicles used; thus contributing to the losses of one force more than the other. Another known problem with the use of FLER as a metric is that the metric only represents the ratio of losses and ignores the magnitude of actual, individual losses. A minor difference in percentages shown on the graph can correspond to both negligible and severe losses [**32**]. Although these problems associated with FLER make its use in the real world unsuitable, in a simulated environment these problems can be minimized, making FLER useful.

As discussed in Section 1.2, the first step in this research was to determine the measurement technique for the impact of sensors. Considering that simulation methodology and parameter sensitivity analysis requires a quantitative measurement for better result generation and interpretation, FLER was used as the measurement technique.

2.2.6 Scenario Utility Modeling and Measurement Integration Tool

This work can be considered an extension of Scenario Utility Modeling and Measurement Integration Tool (SUMMIT). SUMMIT provides an interface to compare two models where each model can be represented by a hierarchical structure as shown Figure 4 [**33**]. The model consists of scenario groups, which are subdivided into scenarios. Each scenario can have several tasks that contain interaction elements. Components form a sub-part of interactions, wherein each component can have some cost-benefit and risk associated with it. In the context of the S&R project, sensors are categorized by components and form an important part of the design. The components have benefits, costs and risks associated with them.

In the absence of standardized metrics to generate the utility of a project, it is important to have methods that can compare different models of the project. SUMMIT provides that kind of interface by comparing models and components on the basis of benefits, costs and risks associated with them.

One limitation of the tool is that the benefits, costs and risks associated with each component need to be quantified by the user. Costs and risks associated with the components are easily identifiable, but quantifying benefits is a difficult task.

Another limitation of the tool is that the benefits, costs and risks in general do not represent a single variable in the real world environment, but a combination of multiple factors. Manual identification of these factors to generate a single variable is time consuming and prone to errors. To mitigate this problem of manually identifying benefits, costs and risks, the current work provided a unique simulation-based approach where human involvement was minimal once the environment was built. Human interaction was only required for providing a set of input parameters that were easily identifiable in the real world. We discuss these variables in detail in section 3.2.7.

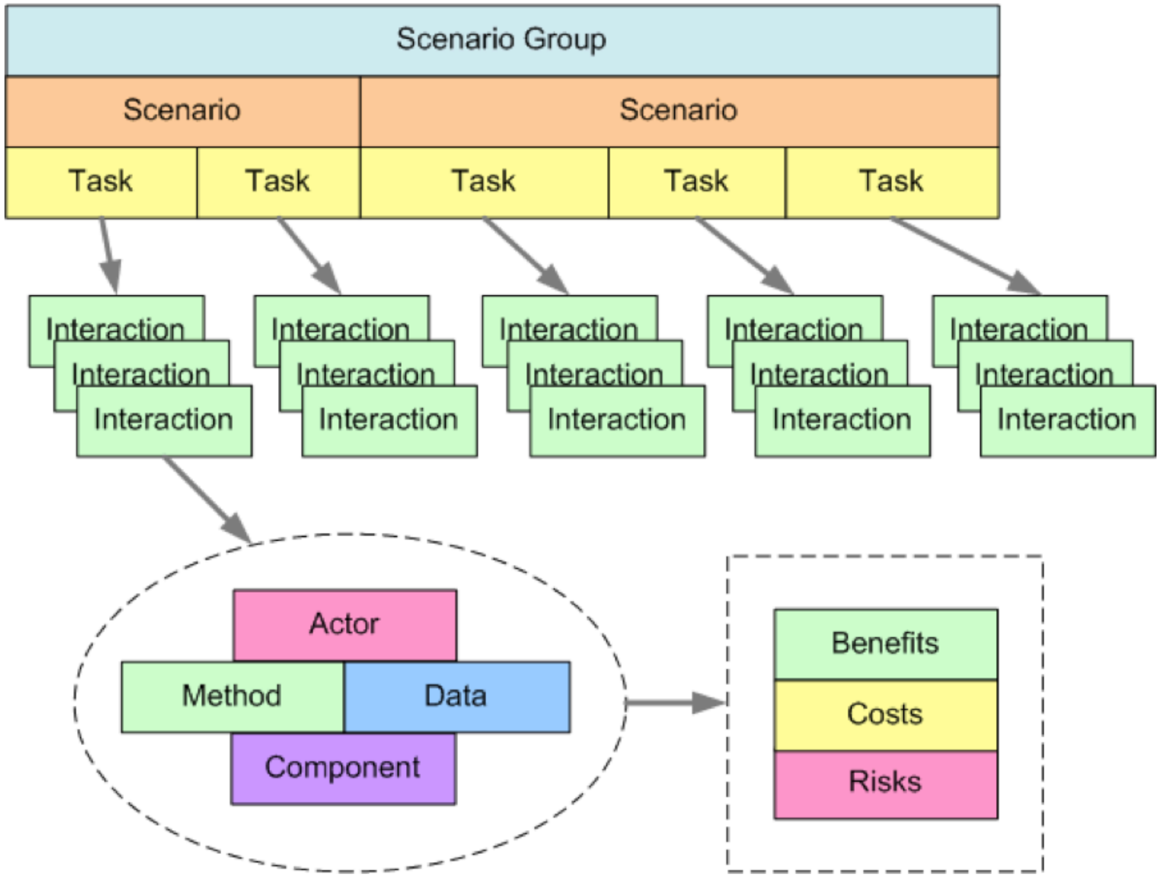

Figure 4: SUMMIT Design (Haynes, 2009)

2.3 Agent-Based Simulations

Before discussing the simulation design, it is important for the reader to be familiar with the concept of an agent. In the current research, the aim is to simulate the real world scenarios where LAVs operate. This includes scenarios such as combat missions, rescue operations and peacekeeping. In such a simulated environment, an agent is the equivalent of a LAV in the real world. A detailed description of the environment will be provided in 2.2.6 with the discussion of the design and implementation of this research.

In the words of Weiss [**34**], "An agent is a computational entity such as a software program or a robot that can be viewed as perceiving and acting upon its environment and that is autonomous in that its behavior at least partially depends on its own experience." Computational entity implies that an agent is a software program that runs on computation devices. The environment is perceived through the use of sensors and the effectors act upon it. The word 'autonomous' implies that agents have independent decision-making capabilities and are capable of working without human intervention. Another important point about agent's decision-making methodology is that its actions are not the same even under similar environments. This means that the agent can react in one of a multiple ways in the same situation. Thus, it can be concluded that an agent's decision is flexible and rational, which is the primary motivation for using agent-based solutions in this work.

Agents featuring this kind of flexibility are commonly referred to as *intelligent agents*. Intelligent agents are different from sensors in the sense that agents have flexibility. To understand flexibility in intelligent agents, one could look at the difference between a street light sensor and a LAV. A street light sensor meets the Wooldridge criteria of reactivity and pro-activeness [**35**] to some extent since the sensor has just one goal instead of multiple goals; it does not have social ability. On the other hand, a LAV is reactive, can have multiple goals and can also be collaborative (discussed in Section 2.3.1.1). In this context, Wooldridge [**36**] defined flexibility as having three characteristics: reactivity, pro-activeness, and social ability [**35**]. Reactivity is defined as the ability to react to the perceived external environment. Pro-activeness is the ability to take actions that help achieve some predefined goals. Social ability is the ability of the agent to interact with other agents and to make a collaborative decision to achieve the predefined goals.

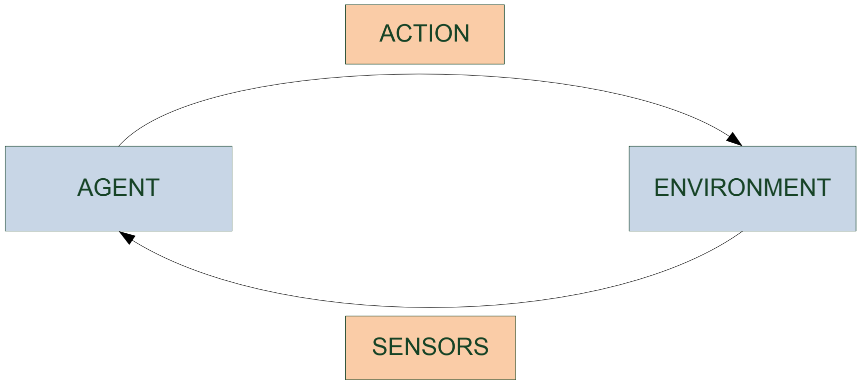

Figure 5: Overview of Agent Based Environment [36]

A basic overview of an agent-based environment is given in Figure 5. An agent perceives the environment through the use of sensors and acts on it through effectors. These actions have only partial control over the environment (i.e., the same action by agents in similar initial environment settings can have different results). Agents must be prepared to face such changes of a non-deterministic environment. An agent that is not ready to act on all possible actions can lead to a faulty behavior like a system crash.

2.3.1 Multiple Agents

This section introduces two concepts related to multiple agents that are relevant to this research. The first concept is coordinated decision making and the second is goals are defined.

2.3.1.1 Coordinated Decision Making

Autonomous behavior is an issue in the case of multiple agents, since agents are not aware of the behavior of other agents. The problem is similar to human decision-making, where a human cannot know the actions of another human unless communicated [**35**]. In order to achieve the goals, agents must communicate their actions to other agents. This kind of agent-agent interaction or agent-human interaction leads to the possibility of coordinated decision making in a simulation environment.

Decision-making can be categorized as independent or collaborative. In individual decision-making, the agent may know the actions of other agents and may disclose its decisions to other agents, but it does not seek the advice of other agents. The agent decides its course of action independently based on the information it has. In collaborative decision-making, agents share their knowledge of the environment and then collectively decide their actions.

This project uses multi-agent solutions with individual decision-making and collective goals. With individual decision-making, an agent's knowledge about its environment is expected to be only partial, but the environment's knowledge of this agent's decision can be complete, partial or none. For simplicity, it was assumed that agents are interacting only with other agents, but make independent decisions. They share the common goal of defeating the enemy.

One of the possible extensions to this research is to use coordinated decision-making. Recognition-Primed Decision enabled agent architecture (R-CAST) as discussed by Fan et al. [**37**] is one of the possible solutions. R-CAST is an adaptive model, and one of the recommended extensions of this work is to move from reactive architectures to architectures which make decisions based on previous knowledge. R-CAST is a good alternate to be used for team-based decision making.

2.3.1.2 Defining Goals

With coordinated decision-making, goals for the agents can be defined either individually or collectively. Individual goals imply that each agent has independent goals and works towards it. An example of an individual goal is the patrolling mission, where one agent may be able to perceive some dubious activity in its surroundings while another agent might fail to do so. If a goal is defined collectively to a set of agents, the success of an individual agent is determined by the overall success of the mission. In other words, irrespective of the fact that an agent was able to perform tasks assigned to it or not, it can still meet both results (i.e., failure or success).

2.3.1.3 Why Use Agent-Based Simulations?

The kind of diversification provided by the autonomous behavior is the key reason for choosing agent-based simulations over other simulation methodologies. Modeling the process of sensor selection in the S&R project is similar to modeling human tendencies in requirement selections such as the ways human think and make decisions. Making the choice of agent-based simulations is consistent with Kleijnen [**38**] who concludes that an agent-based simulation is the best way to match such decision-making. Simulations methodologies using mathematical modeling to formulate a solution and exploring the system in numerical terms fails because no known mathematical model is able to satisfy the large number of variables, a few of which are still unclear to human understanding. With an uncertain understanding of the system, a method is required which can be used even if some of the parameters are unknown and still forms a better understanding of the system as a whole.

Collins et al. [**39**] discusses a tool designed on the principles of expected utility theory, which is used for supporting certain decisions in an automated contracting environment. The first is deciding the composition of the bids and the second is awarding these bids to the right person. The solution shows that agent-based automated systems can be used to augment human decision-making processes if the realities can be incorporated and sufficient infrastructure is available to do so.

Kewley et al. [**40**] point out that "the primary purpose of agent based simulations is to gain insight and a deeper understanding of the system." This makes agent-based simulations an important tool to analyze the contribution of one sensor parameter in the larger combat environment. Section 2.3.2 discusses some examples to support the use of agent-based simulations in this research.

2.3.2 Agent-Based Simulations in Research

Dekker [**41**,**42**], measured the agility of the networked military systems using simulations. In this experiment, he discussed the extensions to the simulation tool CAVALIER used to study the performance of networked military organizations. The environment was used to demonstrate how networked military systems are affected by sensor range, networking and speed of movement.

The results can also be used to study a number of different scenarios quickly. The results achieved in Dekker's simulation are very reproducible and can be proven. The results generated have a direct application in decision making as they can help the Marines study the effect of parameters such as the speed of movement on the final outcome. Thresholds can be found, beyond which investing in increasing the speed of movement will not deliver the required utility. Incorporating the sensors in an environment similar to that used by Dekker can also assist the study of sensor parameters as required.

A conflicting result between two theses provided an insight to the dangers of simulation methodology. The thesis of Erlenbruch [**43**] discusses, in detail, the ways in which simulations can be used in military based environments. It also discusses an experiment to find the best policy to be used by peacekeepers to meet the objectives of defensive tactics.

In Woodaman's thesis [**44**], an agent-based simulation was developed to imitate the environment of military operations that differed from wars. The thesis demonstrated that proactive behavior used by peacekeepers leads to a drop in causalities during riots. In this thesis, the results of Woodaman [**44**] are used while simulating the behavior of agents, for example, the agents have tendency to attack. The results in Woodaman's [**44**] thesis conflicted with those described by Erlenbruch's [**43**]. Though similar scenarios are being modeled, the implementation methodology creates all the difference. These conflicting results motivated repeated reviews of the design in order to find problem areas in the current research. This conflict also showed that results of simulations, though reproducible, are still only indicative and cannot be fully trusted. An important observation is that program managers of the S&R project should use the results obtained in this thesis for practical purposes only after concept and design verifications.

Ritter et al. [**45**] in their work on dTank, discuss a tool, which provides a methodology to analyze various agent architectures used for SA. The tool provided capabilities for scenario development and behavior implementations along with mechanisms like creation of the agents, communication among the agents, etc. The tool was one of the options for this work, but extending the tool to calculate sensor utility in the given timeframe for this project was not possible. To complete the project in due time, dTank was not used.

The work of Erlenbruch, Woodaman and Ritter [**43**] - [**45**] provided an insight into how agent-based simulations have been effectively used for military based operations. Based on the research discussed in Section 2.3.1.3 and these examples, agent-based simulations were considered to be the most appropriate tool.

2.3.3 Agent Architectures

In this discussion, a few possible agent architectures are explored. Chapter 3 discusses in detail the design and implementation of agent architectures used in this project.

2.3.3.1 Functional Systems

Functional systems [**36**] is the first kind of architecture to be considered. These are goal-oriented systems where the agents make a particular decision and follow it to achieve their goals; examples of this type of system include compilers in computer programming or search queries on Wikipedia. These do not take into consideration dynamic changes in the environment while the agent is emulating a particular scenario. If the conditions are not defined to deal with changes in the environment, these systems are highly susceptible to failure.

In large practical systems, this type of programming is generally not acceptable as the environment is too dynamic and uncertain leading to the failure of all assumptions. Considering the frequency of changes in environment where the S&R project is used, a functional system would fail.

2.3.3.2 Logic-Based Architecture

In *logic-based architecture* [**36**], an agent starts with a set of rules and tries to apply these rules to the information it has to generate and to the set of actions that it should perform. These rules form the program while the information it needs comes from the sensors. This type of architecture can be found in e-mail filter programs. Whenever a new e-mail is received, the agent tries to figure out whether or not the email is spam based on a dynamic set of rules. The agent tries to match this e-mail to one of the rules in the set, such as spam e-mail from registrar@psu.edu. Assuming that e-mail is received from registrar@psu.edu, then it is directly sent to the inbox without further queries. Otherwise, the agent continues scanning e-mails with other rules in order to formulate its decision. Now, consider a case where one thousand similar e-mails are received from different IDs; the agent can be programmed to understand such a behavior when a certain number of emails have the exact same content, then such e-mails are considered spam and should be filtered. Though this adaptation was considered useful, a lag exists before such adaptation can actually be used. Due to this lag, the architecture is not considered useful in the current work.

2.3.4 Reactive Architecture

The third agent architecture is reactive architecture [**36**], the approach of acting only on the current known information about the environment. The system has a list of priorities that it tries to meet in the current environment. The deduction is not based on logic from multiple rule definitions like in the previous architecture, but only on the current state of systems and priorities. The system does not take into account past experiences. All of its decisions are based only on the current known state of the environment. This kind of system is computationally much more efficient than logic-based architectures. Sensitivity to the environment made this architecture an important one to be considered for the S&R project.

2.3.5 Belief-Desire-Intention Architecture

In Belief-Desire-Intention (BDI) [**36**] architecture, the agent starts with a particular set of goals and then develops a priority order for pursuing those goals. A set of goals refers to belief; priority order refers to desire and pursuing a goal refers to intention. Agents keep track of the feasibility of the intention, discarding the infeasible intentions and moving to the next goal in priority order. At any instant, agents' actions are determined by intention.

As the agent keeps track of feasibility, this kind of architecture is sensitive towards striking the right balance between working towards a goal and reconsidering the goals. An agent trying to reconsider too quickly may not be able to achieve any goals successfully. Conversely, an agent delineating the job of reconsidering might keep on wasting efforts towards unachievable goals.

Evertsz et al. [**29**], [**5**] used BDI architecture in CoJACK where they provide an environment for cognitive architectures and BDI language by extending JACK [**46**]. Evertsz et al. [**5**] define cognitive architecture as "a computation system that models the structural properties of the human cognitive system." An extension to CoJACK was considered as an option in this research, but being that reactive architecture (section 2.3.4) was more suitable to provide sensitivity analysis for simulations, this was not used.

2.3.6 Layered Architecture

Layered architecture [**36**] is the last architecture considered for this project. Taking into account the required system behavior of reactivity and pro-activeness, the concept of layering comes into view. Multiple layers make decisions and generate suggestions. There are two kinds of layered architectures, horizontal layering and vertical layering. In horizontally layered architectures, each layer has access to both inputs and outputs. Based on the inputs, each layer makes an independent suggestion for what actions should be taken. In the case of vertical layering, only one layer interacts with inputs and the other with outputs. The layers that interact with inputs and outputs can be combined into a single layer. Starting from the input layer, each layer in such architecture makes its suggestions to the next layer and so on. This decision-making can be achieved in one pass, where control passes through each layer only once, or in two passes, wherein control passes through each layer twice.

2.3.7 Design Choice

To answer the fundamental research questions and perform sensitivity analysis of sensor parameters, an architecture that closely relates to the effects of information received from sensors was required. Logic-based architecture built upon a set of rules can make interpretation of results a difficult task since this requires understanding each rule's contribution towards the overall mission effectiveness. Based on this practical problem of interpreting the results, logic-based architecture was not used. BDI architecture was considered more useful when it came to higher level planning and analyzing the response strategies of the S&R system, but was less sensitive towards sensors' inputs. As shown in Figure 6, the responsive nature of reactive architecture towards the information received from sensors was the primary reason for its use in this implementation. Another reason for its use was that this architecture allows easy modular modeling of the sensor inputs and outputs, which was one of the design requirements described in Chapter 3.

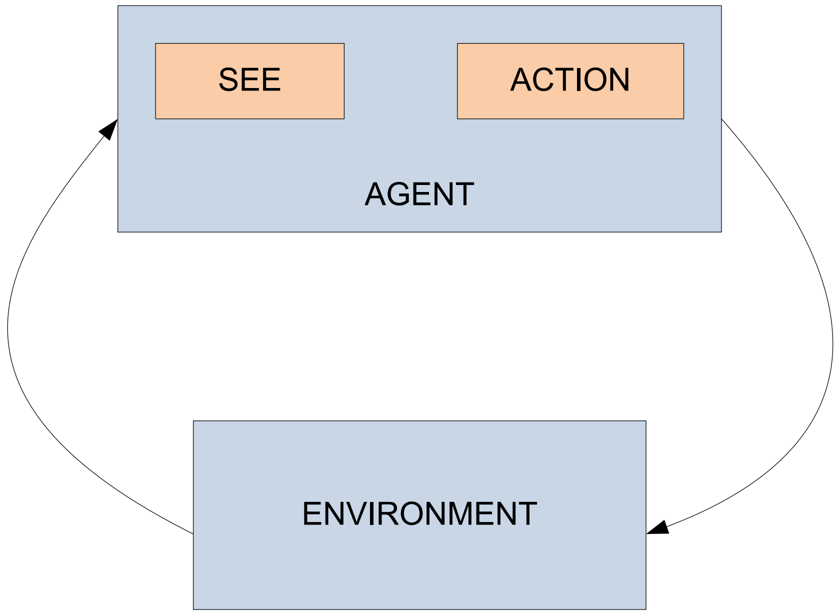

Figure 6: Architecture Used [36]

Chapter 3

Design and Implementation

Developing a modular, agile and high performance simulation tool to imitate an environment where LAVs operate was the primary design goal. This section describes a tool that simulated a scenario where the role of sensors used in the LAVs was clearly observable. Ability to extend this tool for more sensors necessitated a modular design, wherein each module would allow further expansion of code to account for the real world parameters that were not part of the current implementation. Stakeholders want to see impact of change in requirements on the overall system as quickly as possible, thus the tool's performance was also very important. Another design goal was that the tool should be compatible with SUMMIT, so that future versions of SUMMIT that incorporate this tool would be able to provide inputs and get FLER as output. Using AHP for decomposing the S&R project into smaller subcomponents, the project is broken down into environment, events, vehicles and sensors.

3.1 Scenario

The scenario used in this research is similar to the one used by Dekker [**41**] in his experiment involving two kinds of forces: the blue force and the red force. At the start of the simulations, these forces were present in opposite corners of a grid as shown in

Figure 7. All the agents were equipped with sensors, defense weapons and attack weapons. The equipment (e.g., sensors) and intelligence was similar in both the forces. However, the simulation parameters for these equipment (e.g., accuracy parameters for the sensor) could differ between the two forces; thus, creating an imbalance between the capabilities of each force.

A base camp was added to this simulation environment for each force. In this experiment, each force's base camp was placed on corners diagonally opposite on the grid. The purpose of the base camp was to provide agents with ammunition and repair the damages to the defense mechanism of the agent. As long as the base camp had supplies available, it provided services to agents on a first come first serve basis.

The experiment consisted of 10 agents for blue forces and 10 agents for red forces. The initial placement of the blue forces was concentrated towards their base camp whereas the red forces were scattered all around the grid. It was assumed that the blue forces are aware of the red force's initial position and tried to reach the red forces. To keep the implementation simple, agents were purely reactive and did not make decisions based on past knowledge. For example, a red agent would attack a blue agent only when it saw an agent in its vicinity or got the information that an opponent agent was stalled. The information about why an agent got stalled will be discussed in section 3.2.3.

The implementation in this work can be categorized as permissive role of engagement[1], that is, it was assumed that opponents were always hostile. Thus, when either of the forces sensed an opponent in visible range, their default behavior was to attack the opponent.

[1] Rules of engagement as defined in the Dictionary of Military Terms are, "Directives issued by competent military authority that delineate the circumstances and limitations under which United States forces will initiate and/or continue combat engagement with other forces encountered" [**52**]. As defined by Evertsz et al. [**48**], there are five classes of rule of engagement (ROE), namely the right to self-defense, restrictive ROE, permissive ROE, preference ROE and criterion-defining ROE.

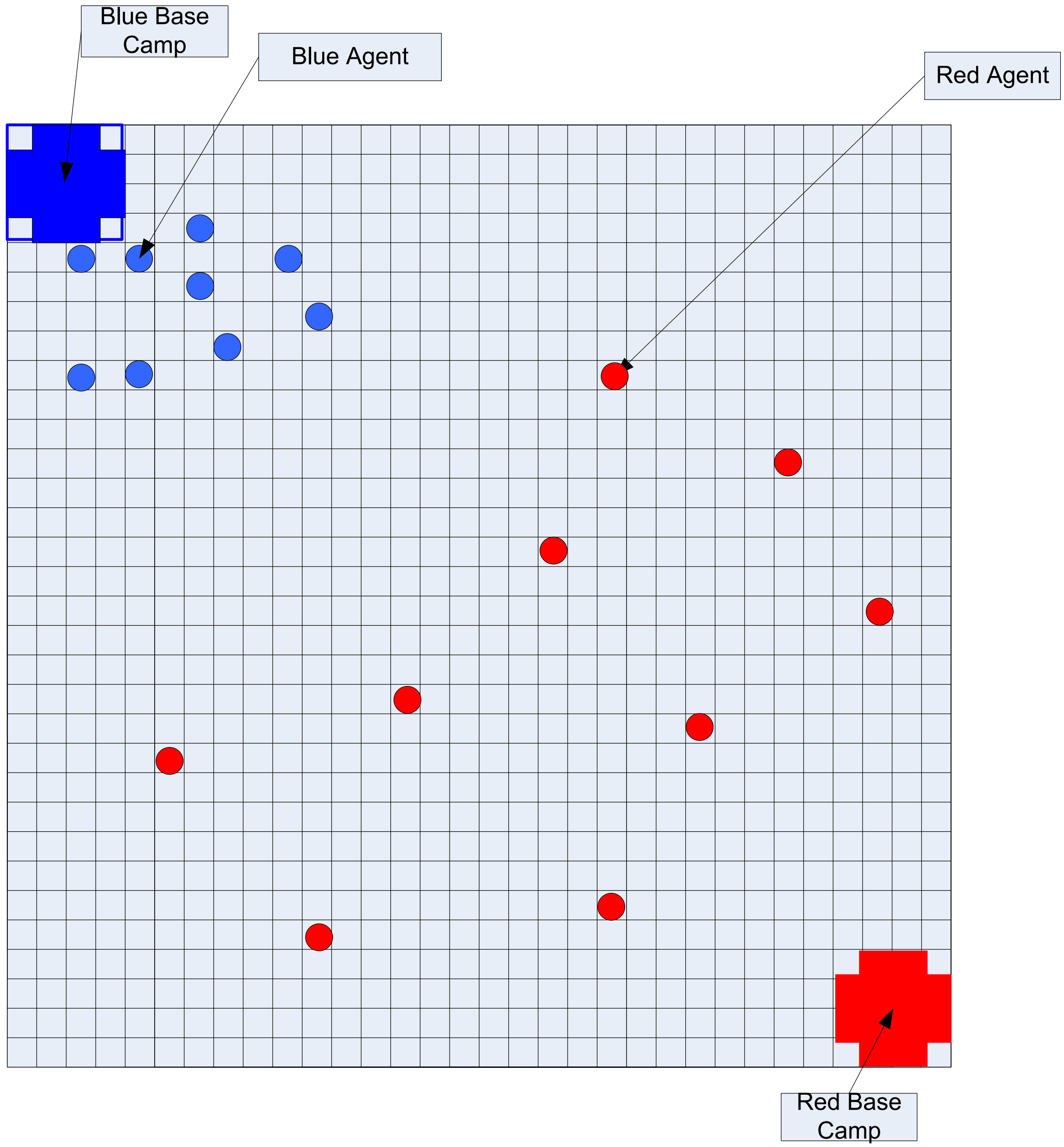

Figure 7: Simulation Environment

3.2 Implementation

This section outlines the major design features of the implementation. It first discusses the design choices, that is, the design is discrete event driven. Next, it describes how a point system was implemented to generate FLER. Lastly, it describes component wise implementation details.

3.2.1 Discrete Event Driven

The first feature of the current implementation is that it was discrete event driven or there was a queue of events sorted by their execution time. Discrete event driven paradigms were chosen over continuous even driven systems for performance reasons [**47**]. Even if continuous event driven systems are easier to program, the unnecessary computations at each time quantum makes them substantially slower.

3.2.2 Point System

Another design requisite was to imitate the real life behavior of attack and defense, accounting for the quantitative losses required in the FLER's measurement. This was achieved by implementing a point-based system wherein each agent had some attack and defense points associated with it. Attack points represented the ammunition available to the agent while the defense points represented the shield associated with the agent.

3.2.3 Events

As described in Section 3.2.1, the system was discrete event driven, that is, simulation was executed as a chronological sequence of events; an event occurred at an exact instant and the state of the system changed based on this event. To understand the modules described in Section 3.2.4, an overview of the events that were implemented is required.

3.2.3.1 Is Visible

Whenever an opponent came into the visible range of an agent, the ‘is visible’ event was triggered. The agent always took an action such as attacking the opposite agent or going back to a safe place based on its current situation. The choice of action depended on a combination of factors including goals assigned to the agent and the agent’s state.

3.2.3.2 Attack

This decision-making process of whether or not to attack the opponent was made by the agent. The occurrence of a successful attack was dependent on a number of factors, including logistics and success rate. In the case of a successful attack, the agent inserted an event named ‘was attacked’ in the opponent’s event queue specifying the points to be deducted from the opponent’s defense.

3.2.3.3 Was Attacked

As described in Section 3.2.3.2, this event was inserted whenever an agent attacked their opponent. Since this was the mark of a successful attack, it always deducted the specified points from its own defense, except when the agent had already resigned, in which case all the events for the resigning agent were removed from the event queue including the ‘was attacked’ event. In this case, the attack from the opponent was wasted, since the opponent attacked a dead agent. Following the point deduction, the agent decided whether to retaliate or retreat according to the logistics and its current mission.

3.2.3.4 Reload

This event was inserted when the agent required either more ammunition or was in need of repair (i.e., when it was running low in attack or defense points). To reload, the agent had to reach its base camp and ask for reloading, in which case the base camp provided both ammunition and repairs to the agent (LAV) irrespective of the requirement, bringing the agent back to its initial strength. In the case that the base camp had run out of ammunition, the agent resigned to avoid any further losses.

3.2.3.5 Heating

The agent was always in one of the four states associated with its temperature parameter. These were heating cool down, breakdown, and cooling. During the 'heating' state, the temperature of the agent's planetary gear was rising. This temperature increase was either dependent on the distance travelled by the agent or the time since the last 'cooling' state.

3.2.3.6 Breakdown

The planetary gear had an operating temperature threshold. If the threshold was crossed, it resulted in vehicle break down. In the case of a breakdown, the vehicle faced two penalties: 1) the vehicle halted with temporary loss of attack capability and was still vulnerable to attacks from the opponents, and 2) constant loss in the vehicle's defense.

3.2.3.7 Cool Down

Every time the agent's planetary gear reached a threshold temperature, it had to halt upon reaching a safe place. However, finding this safe place took time and it was assumed that during this time the vehicle remained vulnerable to attack but was not able to attack the opponent. If the cool down time was set to zero, there was no penalty associated with finding a safe place. A suggested extension to this event is the creation of safe places in the operating environment of agents and the removal of this constant time.

3.2.3.8 Cooling

The cool down event was always followed by the cooling event. During the time of cooling, the agent was protected against attacks, and at the same time, was incapable of attacking the opponents. This event completed the cycle started by a 'heating' event (heating – breakdown (occasional) – cool down – cooling).

3.2.4 Modular design

A high level diagram for the modules is described in Figure 8. The implementation was divided into three modules including mission control, simulation environment and agent. Mission control defined the goals for the agents; simulation environment imitated the real world environment where the LAVs operated; and agents represented the LAVs for both the friendly and adversary forces. As used by Dekker [**41**], the friendly forces were called the 'blue force' while the adversaries were labeled the 'red force'. Figure 9 shows the important classes implemented during the project.

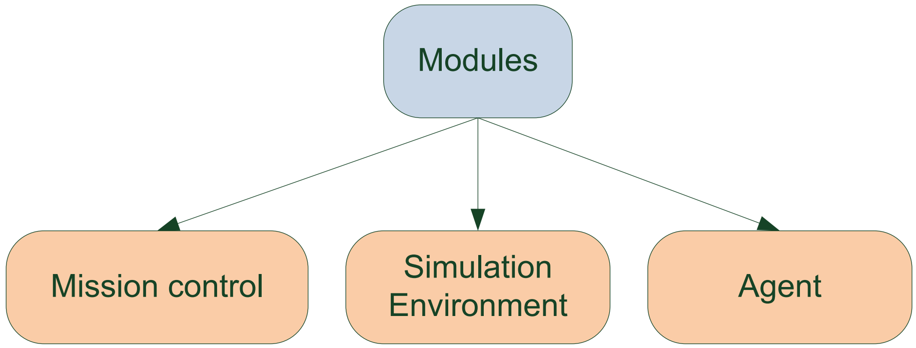

Figure 8: Modules

Simulating reactive architecture required efficient interaction between three components: sensor inputs, decision-making and response. Selection of reactive architecture conflicted with the modularity requirement of the easy addition of new sensors. In order to justify the reactive nature, decision-making needed to be directly dependent on the sensor inputs by making use of a major part of the intelligence received from sensors. However, the requirement of the ability to add new sensors required that the decision making code be independent of the sensors. Creating hooks within the agent code where changes in the value of physical parameters of the agent were followed by registering to a list of sensors solved these conflicting requirements. These sensors make appropriate changes in the perceived value of physical parameters accounting for accuracy errors, resolution or speed of update for the sensor. Details about how to perform these actions are provided in Section 3.2.5.

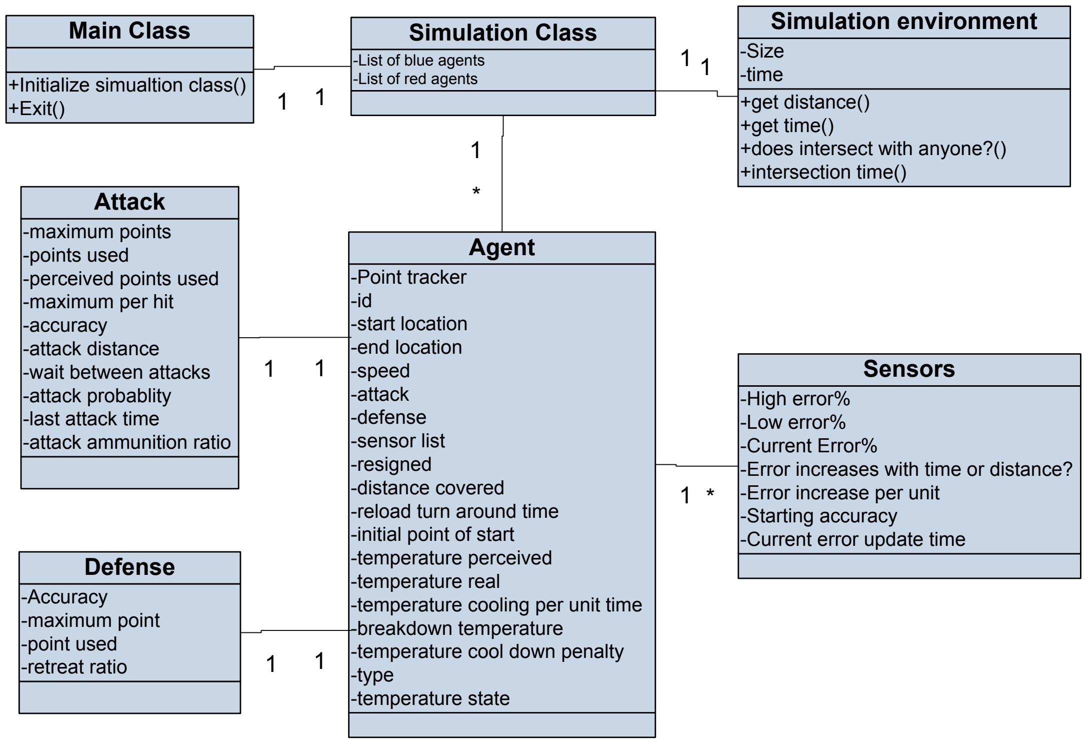

Figure 9: Classes Used

3.2.4.1 Mission Control

Evertsz et al. [**48**] found that results are highly dependent on the scenario chosen. They also found that hard-wiring the ROE into tactics is not a useful option. Based on this research, the simulations implementation supported multiple missions. While the environment supports multiple missions, only one scenario was implemented as part of this research.

The mission control part of the program was responsible for managing and defining the mission, thus performing most of the *off platform* activities discussed in Chapter 1. Mission control affected the outcome by initializing the agents and defining the kinds of actions that needed to be performed by both the blue and the red force. Although the current implementation only linked mission control to the blue force, and the behavior of the red forces was to consistently attack the blue force, the design took into consideration the possibility of linking mission control to the red forces.

3.2.4.2 Simulation Environment

This module kept track of the physical constraints and variables associated with the environment in which LAVs were expected to be in action. An example of physical constraints is keeping track of the grid size (i.e., the boundaries of LAVs operation in the real world environment). An example of variable is simulation time. The environment also provides the functionality to calculate values associated with variables, like the distance between two agents.

3.2.4.3 Agent

As mentioned during the introduction to 2.2.6, agents had the ability to imitate the human thinking process thereby providing the ability to capitalize on the SA gained by the use of sensors like a human would. The project implemented a very simplistic model of human behavior with the use of probabilistic parameters. Each optional action taken by an agent was based on some probabilistic variable, though some of the agent's actions were enforced (for example, the decision to attack the opponent versus the enforced move back to the base camp if the vehicle was out of ammunition). Along with decision-making that was implemented within the agent, the actions were also dependent on two children classes, namely attack and defense.

3.2.4.3.1 Attack

Whenever the attack event occurred, the control directly entered the attack object associated with this agent. In this case, the agent decided if it could and should attack an opponent based on several factors including the physical constraints, logistics, attacking tendencies of the opponent, current mission and probabilistic values generated for the attack. Physical constraints included the distance of the opponent, while logistical constraints included the ammunition available. The nature of the opponent was determined by keeping track of their attack pattern based on prior attack history. The probabilistic variables were accuracy and frequency of attack by this agent.

Finally, if a successful attack was performed, the agent deducted its own ammunition points and created an event named ‘was attacked’ to deduct points from the opponent’s defense, indicating losses to the opponent.

Other events created by this module included ‘attack to the same opponent’ and ‘reload’ if the agent found its ammunition below a certain threshold. As the decrease in ammunition points represented a value change in logistics, the class provided functionality for the registration of the sensors, which was executed following the change in value.

3.2.4.3.2 Defense

The defense module was used to track the agents’ necessity of retreating back to the base camp to repair the damages that occurred on the agents’ shield. Tracking the loss of defense points and triggering a ‘reload’ event when the remaining defense fell below a predefined threshold achieved this. This module also provided an input to the attack module regarding the nature of the opposing agent such as whether the opponent was in attack mode.

3.2.5 Sensors

An abstract class was created that can be extended by sensor developers to add new sensors to the tool. The class provides functionality to maintain the error percentage based on both the time and distance travelled by the agent. A new sensor can be registered to trigger invocation of the sensor's execute functionality whenever the value of the agent's physical or logistics parameter changes. To demonstrate the power of the tool, a few sensors were implemented.

3.2.5.1 Shots Fired

The shots fired sensor kept track of the number of bullets fired from the gun barrel of an LAV. By keeping track of shots fired, an agent knows how much ammunition it has available and can decide if it should continue with the mission or go back to the base camp to get a refill. Assuming that multiple vendors provide a sensor that can track shots fired from the LAV, and that use of shots fired sensor is a requirement for a mission, it is important to prioritize which shots fired sensor to use. Performing utility analysis on the sensor to determine the benefits, cost and risk of using it can do this requirement prioritization. In this work, risk analysis is demonstrated by looking at inaccuracy in measurement of the sensor. The developed tool is extendable to perform benefit analysis.

Sensors, being electronic devices, are subject to having some inaccuracies in their measurement. For demonstrating the usefulness of the developed tool, this work assumes that the shots fired sensor calculated the number of shots fired with some error. The variation from shots counted by the sensor to actual number of shots fired is defined as accuracy percentage for shots fired sensor. For example, a sensor with 90% accuracy can show anywhere between 90 and 110 for 100 shots fired by the agent. This results in one of the following two implications: either the agent ran out of ammunition without its knowledge or it decided to go back to base camp earlier than it should. In both cases, the utility to the agent was expected to decrease. In the first case, it was expected to bear losses on its way back to the base. In the second case, it might have been wasting critical time by going back to the base.

3.2.5.2 Planetary Gear Temperature Sensor

The planetary gear temperature sensor keeps track of the temperature of the planetary gear installed in an LAV. Keeping track of planetary gears temperature is important because it helps keep the LAV running during a mission without suddenly dying due to overheating. To figure out whether a particular planetary gear temperature sensor will meet the requirements of the LAV during a mission, it is important to know how technical specifications of a temperature sensor would affect the mission. As with the shots fired sensor, the utility analysis of the planetary gear temperature sensor was done by looking at the risk of using it. Due to the error in the sensor, the temperature perceived by the agent differed from the actual temperature of the planetary gears. Such changes in perceived temperature value have two implications: either the agent broke down as a result of the real temperature becoming too high or the agent made cooling stops too frequently thereby over paying the cool down penalty associated with each cooling event.

3.2.6 Points Tracker

A class named *points tracker* kept track of attack and defense points available to the agents and at the base camp. Any change in defense or attack forces was updated in the form of the numerical values in *points tracker*. After the simulations were completed (all agents had resigned or the designated simulation time was over), this tracker calculated the value of FLER.

3.2.7 Input Variables

The tool required approximately 40 variables to execute and user inputs were required for all of these variables to run the simulations. A default value was calculated for each variable. The latest version of the simulations environment could execute independently (without user intervention). To perform sensitivity analysis of a sensor parameter, simulations were run to calculate FLER by altering that parameter while keeping the other parameters constant.

The primary source of error was the randomization of certain parameters such as the starting location of agents. To remove these randomization errors and to obtain consistency, it was important to perform multiple runs of the simulations [**49**]. Even though consistent results could be seen in a few runs, with the availability of a fast simulation environment, the simulations were run several hundred times before obtaining each data point plotted on the graphs presented in Chapter 4.

A description of each input variable is provided in the table below.

Table 1: Variable Descriptions

Variable name	Description	Example value
Current mission	An LAV could be used for multiple missions; each mission represented a different scenario. The simulation environment designed in this work can support different types of scenarios. Only rescue missions were implemented as part of this work.	Rescue mission

Stop time	Intermediate results can sometimes be more interesting to the user compared to the end results. Stop time provided the user with that capability. The environment can be forced to end after some time; time is measured in terms of simulation ticks.	10000
Simulation environment range x	The x component of grid size where agents operate.	200
Simulation environment range y	The y component of grid size where agents operate.	200
Number of blue agents	Count of agents belonging to the blue force.	10
Number of red agents	Count of agents belonging to the red force.	10
Defense assigned	As described in Section 3.2.2, point-based mechanism formed the basis for FLER calculations. The LAVs had some defense against attacks made on it. Each force was assigned some defense points, which represented the defense capability of the force. These points were distributed equally among agents at the start of the simulation.	3000
Defense buffer	While 'defense assigned' was equally distributed among agents at the start of the simulations, the defense buffer provided the base camp with the ability to repair agents when they came back to base camp. When the defense buffers reaches zero, and an agent reaches base camp to refill defense points, the agent resigns.	2000
Defense retreat ratio	It was assumed that the agents retreat to base camp while they still had some defense points left. At this ratio (defined as defense points remaining to defense points initially available to the agent), the agent started retreating to the base camp for repair.	0.3
Attack assigned	Similar to 'defense assigned', it was assumed that each force had some ammunition available which was distributed (equally??) among agents. This variable keeps track of the total attack points assigned to the agents of a particular force.	5000

49

Attack buffer	Total attack points available at the base camp of a particular force. When the attack buffer reached zero, and an agent reached base camp to refill attack points, the agent resigned.	3000
Attack max hits	It was assumed that an agent cannot carry more than a certain amount of ammunition in one particular round. This variable limited the maximum number of hits available to an agent, after which it had to reload from base camp to continue.	20
Attack accuracy	It was assumed that not every attack by an agent would be successful as an agent may miss its target. This variable set the accuracy of an agent when it attacks an agent of opposite force. If an agent was unsuccessful in attack, points were deducted from the agents attack but no points were deducted from opponent agents' defense.	85%
Attack distance	This variable limited the distance of agents attack in terms of grid blocks.	4
Attack wait between hits	It was assumed that an agent cannot fire continuously and must wait some minimum time between the two attacks. An agent waited at least this number of simulation ticks between two attacks.	5
Attack probability	An agent could not always attack an opponent and may either have skipped the attack or may have been unable to attack. Due to a simplistic human model, the system imitates this behavior by probability.	90
Reload turnaround time	When an agent changed its stance from fighting to visit base camp (for ammunition reloading or vehicle repair), it had to wait at its current position, thus paying a penalty. During this time, the agent is vulnerable to attack by opponents.	20

Gun barrel lower error percentage	Each sensor had some accuracy error in its readings. It was assumed that this accuracy error increases with time. This term defines the starting value of error.	5
Gun barrel higher error percentage	As above, this variable defines the highest value of error in a sensor.	20
Gun barrel error increase with time or distance	Error in the sensor's accuracy can increase both with time or distance travelled by the agent. This variable defined if the error in sensor was sensitive to time or distance.	Time
Gun barrel error maximum reached	How much distance or how many simulation ticks it took for the error reach its maximum value. This value was used in conjunction with 'Gun barrel error increase with time or distance'. The error was assumed to increase linearly with distance or time.	500
Planetary gear lower error percentage	Each sensor had some accuracy error in its readings. It is assumed that this accuracy error increases with time. This term defines the starting value of error.	10
Planetary gear higher error percentage	This variable defined the highest value of error in a sensor.	30
Planetary gear increase with time or distance	Error in sensor's accuracy could increase either with time or distance travelled by the agent. This variable defined if the error in the sensor was sensitive to time or distance.	Time
Planetary gear error maximum reached	How much distance or how many simulation ticks it took for the error to reach its maximum value. This value was used in conjunction with 'Planetary gear increase with time or distance'. The error was assumed to increase linearly with distance or time.	400
Temperature break down time penalty	Whenever an agent broke down due to an increase in the temperature of its planetary gear, it was assumed that the agent was not able to move for some time and was vulnerable to attack during that period. This variable defined the time for which the agent would stall.	40

Temperature cool down time penalty	Whenever an agent's planetary gear temperatures increased beyond a certain limit, it was assumed that the agent needed to find a safe place and wait for it to cool down. There was a penalty associated with temperature being above threshold that implied that the agent is not operational for a certain amount of time. Agent was vulnerable to attack during this time, but as it was not operational, it could not fight back. This variable defines the time for which the agent was searching for a safe place.	10
Temperature cool down	Temperature at which the agent needs to stall and wait for its planetary gears to cool down.	30
Temperature break down	Temperature at which the agent would stall suddenly and be unable to perform any actions. This variable was higher than 'Temperature cool down' and was only reached due to inaccuracy in the sensor used to measure the temperature of planetary gears.	40
Cooling per unit time	After reaching a safe place, the agent took some time to cool down. This variable defined the rate of the agents' recovery.	5
Temperature increase per unit	It was assumed that the temperature of planetary gears could increase with either time or distance. This variable defined the per unit (distance or simulation tick) increase in temperature.	0.1
Temperature increases with distance or time	This variable defined the factor to which the planetary gears temperature reacts.	Time
Speed	The speed of agents' movement in grid.	0.2
Attack ammunition ratio	Similar to 'Defense retreat ratio' this variable defined the ratio between ammunition left and total available ammunition.	0.2
Break down penalty	Variable defines the points detected from agents' defense system when it broke down.	30

3.2.7.1 Input Variable Effect on FLER

Table 2 shows the presumed impact of input variables on the FLER. This impact was ascertained based on certain assumptions; few of these assumptions are mentioned in the table. A future extension of this work would be to verify these assumptions and the impact of each variable on FLER using the simulation environment implemented during this work.

Table 2: Parameter Presumed Relation to FLER

Variable name	Assumed effect on FLER
Current mission	The impact could not be ascertained beforehand.
Stop time	If stop time were too small, it would have given an inaccurate FLER as the simulations can stop too early. At the start of the simulations, the losses between two forces might not have given accurate information about the scenario. Having a very high value of stop time gave a more accurate output.
Simulation environment range x	The impact could not be ascertained beforehand.
Simulation environment range y	The impact could not be ascertained beforehand.
Number of blue agents	The greater the number of blue agents, the higher the expected value of FLER.
Number of red agents	The greater the number of red agents, the lower the expected value of FLER.
Defense assigned	The impact on FLER could not be ascertained beforehand, but the accuracy of FLER was expected to be directly proportional to the defense assigned.
Defense buffer	The impact on FLER cannot be ascertained beforehand, but the accuracy of FLER was expected to be directly proportional to the defense buffer.
Defense retreat ratio	If the defense retreat ratio was too high, the agents would waste a lot of time coming back to base camp for repairs and would also pay the turnaround penalty. But if the value were too low, the agents would wait for too long and could be forced to resign by opponents even if they could have gotten repairs from the base camp.

Attack assigned	The higher the attack assigned to a force, the more favorable FLER was expected for that force. For example, if the attack assigned to the blue force is increased, FLER should increase.
Attack buffer	The higher the attack buffer assigned to a force, the more favorable FLER was expected for that force. For example, if the attack buffer to the blue force was increased, FLER should increase.
Attack max hits	The effect of the change in FLER with the change in attack max hits parameter was not deterministic because the attack assigned and the attack buffer determined the total attack assigned to each LAV. Attack max hits only determined the size of each attack. With a change in this parameter, either the force had more attack points in each hit but a lower number of hits, or a smaller attack in each hit but a higher number of hits.
Attack accuracy	With an increase in attack accuracy, FLER was expected to be more favorable for the force.
Attack distance	The longer the attack distance, the more favorable the FLER.
Attack wait between hits	With longer wait time between two attacks, the chances of getting hit by an opponent increased, implying an unfavorable FLER.
Attack probability	The impact cannot be ascertained in general, and would be dependent on the scenario. Should be favorable in a battle scenario, but adverse in peacekeeping ones.
Reload turnaround time	With a longer reload turnaround time, FLER was expected to be unfavorable.
Gun barrel lower error percentage	The lower error percentage implied a more favorable FLER.
Gun barrel higher error percentage	The lower error percentage implied a more favorable FLER and a higher error implied that the agent has either run out of ammunition while it expects to have ammunition or the agent goes back to base camp too soon, thus paying a turnaround penalty unnecessarily.
Gun barrel error increase with time or distance	The impact could not be ascertained.
Gun barrel error maximum reached	The higher the value of time or distance taken to reach highest error, the more favorable was the expected FLER as the agent would have been operating at a lower error percentage for a longer period.

Planetary gear lower error percentage	The lower the error percentage implied, the more favorable the expected FLER.
Planetary gear higher error percentage	The lower the error percentage implied, the more favorable the expected FLER as a higher error implies a higher chance of an agent breakdown and the agent would end up paying the steep breakdown penalty.
Planetary gear error increase with time or distance	The impact could not be ascertained because this variable determined if error increased with time or distance. Time and distance, being independent variables, cannot be compared.
Planetary gear error maximum reached	The higher the value of time or distance taken to reach the highest error, the more favorable the expected FLER as the agent would have been operating at a lower error percentage for a longer period.
Temperature break down time penalty	The higher the value implied, the higher the penalties and the more unfavorable the expected FLER.
Temperature cool down time penalty	The higher the value implied, the higher the penalties and the more unfavorable the expected FLER.
Temperature cool down	The higher the temperature limit, the longer the agent could operate and the longer it is able to attack opponent agents. This implied a more favorable the expected FLER.
Temperature break down	The higher the temperature limit, the longer the agent can go without breaking down and thus avoid paying the breakdown penalty. This implied a more favorable FLER.
Cooling per unit time	The more cooling per unit time, the sooner the agent can go back to fighting. This implied a more favorable than expected FLER.
Temperature increase per unit	The greater the increase in temperature per unit time, the longer the agent needs to wait for cool down. This implied a less favorable than expected FLER.
Temperature increases with distance or time	The impact could not be ascertained.
Speed	The faster the agent can move, the faster it could save itself when under attack. This implied a more favorable than expected FLER.

Attack ammunition ratio	If the attack ammunition ratio were too high, the agents would waste time coming back to base camp for reloading and also pay the turnaround penalty. But if the value were too low, the agents would wait for too long and may run out of ammunition.
Break down penalty	The higher the breakdown penalty, the less favorable the expected FLER would be.

3.2.7.2 Impact of Defense Parameters on FLER

Figure 10 represents the impact of defense parameters on FLER. The details for these parameters and the reason for the impact are provided in Table 1 and Table 2 respectively. The figure categorizes the impact into four categories. An example category is increase, which implies that an increase in value of this parameter would directly relate to the increase in FLER. ‘Nominal’ (in parentheses) implies this is true for some range of parameter. ‘High and low’ (in parentheses) implies it is valid for extreme values of this parameter. The impact is shown with respect to the blue force. For the red force, the parameters in categories ‘increase’ and ‘decrease’ would be interchanged. The categories are:

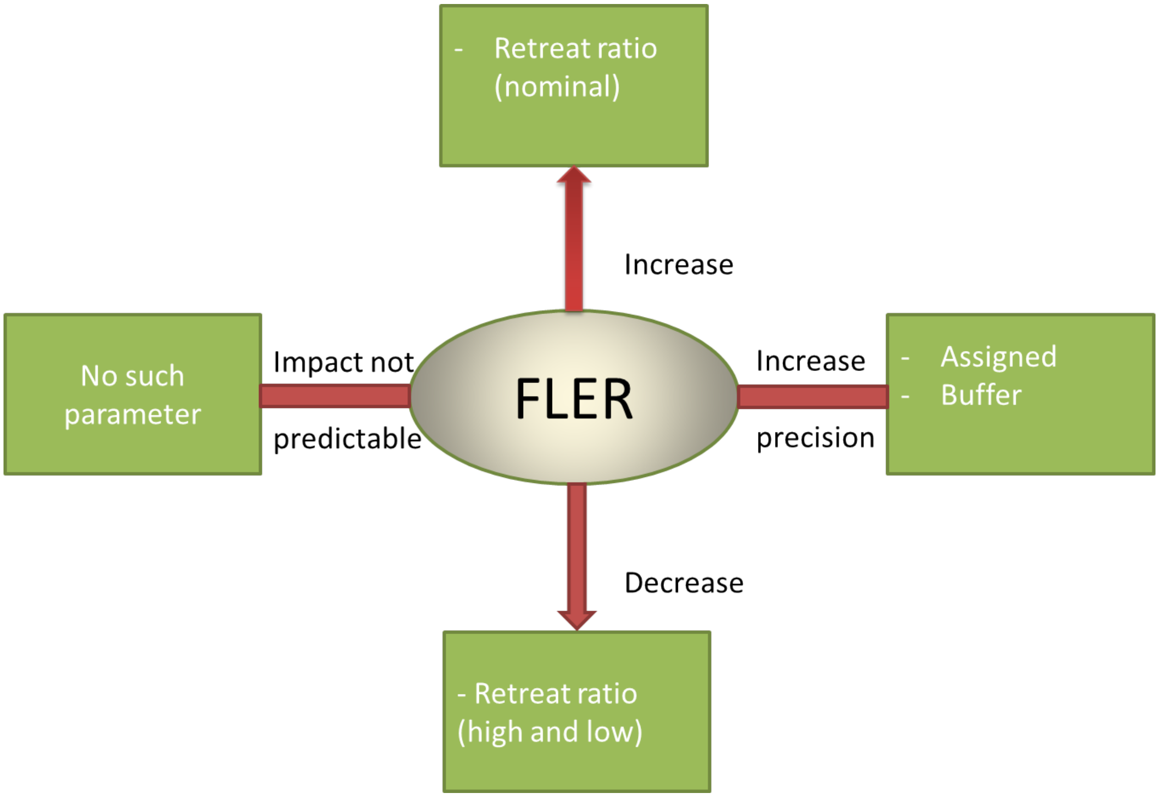

Figure 10: Impact of Defense Parameters on FLER

3.2.7.3 Impact of Attack Parameters on FLER

Figure 11 presents the impact of attack parameters on FLER. The details for these parameters and the reason for the impact are provided in Table 1.

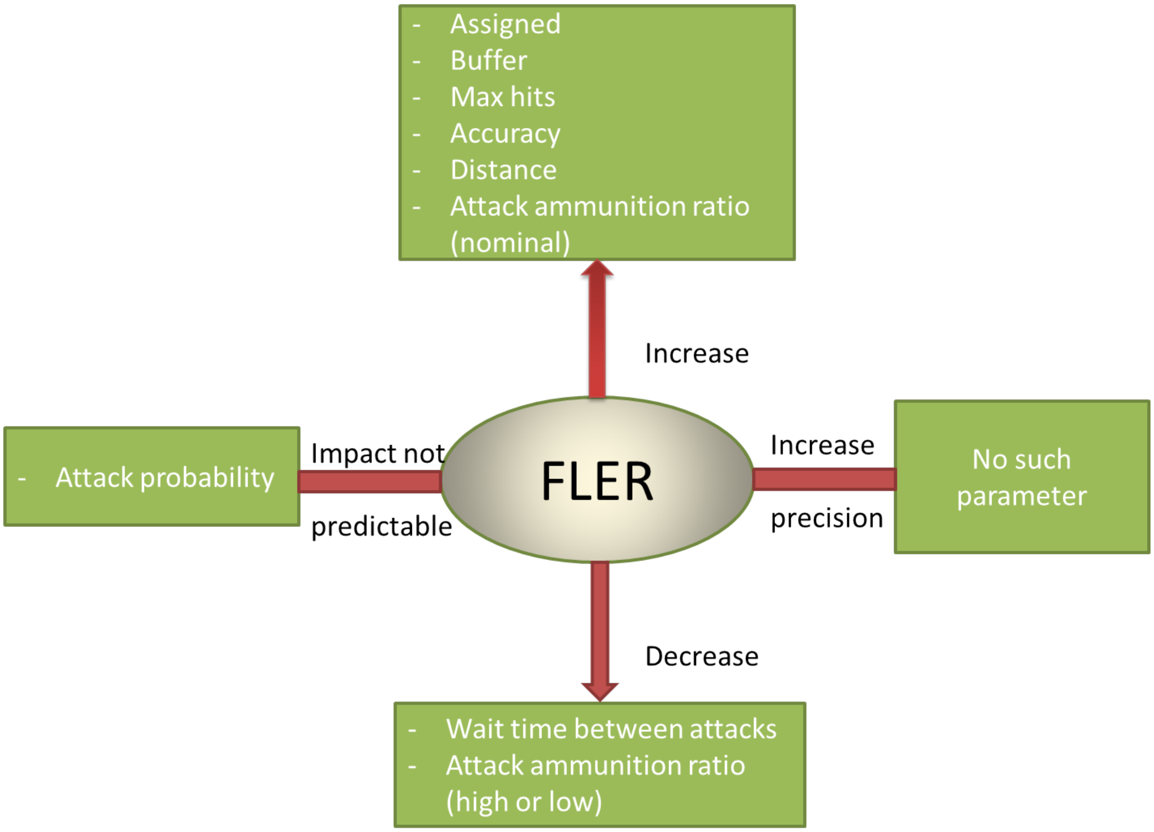

Figure 11: Impact of Attack Parameters on FLER

3.2.7.4 Impact of Sensor Parameters on FLER

Figure 12 represents the impact of defense parameters on FLER. The details for these parameters and the reason for the impact are provided in Table 1 and 2 respectively.

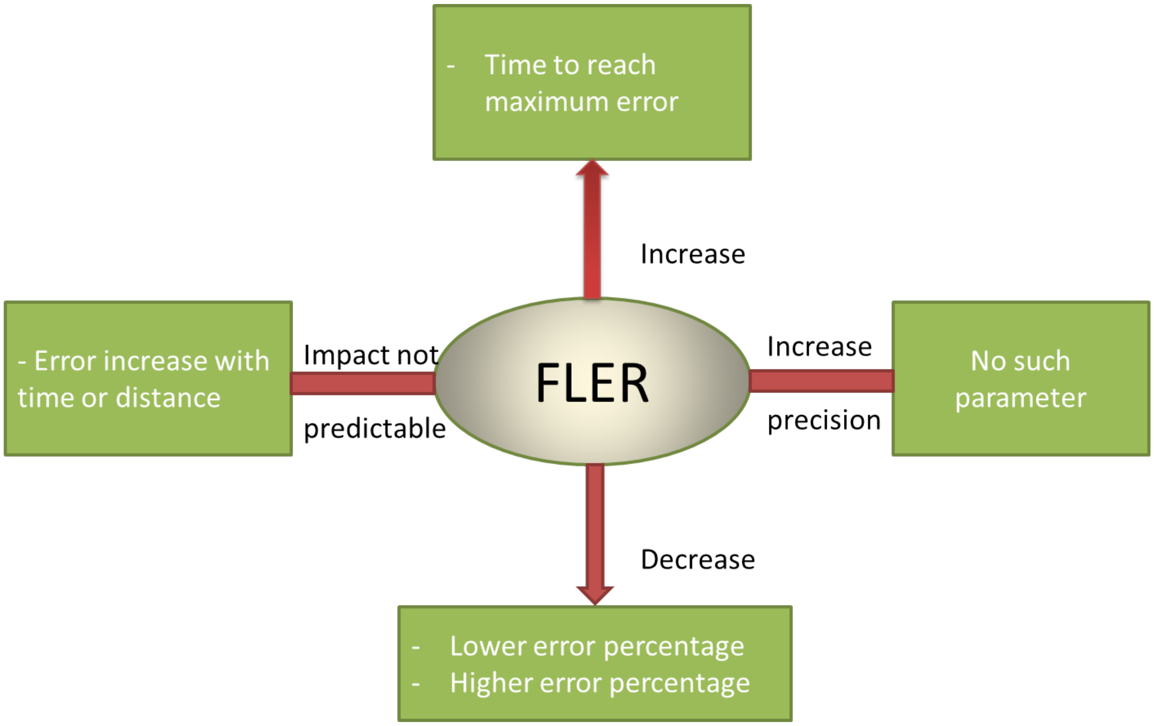

Figure 12: Impact of Sensor Parameters on FLER

3.2.7.5 Impact of Temperature Parameters on FLER

Figure 13 represents the impact of defense parameters on FLER. The details for these parameters and the reason for the impact are provided in Table 1 and 2 respectively.

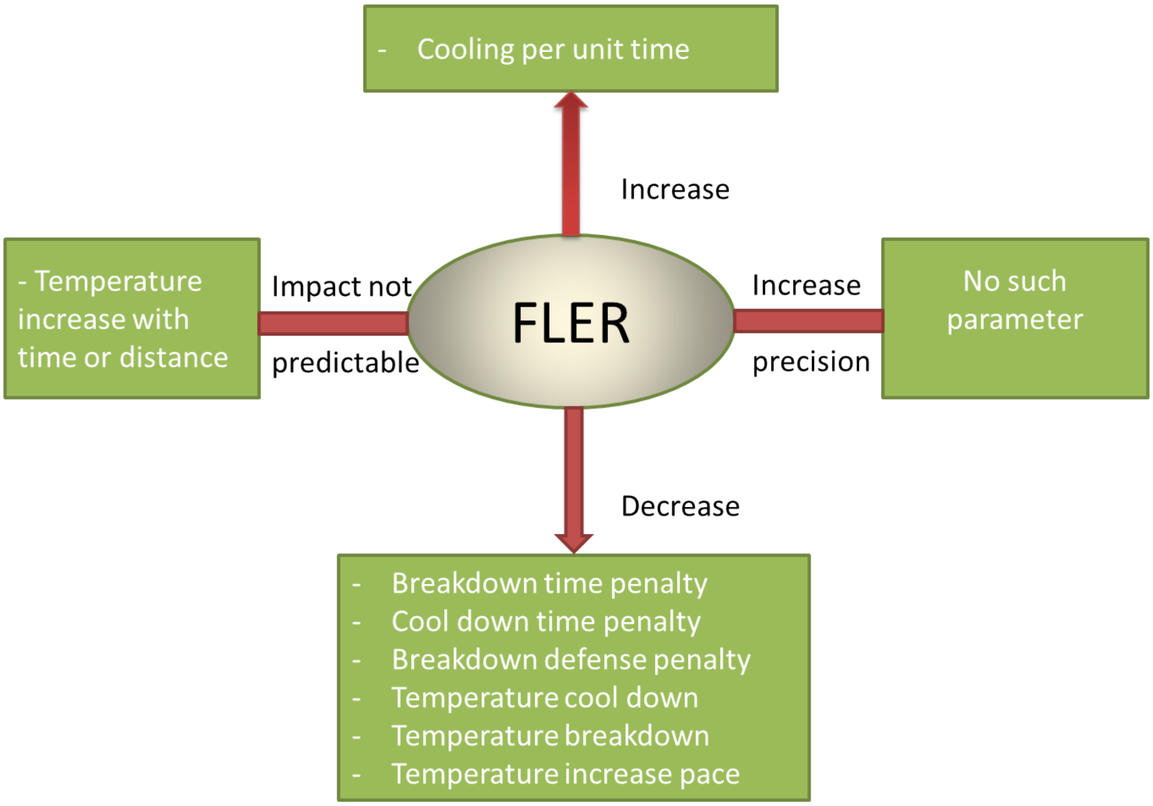

Figure 13: Impact of Temperature Parameters on FLER

3.3 Method

We can now generalize our implementation to a wider class of problems (i.e., 'Requirement Selection') in projects undertaken by industrial enterprises. The solution is currently limited to projects using sensors to track information. The method is a two phase process; the first phase is design and the second phase is implementation.

3.3.1 Phase 1

The first step in Phase 1 is to gather the information obtained by the use of sensors. The next step is to analyze the outcome of this information and to determine the parameters that would be affected by it.

To clarify each of these steps further, Phase 1 is correlated to the implementation described in Section 3.2.5. For the S&R project it was determined that the two sensors: shots fired sensor and planetary gear temperature sensor, track the information regarding ammunition and vehicle health, respectively. This information affects two parameters: the counter that keeps track of the shots fired and the temperature of the LAV's planetary gears. The first parameter is the counter to keep track of the shots fired to determine the time when the LAV returns to the base camp for reloading ammunition; the second parameter is the temperature of the LAV's planetary gears to keep track of the time when the LAV should take a break from its activities to let the planetary gears cool down. In the context of the S&R project, FLER was determined to be the profitability metric.

3.3.2 Phase 2

The second phase correlates to the implementation of a simulation environment, which imitates the real environment of the project. An important factor in this environment is simulating all the important outcomes for the set of parameters determined in Phase 1. Implementing those components associated with the parameters determined in Phase 1 affects the profitability metric determined in Phase 1. With the use of this simulation environment, profitability is calculated for the various sensors that are available. Based on the calculated profitability, a comparison can be made among the available sensor alternatives.

During this work, a simulation tool was developed to facilitate a comparison of the sensor parameters installed on LAVs. Based on what was learned during the development of this tool, a methodology was developed which can be applied to evaluate a broader category of projects. The following chapter discusses the results obtained by the use of this tool.

Chapter 4

Results and Discussion

The results discussed in this section demonstrate the effectiveness of the comparison methodology to aid the decision making process of the USMC. The results are quantifiable and fast to reproduce.

4.1 Interpreting the graphs

As discussed in Section 2.2.5.2, the measurements in the simulations were performed by the use of an attrition-based metric [32] called FLER, whose higher and lower values represent the advantage to the blue force and red force, respectively.

$$FLER = F = \frac{Fraction\ of\ Red\ Losses}{Fraction\ of\ Blue\ Losses}$$

Figure 14 shows an example of the effect of changing one sensor variable while keeping all others constant. In this case, the variable is the accuracy of the shots fired sensor. The slope of the output shows that the accuracy of the shots fired sensor does affect the outcome. When this graph is compared against another sensor's results (also acquired by changing one variable while keeping all other simulation parameters fixed), it can provide a better understanding about which sensor is more important for the mission, thus helping requirement prioritization. As this work is trying to find prioritization by performing sensitivity analysis of a mission towards a sensor, the pattern of the results - as observed in the graphs - is more interesting than their absolute value.

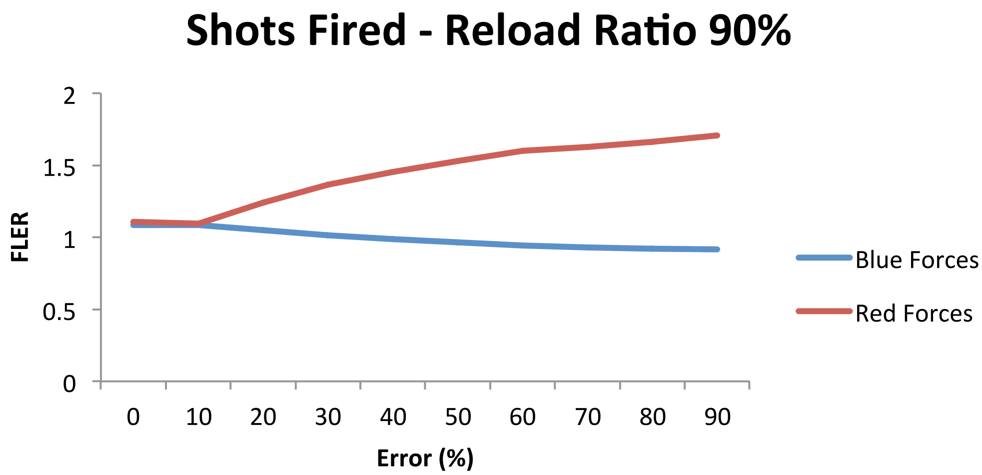

Figure 14: Sample Figure

4.2 Sources of Error

The simulation environment made some assumptions about the scenario and used some randomized values for execution. This introduced some errors into the. This section discusses those errors and the steps that can be taken to mitigate them as an extension to this work.

4.2.1 Initial Value

FLER corresponding to the accuracy parameter being set to zero was observed to be bound in an interval of 0.6 to 1.2 as opposed to the expected mean value of 1. In other words, the simulations are not fair to the forces and favor one force more than the other depending on the input values.

Another important point in the graphs is that for red and blue force graphs, they seemingly have the same value at certain points. This occurs, for example, at 10% error in Figure 15 or at 0% error in Figure 16. In reality, these are distinct points that are not visible at the scale used to represent these graphs in this document. The absolute values of FLER for both blue and red forces are nearly but not exactly equal.

The factors determined to have directly caused these variations from FLER of 1 and distinct data points at 0% error are discussed below.

4.2.1.1 Placement of Agents

The first reason for these variations was the starting point of the blue and red forces. The blue forces were placed around the blue base camp, while the red forces were scattered all around the grid. The exact placement was determined by the use of a random number generator, thus varying the output of each run and making these simulations a stochastic process. This creates a lot of noise in the system, and consequently the results were heavily dependent on the initial placement of the agents. A recommended extension to this system is to use a seeded random number generator that produces the same starting placement of agents during a particular run.

4.2.1.2 Placement of Base Camp

The second reason for the variations was that the placement of the base camps for both the blue and red forces coupled with scattered placement of red agents benefitted the red force specifically. As the base camps were located in opposite corners, blue agents were unable to avoid red agents while moving back to the base camp because the implementation of agents' movement is primitive (i.e., they move in a straight line while returning to base camp). Because of the nature of their movement, the blue agents to come into close proximity with multiple red agents before reaching the base camp. Since this work does not concern itself with the strategy being employed by the agents and limits imitating the use of information received from sensors, it was decided that the behavior should not be modified as it was beyond the scope of the current work. Nevertheless, it can be safely predicted that such implementation is not expected to have affected the sensitivity analysis as all the data points in the analysis would be affected in a similar manner.

4.2.2 Architecture Used

The architecture used during the implementation of the system, although useful, was not perfect for this kind of scenario. In general, military decisions are strategically well-defined and not reactive in nature. For example, an agent may be required to ignore its warning of low ammunition and to hold its position for the time being, if it knows that backup is coming soon. Modeling such advanced behavior was beyond the scope of this work.

4.2.3 Implementation

The implementation of simulations was both a source of error and variation in the initial value of FLER. During the execution, the event queues were not randomized for events happening at the same instance. With the initialization of the simulations, events from one force were inserted before the other. The chain of events continued in the same order giving undue advantage to one force as it pushed opposite forces into resignation quickly. This can be addressed by randomizing the order of event execution happening simultaneously.

4.3 Number of Simulation Runs

To reduce the noise involving stochastic components, Ritter et al. [**49**] suggest that when runs are inexpensive, the simulations should be run until stable predictions are obtained. Evaluating the change in mean and standard deviation between different runs of the simulation and claiming that this change is now negligible define stability of simulation. To remove noise from stochastic components like placement of agents and the probability of attack, the simulations were found to give stable results after less than 100 simulation runs. Ritter et al. [**49**] suggested that in the case of inexpensive runs, simulations can be run more times to have highly stable predictions. The data points shown in these results were obtained after 300-500 runs depending on the kind of sensor.

Each line in Figure 14 was obtained in less than 5 minutes, implying that each data point was obtained in less than sixty seconds. This provides an insight into the effectiveness of this implementation and that it can be integrated into SUMMIT models for component analysis.

4.4 Shots Fired Sensor

The ratings taken during the previous evaluation of the S&R project [**25**] indicated poor reliability and accuracy for the shots fired sensor [**25**], which formed the basis for choosing the sensor's accuracy as the testing parameter in this research. The sensitivity analysis was carried out to justify whether further research and development (R&D) for improving the accuracy of this sensor would improve the mission's effectiveness. Each data point for this sensor was obtained after 300 simulation runs.

Sensitivity analysis for accuracy parameters was performed with two sets of input parameters wherein the notable parameter was the reload ratio that determines when the agent decides to move back to the base camp for reloading.

In the graph shown in Figure 15, the reload ratio was set at 90%. A slight increase in FLER was observed when the error percentage was below 10%, and further testing showed that the reload ratio impacted the shape of the graph. When the reload ratio was 90%, an accuracy of more than 90% implied that the agent never ran out of ammunition - that provided it with a competitive advantage in a one-on-one battle. However, beyond that error percentage, it would run out of ammunition unexpectedly and would incur losses. The shape of the curve is nearly straight, indicating a direct correlation between the accuracy of the sensor and mission effectiveness. The difference between the slopes of the red and blue forces can be accounted for by other test parameters and the strategy used for the randomization implied more losses to the red and less to the blue forces.

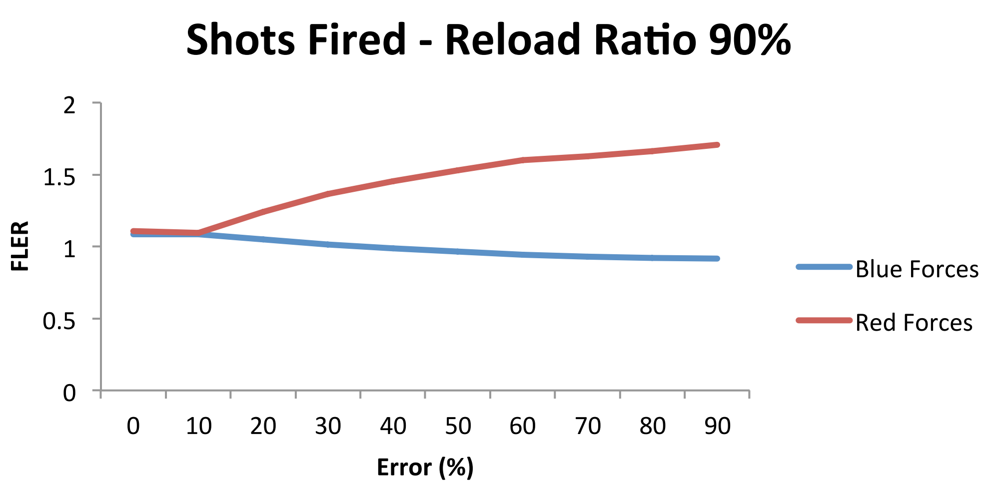

Figure 15: Shots Fired - Reload Ratio 90%

As shown in Figure 16, the reload ratio was set at 100%, which physically implies that the agents would come back to base camp only when completely out of ammunition. Having a pure sensitivity analysis is important to observe an impact on FLER. Another difference is that the graph has a milder slope compared to the one generated with a 90% reload ratio. However, as discussed in Section 2.2.5.2, this actually is misleading and can be accounted for by a change in other parameters. For example, in this test case, attack ammunition assigned to each team was higher without changing the defense assigned. This meant higher losses for both the forces - an increase in both the numerator and denominator. The curve is not strictly monotonic, probably due to randomization.

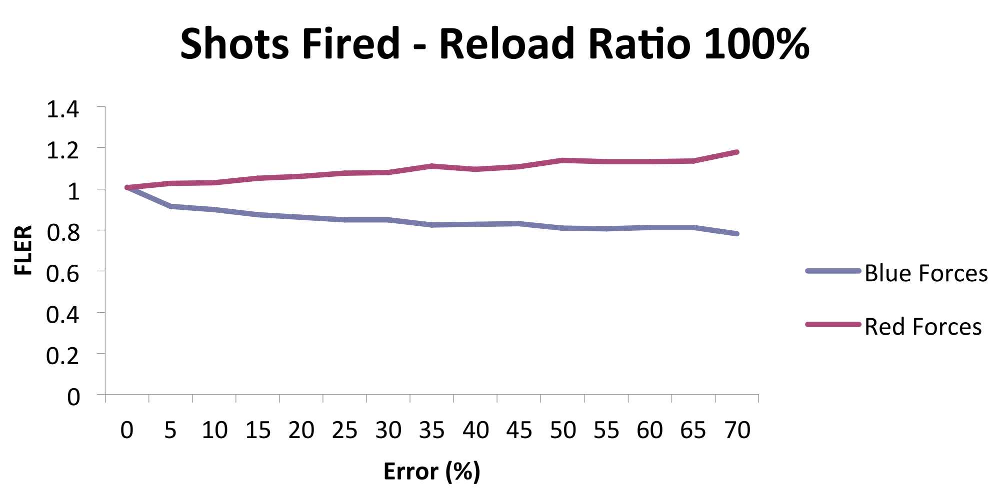

Figure 16: Shots Fired – Reload Ratio 100%

4.5 Planetary Gear Temperature Sensor

As suggested in the previous evaluations of the S&R project [**25**], the planetary gear sensors are also known to have low accuracy and reliability. Accuracy was chosen as the testing parameter and the results are shown in Figure 17. Each data set for this sensor was obtained after 500 simulation runs.

The sensor was expected to demonstrate a pattern wherein a decrease in error should be accompanied by an increase in utility. However, the pattern observed in the results did not indicate this and was far from the expected behavior. Further analysis of the breakdown pattern indicated that the accuracy decrease was inversely correlated with the breakdown frequency. This correlation did not translate into losses because the opponent's agents were not able to attack broken down agent as the system used reactive architecture. Reactive architecture implied that as soon as the agent broke down, opponents would try to reach it and attack. But such an attack would never take place, because on its way to the broken down agent, the opponent would inevitably react to some new information encountered instead of continuing on to find and attack the halted agent. BDI architecture would have been more suitable under such circumstances.

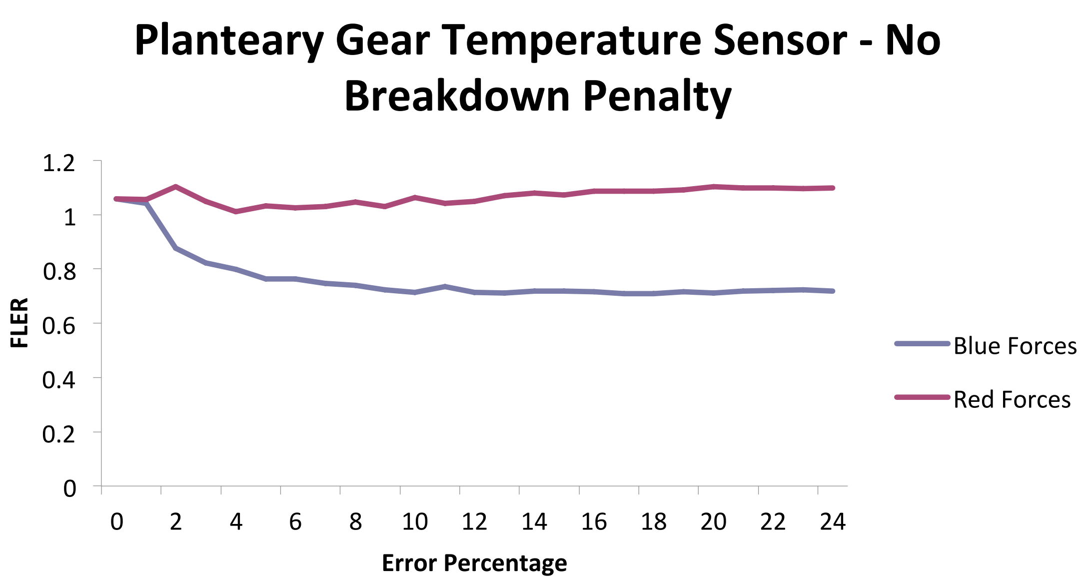

Figure 17: Planetary Gear Temperature Sensor – No Breakdown Penalty

To get a better indication of how FLER changed in the real world environment, another graph, shown in Figure 18, was generated after adding a new parameter to the simulations: the breakdown loss penalty. This parameter imitated the real life behavior of an agent suffering losses during the break down time in a much better way, by adding a constant loss to the defense of the force.

The nature of the curve in Figure 18 brings to light two important conclusions. The first conclusion is that there is a sharp drop in the utility of the sensor with a decrease in the sensor's accuracy at about a 2% error rate. The second conclusion is that decreases beyond a 6% error rate do not affect the mission effectiveness.

A further analysis of the results based on the parameters of the simulations indicated that this decline was the result of one parameter: the ratio between the current temperature and breakdown temperature that decides the time when the agent starts its cooling procedure. If this ratio is less than the error percentage in the temperature sensor, it can avoid losses due to the error in this sensor's accuracy. However, if the agents decide to operate the vehicle too close to the breakdown temperature without taking into account the error in the sensor, the sensor is effectively useless.

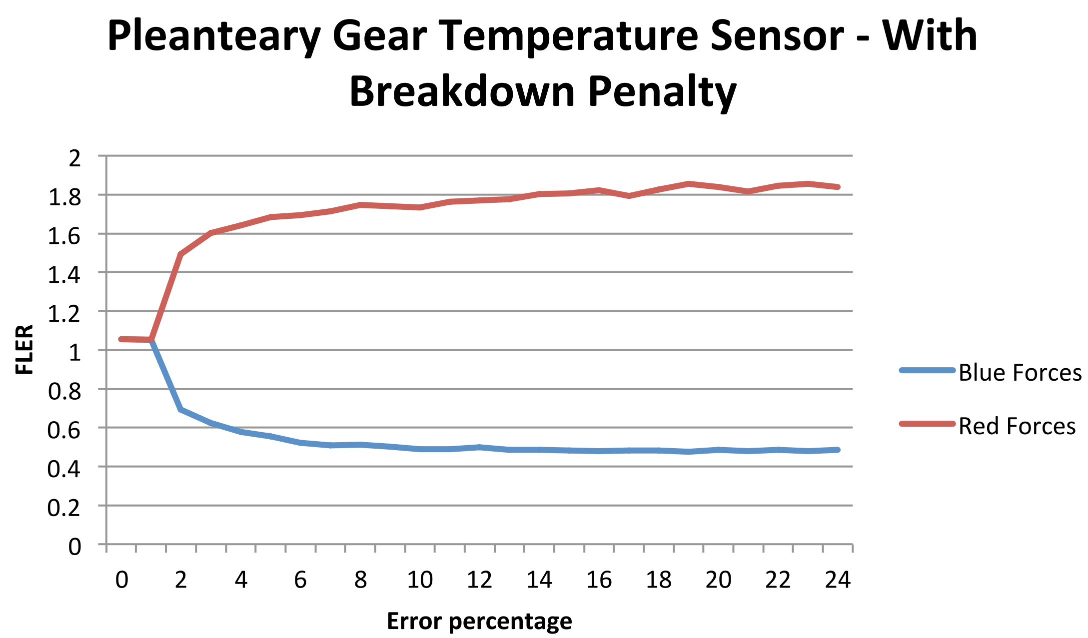

Figure 18: Planetary Gear Temperature Sensor Accuracy with Breakdown Penalty

4.6 Discussion

These results provide insight into the effects of the different variables involved in the decision making process. A seemingly small change in the accuracy of a sensor can have a huge impact on the outcome and thus on the choice of options.

Having a large variable set (Table 1: Variable Descriptions) enables user to customize the simulation environment and gain knowledge about what kind of information can enhance mission effectiveness. Seeing the impact of a variable on outcome can lead to design of new sensors.

One of the fundamental research questions focused on finding the resolution of the information required from a sensor. The tool developed in this work makes it easy to study the effects of changing the resolution of the information on the FLER. For example, to determine the required resolution for the planetary gear temperature sensor, it is possible to re-run the simulation with various resolutions of temperature, like at 5F, 1F or 0.1F and analyze the output.

Another question was related to the refreshing rate of the information. It was assumed that the information is updated infinitely fast, that is, all the required information was available as and when required. This can be easily modified in the simulation environment; the information can be available at discrete intervals. By varying the time difference between these discrete intervals, the sensitivity of the refreshing rate of a particular sensor on mission effectiveness can be determined.

The research also looked into finding the most appropriate way to use the information available from the sensor (i.e., how to respond to the information provided by the sensor). During this work, only one-way of responding was tested: immediate reaction to the information available. It is not the best way to use the information in a real world scenario and it is suggested that this work be extended to use BDI architecture [**5**], [**29**], [**44**]. Changing the behavior of agents towards available information can also help enhance the results.

The results discussed prioritized the sensors used in the S&R project, where sensors represent alternatives in a decision-making process like the requirement selection or the AHP.

Chapter 5

Conclusion

With the multi-billion dollar budget of the USMC and other enterprises, the importance of a fast, reliable, reproducible decision making process was recognized. Requirements engineering and decision-making process were explored and broken down into smaller components. This work makes a modest contribution towards requirements engineering by introducing a method of requirements prioritization, a primary requirement in a justifiable decision making process. The problem of ordering the options was narrowed down by focusing on providing a logical order for choices of systems involving use of sensors in the mechanical systems. The S&R project, undertaken by the USMC, was studied as an example project.

The work explored different methods available for the decision-making processes like requirement selection and the AHP, and concluded that the agent-based simulations are highly efficient in evaluating projects similar to the S&R project with minimal user interaction. Simulation methodologies are time efficient, reliable, have reproducible results, and are easy to extend with changing requirements. They can be used to prioritize order of the sensor development, and justify expenditure in sensors research and development.

Additionally, the research realized the importance of quantifying the outcomes of the requirements engineering. Good qualitative approaches can provide a fair amount of insight into the outcomes, but generally fail to provide an assertive case regarding the extent to which one option is better than another. Also, non-reproducibility of qualitative processes was seen as a hindrance to decision making.

74

Based on the results obtained, the effectiveness of the tool developed in this research was ascertained and it can be summarized that agent-based simulations are an important tool for requirements prioritization.

Bibliography

[1] M. Lubars, C. Potts, and C. Richter, "A review of the state of the practice in requirements modeling," *Proceedings of IEEE International Symposium (Requirements Engineering)*, pp. 2-14, Jan 1993.

[2] Aybüke Aurum and Claes Wohlin, *Engineering and Managing Software Requirements*. USA: Springer, 2005.

[3] K. El Emam and N. H. Madhavji, "Measuring the success of requirements engineering processes," *Proceedings of the Second IEEE International Symposium*, 1995, pp. 204-211.

[4] Jacqueline M. Curiel, Michael D. Anhalt, John J. Tran, and Ke-Thia Yao, "Developing situation awareness metrics in a synthetic battlespace," *Interservice/Industry Training, Simulation, and Education Conference*, 2005.

[5] Rick Evertsz, Frank E. Ritter, Paolo Busetta, and Matteo Pedrotti, "Realistic behaviour variation in a BDI-based cognitive architecture," *Proceedings of SimTecT*, Melbourne, Australia, 2008.

[6] Solidica. Pantheon™ advanced system health, diagnostic and telematic system. 2007

[7] Elizabeth Hull, Ken Jackson, and Jeremy Dick, *Requirements Engineering*, 2nd ed. London: Springer, 2005.

[8] Asa G. Dahlstedt and Anne Persson, "Requirements Interdependencies - Moulding the State of Research into a Research Agenda," *Ninth International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ 2003)*, 2003, pp. 71-80.

[9] Thomas L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Services Sciences*, vol. 1, no. 1, pp. 83-98, 2008.

[10] Milan Zeleny, *Multiple criteria decision making*.: McGraw-Hill, 1982.

[11] Bernanrd Roy, "The outranking approach and the foundation of electre methods," *Theory and Decision*, vol. 31, pp. 49-73, 1991.

[12] Thomas L. Saaty, "How to make a decision: The analytic hierarchy process," *European Journal of Operational Research*, vol. 48, no. 1, pp. 9-26, Sep 1990.

[13] John Doyle and Richmond H. Thomason, "Background to qualitative decision theory," *AI Magazine*, vol. 20, 1999.

[14] Ernest H. Forman and Saul I. Gass, "The Analytic Hierarchy Process- An Exposition," *Operations Research*, vol. 49, no. 4, pp. 469-486, Jul 2001.

[15] Rainer Haas and Oliver Meixner. An Illustrated Guide to the Analytic Hierarchy Process. [Online]. http://alaskafisheries.noaa.gov/sustainablefisheries/sslmc/july-06/ahptutorial.pdf , 2009

[16] Alex Kalloniatis and Paul Wong, "Application of Busniess Process Modelling to Military Organisations," *SimTechT Conference*, Brisbane, Australia, 2007.

[17] Karl T. Ulrich and Steven D. Eppinger, *Product design and development*.: McGraw Hill, 2000.

[18] Jonathan F. Bard, "A Two-Phase methodology for technology selection and system design," *IEEE Transactions on Engineering Management*, vol. 36, no. 1, pp. 28 - 36, Feb 1989.

[19] Stephen Slade, "Qualitative decision theory," *Proceedings of DARPA Workshop on Case-Based Reasoning*, 1991, pp. 339--350.

[20] Craig Boutilier, "Toward a logic for qualitative decision theory," *Proceedings of the KR'94*,

1992, pp. 75-86.

[21] Blai Bonet and Hector Geffner, "Arguing for decisions: A qualitiative model of decision making," *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, Portland, Oregon, USA, 1996, pp. 98-105.

[22] Douglas W. Hubbard, *How to Measure Anything: Finding the Value of "Intangibles" in Business*.: John Wiley and Sons, 2010.

[23] Dapzury Valenzuela and Pallavi Shrivastava. Arizona State University. [Online]. http://www.public.asu.edu/~kroel/www500/Interview%20Fri.pdf

[24] Mick P. Couper, "Web survey: A review of issues and approaches," *Public Opin Q*, vol. 64, no. 4, pp. 464-494, 2000.

[25] Steven R. Haynes and Amie L. Skattebo, "Report on the Sense and Respond Phase II Evaluation," 2007.

[26] Chris Barrett, Stephen Eubank, and Madhav Marathe, "Modeling and simulations of large biological, information and socio-technical systems: An interaction based approach," *Interactive Computation: The New Paradighm*. Berlin: Springer, 2005.

[27] David S. Alberts, John J. Garstka, and Frederick P. Stein, *Network centric warfare*.: CCRP , 2000.

[28] Mica R. Endsley and Daniel J. Garland, *Situation awareness analysis and measurement*.: Lawrence Erlbaum Associates, 2000.

[29] Rick Evertsz, Frank E. Ritter, Paolo Busetta, and Jennifer L. Bittner, "CoJACK—Achieving principled behaviour variation in a moderated cognitive architecture," *Proceedings of the 17th Conference on Behavior Representation in Modeling and Simulation*, 2008, pp. 80-89.

[30] Herbert H. Bell and Don R. Lyon, "Using observer ratings to assess situation awareness," in *Situation awareness analysis and measurement*.: Lawrence erlbaum associates , 2000.

[31] Richard E. Darilek, *Measures of effectiveness for the information-age army*.: RAND Corporation (Issue 1155), 2001.

[32] W L Perry and J Moffat, "Measuring the effects of knowledge in military campaigns," *The Journal of the Operational Research Society*, vol. 48, no. 10, pp. 965-972, 1997.

[33] Steven R. Haynes, "It's What It's In: Evaluating the Usability of Large-scale Integrated Systems," *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems*, New York, 2009 , pp. 3051-3060.

[34] Gerhard Weiss, *Multiagent systems*. Cambridge: The MIT press, 1999.

[35] Xiaocong Fan and John Yen, "Modeling and Simulating Human Teamwork Behaviors Using Intelligent Agents," *Physics of Life Reviews*, vol. 1, no. 173-201, 2004.

[36] Micheal Wooldridge, "Intelligent agents," in *Multiagent systems*. Cambridge, Massachussetts: The MIT Press, 1999.

[37] Xiaocong Fan, Shuang Sun, Michale McNeese, and John Yen, "Extending Recognition-Primed Decision Model for Human-Agent Collaboration," *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS}*, Utrecht, The Netherlands, 2005, pp. 945--952.

[38] J.P.C Kleijnen, S. M. Sanchez, and T. W. Lucas, "A User's Guide to the Brave New World of Designing Simulation Experiments," *INFORMS Journal on Computing*, vol. 17, pp. 263-289, 2005.

[39] John Collins, Corey Bilot, and Maria Gini, "Mixed-initiative decision support in agent-based

automated contracting," *Proc. of the Fourth Int'l Conf. on Autonomous Agents*, 2000, pp. 247--254.

[40] Robert Kewley, Larry Larimer, and William Carlton, "An Agent-Based Modeling Approach to Quantifying the Value of Battlefield Information," Center for army analysis , Fort Belvoir VA, 2002.

[41] Anthony H. Dekker, "Agility in networked military systems: A simulation experiment," Defence Science and Technology Organisation, Department of Defence, Canberra, Australia,.

[42] Anthony H. Dekker, "Measuring the agility of networked military forces," *The Journal of Battlefield* , vol. 9, no. 1, 2006.

[43] Thomas Erlenbruch, "Agent- Based simulation of German peacekeeping operations for units up to platoon level," Naval postgraduate school, Monterey, California, Master's Thesis in Operations Research 2002.

[44] Ronald F. A. Woodaman, "Agent based simulation of military operations other than war small unit combat," Naval Postgraduate School, Monterey, California, Masters Thesis 2000.

[45] E. Frank Ritter, Lewis Bill, and A. Mark Cohen, "dTank updated: Steps towards exploring moderator-influenced behavior in a light-weight synthetic environment," *Proceedings of the 16th Conference on Behavior Representation in Modeling and Simulation*, Norfolk, VA, 2007, pp. 51-60.

[46] P. Busetta, R. Ronnquist, A. Hodgson , and A. Lucas. JACK intelligent agents - Components for intelligent agents in JAVA. [Online]. www.agent-software.com/white-paper.pdf

[47] Paul Davidsson, "Multi agent based simulation: Beyond social simulation," *Workshop on Multi Agent Based Simulation (MABS)*, 2000, pp. 97--107.

[48] Rick Evertsz, Frank E. Ritter, Simon Russell, and David Shepherdson, "Modeling rules of engagement in computer generated forces," *Proceedings of the 16th Conference on Behavior Representation in Modeling and Simulation*, Norfolk, VA, 2007.

[49] Frank E Ritter, Micheal J. Schoelles, Karen S Quigley, and Laura Cousino Klein, "Determining the number of simulation runs: Treating simulations as theories by not sampling their behavior," *Human-in-the-loop Simulations:Methods and Practice*. London: Springer-Verlag, 2011.