COGNITIVE MODELLING OF OPPOSING FORCES AT THE SIMULATION ANALYSIS LAB

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS OF
THE MIDDLE EAST TECHNICAL UNIVERSITY
BY

YELİZ TOPCU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF COGNITIVE SCIENCE

DECEMBER 2021

COGNITIVE MODELLING OF OPPOSING FORCES AT THE SIMULATION ANALYSIS LAB

submitted by **YELİZ TOPCU** in partial fulfillment of the requirements for the degree of **Master of Science in Cognitive Science Department, Middle East Technical University** by,

Prof. Dr. Deniz ZEYREK BOZŞAHİN
Dean, **Graduate School of Informatics** ________________

Dr. Ceyhan TEMÜRCÜ
Head of Department, **Cognitive Science** ________________

Prof. Dr. Deniz ZEYREK BOZŞAHİN
Supervisor, **Cognitive Science, METU** ________________

Assist. Prof. Dr. Murat ULUBAY
Co-supervisor, **Business Administration, AYBU** ________________

Examining Committee Members:

Assist. Prof. Dr. Murat Perit ÇAKIR
Cognitive Science, METU ________________

Prof. Dr. Deniz ZEYREK BOZŞAHİN
Cognitive Science, METU ________________

Assist. Prof. Dr. Murat ULUBAY
Business Administration, AYBU ________________

Assist. Prof. Dr. Umut ÖZGE
Cognitive Science, METU ________________

Assist. Prof. Dr. Gülten ÜNAL
Psychology, AYBU ________________

Date: 09.12.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: YELİZ TOPCU

Signature :

ABSTRACT

COGNITIVE MODELLING OF OPPOSING FORCES AT THE SIMULATION ANALYSIS LAB

TOPCU, YELİZ

M.S., Department of Cognitive Science

Supervisor: Prof. Dr. Deniz ZEYREK BOZŞAHİN

Co-Supervisor: Assist. Prof. Dr. Murat ULUBAY

December 2021, 53 pages

In modern military systems, engagement simulations are used to assist in conceptual design and verification of weapon systems. The quality of these simulations is directly proportional to the level of authenticity of the behavior of the components that are modeled. For this reason, it is important to represent the behavior of human controlled components by accounting for the cognitive aspects including human limitations and flaws. In various studies, the behaviors of these components are modeled using cognitive architectures. These representations of behavior can contain several cognitive processes, but in many studies, a single cognitive process has been modeled and the interacting cognitive processes have not been adequately studied. In this thesis, computational cognitive model of situational awareness and surprisal related cognitive processes were implemented using SOAR cognitive architecture. The behaviors of agents guided by these cognitive models have been observed by performing various experiments in the STAGE simulation environment. Interaction of the two cognitive processes modelled has been discussed.

Keywords: Human Behavior Representation, Human Behavior Modelling, Cognitive Modelling, Constructive Simulations, Computational Cognitive Model, Computational Models of Mind

ÖZ

BENZETİM ANALİZ LABORATUVARINDA KARŞI KUVVETLERİN BİLİŞSEL MODELLENMESİ

TOPCU, YELİZ
Yüksek Lisans, Bilişsel Bilimler Bölümü
Tez Yöneticisi: Prof. Dr. Deniz ZEYREK BOZŞAHİN
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Murat ULUBAY

Aralık 2021, 53 sayfa

Günümüzde geliştirilen silah sistemlerinin kavram tasarımı ve doğrulama aşamalarında angajman simülasyonları yapılmaktadır. Bu simülasyonların başarısı, bileşenlerin davranışlarının ne kadar gerçeğe yakın modellendiğiyle doğru orantılıdır. Bu sebeple, gerçekte insan tarafından kontrol edilen bileşenlerin davranışlarının insanların sınırları ve kusurlarını da içeren bilişsel boyutları da göz önünde bulundurularak temsil edilmesi önem taşımaktadır. Çeşitli çalışmalarda bu bileşenlerin davranışları, bilişsel mimariler kullanılarak modellenmiştir. Davranış temsilinde birden fazla bilişsel süreç yer alabilmekle birlikte, birçok çalışmada tek bir bilişsel süreç modellenmiş ve birbiriyle etkileşen bilişsel süreçler yeterince incelenmemiştir. Bu tezde, SOAR bilişsel mimarisi kullanılarak durumsal farkındalık ve sürpriz süreçlerinin bilişsel modellemesi yapılmıştır. Bu bilişsel modellerin yönlendirdiği ajanların davranışları, STAGE benzetim ortamında çeşitli deneyler yapılarak gözlemlenmiştir. Bilişsel mimari kullanılarak modellenen iki farklı bilişsel sürecin etkileşimi incelenmiştir.

Anahtar Kelimeler: İnsan Davranışı Temsili, İnsan Davranışı Modelleme, Bilişsel Modelleme, Yapıcı Benzetimler, Hesaplamalı Bilişsel Model, Hesaplamalı Zihin Modelleri

To my family

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisors, Murat Ulubay and Deniz Zeyrek Bozşahin, as well as the members of my thesis committee, Murat Perit Çakır, Umut Özge, and Gülten Ünal, for their precious guidance during my study. I also would like to express my gratitude to my manager, Gökhan Bircan, and my employer, Roketsan, for allowing me to attend my master's courses and carry out my research, as well as for providing a simulation environment. I owe a special thanks to my husband for his unwavering support and faith in my abilities; without him, I would not have been able to complete this thesis. For being there for me, I'd also want to thank my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
CGF	Computer Generated Forces
DIS	Distributed Interactive Simulation
IFF	Identification of Friend or Foe
MANPADS	Man-portable Air-Defense System
MWR	Missile warning Receiver
PSCM	Problem-Space Computational Model
RWR	Radar Warning Receiver
SA	Situation Awareness
SAM	Surface-to-Air Missile
SME	Subject Matter Experts
SML	Soar Mark-up Language
WME	Working Memory Element

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

In military domain, modelling and simulation is widely used for analysis, design, and evaluation of weapon systems and equipment. The weapon systems contain components with a high number of design characteristics and interfaces which must be developed, tested, and verified before they can be deployed. It is required to examine "the human engineering concepts, the design of systems and their interoperability with other services or multinational forces, and option prioritization and risk assessment decisions” and to test “survivability, vulnerability, reliability, and maintainability” [4]. The test cases for observing the behavior of developed systems should be the real-world military scenarios, which are composed of friendly, enemy and neutral forces and their engagements between each other compatible with the tactics and doctrines. However, the fighting readiness and capabilities of combatants and other factors cannot be deployed or exercised in peacetime. Therefore, a specific kind of simulations called constructive simulations, in which simulated people operates simulated equipment, are used for analysis applications in general.

Accurate representation of human behavior at the individual, unit, and command levels has become vital and crucial especially in constructive simulations since the human is the focal point for information usage. In modeling and simulating human behavior, it is critical that observable actions reflect realistic decision-making. Human behavior models based on psychological, organizational, and sociological theory must be employed to achieve realism in the simulation. Especially for individual level, it is important to represent the processes underlying the observable behavior, including attention and multitasking, memory and learning, decision-making, perception and situation awareness, and planning [4]. However, it is challenging to simulate human behavior in a manner that is sufficiently accurate. AI techniques have been utilized to make decisions that are as close to reality as possible. The issue with AI is that it aims for perfect intelligence but ends up making predictable conclusions. However, Computational Cognitive Modelling differs from other AI techniques in that it attempts to simulate human behavior with all its biases and limitations. [5]

1.2 Proposed Methods and Models

TacAir-Soar[6] is the most comprehensive study for human behavior representation in military simulations in the context of aerial warfare, using computational cognitive modelling. It utilizes Soar Cognitive Architecture for developing human behavior models, which is capable of real-time reasoning and decision-making on complex goals and plans. Although the project is comprehensive in terms of reasoning and decision-making, the cognitive capabilities of agents are provided by the mechanisms of Soar cognitive architecture and there is no underlying psychological background. Merk [2] has also proposed a comprehensive model for agents in military simulations and this study covers several higher level cognitive processes considering the psychological models for each. In between these two studies, this study concerns a computational cognitive model for representing human behavior in military simulations, developed with Soar cognitive architecture utilizing theoretical assumptions provided by the relevant literature of cognitive psychology .

1.3 Contributions and Novelties

The aim of this study is, using computational cognitive modelling, to model situation awareness and surprise cognitive processes within human behavior representations based on psychological theories. Our research enables the simulated agents to behave more realistically in a simulation environment. Our contributions are as follows:

- The adversary agent behavior representation in a constructive simulation infrastructure
- The integrated cognitive modelling of Situation Awareness and Surprise cognitive processes.
- The integration of Soar Cognitive Architecture and STAGE Simulation Engine.
- Computational cognitive modelling of Data/Frame Theory of Sense-making

1.4 The Outline of the Thesis

In Chapter 2, the important concepts at the core of this study are elaborated. Chapter 3 provides the literature review on current military models of human behavior representation, computational models of surprise and computational models of sense-making. Chapter 4 presents our proposed model and the experiments that are conducted to explain the abilities of an agent using the model. Experiments were carried out in order to demonstrate the exhibition of situation awareness and surprise processes and their impact on decision-making in unexpected scenarios. Finally, the results of the experiments are discussed, the research is summarized, and some recommendations for future work are included.

CHAPTER 2

BACKGROUND

The aim of this thesis is to model the behavior of adversary agents against Surface-to-Air Missile Systems, which are in the subset of Air Defense Systems, at the individual level using cognitive modelling. First, the definition, objective and working principle of Air Defense Systems are elucidated in this section. Second, higher level cognitive processes which are studied in a computational perspective and often consulted during this study are defined and some research on them are mentioned. Third, in which cases the human behavior representations are used in military simulations is examined. The notion of Cognitive Architecture -especially Soar Cognitive Architecture- is emphasised because it is used as the modelling instrument.

2.1 Air Defense Systems

Air defense refers to the process of safeguarding assets against attacks that originate in the air. This involves identifying enemy aircraft and notifying ground forces, eliminating enemy air strike targets, and aiding theater missile defense. Among the components of air defense systems are early warning radars, fighter aircrafts, airborne early warning aircrafts, surface-to-air missile (SAM) systems, and other similar technologies. The SAM systems are the only ones that are covered by this thesis. The aircraft pilots has awareness about SAM systems thanks to Radar Warning Receivers mounted on the aircraft. In this section, Surface-to-Air Systems and Radar Warning Receivers will be discussed.

2.1.1 Surface-to-Air Missile (SAM) Systems

(The infrastructure and operation of SAM systems discussed here is based on TARTAR system which is examined in details in a manual prepared by U.S. Naval Education and Training Program Management Support Activity [7]. Other SAM systems are similar to each other.)

The SAM systems can be broken down into three main parts:

1. Surveillance units that help find, detect, and identify the target at the start.
2. Command and control units that evaluate targets and send commands and control signals to other parts of the SAM system.
3. Fire control units direct and aim the surface-to-air missiles that are fired at the enemy.

Visually, radar, or both may be used to find an air target in the early stages. Visual detection of a target at long range, or even at close range, is difficult when visibility is low. As a result, the first contact with an airborne target is often established using air-search radar. These radars are intended to maintain continuous monitoring of a huge aerial volume.

After detecting a target and determining its approximate position, the search radar must also identify the target and decide whether it is friendly or hostile.

Once the air-search radar identifies and locates the target and the IFF (Identification of Friend or Foe) equipment determines whether it is a friend or foe, the target information is sent to the command and control system through these sources.

As a target approaches, the Command & Control System units evaluate the tactical situation and perform some preparatory work that allows the fire control units to set up a line of fire. These two steps are called (1) evaluating the target and (2) designating a chosen target to the right fire control unit. The goal of evaluation is to figure out what the target's intentions are, how dangerous they are, and what kinds of attacks they might be vulnerable to. The most dangerous targets are then assigned to a fire control system that is capable of dealing with their threat.

Target designation signals coming from the Command & Control System positions the target tracking and missile guidance radars to the designated range (the distance between the radar and the target), bearing (horizontal angle between the direction of radar and the target), and elevation (vertical distance of target to ground). When the radars acquire the target with given information, they lock onto the target and start to track it. As soon as the fire-control computer determines the proper aiming position for the launcher, it transmits signals to the unit that program the operation of the missiles and launch the missile aiming to the target.

There are two kinds of SAM missiles: remote-control and homing. The trajectory of Remote-Control missiles is controlled from the Fire Control System, which continues tracking the target after the launch. Homing missiles has a radar/heat seeking system on board, and mostly guides themselves according to the target information they acquire. In the scope of this thesis, only remote-control missiles will be considered.

There is a different kind of SAM system, which is called Man-portable Air-Defense System (MANPADS), which are launched on the shoulder. These does not have any radars to search or track the target. The military personnel visually acquire the target and launch fire-and-forget missiles. Fire-and-forget missiles does not require further guidance after launch such as illumination of the target, and can hit its target without the launcher being in line-of-sight of the target.

2.1.2 Radar Warning Receiver (RWR)

In modern fighter jets, there is a system called Radar Warning Receiver (RWR) which detects the radio emissions of radar systems, identify various threat types and gives relative azimuth (horizontal angle between the aircraft and the threat) information to enhance the situation awareness. The system will detect search, track and launch signals of SAM systems and warns the pilot using different symbology and audio tones. The warning systems are also capable of selecting a primary threat by following priority logic for the various threats detected, considering the radar mode (search, track, launch), the type of the threat (airborne radar, long-range radar, short-range radar, early warning system, etc), and signal strength. [8]

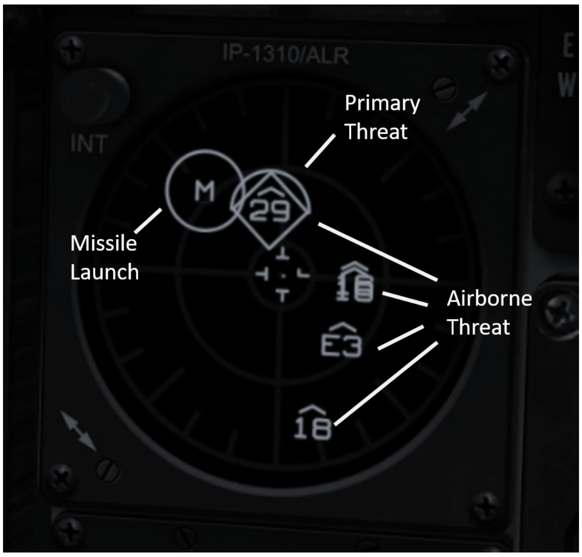

Figure 1: An example Radar Warning Receiver screen

2.2 Higher Level Cognitive Processes

2.2.1 Memory and Learning

Learning is a process about revising existing knowledge, acquiring and encoding new knowledge from instruction or experience, and combining existing components to infer and deduce new knowledge [9]. There are several learning models for human. The following four types of learning models have been extensively developed and tested in cognitive psychology: rule-based learning, exemplar- or case-based learning, neural network learning, reinforcement-based learning. Rule-based learning assumes that all knowledge is represented in the form of condition-action rules and suggests creating a new rule encoding the conditions preceding the problem and the solution as the result of a problem-solving session. Therefore, the next time similar situation is encountered, same solution can be provided. This method is called chunking. Exemplar-based learning proceeds over experiences with past problems to solve new problems through the retrieval of solutions to similar past problems. Neural networks are based on concepts derived from neuroscience. They are networks that consist of neural units and weighted connections between them. Neural units pass over the stimulus they have received to other neural units. The activation produced by stimulus is interpreted as the short-term memory. On the other hand, long-term memory is the knowledge represented by connection weights. Reinforcement learning is the process which individuals learn the most pleasant way to complete a task by trial and error.

To store and retrieve knowledge, which are the functions required for learning, the structure called memory is used. The general agreement for human memory suggests that there are various types of it, but two types are the top level: short-term memory and long-term memory. Long-term memory stores general knowledge or records of experiences. On the other hand, short-term memory holds a limited amount of information, which is retrieved as an activated subset of information in long-term memory, in an accessible state temporarily. There is also a term called working memory, the memory which carries out cognitive operations on the information stored in short-term memory. [10]

Long-term memory can be divided into two types: explicit memory and implicit memory. Explicit (declarative) memory is the memory of things describable in words. Explicit memory has also two types of memory: episodic and semantic. Episodic memory stores the events experienced. Semantic memory, however, stores the general knowledge about the world, such as rules, concepts, meaning of words, etc. Implicit (non-declarative) memory is the opposite of explicit memory as having the memories which are not expressible with words but with other means like motor skills (procedural memory), conditioned reflexes, conditioned emotional responses. [11]

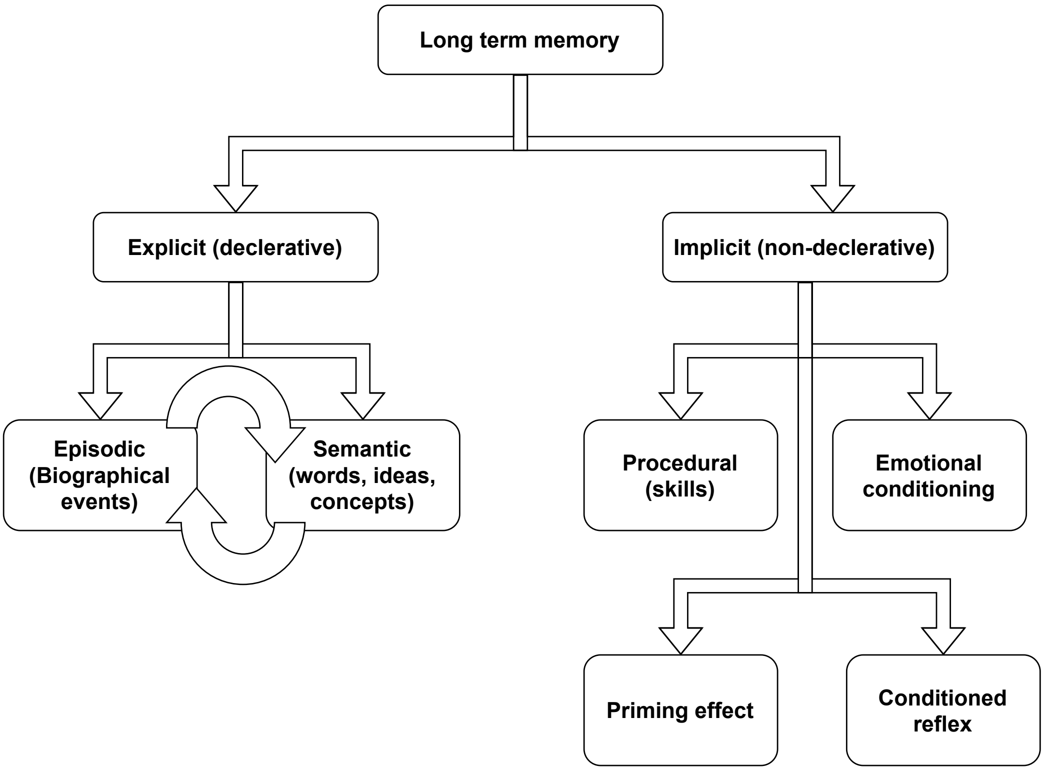

Figure 2: Types of Long-term memory

2.2.2 Situation Awareness

Endsley [12] define the situation awareness as “the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future”. Harwood et al. [13] extend the situation awareness definition for military. From a military point of view, situation awareness encapsulates the knowledge of the spatial relationships among aircraft and other objects, the knowledge of the presence of threats and their objectives, the knowledge of own state, the knowledge of who is in charge – the operator or an automated system-, and the knowledge of the evolution of events over time. Endsley [14] discussed the situation awareness in three phases. First phase is the perception of cues. In this phase, the key elements or events that serve to define the situation is identified. Second phase is the comprehension of the current situation. Last phase is the projection of future status.

Expert systems, case-based reasoning, and belief networks are some of the methods for implementing situation awareness. [4] In the scope of thesis, Situation Awareness will be modelled using Soar Cognitive Architecture, which is an example of an expert system. Expert systems, or production rule systems, have been used to describe situation awareness in computational behavior models since the early 1980s [15] [16]. A situation was assessed using the rule "if a set of events E occurs, then the situation is S." by the situation awareness model developed in these efforts.

Decision cycles conducted within Soar architecture start with perception and assessment of the state/situation and then continue with the perceived/assessed state to drive its own goal-oriented behavior. By using the Soar architecture, an explicit situation awareness approach is possible, consisting of a situational rule followed by a decisional rule of the type:

IF [(event 1 is true) AND (event 2 is true) . . .] THEN [situation 1 is true]

IF [situation 1 is true] THEN [perform maneuver type 1]

This approach allows for higher-level situational evaluations to be formed. To illustrate this idea, Tambe et al. [17] show that by combining a sequence of low-level events (e.g., enemy position and speed), a higher-level situational attribute (e.g., the enemy is presumably a MiG-29) can be generated. When a higher-level attribute is known, it can be used to figure out what other things are likely to happen in the situation that haven't been seen. For example, the enemy MiG-29 probably has fire-and-forget missiles, which can have a big impact on the decisions humans make afterward (e.g., the enemy is likely to attack me right after targeting my wingman).

2.2.3 Decision Making

Decision-making is the process of selecting one of the competing alternatives (a belief or a course of action) based on preferences and beliefs of the decision-maker. The decision-maker makes his choice according to his expectations about possible consequences and uncertainties of decision options. While modelling the decision-making process, such concerns as variability, flexibility and adaptability must be taken into account. Otherwise, the model will be stereotypical, predictable, rigid and doctrine-limited. To promote unpredictability and give possibilities for learning and exploration of new choices in an uncertain and dynamic environment, it is necessary to have some degree of variability. Adaptability and flexibility is important to reevaluate resources and plans with the help of experience in case of unexpected events.[4]

The most known decision-making model is Expected Utility Theory. Expected Utility Theory recommends that one should choose an option based on the gains or losses it causes, which are determined by one's tolerance for risk and personal preferences. The expected utility of an option is determined by the probabilities of the possible outcomes and the gains or losses for these outcomes for that option. Then, calculating expected utilities for all of the options, one chooses the option with highest expected utility. However, the theory always aims for idealized rational decision-making. We know that there are also emotions and biases in human decision-making, and humans are bad at estimating probabilities. There are some alternative theories for human decision-making, such as Recognition-Primed Decision Model [18], Cognitive Continuum Theory [13], Somatic Marker Hypothesis [19], Problem-Space Hypothesis [20] . These theories do not aim for perfect rational decision-making, and consider the inevitable effect of emotions and non-rational aspects.

2.2.4 Planning

Planning is the process of formulation, evaluation and selection of a sequence of actions to reach a goal. Planning is done by internally simulating the effects of actions, i.e. mental simulation, before acting. Widely known cognitive model of planning was proposed by Hayes-Roth & Hayes-Roth [21]. They suggest that individuals simulate the execution of plan mentally, and they do it using different levels of abstraction. Their planning model consists of different categories of decisions, so-called planes. These planes are used for retrieving and recording data by condition-action rules which run the planning process. For example, decisions about explicitly planned activities exist in one plane, while another contains decisions possibly useful data in generating planned activities, and another contains the planning method to approach the problem, etc. Then, the planes are also partitioned into several levels of abstraction. These levels provide a conceptual classification of the decisions made during planning and restrict the number of previous decisions each condition-action rule must consider in generating decisions.

U.S. Army planning process follows the doctrinally specified process detailed in Command and General Staff College publication ST 100-9 [22]. According to this document, planning is the first four-stage part of a five-stage decision-making process: mission analysis, intelligence preparation of the battlefield, development of courses of action, analysis of courses of action, decision and execution. The mission analysis stage is about considering the complete definition of the initial state (current situation) and final goal state (the mission objectives). Endsley's first stage of situation awareness is crucial for initial state specification. The intelligence preparation of the battlefield stage focuses on a detailed assessment of the situation. Assessment of enemy and friendly situations can be done in the scope of second stage of Endsley's SA model, and expectations of operational behavior can be focused in the scope of third stage of Endsley's SA model. The course-of-action development stage is the stage that plans to accomplish the mission are generated. In the course-of-action analysis stage, the generated plans are elaborated, mentally simulated, and evaluated and scored on the specified criteria. The last part course-of-action selection is the decision-making part, in which the highest-rated course of action is selected and executed.

Soar Architecture based on Problem-Space Hypothesis takes this problem as conducting look-ahead search. Look-ahead search evaluates all the alternatives by simulating mentally. Therefore, the assumption is that the agent has knowledge about the situation and alternative actions, about how to simulate the effects of an action mentally, and about how to evaluate the desirability of the outcome before planning process starts. Soar architecture handles mental simulation and the evaluation of mentally simulated alternatives with impasse and sub-state structures. [23]

2.2.5 Surprise

Surprise is the mental reaction to unexpected stimuli. Unexpected stimuli disrupts the integrity and predictability of reality, and it takes time to comprehend the new reality. A comprehensive psychological model of surprise is developed by Meyer, Reisenzein & Schützwohl [24] as an integration and elaboration of previous theories on surprise within the framework of schema theory. It is called “The Cognitive-Evolutionary Model of Surprise”. Schema theory [25] assumes that all the knowledge that a human has is organized into structured units called schemas. Schemas can be seen as a set of beliefs about objects, events, event sequences, situations, and their relationships among each other. Schemas are developed by experiences, used for the comprehension and interpretation of present and the prediction of future events, and changed as new information is received. The surprise model of Meyer et al. assumes that newly acquired information is continuously compared with the currently activated cognitive schemas, which may coincide with the content of the person’s working memory model of her present situation, preconsciously. A "surprise reaction" is generated when there is a certain level of disagreement between the schema and the input. The series of cognitive processes triggered by the surprise-eliciting events is as follows:

1. An event is appraised as schema-discrepant, or unexpected. If the degree of schema-discrepancy (unexpectedness) exceeds the certain threshold, this means that one or more of the person’s current expectations is violated.

2. Attention is drawn back to the unexpected event interrupting the ongoing mental processes, and surprise is experienced.

3. Experienced surprise enables and prepares the analysis and evaluation of the unexpected event, validating its unexpectedness, determining the cause of the event, and evaluating whether a response or change in schemas is required.

4. If the analysis suggests so, immediate reaction is provoked to the unexpected event, as well as updating, extending, or revising of the schema or schemas that caused the discrepancy.

2.3 Human Behavior Representation in Military Simulations

Human behavior representation entails the modeling of the processes and consequences of human behavior at the individual, unit, and command levels. Regarding the simulation users, military simulations can be grouped into several categories according to their aim by using human behavior representation. Training simulations are for training individuals or teams. Mission rehearsal simulations are for preparing operational force members for specific missions. Analysis simulations are for evaluating weapon systems, doctrine, and tactics. Acquisition simulations are for supporting acquisition decisions based on the expected performance of weapon systems. Joint force analysis simulations are for improving the command, control, and communications interoperability of joint forces. Military simulations are also grouped regarding the human behavior represented by a computational model or actual human participants. First are live simulations that real humans operate real equipment, but weapon firing and different types of ordnance are simulated. This type of simulations is used for maintaining readiness and testing new employment concepts. Second is virtual simulations that are characterized by real humans operating simulated equipment in simulated environments. In some cases, opposing forces are operated by real humans; in other cases, a human controller represents the decision-making and tactics of enemy commanders and the execution of battle movements, and engagement behavior is simulated to represent doctrinal maneuvers. Virtual simulations are usually used for training purposes. The last one is constructive simulations that simulated people operating simulated equipment, usually not in real time. Limiting values on parameters are set beforehand by real people, but the models determine the outcomes. All human behavior in the event flow comes from models. In military simulations, integrative cognitive architectures are used to represent human behavior. Detailed information about cognitive architectures is given in the Cognitive Architectures section. Review and comparison of several cognitive architectures used in military simulations can be found in a study published by National Research Council [4].

2.3.1 Computer Generated Forces (CGF)

The comprehensive definition is made by US Department of Defense as "A generic term used to refer to computer representations of entities in simulations which attempts to model human behavior sufficiently so that the forces will take some actions automatically (without requiring man-in-the-loop interaction)". [26]. CGFs operate in synthetic environments. A synthetic environment is defined in the words of Dompke [27] as "Internetted simulations that represent activities at an appropriate level of realism. These environments may be created by within a single computer or over a distributed network connected by local and wide area networks and augmented by realistic special effects and accurate behavioral models." The CGFs are grouped accordingly to their automation levels. Semi-automated forces have automated command processes, which means that the actions executed on a given command and the state conditions required for a command can be employed are preprogrammed. However, commands are chosen under the supervisory control of human. Therefore, there is actually no underlying model of human behavior. Most known semi-automated forces are ModSAF and CCTT SAF developed by U.S Army [28]. Intelligent forces exhibit their behavior to achieve the assigned goal while interacting with the environment and other entities. They are fully automated, which means they are fully responsible of reasoning, problem solving and learning to carry out the task because there is no human supervisor. FWA-Soar and RWA-Soar is the most known examples for this type of forces [29]. Command forces can be considered as a kind of intelligent forces, but these forces are specialized for commanding the battalion of intelligent forces. Soar/CFOR is one example of a development effort for U.S Army [30].

2.3.2 Cognitive Architectures

On-going research is aimed at the development of general computational systems that possess cognitive abilities such as interacting with dynamic complex environments, pursuing multiple goals, utilizing vast amount of information while planning, and continuously learning from experience. Cognitive architectures have arisen as a generic task-independent infrastructure, which coordinates the cognitive abilities stated before, to utilize the knowledge for problem solving and produce the related behavior. [23]

The most used architectures are ACT-R, Soar and BDI. ACT-R is a cognitive architecture that is aimed at understanding human cognition [31]. Soar is a cognitive architecture that exhibits intelligent behavior [32]. The beliefs, desires, and intentions (BDI) model is based on the theory of human practical reasoning developed by philosopher Michael Bratman [33].

2.3.2.1 The Soar Cognitive Architecture

Soar is a cognitive architecture that is used to construct detailed human behavior models and to design integrated artificial intelligence agents that interact with their surroundings in a dynamic manner utilizing these models. The components of the Soar cognitive architecture are designed to support problem-space computational model (PSCM), defined by Newell [34]. PSCM claims that behavior is the result of a sequence of choices made by an agent as it chooses actions to advance toward its objectives based on available information [37]. There are three main concepts in the problem-space framework: the agent's current state, operators, and goals. The agent is constantly in a **state**, which is a representation of the current problem-solving scenario; an **operator** transforms the state (makes modifications to the representation); and a **goal** is the intended result of the problem-solving activity. Until the goal is reached, Soar keeps applying the current operator and selecting the next operator (a state can only have one operator at a time).

In Soar architecture, the present situation, comprising sensor data, intermediate inferences, active goals, and active operators, is all stored in **working memory** for quick access. On the other hand, **production memory** contains the information needed to solve a given problem, including how to select and apply operators to modify the problem's states, and how to recognize when the goal has been reached. This information is represented with rules which are similar to if-then statements: if (condition) then action. When conditions are met in the current situation, so the rule is matched and it fires. This means that its actions are carried out, which changes working memory accordingly. In Algorithm 1, the simplified version of Soar algorithm can be seen.

Once enough information has been gathered, a selection is made on which operator to apply. If rules lack the knowledge to solve problems, Soar is stuck and reaches an impasse. The possible reasons for an impasse can be that:

1. Because no new operators have been proposed, no operator can be chosen.
2. There is no way to choose an operator since multiple operators are suggested and the comparisons are inadequate to identify which one should be chosen.
3. Although an operator has been chosen, it cannot be applied due to a lack of knowledge.

After an impasse, the Soar architecture provides a substate where operators can be used to produce or purposefully obtain the information that was not immediately accessible.

In Figure 3, a structural view of Soar is represented. Input coming in through Perception module and kept in Perceptual Short-Term Memory is converted to symbolic structures and added to Symbolic Working Memory. Symbolic Working Memory reflects the current knowledge of the world and the status in problem solving. Problem solving makes use of symbolic long-term memories. There are three long-term memories, which are independent and have separate learning mechanisms.

Procedural Long-Term Memory, also known as Production Memory, keeps the production rules, which controls the processing. If the conditions of a production rule match against the contents of working memory, the production fires, and the actions are performed. There are two learning mechanisms associated with production memory: chunking and reinforcement learning. Chunking learns new production rules about how the impasses are resolved. Reinforcement learning tunes operator selection knowledge based on a given reward function. Semantic memory stores general facts about the world. Episodic memory contains memories of the agent's experiences. Actions can make changes external environment, or they can modify Perceptual Short-Term Memory through Spatial/Visual System (SVS) that models many of the characteristics of human mental imagery.

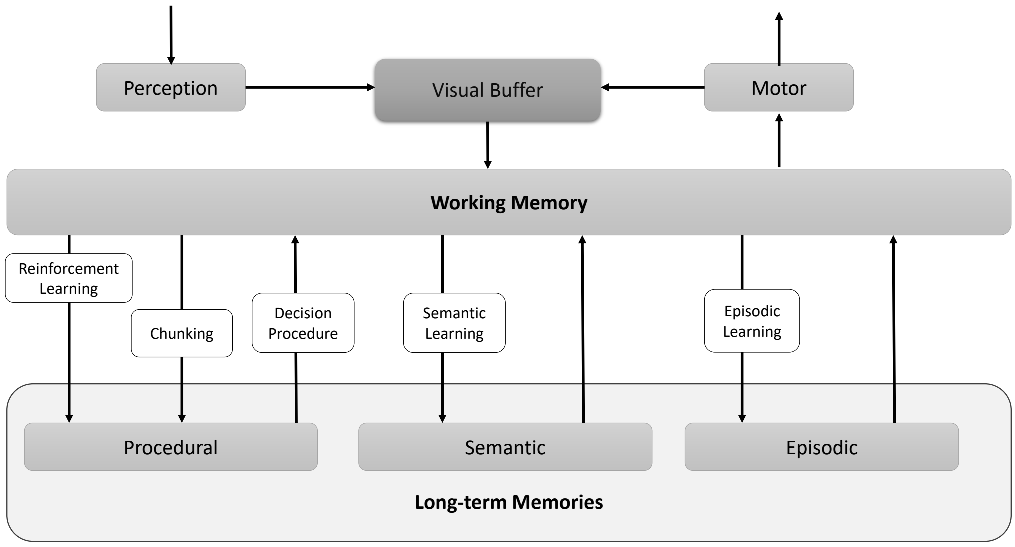

Figure 3: Structural view of Soar

Algorithm 1 Simplified version of Soar algorithm. [35]

```
procedure SOAR
    while HALT≠ true do
        Cycle
procedure CYCLE
    InputPhase
    ProposalPhase
    DecisionPhase
    ApplicationPhase
    OutputPhase
procedure PROPOSALPHASE
    while some i-supported productions are waiting to fire or retract do
        FireNewlyMatchedProductions
        RetractNewlyUnmatchedProductions
procedure DECISIONPHASE
    for each state in the stack, starting with the top-level state do
        EvaluateOperatorPreferences            ▷ for the state being considered
        if one operator preferred after preference evaluation then
            SelectNewOperator
        else                                   ▷ could be no operator available or
            CreateNewSubstate                  ▷ unable to decide between more than one
     until a new decision is reached
procedure APPLICATIONPHASE
    while some productions are waiting to fire or retract do
        FireNewlyMatchedProductions
        RetractNewlyUnmatchedProductions
```

CHAPTER 3

LITERATURE REVIEW

3.1 Current Military Models of Human Behavior Representation

Computer-generated forces are the representations of forces in military simulations. They are supposed to act in a way that is close to how they would in the real world. For realism, it is necessary to reduce the predictability of the model's forces by using human behavior models. It provides more intelligent and adaptable behavior to the complicated nature of the environment, as well as more reasoned action[4].

This section reviews existing military models that have been developed to achieve realistic human behavior representation in computer-generated forces.

3.1.1 Modular Semi-automated Forces (ModSAF)

ModSAF provides a set of software modules for Advanced Distributed Simulation and Computer Generated Forces applications. It aims that a single human operator can create and control multiple entities used for realistic training, test, and evaluation on the virtual battlefield. These entities are sufficiently realistic so that the user is not aware that the entities are being maneuvered by computers. The type of entities varies between ground and air vehicles, dismounted infantry, missiles, and dynamic structures. The entities can interact with each other and with manned individual entity simulators to support training, combat development experiments, and test of evaluation studies. Using ModSAF allows researchers to focus on agent development rather than vehicle simulation concerns like motion dynamics and DIS networking. [36]

The behavior of entities is implemented in line with subject matter experts and doctrine given by the Army Training and Doctrine Command. ModSAF has no underlying behavior model; all behavior is programmed into the finite-state machine. For each state, the machine has a list of commands that may be sent, a list of actions that can be performed, and a list of state requirements that must be met before a command can be executed [28].

3.1.2 One Semi-automated Forces (OneSAF)

OneSAF provides a simulation environment for individual military forces such as soldiers, tanks and helicopters through aggregate units [37]. The forces are operating either in a completely automated mode or under the control of the training audience via their organic command and control systems or role players using the Graphical User Interface. OneSAF has behavior models, which provide cognitive models of units and entities and utilize the XML based behaviors that have been composed by the user for each units and entities of the scenario. Users are given the capability to construct complex behaviors in an XML based Behavior Specification Language. It replaces legacy military simulations such as BBS, ModSAF, JANUS, CCTT SAF, and AVCATT SAF.

All agents in OneSAF communicate via a framework called blackboard. Agents must register to the blackboard to receive data and events. There is also a world model that stores perceptions and inferred facts. The behavior models written by the users are flow-chart like action sequence descriptions. It includes sequential, parallel, looping, and branching and uses predicates to branch. Actions specified by users can use parameters and these parameters can be computed by preceding actions. Behavior models includes if-then rules to assert facts and start reactive behaviors. Behavior models process data coming from blackboard, make inferences, and add facts to world model. They also put command data on blackboard to trigger physical agents.

OneSAF is considered as a simulation environment, whose entity behaviors can be modelled by using Soar Cognitive Architecture[38]. The framework of OneSAF is appropriate to replace all behavior agents with Soar agents and they will trigger OneSAF physical agents using OneSAF blackboard protocol. The infrastructure of blackboard and world model is also replaced by Soar's working memory. Therefore, OneSAF will send input to Soar agent's working memory instead of writing them to blackboard, and will receive triggers from Soar agent. This study resembles with our utilization of STAGE simulation environment to observe the behavior of our proposed model implemented using Soar cognitive architecture.

3.1.3 TacAir-Soar

TacAir-Soar is an intelligent and rule-based system that produces realistic human behavior for large-scale and distributed real-time military simulations. It was started as a research project to determine whether it was possible to replace human controllers with human behavior models. It achieves its goals by integrating a wide variety of intelligent capabilities, including real-time reasoning and decision-making on complex goals and plans, communication and coordination with humans and simulated entities, situational awareness, and the ability to react to new orders on the fly.

TacAir-Soar was developed by using the earlier versions of Soar. With the latest features of the current architecture, some improvements that can be done to build, customize, and maintain the TacAir-Soar with greater autonomy and robustness, are mentioned in Laird's book [23].

Agents from TacAir-Soar took part in the simulation by interacting with the ModSAF simulation environment. ModSAF gives simulated sensor and vehicle information to TacAir-Soar agents and transforms the commands given by the TacAir-Soar agents into commands that may be used to drive the vehicle, fire weapons, or communicate through radio, etc. [6]

3.1.4 Smart Bandits

Smart Bandits [1] is an AI solution of Netherlands Aerospace Center to model the behavior of Computer Generated Forces. Within simulation environments, it allows end-users to build and execute behavior models for autonomous entities. The behaviors implemented in scripts or based on a set of event-action rules considered to be limiting the capability of CGFs in dealing with unexpected situations that have not been anticipated during the scenario design. The authors propose a paradigm shift from scenario-oriented modeling, in which CGF behavior is primarily driven by scenario events and triggers, to agent-oriented modeling, in which CGFs are treated as fully autonomous entities with the ability to perceive and comprehend the world around, to achieve their goals, and to plan their actions in a dynamic environment.

An agent in Smart Bandits is supposed to have two essential functional roles: decision-making and knowledge management. There are default modules to perform these roles, but the end-users can also add or exchange current modules with custom modules to extend the agent's capabilities via Smart Bandits API. Decision-making module is based on the hierarchical finite state machine paradigm. The authors explain why they chose this paradigm as the result of the graphical representations that the hierarchical finite state machine permits and the similarity of these representations to the way that humans express their reasons when executing a mission.

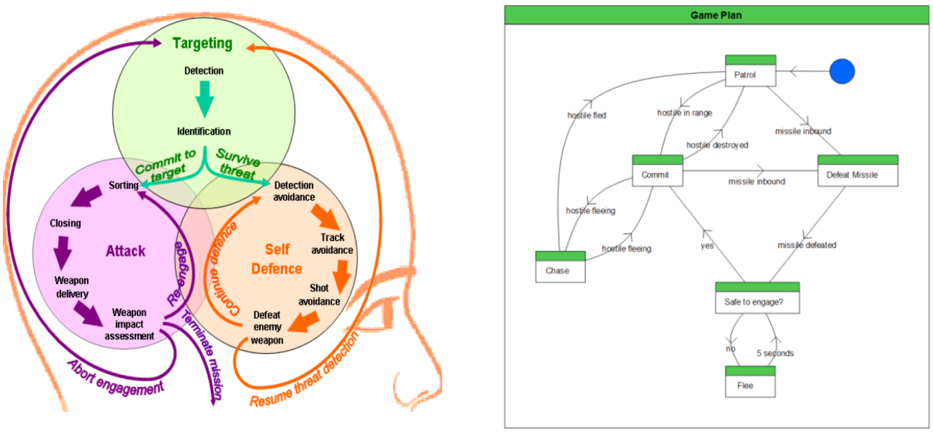

Figure 4: Example fighter pilot model (left) and the model represented in Smart Bandits (right) [1]

Agents store information in two types: observations and entity beliefs. These two types represent the agent's short-term and long-term memories, respectively. The information received from the entity's sensors is represented by observations. They're usually organized by sensor type and updated on a regular basis. On the other hand, entity beliefs are an agent's long-term assumptions about itself and other entities in its environment. Observations are used to infer entity beliefs, which are then timestamped.

3.2 Computational Models of Surprise

3.2.1 Macedo & Cardoso Model

The computational model of surprise proposed by Macedo and Cardoso (see [39] [40] [41] [42] [43] [44]) enhances the surprise model of Meyer et al. [24] supplementing with the proposals of Ortony and Partridge [45] about the choice of the agent's knowledge structures, the implementation of the appraisal of unexpectedness and the computation of the intensity of surprise.

The assumptions the computational model is based on as follows:

- The cognitive system (e.g., artificial agent) that incorporates the surprise model has a memory that stores information or knowledge. The term "knowledge" refers to the type of declarative knowledge known as belief. The agent's declarative knowledge base can include both episodic and semantic knowledge, and it enables the agent to build a model of its world by representing what the world was like in the past, what it is like now, and what it will be like in the future.
- The agent possesses an inference mechanism that allows it to extend its belief base by creating expectations when encouraged to do so by the situation or context (e.g., the acquisition of new information or the contemplation of possible actions). These expectations might be forecasts of future world states or they can be formed during the computation of the current world state, where they fill in gaps in the available observable knowledge.
- The agent possesses a surprise mechanism that compares newly acquired input information to existing beliefs and generates surprise if a discrepancy is found.
- The agent's conduct is guided by the surprise intensity values determined by the surprise mechanism.

Macedo & Cardoso states that the agents' world model is generally both incomplete and partially wrong. It is incomplete because some parts of the environment are inaccessible to the agent. It is partially wrong because the agent's perceptual and reasoning processes are not error-free and may be distorted. However, we know that the agent's actions will be successful if it has reasonably complete information about decision-relevant world states. As a result, even (and especially) when knowledge is incomplete or uncertain, the agent must construct rational and complete models of the world. Humans attempt to overcome these limitations of their cognitive systems by generating assumptions or expectations. Assumptions or expectations are beliefs that fill gaps in an agent's knowledge of the world, allowing the agent to approach the ideal model of the world.

Macedo & Cardoso accepts all the definitions for three causes for surprise made by Ortony and Patridge [45] based on two types of beliefs such as active and passive expectations:

- **Active expectation (Prediction) failure:** It happens when there is a conflict between the input and what the agent predicts motivated by the situation or context.
- **Passive expectation (Assumption) failure:** It happens when there is a conflict between the input and what the agent implicitly knows and believes.
- **Unanticipated incongruity:** It happens when there are no explicit expectations (passive or active) with which the input proposition could conflict.

According to the assumption of having a surprise mechanism, there should be a mechanism that generates surprise with a certain intensity in case of mismatch between input information and existing beliefs. All the expectations and their subjective probabilities attached, which is computed based on the absolute frequency values associated with the relevant cases stored in the agent's episodic memory, form the expectation set of the agent. According to the example given, if all houses represented in episodic memory have a pentagonal shape, the expectation "houses have a pentagonal shape" has a probability value of 1; whereas if only half of the houses have square windows, the expectation "houses have square windows" has a probability value of 0.50. Macedo et al. [40] also shows that the intensity of surprise depends not only on the probability of the actual outcome, but also on that of alternative outcomes. The model does not tell anything about what happens after the surprise intensity has found.

3.2.2 Lorini & Castelfranchi Model

The model of surprise proposed by Lorini & Castelfranchi (see [46] [47]) tries to model surprise as a belief-based phenomenon integrating with a formal model of belief change, considering the models proposed by Ortony and Partridge [45], Meyer et al. [24] and Macedo and Cardoso [39].

Lorini & Castelfranchi distinguishes the forms of surprise based on different kinds of expectation than Macedo & Cardeso. According to their view, an agent has a representation under scrutiny or scrutinized expectation, i.e., the expectation on which the agent is focusing its attention and seeking to match with the perceptual data. Moreover, they introduced two mental operations: 1) "retrieve" for introducing new expectation into the mental Test (scrutiny) space of the agent; 2) "perceive" for modifying the agent's perceptual data. There are also representations and expectations in the background or background expectations which reside at an unconscious level of processing. They are either the product of priming or part of the presupposed frame which represents its unproblematic understanding of the situation in which its action and perception are located, and which supports all (focused and background) expectations. Scrutinized expectations, background expectations and presupposed frame are in the set of explicit informational mental states. The other set of informational mental states called implicit expectations which are all those beliefs and expectations that can be inferred from explicit beliefs and expectations.

Based on all informational mental states defined, Lorini & Castelfranchi argues two different kinds of surprise:

- **Mismatch-based surprise:** In this case, the agent actively checks whether a certain event happening, she has an explicit representation about the next incoming data (scrutinized expectation) and attempts to match the incoming data against it. When there is conflict between the two representations, there occurs surprise. One can say that the cause of mismatch-based surprise is misexpected events. The intensity of mismatch-based surprise depends on the agent's subjective probability assigned to the expectation which conflicts with the input. Moreover, the absence of any object or event that the agent is expecting and scrutinizing is the example for misexpectation. The more certain was the scrutinized expectation, the more probable the event is, the more surprised the agent will be when there is mismatch with the input.

- **Astonishment or surprise in recognition:** The agent perceives a certain fact and recognize the implausibility of this. The event is unexpected for the agent. The more unpredictable, the more unexpected the event is, the more astonished the agent will be. There are two different kinds of mental processes to recognize the implausibility:
  - **Retrieval-based Astonishment:** After perceiving a certain fact that the agent was not actively expecting, she can look at her background knowledge to find the subjective probability for that fact and concludes that she would rather have expected the opposite of the perceived fact to occur. The intensity of astonishment depends on the agent's subjective probability assigned to the expectation for negation of the perceived fact. It is also called degree of unexpectedness. There is a pre-existing expectation about the event or the negation of the event, and the agent retrieves that from her background knowledge.
  - **Inference-based Astonishment:** After perceiving a certain fact, the agent realizes that she would rather have expected the opposite of the perceived fact to occur inferring from her explicit beliefs. There is no expectation about the event or the negation of the event before perceiving it.

Lorini & Castelfranchi also attempts to clarify under what conditions belief revision should be triggered unlike Macedo & Cardoso. They pointed out that accurate belief revision requires time and considerable computational costs, belief change in realistic cognitive agents should be triggered by the surprises which have particular amount of intensity. Another factor that affects the belief change is motivational relevance, i.e., the agent will be more likely to revise her beliefs when she perceives something relevant to her motivations than when she perceives something which are completely irrelevant. Moreover, the agent is more prone to revise her beliefs when she considers her sensors to be reliable than the times that she considers her sensors to be unreliable.

3.2.3 Merk (NLR) Model

Merk [2] assembles multiple theories and research to develop his surprise model.

First of all, Merk concludes five factors that influence the intensity of surprise:

- Expectation disconfirmation [48]
- Importance of observed event [49]
- Whether the observed event is seen as positive or negative [49]
- Difficulty of explaining / fitting in schema [50]
- Novelty (contrast with earlier experiences) [51]

Besides the intensity of surprise, Merk defines two possible effects of surprise unlike the other models of surprise:

- Interrupted decision-making and response delay to the surprising events [52] [48]

- Reduced response quality because of time pressure [53]

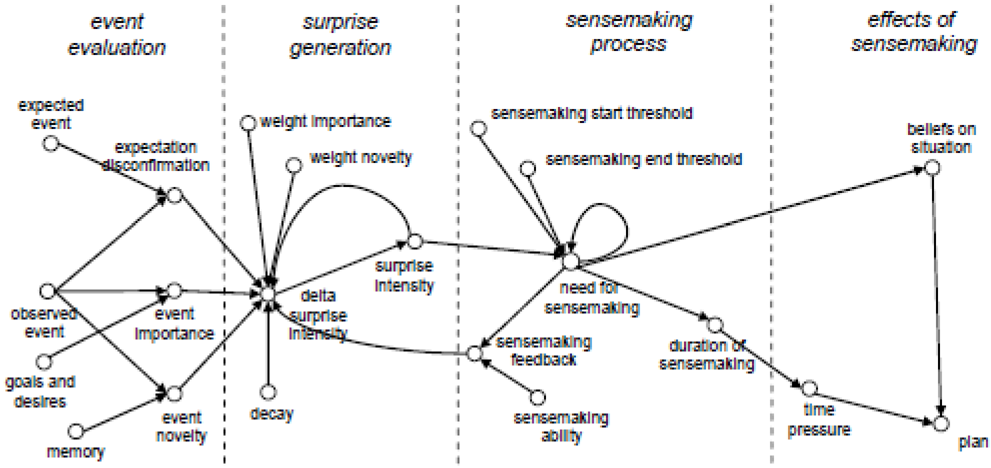

Figure 5: Computational Model of Surprise proposed by Merk. [2]

Merk divides the surprise model into four parts:

1. **Event Evaluation:** Events are continuously evaluated to determine the degrees of factors which influences the intensity of surprise

2. **Surprise Generation:** The intensity of surprise is computed based on the factors

3. **Sensemaking Process:** Sensemaking process tries to lower the surprise intensity over time. If the intensity of surprise is above the sensemaking start threshold, sensemaking processes start until the intensity of surprise is lowered below the sensemaking end threshold

4. **Effects of Sensemaking:** The more time the sensemaking process takes, the more the agent will feel time pressure on him, and the lower the quality of her response will be

Merk computes the change in surprise intensity in means of expectation disconfirmation, event importance, event novelty, sensemaking feedback, i.e., the degree of success of explaining the surprising event, and decay, i.e., non-cognitive factors that reduce the intensity of emotions over time. Expectation disconfirmation, event importance, and event novelty are the factors that increases surprise intensity. On the other hand, sensemaking feedback and decay are the factors that decreases the surprise intensity. The surprise intensity value for the next time can be computed the delta surprise intensity value added to current surprise intensity.

Merk does not mention about how the sensemaking is done. He abstracts the process and only mentions that the parameter of sensemaking ability can be used to differentiate between experienced pilots and less-experienced pilots. He is more interested in the effects of sensemaking process on the response quality and response time among the people with different sensemaking abilities.

3.2.4 Conceptual Model of Startle and Surprise

Landman et al. [3] make a differentiation between the terms of “startle” and “surprise”. Startle is fast and highly physiological reaction to a sudden, intense, or threatening stimulus; whereas, surprise is an emotional and cognitive response to unexpected events that are difficult to explain, forcing a person to change her beliefs. Understanding this differentiation between the surprise and startle and their effects is important to prepare flight crew for unexpected events in flight. Landman et al. proposes a model as the synthesis of:

- The cognitive evolutionary model of surprise [24]
- Perceptual cycle model [54]
- Data/Frame theory of sensemaking [55]
- Literature on startle and acute stress

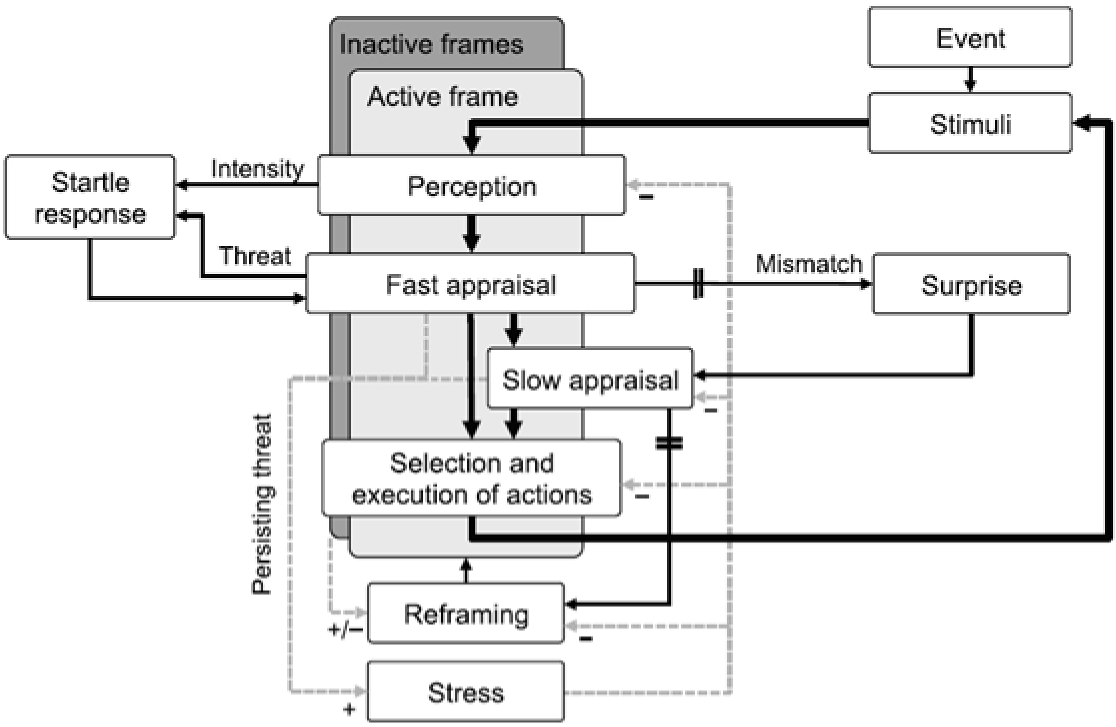

Figure 6: Conceptual Model of Startle and Surprise proposed by Landman et al.[3]

The perceptual cycle is the cycle that an agent perceives stimuli, interprets these stimuli, makes an assessment of the situation, selects and executes actions which (probably) make changes in the environment and causes new stimuli.

The concept of frames is useful in explaining the causes and effects of surprise. Frame is defined as an explanatory structure which synthesizes all types of knowledge structures in long-term memory to describe generic or specific situations. Frames are created based on previous experiences. A matching frame may be activated and applied whenever a circumstance arises in which frame-related knowledge can be used.

The startle response is a fast, sometimes unconscious, appraisal of a stimulus as threat related. Surprise, anyhow, occurs when there is a mismatch between appraisal of the situation and active frame, given that the mismatch exceeds a certain assumed threshold.

Sensemaking is an explorative process that is active, analytical, conscious. Sensemaking activities can be categorized into three groups:

1. If the surprising data is determined to be the consequence of a misunderstanding, the active frame can be kept.
2. If the surprising data are being judged as correct, the active frame may not be detailed enough to account for them, and it can be enhanced.
3. If the data are considered to be valid yet fundamentally contradict the active frame, a paradigm shift is required, and a new frame should replace the active frame, i.e., Reframing.

3.3 Computational Models of Sensemaking

Sense-making is a reaction to surprise. As soon as there is a difference between what is seen and what is expected, the process of sensemaking is triggered. Data/Frame Theory was suggested by Klein et al. [55] as a theory of Sensemaking. Long-term scripts and scenarios that have been learned and causal processes that explain how things function are considered as "mental models". On the other hand, the term "frame" is used to describe processes that are in progress or that are temporary in nature. Similar to Endsley's situation assesment [14], the frame serves as a hypothetical model within which to analyze and adjust one's understanding of the world and the state of events. Sensemaking, according to this view, is a dynamic process that involves fitting data into a frame and fitting a frame around the data in both directions. There is no such thing as a starting point for the data or the frame. The data provoke and assist in the construction of a framework, and the framework defines and links the data. It is possible to fit data into a frame or to question, develop, compare different frames, revise, or replace the frame itself if the data or the frame appear insufficient.

The concepts of data frames, and accounts are not described with the precision required to properly assess and model sensemaking by Klein et al. [55]. From a computaional modelling and implemantation point of view, what a "frame" is not clearly discussed or described. According to the theory, it may be in a form of a story, a map, a script, or any other explanatory knowledge structure. A computational specification of mental representations and processes involved in producing sense is not provided by data-frame theory. Because of this, although there have been efforts to model the sensemaking process computationally, there is unfortunately no widely accepted complete model of sensemaking. Among them, three attempts for computational modelling of sensemaking based on Data/Frame Theory will be discussed and examined. Two of them belongs to the program called ICArUS (Integrated Cognitive-Neuroscience Architectures for Understanding Sensemaking) conducted by the IARPA (Intelligence Advanced Research Projects Activity), which attempts to develop and test computational models of sensemaking. The goal of the program is to build a model of sensemaking that is neurally plausible and takes into account cognitive biases in the context of intelligence analysis. The first one use Bayesian inference rules to model sensemaking, second one facilitates ACT-R Cognitive Architecture for modelling purposes. The third one attempts to apply Finite State Automaton to modelling sense-making.

Burns [56] reported that an attempt has been made for modeling sensemaking using Bayesian inference rules in the scope of this program. They define sensemaking as a recurring cycle of obtaining evidence and updating confidence in competing hypotheses, to explain and predict an evolving situation. Frames are knowledge structures encapsulating hypotheses, evidence, confidences in hypotheses, and likelihoods of evidence. This definition goes far beyond that of the data-frame theory, in this case, a frame always represents data, but it also represents other knowledge and beliefs that help us make sense of data (like hypotheses, likelihoods, and confidences). The likelihoods are used to figure out how confident you are in your hypotheses, and they always refer to data (evidence). This is important because the "data-frame" theory doesn't talk about or model likelihoods, but they are important parts of frames (along with hypotheses, evidence, and confidences). Any sensemaker must mentally represent likelihoods, at the very least implicitly. People don't often say or measure numerically how likely something is. That means it's not possible to model and measure sensemaking in terms of normative (Bayesian) standards. To use this sensemaking model, you must either give people (and neural models) likelihoods as inputs, or you must measure how those people and models use those likelihoods when they make sense of evidence. People aren't able to report all possible outcomes in each sensemaking cycle because it's troublesome for them. In fact, even if human subjects were willing to do so, it is not possible for researchers to measure "average" human performance when all subjects use their own personal estimates of how likely it is that they will do something. Finally, even if it were possible, each person's likelihoods would be influenced by the real-world knowledge that he or she brings to an experiment.

Another endeavor within the scope of the program is to model sensemaking using the ACT-R cognitive architecture [57]. They help build a generic cognitive model of sensemaking that can predict human performance. They based their theory on information flows defined by Pirolli & Card [58] through two interconnected processing loops: the foraging loop and the sensemaking loop.

How data is gathered, filtered, and combined into structured evidence is called the "foraging loop." Learning, recalling, and assessing a frame, generating hypotheses, getting more data, and reframing are all part of the sensemaking loop.In ACT-R, the difference between frames and hypotheses is what kind of information is stored in the chunks. A frame can be a single chunk or a group of related chunks that hold rules for organizing and interpreting data-chunks (e.g., evidence files) into testable hypotheses. When people are making sense of things, they make hypotheses in the form of either an estimate of how many choices they will have to make or how likely it is that they will be right or wrong about something in the environment. A hypothesis can be thought of as a chunk that includes a representation of a possible response.

Feedback happens when you compare your current hypothesis to a normative (i.e., externally-driven) solution and then change the way you look at the data. Reframing can be used to change the hypothesis for the current data set and to come up with better hypotheses for future data sets. In the beginning, early evidence shapes the way people think about a frame. This frame can then influence how people think about future evidence through base-level activation mechanisms and spreading activation mechanisms.

Another attempt for computational modeling the Data/Frame Theory of Sensemaking was using a Finite State Automaton [59]. Using an FSA for the Data/Frame Theory simulation was chosen because of its properties for simulating changes in system behaviors and transitional states that are similar to the dynamic information changes seen during unstructured and dynamic situations.

CHAPTER 4

METHODOLOGY

4.1 Proposed Computational Model of Agents

The main concern of this thesis is the cognitive modelling of the behavior of adversary agents against Surface-to-Air Missile Systems. While doing this, Situation Awareness and Surprise cognitive processes are taken into account. The model attempts to abstract and simplify SA and surprise processes. This does not imply that the human mind functions on the basis of such computing, nor does it imply that basic processes, such as those happening at the level of the brain, are represented by the model. The model, on the other hand, attempts to replicate the tactical behavior of the CGF. Managing radar alerts caused by surface-to-air missile systems falls within the purview of this level of behavior.

Surprise sub-model is mostly based on the model proposed by Macedo & Cardoso [41]. The model meets the assumptions of declarative knowledge base (semantic memory which stores the beliefs), and surprise mechanism that compares the newly acquired input to existing beliefs and generates surprise if a discrepancy is found, which the Macedo & Cardoso model is based on. On the other hand, there is no inference mechanism which extends its belief base by creating expectations when the knowledge about the world is incomplete and uncertain. In addition, there is no probability attached to the expectations, which means all the expectations are equally likely to happen, and any unexpected state is always assumed to create maximum surprise intensity and make the agent experience surprise and provoke the sensemaking process. To sum, the model only checks for the violation of assumptions (passive expectations).

For sensemaking process, the model is based on Data/Frame theory, which is widely accepted as a comprehensive theory for sensemaking. Considering the challenges in defining the frames computationally, a simple approach is followed. In the model, the frame is represented as a state which the agent transitions to when the required conditions are met. The frames can be enhanced adding other functionalities about what will happen when the agent transitions to that state and activates the relevant frame. The frame can be considered as the collection of operations and actions under specific conditions. Semantic memory also stores assumptions of states that is known to be transitioned in a perfect world, for each frame.

On the other hand, the Situation Awareness sub-model is based on Endsley's Situation Awareness definition [14]. Situation awareness is provided by detecting the changes in inputs (level 1), matching them to the most relevant state (level 2), and retrieving the expectations for next possible states (level 3).

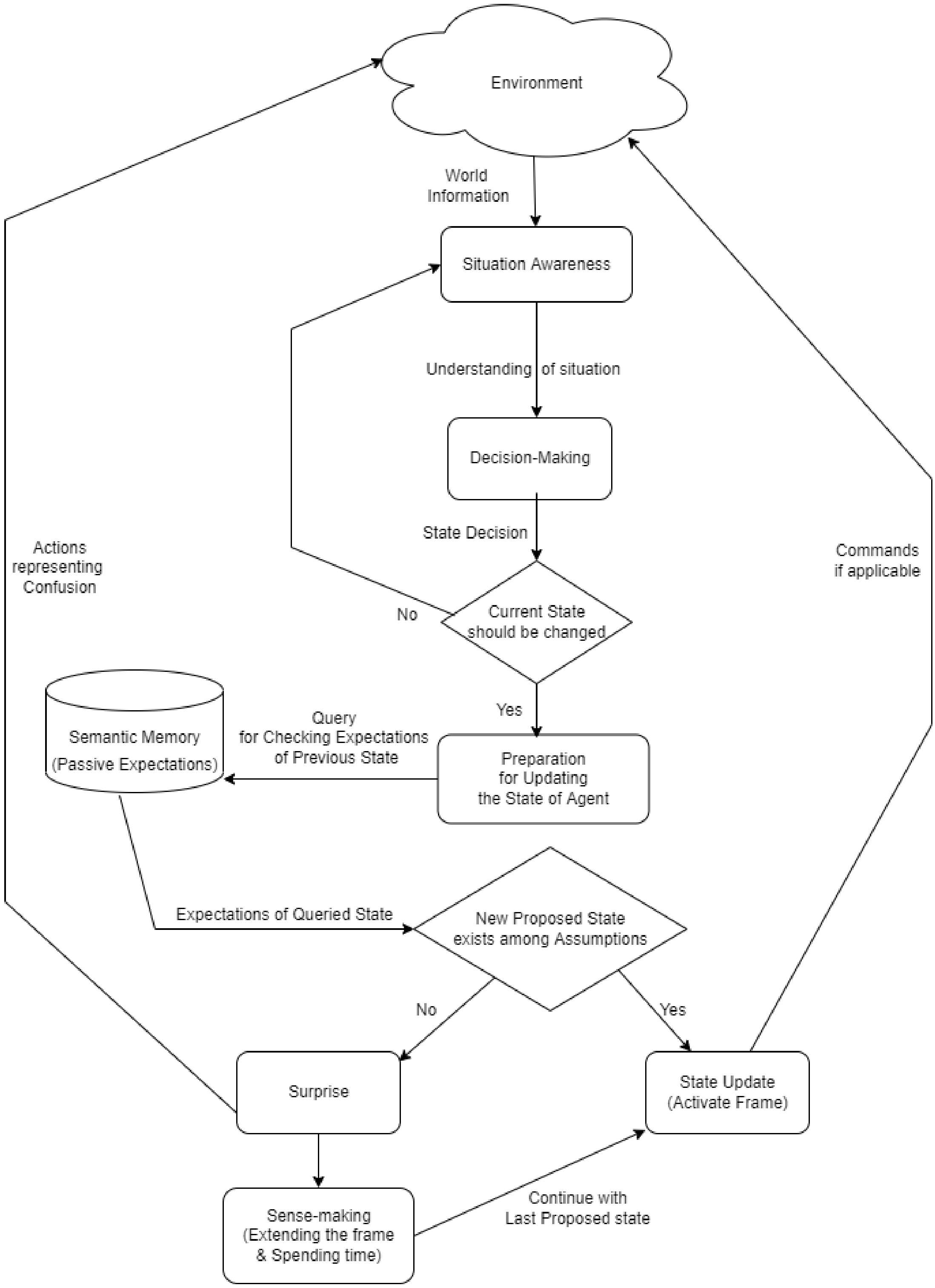

Figure 7: The flow chart diagram of the proposed model

4.2 Implementation and Simulation of the Model

4.2.1 Soar Cognitive Architecture for Modelling Human Behavior

This thesis concerns about the realistic representation of adversary pilot behavior against Air Defense Systems. Pilots are adaptable and utilize their expertise flexibly. This involves coping with unforeseen events, adjusting to new conditions, learning from experience, and juggling various goals (e.g. protecting a position, intercepting the enemy, and surviving). Pilots also balance mission-critical decision-making with quick responses to changing conditions and dangers. To create a more realistic training environment, automated agents should be indistinguishable from human pilots. Therefore, agents are expected to have these capabilities to imitate pilots realistically. On the other hand, the same vulnerabilities must be found in intelligent agents as humans. This includes things like attention and cognitive load, as well as physical limits, such as being under high stresses and being unable to process information properly.

Soar is ideally suited as a cognitive architecture with the objective of creating such intelligent agents that act similarly to humans. Soar is a cognitive architecture that is used to construct detailed human behavior models and to design integrated artificial intelligence agents that interact with their surroundings in a dynamic manner utilizing these models. Soar is the cornerstone for a proposed unified theory of human cognition, and hence fits well with many of the cognitive topics studied.

Deep learning or other AI techniques are not used in the scope of this thesis. Although Google DeepMind, Facebook AI Research (FAIR), and others have lately stated that deep learning can solve AI, these approaches are not employed for models with psychological backgrounds or for integrated modelling since they are specialized for a single purpose. For example, DeepMind research [60] focuses on major AI topics such natural language processing, perceptual processing, general learning, and AI evaluation. FAIR's research [61] interests include visual processing, data mining, natural language processing, and Human-Computer Interaction, as well. A unified model of intelligence is not yet possible, despite the fact that several models have shown cognitive ability in specific fields. However, deep learning technologies will certainly play a key part in future cognitive architectures.

4.2.1.1 The Model Implementation using Soar Cognitive Architecture

There are four simple states defined based on the warnings seen on RWR (radar warning receiver) screen: normal, search, track, and launch. As the input warnings change, the agent goes into the most relevant state.

1. **Normal**: Normal state is when no threat agent sends radar emissions representing the Search, Track or Launch mode of operation, and there is no primary threat. In other words, there are no detectable radar emissions on the RWR side. The expected state after this state is "Search" state.

2. **Search**: There is a search warning for primary threat. The aircraft is within enemy search radar range, and assumed to be detected and monitored by enemy air defense systems. The expected states after this state are "Normal" and "Track" states. If the aircraft exits the SAM system's search radar range, there will be no warning for that SAM system on the RWR, thus back to

"Normal" state. If the aircraft continues flying approaching to the SAM system, it will enter the track radar range and the SAM system starts to track it ("Track" state).

3. **Track**: There is a track warning for primary threat, but there is no incoming missile. Enemy air defense system forwarded the information about detected aircraft to the fire control systems. Fire Control Systems are tracking the target now. The expected states after this state are “Search” and “Launch” states. If the aircraft is quick enough to get away from the track radar range, it will possibly be in search radar range. Otherwise, the launcher gets the information of the aircraft and launches a missile.

4. **Launch**: There is a launch warning for primary threat or the missile launch happened in pilot’s visual range. As soon as the fire-control computer determines the proper aiming position for the launcher, a missile is launched aiming to the target. The expected state after this state is “Track” state. The SAM system will continue to track the aircraft after launch to guide the missile.

In the scope of thesis, the frames are only defined with its activation conditions, which are warning events and threat-in-visual-range information, and the track frame has a specific action for indicating the evasion maneuver for testing purposes.

In our model, when the new transitioned state is not within one of the expected states of the previous state, the agent goes into the surprise state. For the sake of simplicity and testing purposes, the agent spends some time, commands an action which indicates the confusion, and modifies the expectations of previous frame by appending the unexpected state (sense-making). After the sense-making is over, the agent comes out of the surprise state, and continues with the last state it was in.

For example, when the aircraft flies through its predefined route and there is no threat indicated on RWR, the pilot will be surprised if there is a tracking system appears on RWR. It is because “Track” state is not among the expected states of “Normal” state. The track warning may come without search warning before because of several reasons such as an anormality in the RWR system or SAM-traps. SAM-traps is a counter-attack tactics where a group of SAM sites remain hidden by keeping its radar deactivated, so that they are not seen on RWR. When the opponent aircraft flies within the weapon range of the SAM site, the SAM site operator is warned by another radar station, activates its radar and fires a missile at the opponent CGF. Another example of this suprise situtation is MANPADS (Man-portable Air-Defense System) which are launched on the shoulder. If the aircraft does not have any MWR (Missile Warning Receiver), the pilot does not get any search, track and missile warnings on RWR when an IR-guided missile launched from a MANPADS. Because RWRs show information about the radar emitters and radar frequency changes. If the aircraft has MWR, it only gets the launch warning when a missile is launched from a MANPADS, without any prior search or track warning. When the aircraft flies through its predefined route and there is no threat indicated on RWR, the pilot will be surprised if there is a launch warning. It is because “Launch” state is not among the expected states of “Normal” state.

The Soar production rules for the agent model can be found in Appendix A.

4.2.2 STAGE Simulation Environment

Presagis STAGE is a Computer Generated Forces (CGF) tool that includes various simple simulation models out of the box. Each of these simulation models is modifiable or even replaceable. The

Scenario Manager module can be expanded to supply and show information supplied by the user to and from the simulation. One can use the STAGE Development Kit to develop new simulation models and integrate them into the Simulation Engine. The data entered by the Scenario Designer module is used to power these new models.

4.2.3 Modelling SAM System in STAGE

For the sake of simplicity, the SAM system is designed consisting of a search radar, a radar which tracks the target and guides the missile, and a launcher which launches SA-2 missiles, which is a kind of remote-control missile and already defined in Stage Simulation Environment. Specifications within STAGE program can be seen at Figure 8.

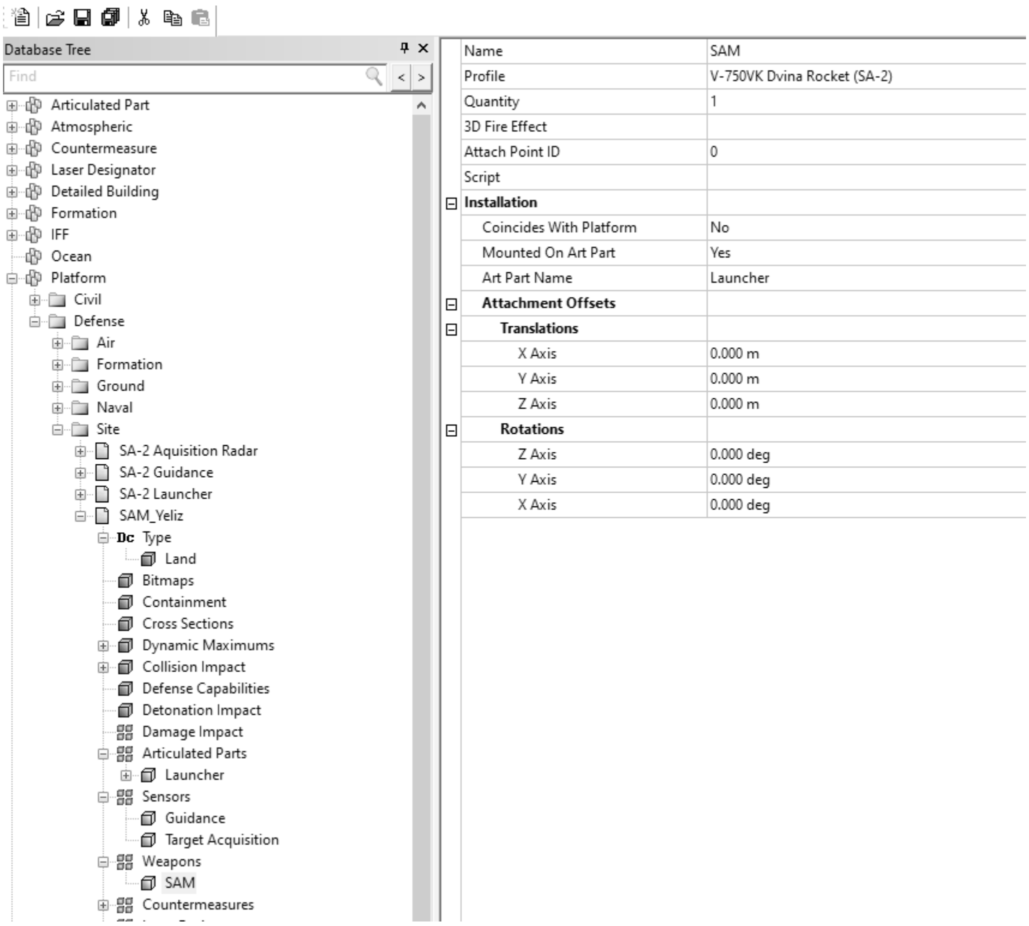

Figure 8: SAM site specifications

The SAM system is modelled without a complex evaluation logic but just checking the ranges of possible targets for the system. There is no separate command and control system between search radar and track & guide radar because Stage procedures already handles the target detection and assignment as opponent. There is no Stage entity which simulates the RWR or no procedure that simulates a change in radar frequency and also handles it on the plane side; therefore, SAM system sends Normal, Search, Track, Launch events to the plane to simulate the real world scenario. The detailed behavior of SAM system can be seen at Figure 9.

Mission Context

Database: default   Scenario: Platform:

Task Group	Type	Logical Oper.	Left Term	Oper.	Right Term	Action
Init	INIT					Track Cycle On
Search	AT		SELF EVENT	=	TRACK_NEXT	
		AND	Reason	=		
	IF		Track Hostile	=	true	Send SEARCH TO PlaneAgent
		AND	Track Range	<=	99000 m	Report
						Opponent Assign From Track
						SUSPEND Track Cycle On
Track	IF		Opponent Present	=	true	Auto Aim At Target_Weapon
		AND	Opponent Range	<=	45000 m	Report
						Send TRACK TO PlaneAgent
Launch	IF		Is Locked On Target	=	true	Report
		AND	Is Auto Aiming	=	true	Weapon Launch By Type
		AND	Opponent Present	=	true	Send LAUNCH TO PlaneAgent
		AND	Opponent Range	<=	40000 m	
Start Searching Again	IF		Opponent Destroyed	=	true	RESUME Track Cycle On
		OR	Opponent Range	>	45000 m	Drop Opponent
Normal	IF		Track Range	>	99000 m	Send NORMAL TO PlaneAgent
Track After Launch	IF		Opponent Locked	=	true	Send TRACK TO PlaneAgent
End	END					

Figure 9: SAM Site Mission- Mission Editor

4.2.4 Interface between Soar and STAGE

Determining how to mimic intelligent behavior in dynamic situations that need domain knowledge is a difficult task. It is far too costly to create a vehicle or robot for the purpose of operating in environments, as well as the cost of developing sensory hardware which takes one's emphasis away from behavior modeling. An ideal testbed for this type of work would be a simulator, which would address these issues. This simulator should create a vivid, high-fidelity world where pilot behavior may be modelled without simulation artifacts. STAGE is a commonly used simulation engine on military field since it allows researchers to concentrate on improving plausible agent models, rather than on difficulties related to vehicle simulation, such as motion dynamics and Distributed Interactive Simulation (DIS) networking. Although there are several cognitive abilities required to properly mimic agent behavior, integrating these abilities into a cohesive whole is best facilitated by the unifying approach of Soar.

The STAGE simulation environment is interfaced to the Soar kernel with the help of Soar Mark-up Language (SML). The simulation environment consists of a fixed-wing entity (F-16) and Air Defence System entities. The fixed-wing entity is controlled by a Soar Agent. The Soar kernel is capable of developing and maintaining the agent which can have its individual behavior based on the Soar production rules loaded in that agent. Both systems have schedulers, but only one is responsible for managing primary scheduling, STAGE should call on Soar when required. The sensory inputs received by the fixed-wing entity in the simulation environment such as radar displays, vehicle status signals, and visual observations are fed to the Soar agent, and the output commands to control vehicle motion provided by Soar agent are conveyed to the simulation environment using SML within the STAGE user plugins written in C++.

In this study; an additional Reduced Situation Awareness Layer is implemented. By means of this layer a distortion in warning signals and a distortion of the visually acquired incoming missile information is simulated, according to given randomness level. This way, the behavior of agent will be observed in the cases of incomplete or inaccurate information to simulate a context of reduced situational awareness.

4.3 Model Evaluation

There are different evaluation techniques for cognitive models:

- Wray and Chong [62] distinguishes between cognitive models and human behavior models. According to their differentiation, our modelling effort belongs to human behavior modelling. Therefore, they suggest that the human behavior models should be evaluated via asking subject matter experts (SME) to review system behavior and evaluate it according to their understanding of the task. As an example, Merk and Roessingh [63] used this evaluation method for evaluating their agent models. They ran several scenarios with the agents, and asked questions to SMEs about the performance of the agents.
- Arrabales et al. [64] evaluated FPS agents' human-likeness according to a scale called ConsScale. The levels of this scale are: 1(decontrolled), 2 (reactive), 3 (adaptive), 4 (attentional), 5 (executive), 6 (emotional), 7 (self-conscious), 8 (empathic), 9 (social), 10 (human-like), and 11 (super-conscious).
- Yule et al [65] suggest that one should use simulation to evaluate a model based on a pre-existing theory. If the target behavior is consistent with the theoretical assumptions, it is sufficient. Otherwise, the modeller should again use simulation to discover the definition of the problem, the possible solutions, and existing theory.

In compliance with Yule et al, we use simulations to evaluate our model as modellers to check whether the target behavior is displayed by the model.

4.4 Experimental Design

One fighter aircraft, operated by an agent with an integrated model, and two Surface-to-Air Missile systems, whose behaviors are built using the STAGE Mission Editor, are included in all scenarios. Surface-to-Air Missile (SAM) systems are placed in such a manner that the aircraft will eventually be within their search and tracking range. SAMs will issue a Search alert if the aircraft enters its target acquisition range. A SAM site will issue a Track Warning first, and then a Launch Warning when it is ready to fire, which will result in the launch of a missile that will kill the aircraft if no action is taken by the agent. To make things as simple as possible, the agent will perform a right-handed 90-degree maneuver to avoid the missiles.

The Radar Warning Receiver is not functioning correctly, and the incoming signals are being distorted with a probability of 12 percent. Similarly, the agent's visual perception is altered with a probability of 12 percent. (Though it is a high likelihood of distortion, the probability has been set deliberately to be high in order to witness the impacts of the distortion more often.) If the agent receives an unexpected warning or if the agent loses an unexpected warning, the agent will be surprised. For the purpose of simplicity, the agent will perform a 45-degree turn to the left before continuing.

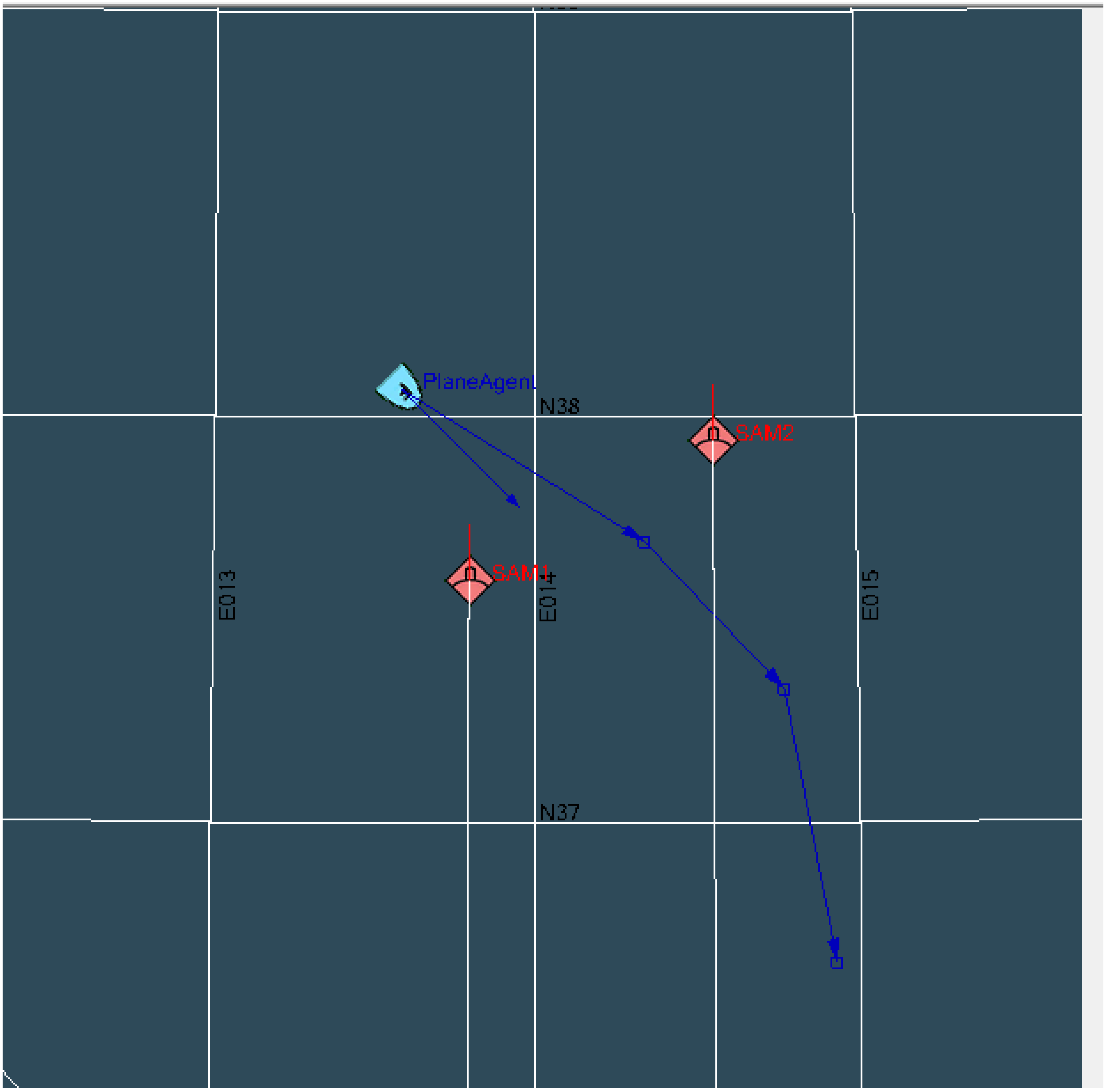

Figure 10: Scenario Scene

4.5 Results

To illustrate the findings, Figures 11 through 13 are the screenshots of the STAGE simulation environment executing the specified scenarios. The blue aircraft, controlled by the Soar agent, departs from the west and attempts to reach the southeast target. The blue line represents the fighter aircraft's actual flight route. The black line represents the aircraft's flight path as determined by external influences.

Figure 11 shows how the Radar Warning Receiver operates without distortion. At point A, the agent gets a Search Warning and enters the Search state. When the agent reaches point B, it gets a Track Warning, enters the Track state, and performs the evasion maneuver to avoid missiles. The agent is able to escape away from the Tracking Range with the assistance of this maneuver, and the SAM system is prevented from firing a missile.

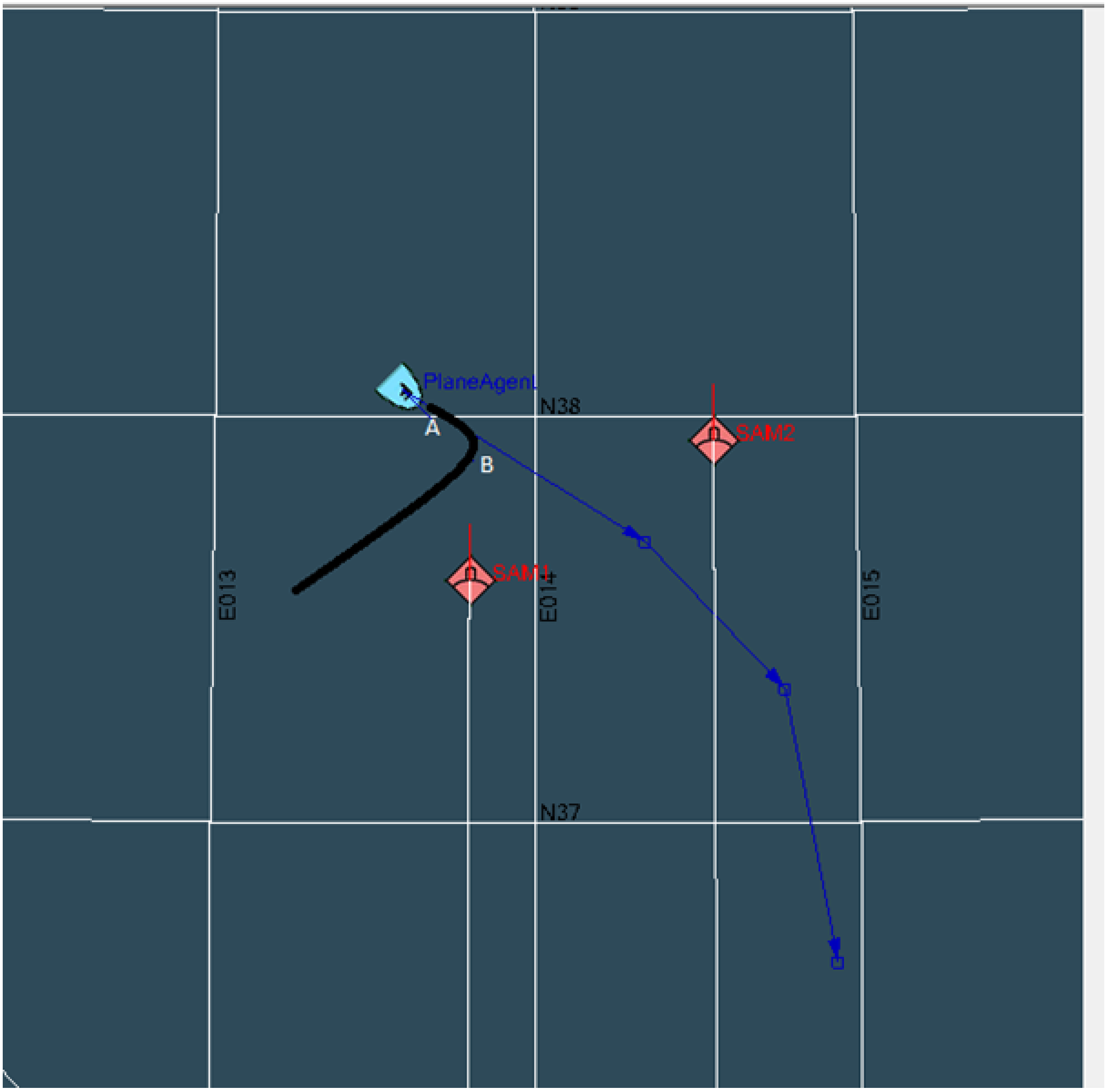

Figure 11: Scenario 1

In Figure 12, it can be seen that the agent got surprised by the distortion of visual perception. At point A, the agent gets a Search Warning and enters the Search state. At point B, the agent's vision is distorted and he thinks there is a missile coming at him. Due to the fact that an incoming missile is associated with launch state, the agent attempts to enter the Launch state. However, it does not seem to be among the expectations of the Search state. As a result, the agent got surprised and exhibit the surprised maneuver.

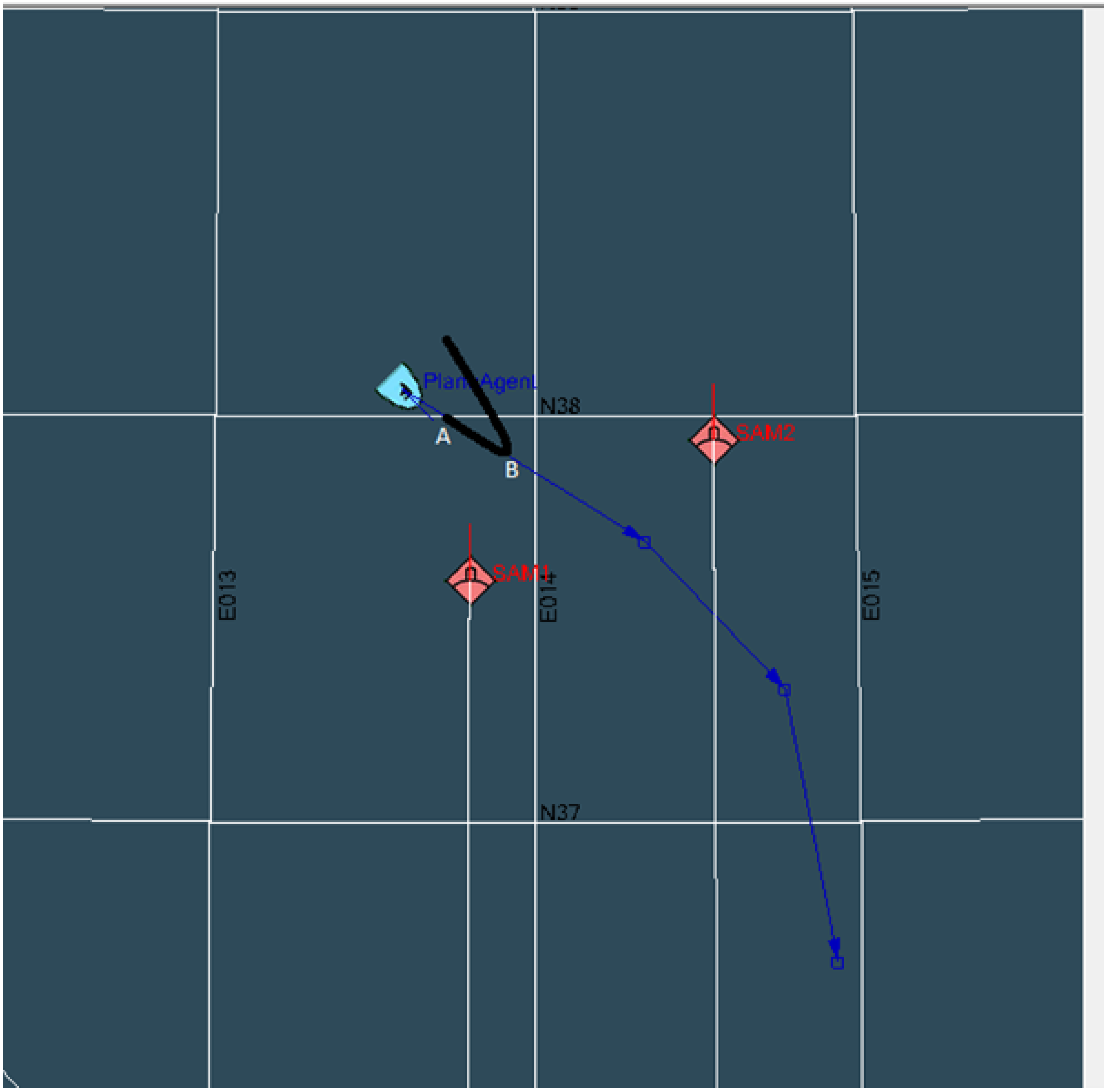

Figure 12: Scenario 2

The effects of Radar Warning Receiver distortion on the agent are shown in Figure 13. At point A, the agent gets a Search Warning and enters the Search state. At point B, the Radar Warning Receiver (RWR) is distorted, and the Track Warning message appears on the RWR screen. The agent enters the

Track state. Track state is one of Search state's expectations. As a result, the agent did not get surprised and engaged in an evasive maneuver. At point C, the SAM system sends the Track Warning, but the agent was already in the Track state, thus it has no effect. At point D, the Radar Warning Receiver becomes distorted and displays a Launch Warning. The launch state is one of the expectations for the Track state. The agent is not interrupted as a result, allowing him to continue with his movement. At point E, the Radar Warning Receiver becomes distorted, causing the Launch Warning to be retracted from the screen. At this time, there is no threat shown on the screen. This is connected to the Normal state and is not among the expectations of the Launch state, as a result, the agent was surprised by the situation. This relates to the Normal state and is not expected in the Launch state, thus the agent got surprised.

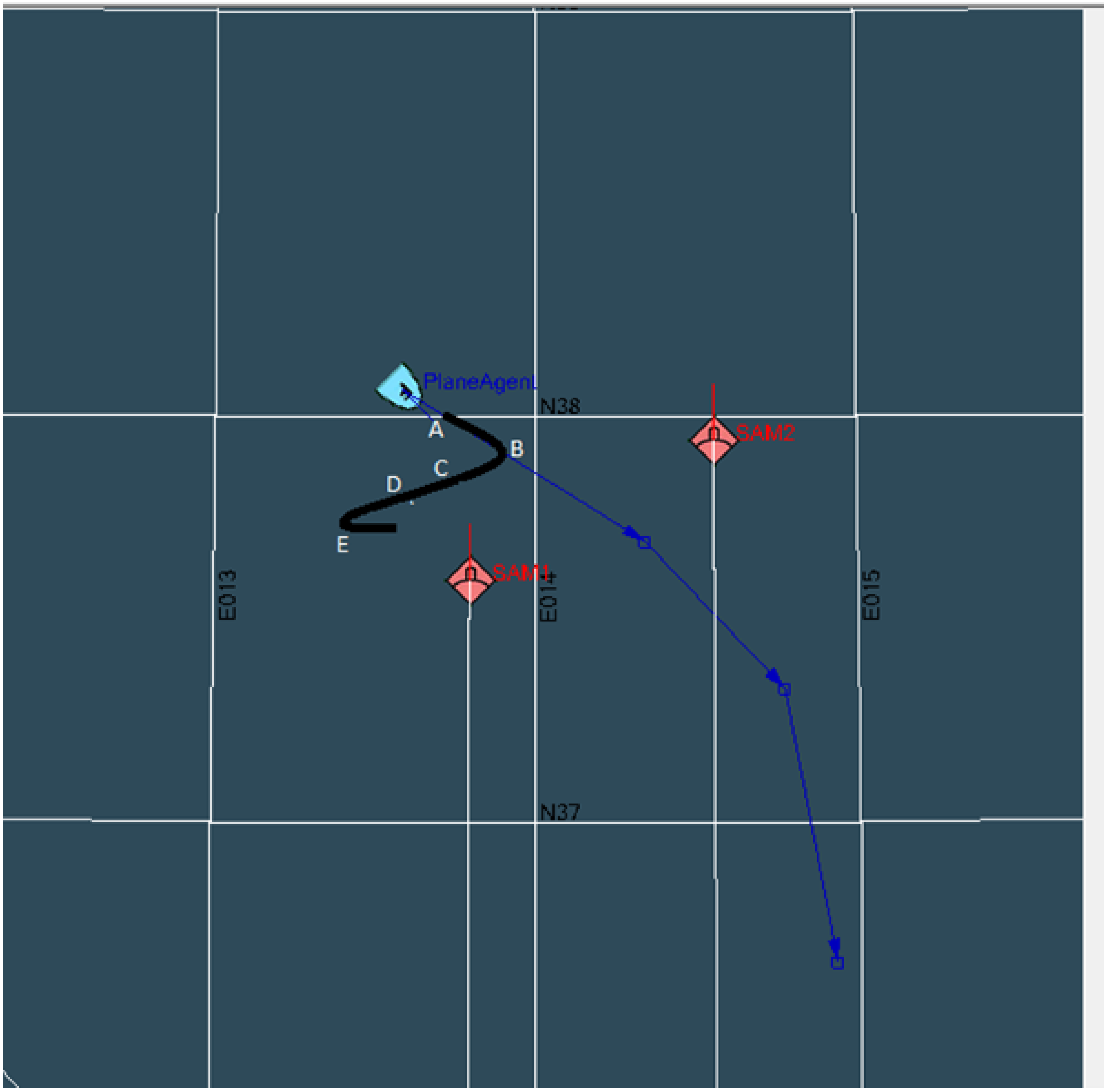

Figure 13: Scenario 3

CHAPTER 5

DISCUSSION, CONCLUSION & FUTURE WORK

5.1 Discussion & Conclusion

Behaviors as possible products of internal cognitive processes are overlooked in the research of Computer Generated Forces (CGFs). This study examines the impact of representations of two crucial internal processes in CGFs depicting fighter pilots, namely (1) Situation Awareness (SA) and (2) the ability to be surprised. For example, a drop in a person's Situation Awareness may have a significant impact on their ability to perform, especially in high-pressure situations. Similarly, when a person is taken by surprise, their current course of action is disrupted. Surprise is a crucial mechanism for a person to identify when SA does not adequately portray reality. The usual cognitive operations slow down in order to preserve SA, such that reactions are delayed as a result of surprises.

To achieve the aim of this research, we have developed a computational agent model, which integrate models for situation awareness and surprise based on psychological theories, using the Soar cognitive architecture in this thesis. The crucial element in the integration of both models is a mechanism that provokes surprise mechanism when the expectations about the future world state(s) are extracted using state information gathered from Situation Awareness rules are violated. The sensemaking has been performed by appending unexpected but experienced states to the known expectations of the last state which the agent experienced the surprise. The situation awareness model is based on Endsley's three-phase theory on situation awareness. The surprise model is inspired by the Macedo& Cardoso's model and the sensemaking part tries to mimic Data/Frame Theory in a simple way. In addition to eliciting particular responses, surprise has an influence on behavior by slowing down the regular cognitive activities required to maintain situational awareness.

The Soar architecture is selected to represent human cognition because of its successful applications in representing human behavior in the military domain (especially the TacAirSoar). Moreover, recognizing patterns in the environment and proposing applicable operators are the characteristic of Soar, and if Soar has sufficient knowledge then it already behaves like Level 1 and Level 2 Situation Awareness, with only one problem that Soar does not allow partial matching of conditions for recognition of a production rule. For the Situation Awareness, Soar's elaboration rules are also facilitated. Some inference rules are used to understand the situation and extract cues for situation recognition, such as if there is a missile launch warning or there is a threat in agent's visual range, it means that there is a missile coming exactly to the agent. In addition, Semantic Memory infrastructure of Soar enables the expectation storage for each situation/state/frame. Soar also supplies the infrastructure of Episodic Memory and Spatial Visual System, which will be discussed in the future work part.

In the scope of the study, an agent is developed to work in a simple simulation environment so that it can be observed how realistic human decision-making affects the outcome of battle simulation games. More realistic human behavior in military simulations can alter the results, and the mathematical and probabilistic solutions for combat modeling can be used to validate the starting point or base line of simulations including human behaviors. Accordingly, the agent is integrated to Presagis STAGE simulation environment, which includes various simple simulation models out of the box, and these models are modifiable and replaceable. The developed model is integrated to the STAGE Simulation Engineer using API. As the agent takes higher level decisions, the STAGE entities implement the command in the simulation environment. The changes in and events coming from the environment is delivered to the Soar agent using STAGE API. The proposed implementation is tested in a way that is similar to how command decisions are made in tactical situations. Experiments are made to see if the proposed implementation is flexible in how it makes decisions, how it behaves, and how it can change. The results show that the model has the potential to make virtual agents more realistic. In addition, results show that manipulating levels of situation awareness and surprise allows the cognitive modeler to come up with a wider range of behaviors without explicitly implementing all the combinations. In all experiments, reduced situation awareness was the reason for being surprised. Although the proposed model is based on widely accepted psychological theories, there is no example to make comparison because there are a few studies to computationally modeling them. The closest attempt was Merk's study, but this study's modelling of behavior is done without using any cognitive architecture, just formalizing and implementing all the steps by the researcher and discussed the results of the simulation experiments made with the agents equipped with these models. In this study, Situation awareness model (belief network) and surprise model (expectation model) were integrated by matching their beliefs about the current situation and their expectations about the world. Matching results in an expectation disconfirmation level, which (when paired with novelty and significance values) results in a certain degree of surprise.

The major contribution to recent developments is trying to represent human behavior in military simulations using Soar cognitive architecture. The last comprehensive study for this purpose was TacAir-Soar [6] and it did not consider surprise and their effects on the capabilities of the agents. Soar cognitive architecture allows greater flexibility and generality, requiring less programming into a symbol system. In order to be more flexible and adaptable, AI systems have shifted away from a sequential execution of a predetermined set of operations and toward a selection-based approach [23]. Soar cognitive architectures have made it possible for us to deal with real-world events by simply establishing frames and associated expectations. There was no need to manage every circumstance. In Scenario 3, for instance, the agent deals with a hypothetical event that may arise in the actual world. An SAM system detected him and launched a missile at him, giving him a Launch Warning. Then, as if there was no danger at all, the RWR suddenly perceives no radar emissions. This makes the agent believe that he is in danger or that something is wrong since it is hard to dodge so rapidly. The agent will be surprised and will attempt to make sense of the situation at some point. Our model does not attempt to implement all of these hypothetical situations, but rather intends to cover them with the logic underneath. According to the model's fundamental logic, being surprised is caused by the consecutive states' expectations being violated. In pure logical sense, returning to a "normal" state from states including threat agents and actions for threat avoidance should look like a positive place to be, and does not imply showing surprisal effect. But in real life it's not the case, unexpected absence and disappearance of threat signals would cause some disturbance in human agents as a reaction to anomaly. In our case, Reduced Situation Awareness disrupts the state sequence, causing surprise. The model's behavior is compatible with the real-world situation, as shown by the fact that agent is surprised when it transitions from the Launch state to the Normal state, and this is an indication that the model is effective in exhibiting complexities of realistic behavior in simulations.

5.2 Future Work

Here are some of the limitations we faced during model development and recommendations that can improve our model in the future. The limitations can be divided to three categories: software limitations, practical limitations and theoretical limitations.

Software limitations are based on the utilization of Stage API caused by the lack of knowledge. With more information and competency, different evasion maneuvers and different counter-measures can be defined, implemented and exhibited depending on factors such as the aircraft's location and speed, the kind of missile, the type of the mission, etc. Moreover, there was no predefined model for Surface-to-Air Missile Systems and Radar Warning Receivers. Therefore, the operational concept of these systems was implemented fundamentally based on the information found. If a complete model for these systems can be developed or found, the agent model may be extended in accordance with it.

Practical limitations are caused by the lack of the representation of real-world scenarios. For example, the suggested model depicts the mental state of a single pilot. When pilots fly missions in the real world, they do it as a group of pilots, and they always have the assistance of ground support forces. It is possible to take into account the information provided by the other pilots and ground support personnel. Another example is that the interaction between the pilot and the surface-to-air missile systems is the main concern of the proposed model. However, in the actual world, the pilot must manage a large number of tasks at the same time, such as aviating the aircraft and completing a mission, as well as monitoring a large number of indicators. These concerns may be regarded as additional frames, and the agent model can be modified to account for them.

Theoretical limitations are caused by the lack of proper and complete theories for computational cognitive modelling. In order to demonstrate human-like behavior, it is necessary to examine the integration of various higher-level cognitive processes, such as learning, planning, theory of mind, and so on. However, the field of computational cognitive modeling of higher-level cognitive processes and their interaction is still in its infancy. The expectation disconfirmation and sensemaking processes are addressed in a simplistic manner by our surprise model. The problem is that there are far too many points that have not been addressed at all. According to the Cognitive-Evolutionary Model of Surprise, surprise is experienced when a particular threshold of schema-discrepancy is exceeded. There is a metric known as surprise intensity to measure the degree of schema-discrepancy, and it is often computed using probabilities, which express the likelihood of an event occurring based on one's expectations. The probability calculation varies depending on the model being considered. In line with the core computational model of surprise that has been selected, the surprise model may be further developed by calculating and employing expectations as probabilities linked with the events or with the situation, and then incorporating them into the surprise model.

At the time this study was done, there was no widely-accepted computational model of Data/Frame theory of sense-making. A "frame" is not defined in a computational manner either. There are only a few things we know about the frame. It can be a set of schemas, mental models, scripts, or data structures that are in long-term memory. The frames in the proposed model are based on the states of a surface-to-air missile system, and they entailed no or only one action, and expectations about upcoming frames. The states of the system are represented by the states of the proposed model. There are a few suggestions for future improvements on these frames:

- The amount of time spent on sense-making has been set as 5 seconds before the frame transition. With the complete implementation of Data/Frame Theory, it is feasible to generate time cost in a natural way. Finding a more appropriate frame, and if one cannot be identified, establishing a new frame and re-framing are all time-consuming tasks in the context of sense-making.
- Data/Frame Theory asserts that the frame determines the data one will seek, and the data determines which frame one will be in. "The data one seeks" indicates that some set of data maps to some frame, and one can only make sense of that collection of data. There are changes made to the frames in a way that changes the data one looks at or the goals and expectations one has for the frame. According to an inexperienced individual, an experienced person's frames are more accurate and thorough since the frames are examined and modified more often by them. In the model, this degree of experience has been reflected as the change in frames, but not in the data one seek. It is possible to do this via the use of the episodic memory and chunking capabilities of the Soar architecture.
- In proposed model, only the Situation Awareness sub-model is responsible for determining a frame shift, and not the Surprise sub-model. This is due to the fact that the presented model simply makes adjustments to expectations in the frames, rather than creating new frames or selecting more relevant frames. The modifications in the model may be considered after the Data/Frame Theory has been fully implemented.

REFERENCES

[1] "Smart Bandits," white paper, Netherlands Aerospace Centre, 2018.

[2] R.-J. Merk, "A computational model on surprise and its effects on agent behaviour in simulated environments," in *Advances in Intelligent and Soft Computing*, pp. 47–57, Springer Berlin Heidelberg, 2010.

[3] A. Landman, E. L. Groen, M. M. R. van Paassen, A. W. Bronkhorst, and M. Mulder, "Dealing with unexpected events on the flight deck: A conceptual model of startle and surprise," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 59, pp. 1161–1172, Aug. 2017.

[4] National Research Council, *Modeling Human and Organizational Behavior: Application to Military Simulations*. Washington, DC: The National Academies Press, 1998.

[5] J. E. Laird, "An exploration into computer games and computer generated forces," 2000.

[6] R. M. Jones, J. E. Laird, P. E. Nielsen, K. J. Coulter, P. Kenny, and F. V. Koss, "Automated intelligent pilots for combat flight simulation," *AI Magazine*, vol. 20, p. 27, March 1999.

[7] G. Nowaczewski, *Gunner's Mate (Missiles) First Class*, 1987.

[8] "RWR - UOAF Codex." Available at `https://codex.uoaf.net/index.php/RWR`.

[9] H. C. Lane, "Cognitive models of learning," in *Encyclopedia of the Sciences of Learning*, pp. 608–610, Springer US, 2012.

[10] N. Cowan, "What are the differences between long-term, short-term, and working memory?," in *Progress in Brain Research*, pp. 323–338, Elsevier, 2008.

[11] B. Dubuc, "The Brain From Top To Bottom - Memory and Learning." Available at `https://thebrain.mcgill.ca/flash/a/a_07/a_07_p/a_07_p_tra/a_07_p_tra.html`. Accessed: 2021-03-21.

[12] M. R. Endsley, "Design and evaluation for situation awareness enhancement," *Proceedings of the Human Factors Society Annual Meeting*, vol. 32, pp. 97–101, Oct. 1988.

[13] K. R. Hammond, R. M. Hamm, J. Grassia, and T. Pearson, "Direct comparison of the efficacy of intuitive and analytical cognition in expert judgment," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, pp. 753–770, Sept. 1987.

[14] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, pp. 32–64, Mar. 1995.

[15] S. Baron, R. Muralidharan, R. Lancraft, and G. Zacharias, "Procru: A model for analyzing crew procedures in approach to landing," tech. rep., National Aeronautics and Space Administration Ames Research Center, 1980.

[16] P. Milgram, R. Vanderwijngaart, H. Veerbeek, and O. Bleeker, "Multi-crew model analytic assessment of landing performance and decision-making demands," pp. 373–399, 1984.

[17] M. Tambe, W. L. Johnson, R. M. Jones, F. Koss, J. E. Laird, P. S. Rosenbloom, and K. Schwamb, "Intelligent agents for interactive simulation environments," *AI magazine*, vol. 16, no. 1, pp. 15–37, 1995.

[18] G. A. Klein and J. Sullivan, "Sources of power: How people make decisions," *Leadership and Management in Engineering*, vol. 1, pp. 21–21, Jan. 2001.

[19] A. R. Damasio, *Descartes' Error. Emotion, reason and the human brain.* New York: Avon Books, 1994.

[20] A. Newell and R. Nickerson, *Reasoning, Problem Solving, and decision processes: the problem space as a fundamental category*, pp. 693–718. Cambridge, MA: Harvard, 1980.

[21] B. Hayes-Roth and F. Hayes-Roth, "A cognitive model of planning.," *Cognitive Science*, vol. 3, no. 4, pp. 275–310, 1979.

[22] US Army Command and General Staff College, Fort Leavenworth, KS, USA, *Student Text 100-9: The Tactical Decisionmaking Process*, 1993.

[23] J. E. Laird, *The Soar Cognitive Architecture*. The MIT Press, 2012.

[24] W.-U. Meyer, R. Reisenzein, and A. Schützwohl, "Toward a process analysis of emotions: The case of surprise," *Motivation and Emotion*, vol. 21, no. 3, pp. 251–274, 1997.

[25] D. E. Rumelhart, "Schemata and the cognitive system," in *Handbook of Social Cognition* (R. S. Weyer and T. K. Srull, eds.), vol. 1, pp. 161–188, Hillsdale, NJ: Erlbaum, 1984.

[26] U. Dompke and K. Scheckeler, "Final report on long term scientific study (ltss/48) on computer generated forces (cgf) technology," *AC/243 (LTSS) TR/48, Brussels*, 1998.

[27] U. Dompke, "Human behaviour representation-definition," tech. rep., NATO C3 AGENCY THE HAGUE (NETHERLANDS), 2002.

[28] S. Downes-Martin, "A survey of human behavior representation activities for distributed interactive simulation," *Client report, Defense Modeling and Simulation Office, Alexandria, VA*, 1995.

[29] J. Laird, K. Coulter, R. Jones, P. Kenny, F. Koss, and P. Nielsen, "Review of soar/fwa participation in stow-97," *Published electronically at< http://ai. eecs. umich. edu/ifor/stow-review. html*, 1997.

[30] J. Gratch and R. Hill, "Continuous planning and collaboration for command and control in joint synthetic battlespaces," in *Proceedings of the 8th Conference on Computer Generated Forces and Behavioral Representation*, 1999.

[31] J. Anderson, *Rules of the mind.* Hillsdale, N.J: L. Erlbaum Associates, 1993.

[32] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: An architecture for general intelligence," *Artificial intelligence*, vol. 33, no. 1, pp. 1–64, 1987.

[33] M. Bratman, *Intention, plans, and practical reason*. Harvard University Press, 1987.

[34] A. Newell, *Unified Theories of Cognition*. USA: Harvard University Press, 1990.

[35] J. E. Laird, C. B. Congdon, M. Assanie, N. Derbinsky, and J. Xu, "The Soar User's Manual Version 9.6.0," 1998.

[36] A. Ceranowicz, "Modular semi-automated forces," in *Proceedings of Winter Simulation Conference*, pp. 755–761, 1994.

[37] R. Wittman and C. Harrison, "Onesaf: A product line approach to simulation development," 2001.

[38] D. Reece, "Integrating soar into the onesaf models framework." Proceedings for the 22nd North American Soar Workshop, 2002.

[39] L. Macedo and A. Cardoso, "Modeling forms of surprise in an artificial agent," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 23, 2001.

[40] L. Macedo, R. Reisenzein, and A. Cardoso, "Modeling forms of surprise in artificial agents: empirical and theoretical study of surprise functions," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 26, 2004.

[41] L. Macedo, A. Cardoso, and R. Reisenzein, "A surprise-based agent architecture," in *Proceedings of the 18th European Meeting on Cybernetics and Systems Research*, pp. 583–588, 2006.

[42] L. Macedo, A. Cardoso, R. Reisenzein, and E. Lorini, "Artificial surprise," *Handbook of research on synthetic emotions and sociable robotics: New applications in affective computing and artificial intelligence*, pp. 267–291, 2009.

[43] L. Macedo and A. Cardoso, "The exploration of unknown environments populated with entities by a surprise–curiosity-based agent," *Cognitive Systems Research*, vol. 19, pp. 62–87, 2012.

[44] L. Macedo and A. Cardoso, "A contrast-based computational model of surprise and its applications," *Topics in cognitive science*, vol. 11, no. 1, pp. 88–102, 2019.

[45] A. Ortony and D. Partridge, "Surprisingness and expectation failure: what's the difference?," in *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, vol. 1, pp. 106–108, 1987.

[46] E. Lorini and C. Castelfranchi, "The unexpected aspects of surprise," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 20, no. 06, pp. 817–833, 2006.

[47] E. Lorini and C. Castelfranchi, "The cognitive structure of surprise: looking for basic principles," *Topoi*, vol. 26, pp. 133–149, Feb. 2007.

[48] J. Stiensmeier-Pelster, A. Martini, and R. Reisenzein, "The role of surprise in the attribution process," *Cognition & Emotion*, vol. 9, no. 1, pp. 5–31, 1995.

[49] G. H. Gendolla and M. Koller, "Surprise and motivation of causal search: How are they affected by outcome valence and importance?," *Motivation and Emotion*, vol. 25, no. 4, pp. 327–349, 2001.

[50] R. Maguire and M. Keane, "Surprise: Disconfirmed expectations or representation-fit?," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 28, pp. 1765–1770, Cognitive Science Society, 2006.

[51] K. H. Teigen and G. Keren, "Surprises: Low probabilities or high contrasts?," *Cognition*, vol. 87, no. 2, pp. 55–71, 2003.

[52] G. Horstmann, "Latency and duration of the action interruption in surprise," *Cognition & Emotion*, vol. 20, no. 2, pp. 242–273, 2006.

[53] L. Mann and C. Tan, "The hassled decision maker: The effects of perceived time pressure on information processing in decision making," *Australian Journal of Management*, vol. 18, no. 2, pp. 197–209, 1993.

[54] U. Neisser, *Cognition and Reality: Principles and Implications of Cognitive Psychology*. Books in psychology, W. H. Freeman, 1976.

[55] G. Klein, J. K. Phillips, E. L. Rall, and D. A. Peluso, "A data-frame theory of sensemaking," in *Expertise Out of Context: Proceedings of the Sixth International Conference on Naturalistic Decision Making*, p. 113, Psychology Press, 2007.

[56] K. Burns, "Integrated cognitive-neuroscience architectures for understanding sensemaking (icarus): A computational basis for icarus challenge problem design," tech. rep., MITRE Corporation, 2014.

[57] R. Thomson, C. Lebiere, M. Rutledge-Taylor, J. Staszewski, and J. Anderson, "Understanding sensemaking using functional architectures," 03 2012.

[58] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," 01 2005.

[59] E. A. Codjoe, *A Finite State Automaton Representation And Simulation Of A Data/Frame Model Of Sensemaking*. PhD thesis, North Carolina Agricultural and Technical State University, 2011.

[60] "DeepMind - Research." Available at `https://deepmind.com/research`. Accessed: 2021-12-06.

[61] "Meta AI." Available at `https://ai.facebook.com`. Accessed: 2021-12-06.

[62] R. E. Wray and R. S. Chong, "Comparing cognitive models and human behavior models: Two computational tools for expressing human behavior," *Journal of Aerospace Computing, Information, and Communication*, vol. 4, no. 5, pp. 836–852, 2007.

[63] R. Merk and J. Roessingh, "Assessing behaviour of cognitive agents in a flight simulator with fighter pilots," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 004842–004847, 2016.

[64] R. Arrabales, A. Ledezma, and A. Sanchis, "Characterizing and assessing human-like behavior in cognitive architectures," in *Biologically Inspired Cognitive Architectures 2012*, pp. 7–15, Springer, 2013.

[65] P. G. Yule, J. Fox, D. W. Glasspool, R. P. Cooper, *et al.*, *Modelling High-level Cognitive Processes*. Psychology Press, 2002.

Appendix A

THE AGENT MODEL

```
sp {propose*initialize-stage-agent
   (state <s> ^superstate nil
             -^name)
-->
   (<s> ^operator <o> +)
   (<o> ^name initialize-stage-agent)
}

sp {apply*initialize-stage-agent
   (state <s> ^operator <op>
              ^smem.command <cmd>)
   (<op> ^name initialize-stage-agent)
-->
   (<s> ^name stage-agent
        ^query-checked no)
   (<cmd> ^query <cue>)
}
```

Listing A.1: Initialize STAGE agent

```
sp {apply*look-for-expectation
   (state <s> ^name stage-agent
             -^surprise yes
              ^operator <o>
              ^last-operator <last-op-name>
              ^smem.command <cmd>)
   (<o> ^name <> <last-op-name>)
   (<cmd> ^query <q>)
-->
   (<cmd> ^query <q> -
          ^query <new-q>)
   (<new-q> ^name <last-op-name>)
   (<s> ^query-checked yes -
        ^query-checked no)
}

sp {apply*look-for-expectation-after-surprise
   (state <s> ^name stage-agent
              ^surprise no
              ^operator <o>
              ^last-operator <last-op-name>
              ^smem.command <cmd>)
   (<o> ^name <> <last-op-name>)
   (<cmd> ^store <st>)
-->
   (wait 5000)
   (<cmd> ^store <st> -
          ^query <new-q>)
   (<new-q> ^name <last-op-name>)
   (<s> ^query-checked yes -
        ^query-checked no)
}

sp {query-retrieved*success
   (state <s> ^operator <op>
          ^smem <smem>
          ^query-checked no)
   (<op> ^name <name>)
   (<smem> ^command.query <q>
           ^result <r>)
   (<r> ^success <q>
        ^retrieved <ret>)
   (<ret> ^{<> name} <name>)

-->
```

```
   (write (crlf) |Hey|)
   (<s> ^query-checked no -
        ^query-checked yes)
}

sp {query-retrieved*failure
   (state <s> ^operator <op>
          ^smem <smem>
          ^query-checked no)
   (<op> ^name <name>)
   (<smem> ^command.query <q>
           ^result <r>)
   (<r> ^success <q>
        ^retrieved <ret>)
   -(<ret> ^{<> name} <name>)

-->
   (write (crlf) |Ney|)
   (<s> ^query-checked no -
        ^query-checked yes)
   (<s> ^surprise yes)
}

sp {surprise*extend-frame
   (state <s> ^operator <op>
              ^last-operator <name>
              ^smem <smem>)
   (<op> ^name surprise)
   (<smem> ^command <cmd>
          ^result.retrieved <lti>)
   (<cmd> ^query <q>)
-->
   (<cmd> ^query <q> -
          ^store <lti>)
   (<lti> ^expectations <name>)
   (<s> ^surprise yes -
        ^surprise no)
   (<s> ^last-operator <name> -
        ^last-operator surprise)
}
```

Listing A.2: Semantic Memory Operations

```
sp {frames*propose*normal
   (state <s> ^name stage-agent
              -^io.input-link.primary-threat)
-->
   (<s> ^operator <op> + =
         ^query-checked yes -
         ^query-checked null)
   (<op> ^name normal)
}
```

Listing A.3: Normal state

```
sp {frames*propose*search
   (state <s> ^name stage-agent
              ^incoming-sam-missile no
             # ^in-sam-range no
              ^io.input-link.primary-threat.search-warning yes)
-->
   (<s> ^operator <op> + =
        ^query-checked yes -
        ^query-checked null)
   (<op> ^name search)
}
```

Listing A.4: Search state

```
sp {frames*propose*track
   (state <s> ^name stage-agent
              ^incoming-sam-missile no
              ^io.input-link.primary-threat <threat>)
   (<threat> ^track-warning yes
         #^in-sam-range-info yes
                            )
-->
   (<s> ^operator <op> + =
        ^query-checked yes -
        ^query-checked null)
   (<op> ^name track)
}

sp {apply*track
   (state <s> ^query-checked yes
              -^surprise yes
              ^operator <op>
              ^io.output-link <ol>)
   (<op> ^name track)
-->
   (<ol> ^commmand <c>)
   (<c> ^action evade)
   #choose evasion maneuver
   #control for the guidance-type of missile
}
```

Listing A.5: Track state

```
sp {frames*propose*launch
   (state <s> ^name stage-agent
              ^incoming-sam-missile yes)
-->
   (<s> ^operator <op> + =
        ^query-checked yes -
        ^query-checked null)
   (<op> ^name launch)
}



sp {apply*launch
   (state <s> ^operator <op>)
   (<op> ^name launch)
-->
   #if did not choose an evasion maneuever, then choose
   #control for the guidance-type of missile
}
```

Listing A.6: Launch state

```
sp {propose*surprise
   (state <s> ^name stage-agent
              ^surprise yes
              ^operator <o> +)
   (<o> ^name <> surprise)
-->
   (<s> ^operator <o> -
        ^operator <op>)
   (<op> ^name surprise)
}

sp {apply*surprise
   (state <s> ^operator <op>
              ^io.output-link <ol>)
   (<op> ^name surprise)
-->
   (<ol> ^commmand <c>)
   (<c> ^action zigzag)
}
```

Listing A.7: Surprise

```
sp {elaborations*elaborate*incoming-sam-missile
   (state <s> ^name stage-agent
         ^io.input-link.primary-threat <threat>)
   (<threat> ^threat-in-visual-range yes
         ^missile-launch-warning <mlw>)
-->
   (<s> ^incoming-sam-missile yes)
}

sp {elaborations*elaborate*incoming-sam-missile2
   (state <s> ^name stage-agent
        ^io.input-link.primary-threat <threat>)
   (<threat> ^missile-launch-warning yes
         ^threat-in-visual-range <tivr>)
-->
   (<s> ^incoming-sam-missile yes)
}

sp {elaborations*elaborate*incoming-sam-missile3
   (state <s> ^name stage-agent
        ^io.input-link.primary-threat <threat>)
   (<threat> ^missile-launch-warning no
         ^threat-in-visual-range no)
-->
   (<s> ^incoming-sam-missile no)
}
```

Listing A.8: Threat Elaboration

```
sp {record-last-operator
   (state <s> ^name stage-agent
             ^operator <o>)
   (<o> ^name <name> <> surprise)
-->
   (<s> ^last-operator <name>)
}

sp {remove*old*last-operator
   (state <s> ^operator <o>
              ^last-operator <name>)
   (<o> ^name <> <name> <> surprise)
-->
   (<s> ^last-operator <name> -)
}
```

Listing A.9: Last operator elaboration