2007

On Advanced Template-based Interpretation As Applied To Intention Recognition In A Strategic Environment

Cameron Akridge
University of Central Florida

ON ADVANCED TEMPLATE BASED INTERPRETATION AS APPLIED TO INTENTION RECOGNITION IN A STRATEGIC ENVIRONMENT

by

CAMERON AKRIDGE
B.S. University of Central Florida, 2005

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Engineering
in the School of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2007

ABSTRACT

An area of study that has received much attention over the past few decades is simulations involving threat assessment in military scenarios. Recently, much research has emerged concerning the recognition of troop movements and formations in non-combat simulations. Additionally, there have been efforts towards the detection and assessment of various types of malicious intentions. One such work by Akridge addressed the issue of Strategic Intention Recognition, but fell short in the detection of tactics that it could not detect without somehow manipulating the environment. Therefore, the aim of this thesis is to address the problem of recognizing an opponent's intent in a strategic environment where the system can think ahead in time to see the agent's plan. To approach the problem, a structured form of knowledge called Template-Based Interpretation is borrowed from the work of others and enhanced to reason in a temporally dynamic simulation.

Dedicated to Him

ACKNOWLEDGEMENTS

My parents and my grandparents have always supported me through this endeavor. I cannot thank them enough for that. I also thank my friend Brian who selflessly helped me through the debugging and optimization of my code (a very stressful experience). I am very grateful to my friend, Jimmy, who let me use his quad-core machine for the testing phase. I also thank all of my friends (Brian, Chris, Jimmy, and James) for their patience as test subjects. Without you guys, this would not have been possible. Finally, I thank my advisor, Dr. Gonzalez for getting me there from start to finish.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

This thesis addresses the application of Template-based Interpretation to strategic environments. It represents an extension of previous work by Akridge (2005), Drewes (1997), and Gerber (2001). Before a complete exploration is done, a description of Intention Recognition and the history of AI in games are provided.

1.1 Thesis Goal

The goal of this thesis is to detect the tactics being implemented by an observed agent in a competitive environment. The environment chosen in this thesis is the game of chess, because of its closeness as a battlefield analogy. The IRS is therefore tasked with the detection of an agent's tactics being implemented during the game. It does so by evaluating not just the current state of the environment, but future states as well by using an enhanced form of a paradigm known as Template-based Interpretation (TBI). This paradigm, devised by Drewes (1997) and later expanded by Gerber (2001) is a form of structured knowledge that has been used in the past to model human behavior. The templates of TBI are used in this thesis to represent the various tactics of the observed agent in the chess environment. Each template structure signifies one chess tactic, though they are not all the same size since some tactics involve more chess pieces than others. A chess tactic could be viewed as anything from a single piece capture to the desire to control a certain part of the board. The enhanced form of the TBI paradigm, known as Advanced Template-based Interpretation (ATBI), considers future states as well as the current one. This concept is discussed in greater detail in Chapter 4.

1.2 Intention Recognition

Conceived by Charles Schmidt (1978), Intention Recognition is the process of identifying an entity's plans or overall intentions. By observing its actions, the observing agent assimilates the individual actions of the entity as progressive phases of the plan, and translates these actions into a type of psychological understanding of the entity's plans. These plans could range anywhere from a basic linear plan of retrieving an object from across a room to more complex intentions such as strategies or plans of attack, which are often nonlinear with several feasible combinations of sub-plans or sub-goals.

Since this area of research began in the late 1970's, there has been a considerable amount of research pertaining to Plan Recognition for agents with malicious intent. Plan Recognition is treated as the same concept as Intention Recognition since the use of strategy involves some form of planning to reach a goal (defeating the opposition).

The problem addressed in this thesis is the recognition of an agent's intentions in environments where the elements of strategy and competition are present. When two entities directly oppose one another, the intention being inferred then becomes the observed agent's strategy. This sort of Intention Recognition System (IRS) is meant to be used to help a user as an intelligent observer. The question of whether an IRS can be used as an effective aid in a competitive environment is investigated in this thesis.

This chapter first defines the base concept of Intention Recognition, and then it defines the general problem being addressed in works pertaining to the topics of Intention Recognition and Plan Recognition. The relevance of the chess simulation to a battlefield is then described.

Schmidt (1978) first defines Intention Recognition as a cross between Psychology and Artificial Intelligence. The implication is that machines must be able to reason like human beings if they are to approximate human capability. A computer must somehow observe a human's actions, convert the input into a meaningful format, then decide how the observations apply to a possible plan. While recognition of plans does not require actions on the part of the observed entity to be physical (movements on a chess board could also be observed), Schmidt (1978) uses them to better explain the mechanics of Intention Recognition. He uses the example of a person retrieving and playing a record from the cabinet. The person starts from some point in a room, presumably without any previously detected actions, and formulates a plan to play a record of his choosing. The computer observing this individual is tasked with analyzing the person's actions, and deducing possible overall objectives as possible outcomes of these actions.

A1. Steve walked to the cabinet.
A2. He opened the cabinet.
A3. He took a record from the cabinet.
A4. He took the record out of the record jacket.
A5. He dropped the record.

The above list details the execution of the person's plan. With the exception of A5, each action further clarifies the entity's intentions of playing a record. Assuming the act of dropping the record was not done intentionally for the purpose of confusing or misleading the observer, the incident may be discarded as accidental, and therefore not be considered in the judgment of intent.

The concept of Intention Recognition has been applied to many different applications. There are two types of application in particular that are relevant to this thesis.

There have been several instances of Intention Recognition being applied as an observer. More specifically, the recognition system acts in the background of the simulator, and plays no essential role in its operation. This observer role could be seen as the base for Intention Recognition, as described in Schmidt (1978). In his example, the Intention Recognition System (IRS) remained invisible to the observed agent (the boy fetching the record). Such a system has been also discussed in the works of Goodman and Litman (1984), Goodman and Litman (1990), Gross (1991), and Ming-Hao et al (2004).

This thesis also examines the use of an IRS as a means of aiding the user of the chess simulation. By using environment data, an IRS could provide valuable information pertaining to agent intentions. The system from Imura et al (1993) approximates user intention as a means to facilitate human-machine interaction with a user interface. In Goodman and Litman (1990), the IRS provides feedback to an engineer designing an industrial plant and offers advice when necessary. In Litman and Allen (1984), the agent’s input is intelligently processed and fed back to him in such a way that solves the agent’s problem. These systems and others are described in greater detail in the next chapter.

1.3 Relevance of Chess Environment

Because the type of environment chosen (the game of chess) is of great significance for this project, a brief discussion of the relevance of chess to real life situations is now presented. While playing the game is not the focus of this project, an understanding of its relevance is still important for a complete introduction.

For example, the chess board can be perceived as a representation of a battlefield. The squares could be viewed as partitions of a large area of some war zone, be it urban, desert, jungle, or any other sort of terrain. A chess board is simply the space in which the pieces can move, just like how the real world terrain is the space in which soldiers move. Of course, the rules of chess do not apply to such environments, since actual soldiers are not limited to the same rigid movements to which chess pieces are bound.

Despite their differences, there are still some similarities between chess pieces and actual soldiers that should be observed. The pieces of a chess game must abide by a set of rules that govern their movement. These rules and all others pertaining to chess are discussed later in the thesis, but for now it is only necessary to understand that each piece has its own set of limitations. Each type of piece (pawn, knight, bishop, rook queen, and king) has its own unique pattern of movement. Actual soldiers in a battlefield are not burdened with such artificial restrictions. They are, to an extent, freethinking individuals capable of fully exploring the environment space in any manner that they wish. Of course, their commanding officer likely limits their freedom of movement to some section of the space, depending on the type of mission that they are on. Also, the type of gear soldiers carry often dictates their behavior to a large extent. For instance, medics are usually equipped with first aid supplies, which limit their capabilities as fighters. Other types of soldiers carry heavier weapons in lieu of such equipment, which makes them more suitable for fighting. Chess pieces behave similarly in that each type of piece has strengths and weaknesses in its rules of movement. In addition, some pieces can be considered more powerful (more valuable) than others. For instance, a queen is far more valuable than a pawn because it possesses much greater mobility. Similarly, an argument could be made that a soldier driving a tank or other kind of assault vehicle is more valuable than a foot soldier, since the vehicle's attack power and mobility surpass that of the foot soldier.

1.4 Summary

While many of the aforementioned papers utilizing Intention Recognition are aimed at assisting their user either by warning against mistakes, or suggesting future actions, none of them choose an environment where the decisions they make are of tactical significance. This problem is further defined in Chapter 2, when the concept of Intention Recognition is fused with strategy.

The remainder of this thesis is focused on identifying the problem being addressed, presenting the ATBI approach that solves it, and validating the resultant solution. Chapter 2 reviews past and recent works and the problems that they address. Chapter 3 uses that information to define the specific problem addressed in this thesis. Chapter 4 defines the method used to solve the problem and presents the mechanics of the IRS. Chapter 5 presents the prototype of the solution, describing the exact functionality of the templates. Chapter 6 presents the validations the system through a series of experiments designed to test the legitimacy of the approach. Chapter 7 reflects on the meaning the test results and from them makes general conclusions and defines some future work that could be done to improve the system's performance.

CHAPTER 2: STATE OF THE ART REVIEW

This chapter goes into depth on the state-of-the-art in Intention Recognition, describing relevant research papers. The similarities and differences between these papers and the work of this thesis are explained. The purpose of this analysis is to better establish the specific problem that is being addressed by this thesis.

2.1 General Applications in Intention Recognition

The following section reviews literature pertaining to Intention Recognition and Plan Recognition in a broad spectrum of applications. From these works, the problem of using Intention Recognition as a helpful advisor is revealed.

2.1.1 Intention Recognition as a Tutor

Conati and VanLehn (2005) use Bayesian networks to fabricate an Intention Recognition System (IRS) that can intelligently tutor physics problems to students through "unsolicited hints" and setting up the problem step by step or "scaffolding" (Conati and VanLehn, 2005). Of course, in order to properly assist the students with hints, the system must first be able to infer the method and reasoning utilized by the students to solve the problem. Once the system has determined the path it believes a student is following, it may then generate appropriate hints to better assist the student (Conati and VanLehn, 2005). Additionally, the authors use a network of plan and action nodes to better map the path that the students are taking, and also to include learning in the system. For example, if a student can demonstrate he/she has developed an understanding of Newton's 2nd Law of Motion from previous uses of the system, then when that student reaches a problem that involves that Law again, the IRS will not force the student to show his/her work for the segment of the problem that concerns it.

2.1.2 Intention Recognition as a Helpful Interactive Agent

The authors extend the base concept of intention recognition from a series of physical actions to a discourse between two individuals, where the one seeking information is the observed entity. In this paper by Litman and Allen (1984), the plan recognizer seeks to determine what kind of information the inquiring agent seeks by intelligently associating recognized terms together to infer what is the next step in the agent's plan of action. A discourse transpires between two individuals, one who seeks knowledge, and the other who possesses it.

The underlying context of this paper is a form of cooperation between the two agents. Neither acts to deceive the other with false information, although the inquiring agent does not always have exact information itself. Still, the two exchange information with one another, building from the discourse a plan that can be used to resolve the problem. Additionally, the recognition system is not hidden from the user as a passive observer, but rather actively participates in the recognition of the observed agent's plans by prompting it for input. In contrast, the system defined in this thesis takes the role of a passive observer, only offering advice when deemed appropriate.

Additionally, the agent's goal is predefined at some level, because the IRS in this case assumes the context of a specific environment when it evaluates the inquiring agent's intent. This project similarly makes its evaluations with the assumption of a certain context.

2.1.3 Intention Recognition with User Interfaces

Goodman and Litman et al (1990) present a "domain independent assessment of plan recognition" as it applies to the enhancement of user interfaces. This paper serves as an expansion from the basic concept recognition of human intent into the realm of active interaction with the user itself. The authors demonstrate how intention recognition can be used not only to infer a person's actions, but to also assist. In an extreme case, the system might even intervene when it believes the user is acting in error.

When the users interface with the system, they perform base-level actions from which the system abstracts high level plans. If it cannot reach a high level of plan abstraction and construct an end node, then the process fails. By acting as an advisor, the plan recognizer can make suggestions of where to place parts or to recognize flaws or errors in the designer's decisions.

This recognition system functions similarly to that of Litman and Allen (1984) in that it operates in cooperation with the same goal as its user. It observes the input of the user and infers intent for the purpose of offering helpful advice. While the user's input is captured by the system, the two are not dependent on one another as in Litman and Allen

(1984). To the contrary, the system is virtually an invisible entity until it can ascertain user intent. Until then, it simply acts as a passive observer.

From this work, the problem of abstracting high level intention from low level action is identified. The interpretation of high level strategic intention could also be achieved from the same thought process. If a strategy could be defined from a series of low level actions or observations, a method for the recognizing that strategy could similarly be devised.

Thus, Litman and Allen's (1984) design is similar to the goal of this thesis, in that it seeks to recognize intent of an agent from a series of base actions. Additionally, the goal of their IRS is to provide aid by acting as a silent observer for the most part. The problem of providing output that facilitates the decision making process that the interacting agent goes through is also one to be addressed.

Using a method that is somewhat similar to that of Goodman and Litman (1990), Huff and Lesser (2000) use multiple levels of abstraction to construct plans that describe a software developer's intentions, specifically the software that they intend to produce. They define a plan as a "hierarchal, partial order of operators (with bound parameters) that achieves a goal given an initial state of the world" (Huff & Lesser, 2000). They also state that there are two algorithms, *planning* and *plan recognition*. The former works with only a goal and initial state to construct a process, whereas the latter infers a plan from a sequence of actions and an initial state. Furthermore, with planning, the system is actively building a goal, whereas with plan recognition, it acts more like a passive observer, looking over the shoulder of the programmer and jumping in only when it spots an error.

This system functions in a similar context as Goodman and Litman et al (1990) in that it operates in a cooperative manner with the user in a productive environment. As before, the agent's actions are passively observed and then abstracted upon to arrive at the agent's high level of intention. The use of abstraction here further prods the question of whether it can be effectively used as a means of approximating high level strategic intent.

Imura et al (1993) address the desire to provide "smooth communication between a user and a system," being such that it does not disrupt the flow of the program interface. They argue that this goal may be accomplished if this system understands the intentions of the user (Imura et al, 1993). They believe in using Fuzzy Set Theory by virtue of its relation between the logic-based world and the real world, and because it allows for a flexible interface. Because of complex knowledge processing, however, Fuzzy Set Theory has trouble with differentiating between meanings of concepts.

Thus, the authors present Conceptual Fuzzy Sets (CFS), which is described as a "distribution of labels that correspond to concepts" (Imura et al, 1993). The labels define the context, which shifts depending on what labels are activated. The "propagation of activations," they reason, "achieves logical operations and reasoning as well as the representation of meanings" (Imura et al, 1993). In other words, when the logic nodes are activated, they form a representation of the user's input, which in turn is abstracted to represent a higher order of meaning or a plan. The distribution of activated label nodes ultimately denotes the user's intent.

The goal of the connectionist logic of (Imura, et al 1993) agrees with that of this thesis in that it seeks to assist a user agent, though not in the same way as with most of the previously reviewed papers. They address the problem of inferring high level intent from low level input, though they do not do so as a means of detecting any sort of strategy. The goal of Imura, et al (1993) is not that dissimilar with that of Litman and Allen (1984), in that the end goal is largely the understanding of user intent with minimal feedback. This thesis, on the other hand, seeks not only to realize the observed agent's intent, but to also inform the user with meaningful results. Additionally, the observations to be made by this thesis's recognition system are of tactical significance and take place in a competitive environment. This setting is in contrast to that of Litman and Allen (1984) which presumably occurs in a noncompetitive setting.

The authors do, however, introduce the idea of grouping similar actions to give them greater meaning. This idea could conceivably be incorporated into tactical recognition by taking similar actions, and associating them with the same strategy.

2.1.4 Goal Recognition with Bayesian Networks

Ming-Hao et al (2004) create an application for intention recognition that does not rely on any predefined library of possible intentions. Furthermore, their system is designed to recognize a plan with imperfect information and to make suggestions should there be room for improvement, which requires the observed agent's cooperation. They define their method as being similar to the paradigm known as Goal Graph, but with two key differences. The first is that, unlike Goal Graph, the authors' approach is capable of solving problems when there are some unobserved actions made by the agent. For example, in the act of a robot picking up and moving some boxes, every individual action would need to be recorded by a goal graph system otherwise it cannot make a complete analysis. Their model on the other hand allows for uncertainty and can hypothesize intention even when there are some gaps in the data. The second difference is that Goal Graph makes the assumption that "all goals explicitly explain the actions observed," meaning every action executed is linked to some known goal (Ming-Hao et al 2004).

They incorporate these differences into their paradigm, called Constraint Parallel Goal Graph (CPGG), a directed mapping of the observed entity's possible plans. This map is composed of three interconnected layers of nodes, labeled proposition, action, and goal layer. Once all viable actions are recorded, and all possible plans are mapped to the CPGG, the algorithm begins a backward-chaining process of validating each goal. For each possible plan on the goal layer, the algorithm moves up a level to determine whether the requisite actions are present to fulfill this plan for the goal, and if so it is presented; else the algorithm moves to the next goal. Like many of the papers reviewed in Chapter 1, this one also uses some form of multi-layered abstraction to infer overall intention from actions.

The application of this method seems limited to cooperative plan recognition where questions may be asked for clarity concerning incomplete information regarding an individual's action. Although this system is capable of operating with incomplete information, it discards any ill-defined actions. Thus, the CPGG method would not be desirable in an environment where the observed agent may not wish to clarify its intentions or even seek to purposefully mislead the observer with inaccurate information.

The authors approach the problem by using a Bayesian Network to integrate a student's knowledge with his/her plans of the available plans (Conati and VanLehn, 2005). This concept relates to Intention Recognition in that it extracts from human subjects the knowledge of their intentions and uses it to inform the system's user of possible future agent actions. The authors' approach differs somewhat from that to be presented in this thesis in that it uses Bayesian Networks, whereas this the research composing this thesis utilizes a more structured form of knowledge retention (Template-Based Interpretation).

2.2 State-of-art in Specific Problem

Having laid a foundation for Intention Recognition that examines the problems it has been applied to, an assessment of relevant works can now be made to narrow the scope of this project's IRS as a helpful observer and warning mechanism. The goal of this section is to provide evidence of a problem with Intention Recognition in tactical and adversarial environments.

2.2.1 Intention Recognition with Hostile Agents

Suzić (2006) describes the application of intention recognition using embedded simulations to support on-line inference. He explains how embedded simulations can be used to model behavior of multiple agents, hypothesize the intentions of each, and from that belief project the most probable course of action that the agent cluster will undertake. His purpose is to use the expected behavior of a rioting crowd to approximate their intentions. He accomplishes the task by incorporating the microeconomic aspects of planning into his plan recognition system, which provides a sort of utility to his method. Suzić recognizes that their strategy is governed at some level by the effect its members have on their economy. The economy in this case is represented by the members' motivation and fatigue levels versus the profitability of their plans. For example, positive factors such as amount looted could be matched against the effort required and risk involved in acquiring it. Similar negative factors could be weighed against any political, civil, or otherwise non-monetary agendas.

Suzić's (2006) purpose is to illustrate that many agents' plans have strong dependencies on their effects. He uses curves that depict the price paid in confronting a conflicting force versus the desired effects achieved. Unlike Suzić's (2006), the work presented in this thesis does not rely on any economic factors as a means of governing agent behavior. However, it does utilize a similar utility approach in that it weighs the positive aspects of a possible plan versus the respective negative effects of its outcome.

The task of weighing positive and negative factors of decision making is a problem that could be researched further in other tactical settings, specifically one where the observed agents are conflicting with other entities in direct combat. Suzić (2006) shows that it is possible to use utility as a means of modeling human intention in a riot scenario, but its application other environments, such as simulated battlegrounds, could be researched further.

Wen-Xiang et al (2005) also addresses the issue of recognizing an opposing agent's hostile intentions, specifically in the realm of computer hacking. They undertake a pseudo neural networking approach to the intention recognition problem, breaking down a person's intentions into a series of actions. Each action is handled as a possible step in a known pattern of attack on some system. These individual actions are further broken into a series of coefficients each represented by some numerical confidence value. Each of these numbers handles some important property of the action, such as the time or the place it is performed. The authors then send these numbers as vectors to the known action library to be compared against known actions and later against known enemy plans.

Additionally, the authors describe a method to counter the attacker's plan by implementing an oppositional planning mechanism. This component acts to either neutralize the attacker's actions to the point that they become harmless activities or to weaken them to a point where the damage caused would be tolerable. These countermeasures are aimed at reducing the threat levels indicated by the action coefficients of the attacker, where a value of '1' represents and absolute threat and a value of '0' indicates no danger. The opposition action may be required to act against only one offensive action, known as the primary action, or it may have to neutralize several, known as secondary action(s).

The authors' use of discrete values to signify the degree of match that an action has to any recognizable hostile activity closely mirrors the methodology implemented in this thesis. The itemization of several numerical attributes to converge on an assumption of an agent's intent strongly resonates with the design of the Template based Interpretation approach, which is discussed later. Additionally, the authors further define the problem of recognizing human intent as an attack on some intangible entity. In this case, the attack is against a computer, though the method of attack described by Wen-Xiang et al is not that dissimilar to one person attacking another. Further work could be done to apply their type of methodology in a different environment.

Another example of hostile agent intention recognition comes from Mao and Lee's (2004) research involving the detection of human attacker attempts on a computer network's security. The goal in this work is to recognize the intention of the adversary based on his actions. Unlike other plan recognition systems in which the human behaves normally without intentionally misleading the observer, an attacker could use dynamic plan patterns aimed at giving the Intention Recognition System (IRS) false information regarding his intentions, thereby creating another obstacle that must be overcome.

Similar to Conati and VanLehn's tutoring model (2005), their system relies on the use of Bayesian Networks to establish a relationship between the various plausible actions of the attacker. It monitors low level actions of the monitored entity, such as querying for an IP address or checking the security system's firewall status, and infers higher order intentions such as the use of a Trojan virus to illegally transfer data to and from the system.

The goal of Qin and Lee's attack recognition system relates to that of this thesis in that it is designed to identify the strategy of an attacking entity. The main difference between the two goals, however, is that the entity being attacked is also an intelligent entity capable of fighting back against its aggressor. The authors help to narrow the problem scope of inferring agent intent by defining the intent as malicious. Furthermore, the agent being observed had a strategy that he was using to accomplish his goal, which further narrows the problem scope.

2.2.2 A Utilitarian Approach to Intention Recognition

Suzić (2005) utilizes a combination of Bayesian Networks (BN) with fuzzy membership functions to create an Intention Recognition system with multiple levels of abstraction. Furthermore, he incorporates models that evaluate the utility of possible plans as a means of determining tactical intent.

Suzić's method is based on Multi-Entity Bayesian Networks (MEBN) consisting of several context sensitive graphical models. Together, these individual "nodes" signify a distribution of probability for each "hypothesis" or possible plan given the current state of the environment, quite similar to the mechanism of Imura et al (1993).

To filter the input sets to the MEBN, Suzić uses fuzzy membership functions to transform the sensor data fed to the system into classes of incomplete empirical knowledge. These classes are, in turn, seen as states in BN node form, which the MEBN compares against its numerous unbound set of hypotheses to determine intent. In other words, before the base or "action" level can be analyzed, the system must first process the raw sensor input and transform it into a meaningful format (e.g. unit movements, positions, etc) that can then be reasoned upon.

The last contribution Suzić mentions is the assumption that a rational agent will choose the plan that realizes the highest overall utility value. Once the system has made its decision of what plans are possible, it must then choose from which are most probable given the outcome of each. This facet of his system is interesting because it uses utility as a means of determining intentions of soldiers on a battlefield, which is quite relevant to the topic of this project. Further study could be done to incorporate the utility of a decision as a means of predicting whether or not it shall be made.

2.2.3 Frame-based Intention Recognition

Gross (1991) uses a frame-based system of reasoning to determine the strategic intent of an opposing team in a game of American football. His methodology can be likened to the device an optometrist uses to determine a patient's prescription. He starts with a shell of a play that contains elements common to all plays, and proceeds to use environment data to define facets of the play step by step. A sample tree of possible paths the system can take is shown below.

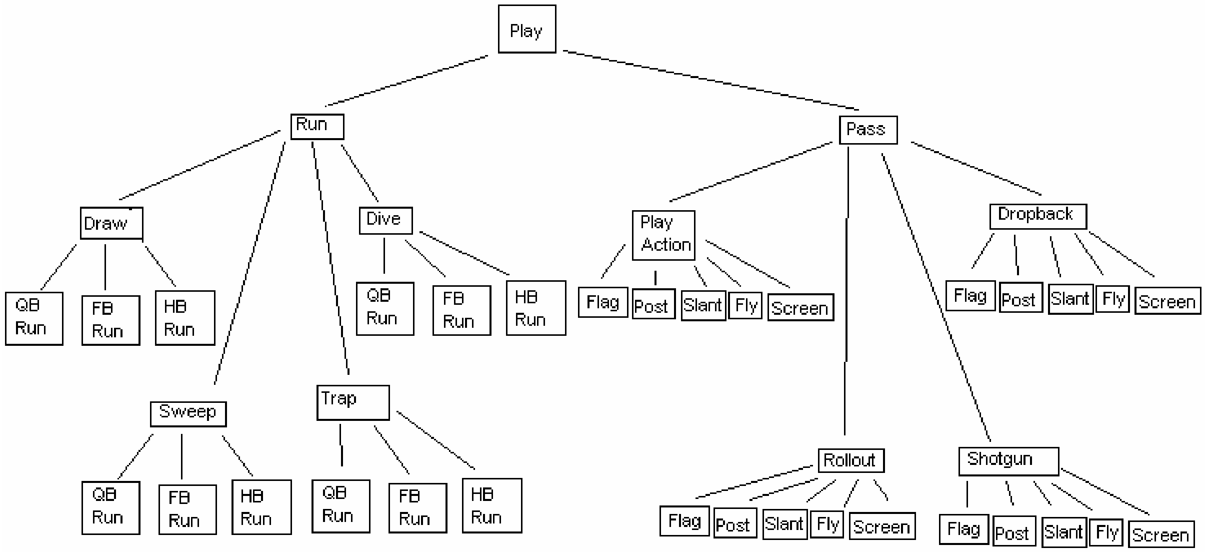


Figure 1: Example of Frame Structure Tree (Gross, 1991)

His goal is to correctly predict the play that the opposing team's coach will choose so that it may be effectively countered. This method is similar to the papers in the previous section that use abstraction to build to a solution, in that the system begins with low level knowledge and intelligently correlates it to agent intent. In Gross's case, environmental parameters include the down, the yards required for a first down, time on the play clock, and other important factors in deciding a play. The difference between his Frame-Based strategy for recognition and the papers in the previous section, however, is that Gross does not generalize agent intent, but rather zooms in to a specific type of play that is to be executed.

Gross identifies a very relevant problem of using intention recognition as a means of recognizing agent intent in an environment of strategic importance. His work shows that the paradigm can be applied to situations where there are two entities in direct competition with one another. This concept could be further studied with the use of a different approach.

2.2.4 Intention Recognition and Behavioral Modeling with Template-based Interpretation (TBI)

Drewes (1997) uses an approach known as Template Based Interpretation (TBI) as a means of identifying the intention of a pilot in training. His paradigm is a structured form of knowledge not too dissimilar to the connectionist logic of Wen-Xiang et al (2005) in that it uses smaller structures within its templates as its relevant data repository. These smaller objects, known as attributes, contain the procedural components called daemons that serve to update the attributes and in turn the templates themselves. Instead of carrying single bits, however, attributes have weights that can affect the confidence of the template depending on the outcome of their daemons. If the daemon returns a positive result to the attribute, then it is "checked-off" or activated, and its weight is contributed to its parent template. Since each template is the representative of a specific intention, its attributes are therefore the facets of that intention it represents.

For example, there may be template for the intention of landing a plane. Its attributes could then be checklist items like distance from the airstrip, status of the landing gear (up or down), altitude, and airspeed. Each of these items could hold a certain percentage of confidence that this template is indeed a representative of the pilot's intent. If enough of the items return positive, then the template "speaks up" as a viable candidate for the agent's intentions.

The system can then decide from among those that feel confident which should be representative of the pilot's real intention. The system can then choose to select only the top ranking template (the one with the highest confidence) or any number of those that meet the minimum level of confidence. This minimum level is typically defined by the system as a global threshold value.

The TBI paradigm is further discussed and expanded in Chapter 4 as the basis used in this project for determining agent intent. Its applicability to the recognition of tactical intent, specifically recognizing intentions in advance, is addressed.

Akridge (2005) lays the foundation for this thesis in his work pertaining to Intention Recognition in environments with mutual aggressors. The unique feature of his work is that it applies to environments where both sides of the conflict have similar resources and goals. His chess simulation embodies this idea, with both sides possessing the same pieces and sharing the same overall goal of capturing the opposing king.

The approach he uses, Template Based Interpretation (TBI), is a structured paradigm where each template represents an encapsulated plan that the observed agent could implement. Each plan, or template, is further partitioned into a group of attributes, each bearing a decimal weight of their own that contributes to the overall strength of the template. This arrangement is not dissimilar the CPGG discussed earlier (Ming-Hao, 2004), although TBI is not a multi-layered paradigm unlike theirs.

While the two approaches are quite similar, this thesis implements previous and possible future actions as factors in the intention detection phase. This concept, called temporal template based interpretation (Gerber, 2001), is described in Chapter 3, and is the means through which this thesis offers its contribution. However, this thesis is founded on Akridge's work (Akridge 2005), and as such there are many areas in their respective methodologies that run parallel.

However, the lack of any knowledge beyond the current state of the environment leaves Akridge's system incapable of seeing beyond the already existing patterns of pieces on the chess board. This deficit rendered the system unable to spot simple tactics that were a single move away from being realized. An augmentation of the TBI paradigm to gain some degree of foresight could conceivably alleviate this problem.

Gerber (2001) expands upon the work of Drewes (1997), by providing a means to automatically adjust the weights of a template's attributes to model the agent being observed. He uses the example of a tank driver in a non-combat scenario, where the system is charged with the task of identifying the pilot's movement tactic.

The weights of the attributes correlate with the skill level of the individual driver operating the tank. His reasoning is that drivers of different skill will naturally act differently, forcing the system to adjust how it correlates aspects of the agent's input to the various known strategies incorporated into its knowledge-base.

The contribution of Gerber to the TBI paradigm and this thesis is the demonstration that weights can be changed dynamically at runtime to improve system performance. A similar tact is described in chapter four, when the approach and adjustment of TBI is fully described.

2.3 Summary

From this chapter, the relevant problems to this thesis have been identified in past and recent literature pertaining to Intention Recognition. The first section presented general applications on the topic, and identified the usefulness of an Intention Recognition System (IRS) as an observer and an aid to a friendly agent. The state-of-art section narrowed the scope of the problem to that of Intention Recognition in environments where agents use some form of strategy as their behavior. From this section the application of an IRS in tactical situations was identified. Additionally, the use of Template-based Interpretation as a means of modeling human behavior was presented. This paradigm is discussed further in upcoming chapters. In the next chapter, the problem addressed by this thesis is explicitly stated as well as the hypothesis and expected contributions.

CHAPTER 3: PROBLEM DEFINITION

In this chapter, the general and specific problem being addressed is presented. A hypothesis describing the goal of this thesis is thereafter established, followed by the contributions that this investigation makes to the state of the art.

3.1 General Problem

Intention Recognition has received much attention over the recent years with research efforts aimed at aiding its user directly rather than acting as a passive observer. Investigations by Goodman & Litman (1990) and Huff & Lesser (2000) offer applications of the early concept of Charles Schmidt (1978) by offering the benefit of an intelligent assistant that provides suggestions and help the user avoid erroneous decisions. Many of the more contemporary research efforts reviewed in the previous chapter relate to the general problem of this thesis closely because the observed agents all act maliciously. In the case of Mao and Lee (2004), the recognition system spots intruders in a computer network that use strategies to attack the stability of the network. Suzić (2005) uses a utility based algorithm that approximates the intentions of observed agents in a military setting. Suzić takes a different approach in a later paper (2006), utilizing the observed agent's local economy as a means of inferring its most probable intentions. These papers all address the common problem of recognizing the plans of mischievous agents. Thus, the general problem becomes the assessment of the threat posed by an agent whose intentions conflict with the observer.

3.2 Specific Problem

The specific problem is presented as both the application of intention recognition in a strategic environment as well as the enhancement of the TBI paradigm. These two problems are now discussed separately.

3.2.1 Strategic Intention Recognition

The specific problem that this thesis addresses is the recognition of opponent intent, specifically when two parties are in an environment where they oppose one another. In the case of Wen-Xiang et al (2005), the agent hacks into computer systems with the goal of disrupting a country's communication capabilities. Similarly, the target of the attacker in Mao and Lee (2004) is a computer network, with the intent of accessing private information. These types of research leave room for recognition in other types of environments, specifically those designed for conflict, such as in the military domain.

Addressing this issue are the previous works of Gross (1991), Akridge (2005), and Suzić (2005). They all demonstrate the existence of a problem with recognizing malicious intent, specifically in environments inherently identified as strategically significant. This thesis, therefore, addresses the specific task of recognizing the opposing tactic of an agent acting against another.

3.2.2 Template-Based Interpretation

Drewes (1997) first uses the paradigm in a temporally static environment (only the present state is observed) with the purpose of modeling the behavior of a pilot trainee.

Gerber (2001) then applies it to a type of environment relevant to this thesis, a military environment, and he also makes it temporally dynamic (multiple states are observed) from a historic perspective. The key difference however, is that neither of them address the concept of two intelligent agents strategically acting against one another. Gross (1991) and Akridge (2005) implement this scenario, though they do so in a temporally static environment, just like Drewes (1997). Problems have been identified in Akridge (2005) that point to his system's inability to operate beyond the present state of the environment. Thus, there arises the need to test the TBI paradigm in a competitive and temporally dynamic environment. This is the specific problem addressed in this thesis.

3.3 Hypothesis

The hypothesis is stated as "The TBI paradigm can be successfully applied to intention recognition in a strategic, temporally dynamic environment."

3.4 Contributions

- A dynamic advanced TBI method enhancement capable of better evaluating opponent intentions in tactical environments
- A prototype system that incorporates the dynamic TBI method. This system can be used by others for further research
- Result data from the evaluation of the prototype

CHAPTER 4: CONCEPTUAL APPROACH

This chapter expands on the method utilized in the solution to the problem stated in Chapter 3. The following sections first describe the TBI paradigm and its enhancement for operating in time as well as space. Finally, a high level concept of the entire system's mechanics is laid out.

4.1 Existing TBI Method

In this section, the existing TBI paradigm that is used to address the problem is described. Additionally, the enhancement of TBI, (called Advanced Template Based Interpretation), is presented. This extension of TBI enables the IRS to search future states in its effort to recognize possible present agent intent as well as strengthen its beliefs of currently recognized intentions.

4.1.1 Introduction to Template Based Interpretation (TBI)

Templates in the TBI paradigm can be thought of as a form of structured knowledge, where each is generally capable of making an estimate of its own fitness in response to inputs from the environment and user. The data collected from the input sets is collected and stored in smaller structures called attributes. These smaller objects act similarly to a node in a neural network in that they hold weight values, and an "active" status that is updated every time an input is presented to the IRS system (Drewes, 1997). Each template can hold any number of these objects, which in turn consider various aspects of the environment when deciding whether their weight quantity should be considered.

These weights then sum up to equal their template's overall fitness. In the next few sections, the template is examined further, with each aspect being more thoroughly discussed.

4.1.2 Attributes

The attributes are the repository for attribute data. Its structure can be broken down into three components: the daemon, the activation flag, and the weight.

4.1.2.1 *Daemon*

The Daemon of an attribute is the procedural component called by the template when it is being refreshed. The function can check any aspect of the environment and return a simple Boolean expression to the attribute. The value returned by the daemon determines the state of the activation flag, which will be discussed next.

4.1.2.2 *Attribute Activation*

The attributes of the template are in a dormant or inactive state until its daemon returns a true value, at which point its activation flag is set to true. The activation of an attribute can be likened to turning on a light inside a clear box, where its brightness varies according to the value of the attribute's weight. As more attributes are activated within the template, more lights are switched on, increasing the overall brightness of the box, thereby making it more visible in the "library" of boxes. Thus, the activation of attributes within each template is the key for its visibility by the template selector. The section on template selection discusses the concept further.

4.1.2.3 *Attribute Weights*

The values of weights within attributes must be proportional to the relative importance of the attribute relative to the others within their respective template. For example, given a person preparing a type of festive meal, there could be a template that represents the intention of preparing it. The individual weights in the attributes would then signify the relative significance of each component of the meal being prepared. Say for instance the IRS observing the person contained templates for a Thanksgiving Day feast, a Christmas Feast, a 4th of July cookout, and a Birthday Party banquet. These templates could then be represented by Figures 2-5 below.

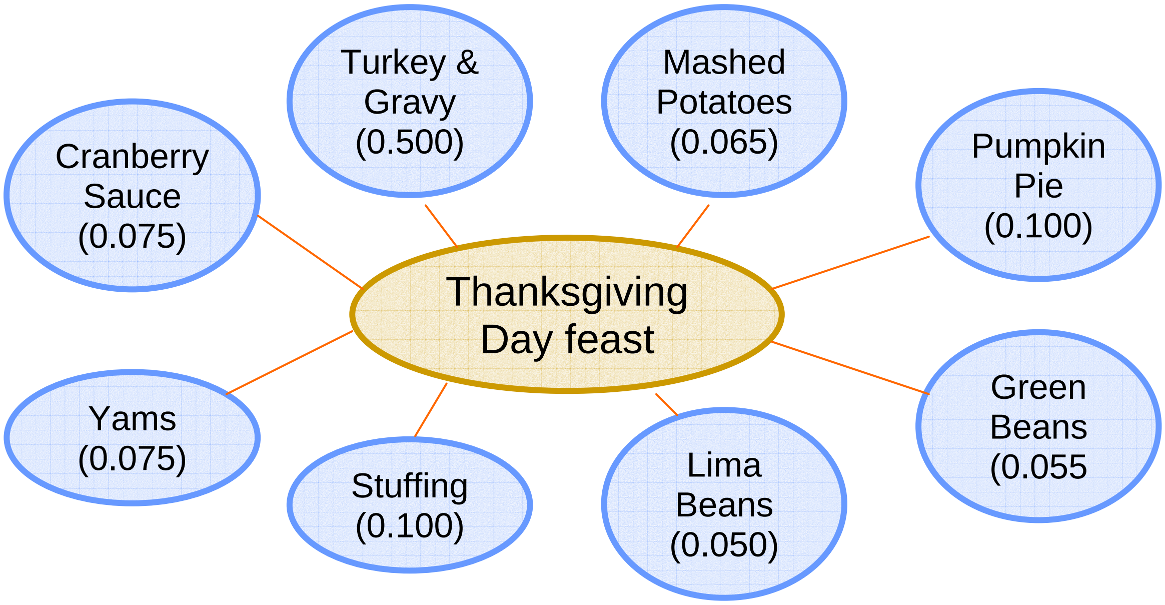


Figure 2: Example template with surrounding attributes and their respective weights

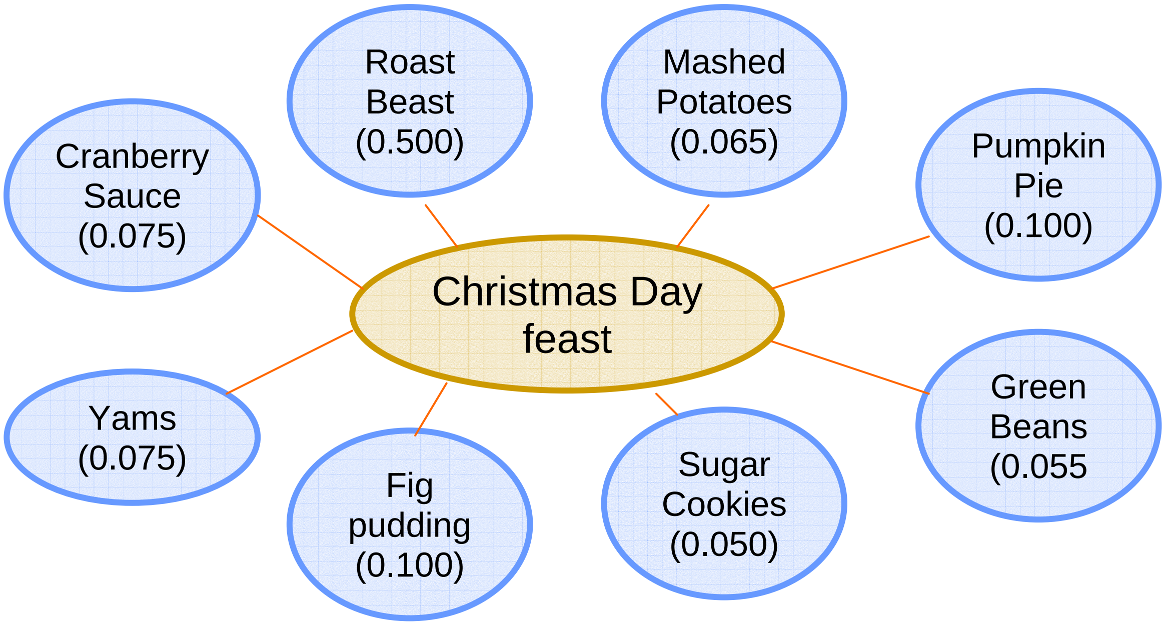


Figure 3: Christmas Feast Template

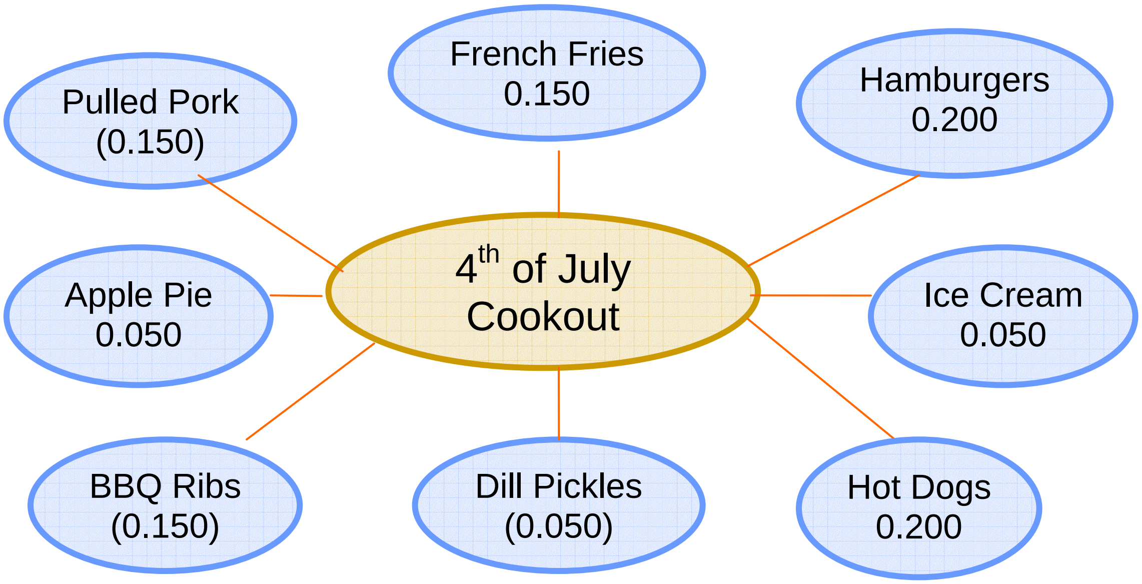


Figure 4: 4th of July Cookout Template

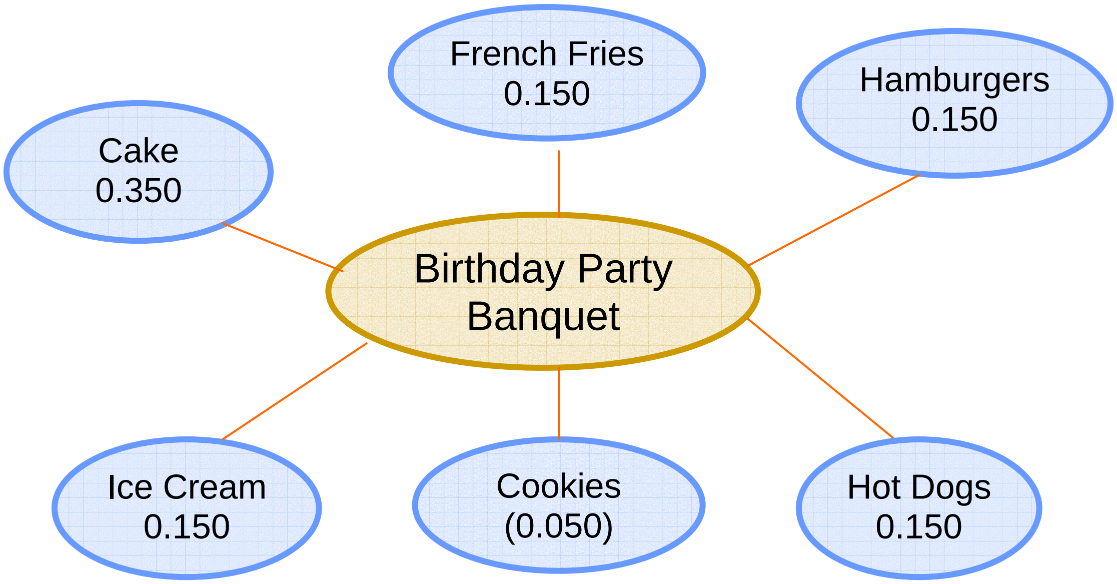


Figure 5: Birthday Party Banquet Template

Now assume that the person is intending to make a Thanksgiving Day Feast. He could start with the potatoes, green beans, and pumpkin pie, but until he gets around to the turkey, the IRS is not entirely certain if he is preparing a Thanksgiving Day Feast or a Christmas Day Feast. When the turkey comes into play, however, the two templates are set far apart in confidence, and the Thanksgiving Day Feast is the clear choice. Note that the turkey carries the largest value of the eight weights. The reason is that the turkey is generally considered to be the component that most strongly correlates with Thanksgiving. Since it is one of few traditional meals that feature a turkey, the chance that the observed agent could be planning something else is not highly likely.

Thus, the weights of the attribute are like shareholders in a corporate shareholder meeting, and the template is the motion that is being voted on. One key difference here, however is that 51% of the votes are not always enough for a majority decision (this concept is discussed later). While every member has a voice, they are not all equal, and the power of one member's vote, say the CEO could dwarf that of the average stockholder and be heard the most. It is true, however, that there may be enough votes to pass or nominate the motion. Again, however, there is an exception that TBI holds from this analogy, in that there are some attributes that are mandatory, meaning they must be activated for the template to be nominated. So for some templates, the percentage of active weights would be for naught if a mandatory attribute remained inactive.

4.1.3 Template Selection: Choosing a Plan representative

Having considered the role of the attribute, the method of selecting a template may be expressed. As aforementioned, attributes act as a source of light to the template library in that they determine its level of visibility among all of those that are also relevant to the agent's observed behavior. The sum of all attribute weights within each template defines their respective strengths relative to one another, though a template has a chance of not being seen at all by the rest of the library should the "light" be too dim. This effect is due to the threshold value set by the template library. The threshold can be thought of as a bar that each template must hurdle if it is to be considered as a viable candidate for selection. The level that each template jumps is measured as the sum value of the weights of its activated attributes. If the collective confidence of one template is not sufficient, the template "trips over the bar," thereby preventing itself from being a candidate for the observed agent's intent.

It is easy to see the benefit of this feature by removing it and scaling the size of the library to some arbitrarily large quantity. If the number of templates is too great, the time required to process all of them for the final selection would be considerable. With the threshold in place, however, the number to be considered is reduced to a select few, making the final decision much more manageable. This way, if the template library is large, valuable time can be saved in the search process by only processing the templates that are in contention. For example, assume there are 1,000 templates in a TBI library with a threshold of 0.70. If only 100 of the templates have an overall confidence level of 0.70 or greater, then the system will only consider those 100 templates for presentation to the user, and can avoid the needless presentation of 90% of the library.

In making its final decision, the system has some options available. It may simply select the template that has the highest confidence rating and present it to the user in some meaningful format. This method is acceptable in many situations. Another option, though, would be to select the top few and present cases for each. This option makes sense, especially if one template is more appealing in a one context than it is in another. The system could then select the strongest templates over multiple cases and present them all to the user as probable agent intentions.

4.2 Advanced Template Based Interpretation (ATBI)

ATBI is the main contribution of this thesis. As mentioned in Chapter 3, ATBI is not limited to operating in the present state of the environment as with Akridge's work

(2005). It must be able to reason from possible future environment states to attain a more fitting representation of the opposition's future plans.

4.2.1 Introduction and Purpose

This thesis offers a modification of the original paradigm by (Drewes, 1997) that takes advantage of environments of perfect knowledge. Given scenarios such as a chess game, future game states can be generated and analyzed by the IRS as it would a current game state. In doing so, searching for a solution is no longer limited to a single timeframe, and may now construct templates that reason not only in the spatial domain but the temporal as well. However, the original model of TBI is designed to function with input sets from a single timeframe. The existing model of the attribute does not allow for the measuring of agent intent over agent actions that have not yet transpired. What must follow then is a high level explanation of how the TBI paradigm is altered so that it may reason through time as well as space.

The inspiration for this approach primarily stems from the future work proposed by Akridge (2005). In his Honors in the Major thesis he concludes that the use of an Intention Recognition algorithm that only searches in the present state for strategies is short sighted. The system cannot see even the simplest of attacks if they are more than one move away from happening. Thus, he reasons that an IRS that is capable of "virtually reaching" into the future to see such attacks would be far more beneficial to one that operates in a temporally static domain. This thesis solves the problem by manifesting an approach to the Strategic Intention Recognition (SIR) problem that allows for the detection of these attacks in advance.

4.2.2 Attribute Adjustment

Each future state read by the system incurs a penalty, the size of which is based on the number of leaps from the present is required to reach that state. This "dampening effect" prevents any possible future intention from being perceived by the recognition system as if it was currently realized. It does so by reducing the active weight value of the template's attributes by fixed percentage with every step into the future. However, the sum of the active attribute weights within each template will still represent the current total of its confidence, and thus the sum is generally significantly less than it would be for a template whose attributes are satisfied in the present state. The reduction of a template's overall confidence thus varies depending on the gap between the present and the time they become active. For many templates, the threshold for candidacy will filter them out due to their reduced attribute weight contributions while others may maintain satisfactory confidence levels.

This dampener value should be kept at a low enough level to allow for future weight contribution to count for a reasonable quantity. Doing so allows agent strategies that still need one or two moves to acquire enough confidence to "speak up" for candidacy and thus possibly be selected by the IRS. However, the value should be kept low enough to block unwanted templates from being candidates. Their future weight contributions should be kept to a level such that if they decide to "speak up" as a candidate for agent intent in a future state, they will not be received as well as one that has already done so in the current state.

4.2.3 Example of Approach Applicability

As a means of explaining this alteration, assume that there is a soldier alone on a battlefield that is equipped with a small computer that sees everything he can see. Now say that this soldier notices in the distance an opposing force too powerful to fight. He notes the different types of enemy units and their formation, but cannot ascertain their intentions quickly enough to decide where to hide. There is not much time to act before he is spotted by the enemy, so he must act fast to remove himself from their line of sight. There are some ravaged structures nearby that offer shelter, but none of them offer complete protection from all possible enemy strategies. If he knew what the enemy's intention was, he could select the spot that best protects him from the opposition, but he only has the time to move to a single location.

Fortunately, his computer has fully scanned the environment and has been virtually testing all possible tactics the enemy could be implementing in several future instances. Equipped with a recognition system that utilizes ATBI, it has processed these future states and reasoned several possible enemy intentions. From here, the soldier can make a more educated decision and move to avoid disaster.

4.3 High Level Design & Algorithm

An algorithm of the overall mechanics of the recognition system is now presented. First, the system components are presented as a flowchart and each module is described. Then, the algorithm for future state generation and template moderation is presented.

4.3.1 High Level Design

As a precursor to the algorithm, a block diagram illustrating a high level design is presented and then explained to the reader.

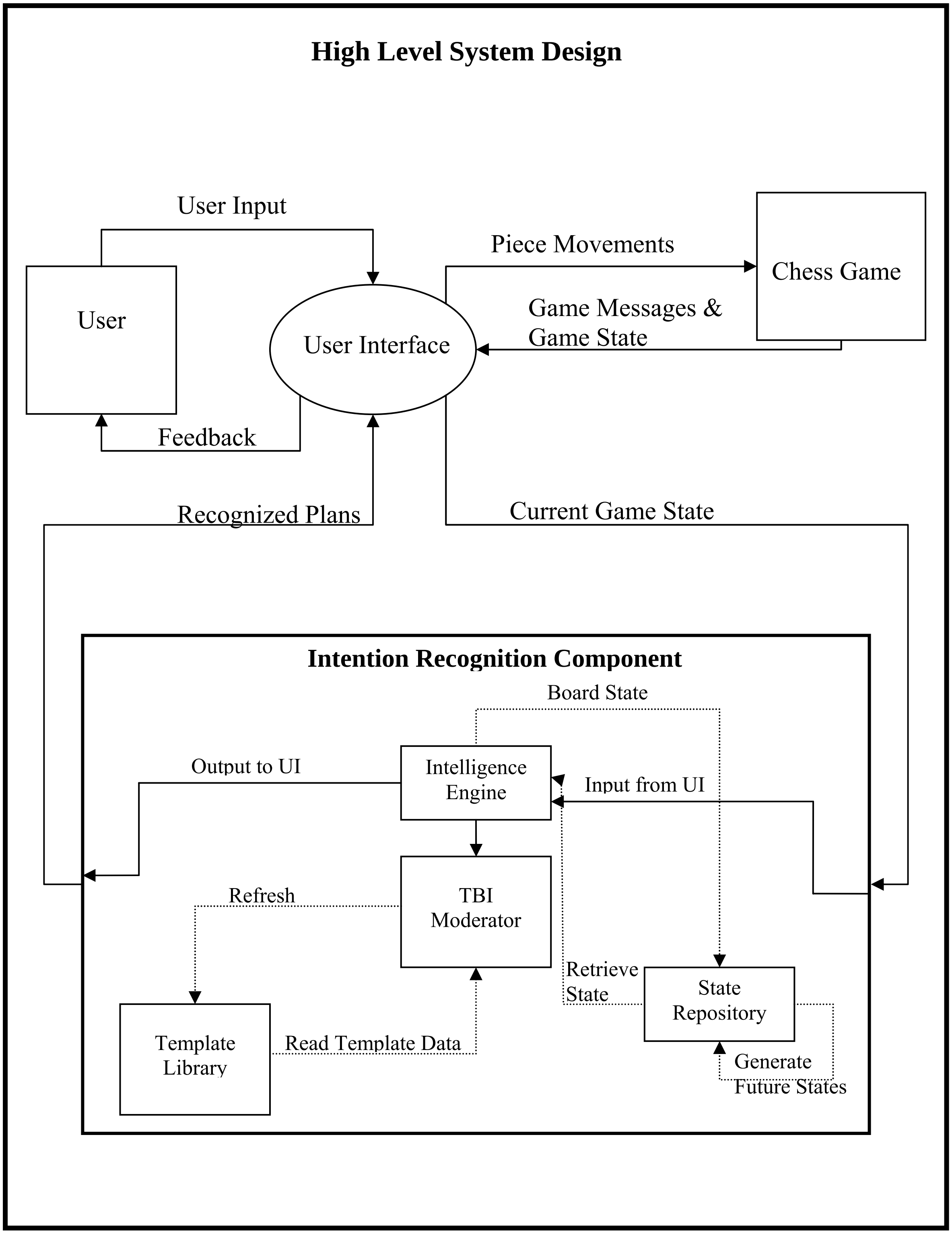


Figure 6: IRS High Level Design

4.3.1.1 *User Interface*

The user interface component serves as the communicator between the user and the system. It initializes instances of both the chess game and the intelligence module. The game is played through this interface, and when a move is recognized by the game component, it sends an updated game state to the intelligence module.

4.3.1.2 *Chess Game*

This module contains the testbed environment of a chess game. When the user makes a move through the interface, it is sent to the game, where all of the rules and regulations of the game are programmed. The game module then decides if the move is in fact legal, and if so, makes a change to the current game state.

4.3.1.3 *Intention Recognition Component*

Considered the overseer of the intelligence component, the Intelligence Engine has its own pseudo UI that allows the user to view the results of the intelligence algorithm. Also, it houses the TBI Moderator, which in turn contains the template library as well as the threshold and dampener settings that govern template candidacy. The Intelligence Engine also controls the production of future states, which are generated using the state passed in from the User Interface. Upon reception of a new state, the templates in the library are reset, and then refreshed for each ply as they are generated. A logic flow diagram is presented next to better illustrate the algorithm.

4.3.2 Intelligence Algorithm

The intelligence engine receives the latest chess board state every time a move is made. Apart from that, it operates independently of the rest of the system. The system begins the process of recognizing player intentions by using the initial board state. All templates are refreshed with the initial board state and the TBI moderator decides whether a template is to be selected for presentation by simply checking to see if their confidence values are above the threshold that the TBI Moderator has set. From the list of legal moves provided by the chess game, the engine then propagates the corresponding number of future states of the board, each reflecting how the state it emanated from would look if the next move had been made.

Also associated with this progression (as mentioned before) is the "dampener effect" that is a factor in determining the net weight yielded by attributes that are activated after the current state. For each template attribute that is activated in a future state, its maximum weight value is reduced by a percentage, the amount of which is predetermined at each ply. The resultant effect is that templates activated in more current states receive a positive bias for selection over than those farther in the future. The amount to be deducted from attributes increments by a fixed rate at the outset of each ply, thus ensuring a linear inverse relationship between the depth of the search and the increase in template confidence.

In compiling the total confidence for each template, the library moderator uses the states that yield the highest value for each template. In other words, given multiple combinations of movements, the recognition system chooses the moves which maximize not just the number of activated attributes for a template, but more importantly the combined weight value for them. The algorithm can be expressed as:

```
Process Template library of N Templates and select one
    FOR EACH template (n) in Template library
        FOR each step into the future or ply (y)
            Sort states of ply (y) with respect to highest confidence for template (n)
            FOR EACH state (s) in ply (y)
                IF confidence (c) of template (n) is above threshold
                    Record ply (y) and confidence (c)
                END IF
            END FOR
            IF confidence (c) is above threshold
                END FOR
            END IF
        END FOR
        Record (c) as overall confidence of template (n)
    END FOR
    Select template of highest confidence with respect to ply
END
```

Since this type of search through the state space may result in a high number of templates over the confidence threshold, templates that cross the threshold closer to the current state are favored greatly over those that do so later. The reason for this dampener (as aforementioned) is to address current enemy intentions rather than those that are not actually recognized yet. Any future intentions of an observed agent have room for change because of events that precede their activation. Agent actions made at the present could very well have an adverse impact on possible tactics that have not yet been fully developed. Thus, the recognition system sorts immediately recognizable intentions from the rest and presents those first, though it also presents those satisfied in the future and notes the minimum number of moves the opposition requires to bring it to fruition.

There is a diagram below illustrating the algorithm that was described in this section. The diagram is then explained to provide a low level understanding of the IRS's process.

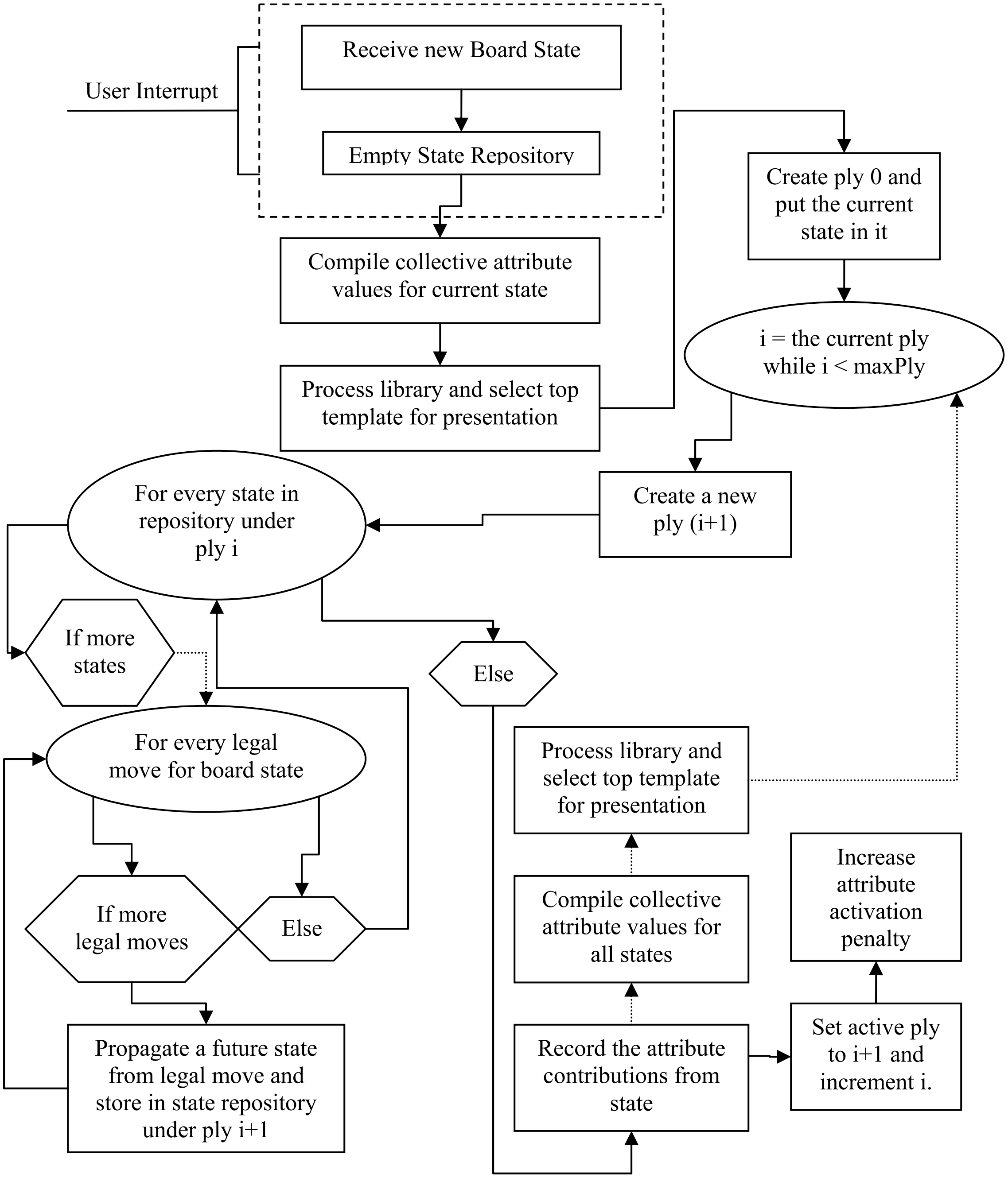

User Interrupt
Receive new Board State
Empty State Repository
Compile collective attribute values for current state
Process library and select top template for presentation
Create ply 0 and put the current state in it
i = the current ply while i < maxPly
Create a new ply (i+1)
For every state in repository under ply i
If more states
Else
For every legal move for board state
If more legal moves
Else
Propagate a future state from legal move and store in state repository under ply i+1
Process library and select top template for presentation
Compile collective attribute values for all states
Record the attribute contributions from state
Increase attribute activation penalty
Set active ply to i+1 and increment i.

Figure 7: Intelligence Algorithm

4.3.3 Low Level Process

When the simulation begins, the Intention Recognition Component is empty. The chess game is first instantiated and the initial list of legal moves is created. When the chess game finishes this process, it creates a game state, adds the legal moves list to it, and passes it to the Intention Recognition Component via the UI. The intelligence engine then begins to operate. Figure 7 above illustrates the low level process that the IRS goes through to convert the current board state into an advanced look at agent intent. The top two boxes are skipped for now since they only come into play when a move is made.

The Intelligence Engine first instructs the TBI Moderator to refresh its templates, and presents the IRS's beliefs for the current state (ply 0). This process was discussed in the original TBI approach so there is no need to mention it again here. Once all templates are refreshed and the results are displayed, the system begins generating the next ply (ply 1).

Every legal move available to the active player is "virtually" made, and the resulting state is added to a list of states that eventually becomes a complete list of game states one move into the future (or ply 1). Once this ply is assembled, the process of updating the template library to account for the future states begins.

Each template takes the responsibility of maximizing its overall confidence level. The high level mechanics of this maximization process is discussed in Chapter 5. All that must be understood now is that each template begins with the current state of its attributes and finds the state in the next ply (ply 1) that provides the largest increase to its confidence. Also, it should be understood that the base value of the dampener (as defined in the TBI Moderator at instantiation) is applied here. The attributes are then updated to reflect the increased confidence and the process repeats for every template in the library.

Once the TBI Moderator finishes updating templates to account for the new ply, the system again displays its results to the user. Also at this time the current ply is incremented (ply0 → ply1) and the dampener value to be applied is increased by a power of 1. In this fashion, the dampener value (n) is n^0 for ply 0, n^1 for ply 1, n^2 for ply 2, and so on. The IRS then loops the process of generating future states from legal moves, only this time there are multiple states to generate moves from. The whole process continues for however many plies the user specifies as the limit.

4.4 Intention Recognition System

Expanding from the high level concept presented earlier in Chapter 4, this section goes into detail with the discussion of the template library moderator. This mechanism can be perceived as the brain of the library. Thus, a thorough breakdown of its role in the system is necessary if a complete understanding of the intelligence engine is to be attained. Following its analysis is an examination of the different templates actually implemented. The purpose of presenting them here is to prepare for their testing and evaluation in the following chapter.

4.4.1 Template Library Moderator

The moderator can be thought of as an employer sifting through resumes. He has the responsibility of deciding from the innumerable candidates which is best suited for the position, just as the moderator must decide from the seemingly countless templates which bets fits the current scenario. Fortunately, each template carries a set of attributes, much like how a resume contains several skills learned by the hopeful applicant. This section expands on the part of the flow diagram of Chapter 4 related to the processing and selection of templates.

4.4.1.1 *Template Processing*

Starting from the current state, the system looks at every future state possible, and imprints the entire library's confidence levels on each. Every template first determines which attributes were not yet activated from a past state or "ply" of movement, and runs their respective daemons. If the Boolean value returned from the attribute's daemon is true, then the future activation dampener is applied, and the remaining weight is recorded for the template at that state and that state only. For other states on the same ply, the process starts over, with each template's change in attribute contributions being marked on each state, until the entire ply has been processed. This concept can be visualized as a tree of state nodes, with the root being the present state. Such an illustration is shown below, with the library consisting of a single template for simplicity. Note how the template gains more confidence with each jump in ply. This is because it accumulates the weight additions from the previous ply as well.

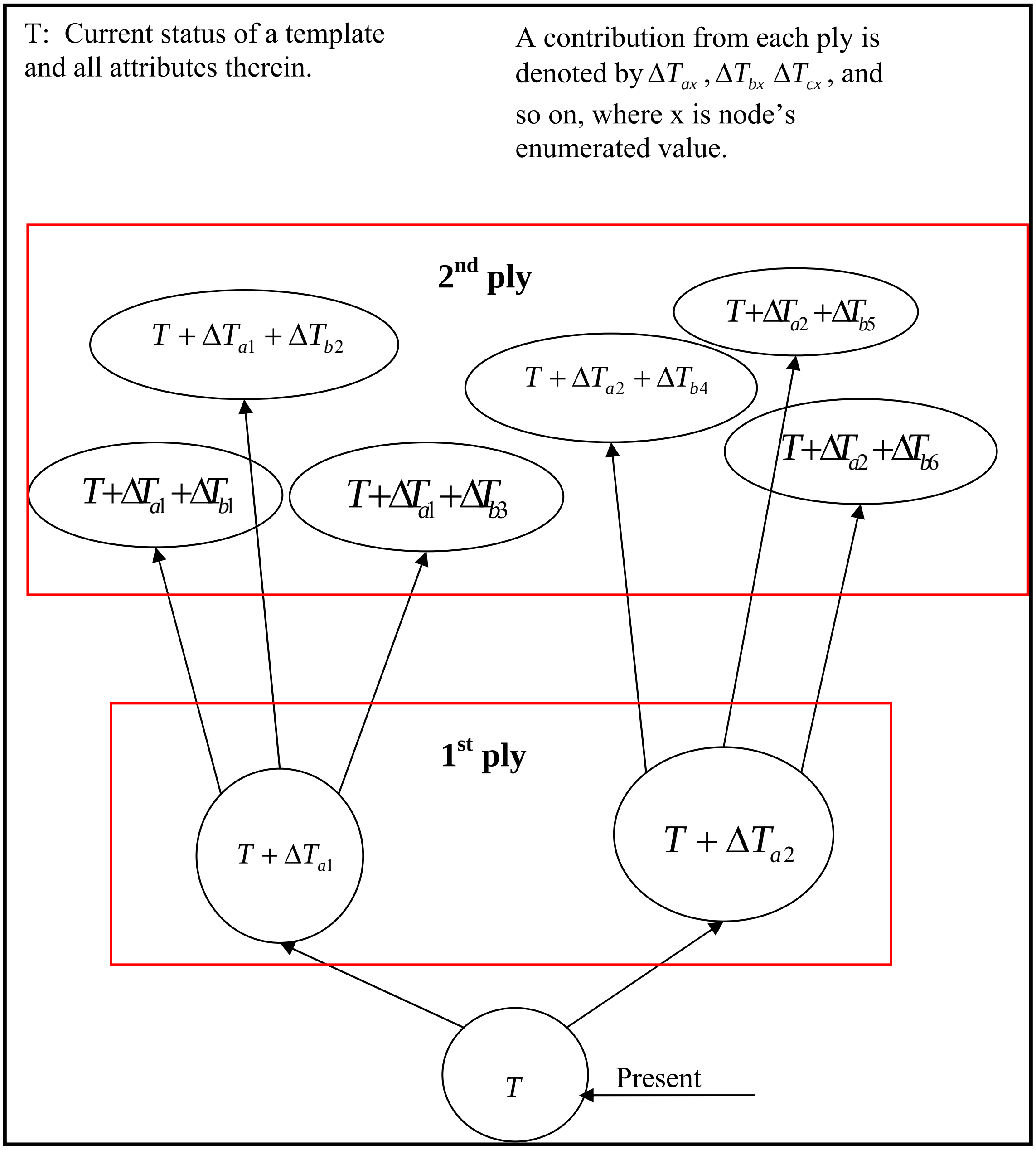

T: Current status of a template and all attributes therein.
A contribution from each ply is denoted by ΔT_{ax}, ΔT_{bx} ΔT_{cx}, and so on, where x is node's enumerated value.
2nd ply
$T+\Delta T_{a1}+\Delta T_{b2}$
$T+\Delta T_{a2}+\Delta T_{b5}$
$T+\Delta T_{a2}+\Delta T_{b4}$
$T+\Delta T_{a1}+\Delta T_{b1}$
$T+\Delta T_{a1}+\Delta T_{b3}$
$T+\Delta T_{a2}+\Delta T_{b6}$
1st ply
$T+\Delta T_{a1}$
$T+\Delta T_{a2}$
T
Present

Figure 8: Template refreshing with future states

Should a template be fully activated at any ply, there are no attribute activation checks for any states further in the future because that template has reached its goal of finding the earliest possible point of complete confidence.

While it is theoretically possible to propagate countless numbers of future states, for practical purposes and limited processing power of the machines used to test the system, the recognition engine will only seek up to the second ply. There are two logical reasons for this decision.

The first is that the confidence added by any strategy that the agent may be implementing far beyond that will be dampened to the point that it becomes negligible. For example, if the dampener was set to 0.5, any template attribute that becomes active in ply 0 has all of its weight added to its template's confidence $\left((1-0.5)^0 = 1\right)$, whereas any attribute activated in ply 1 receives a 50% reduction $\left((1-0.5)^1 = 0.50\right)$, and one activated in ply 2 offers only 25% $\left((1-0.5)^2 = 0.25\right)$. A third ply, however, would yield a negligible 12.5% of the maximum possible weight $\left((1-0.5)^3 = 0.125\right)$. Thus, a third ply does not offer enough weight to merit another step into the future. Appendix C contains a table that further illustrates the insignificance of adding another ply.

Second, the exponential growth in states per ply could mean utter confusion for the template moderator if even a small fraction of them are responsible for another template "speaking up." The proof of the concept of ATBI is believed to be obtainable with 2 ply of search.

4.4.1.2 *Template Selection*

For the purpose of operating as close to real-time feedback as possible, the library moderator performs a reevaluation after every ply is processed. There are two methods that are used as a bias for selection. Should one template break the threshold on the present state, and another do so on a future ply, the former receives a strong bias because of the immediacy of its applicability versus the more uncertain suitability of the latter. Furthermore, some templates hold a relevant importance over others. For instance, suppose there are two templates, both being candidate tactics for selection, one representing an imminent attack on the queen, and the other representing an attack against a pawn. Because of the great value of the queen, the attack on it is by far the most likely, and hence is selected over the push against the pawn. However, if that queen is in jeopardy not in the current state but one ply in the future, the need arises to weigh the relative importance of current versus future states. Here again for the sake of simplicity, the system shall always choose the template of immediate candidacy in such a situation.

Aside from these two biases, the moderator selects the representative template by of simply choosing the one with the highest confidence rating.

4.5 Summary

The reader should now have a low level comprehension of how the recognition component functions, and be ready to understand its high level functionality. The next chapter discusses the implementation of the system prototype. The actual templates utilized in the IRS are also presented as part of the discussion.

CHAPTER 5: SYSTEM PROTOTYPE

The purpose of this chapter is to describe the implementation details of both the chess environment and the implementation of the templates in the IRS. First, the environment simulator is presented, with its functionality described in detail from input to output. Then a detailed analysis of the IRS's templates is done. The individual templates are explained as well as the library moderator that governs their selection and presentation. Communication between the environment and the recognition system is also presented.

5.1 Simulation Environment

While the origin of chess is debatable, the concept it portrays of a struggle between two forces is one that has been observed for ages. It is a game of logic as well as tactics and planning. Just like with any other game, there are rules and boundaries that must be observed. The following sections discuss the nature of the game along with the limitations by which the players must abide.

5.1.1 Introduction

The game of chess takes place on an eight-by-eight grid of squares, 32 of which are initially occupied by chess pieces. At the outset of the game, each player occupies the first two rows on opposite ends of the board. The starting configuration of the board is constant, with the pieces for each side starting in the same positions, as illustrated below.

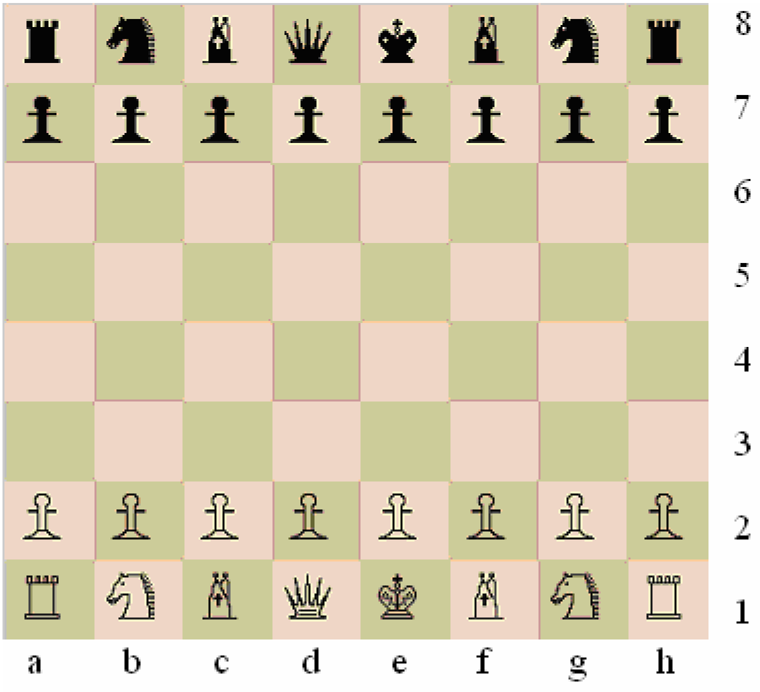


Figure 9: Starting configuration for chess board

Starting with white, the two sides take turns making one move each until one player captures the opposing player's king. Should a player's king be in a position such that an opposing piece could capture it in one move, that king is said to be in check. The controlling player must remove his king from check for the game to continue. If a king is found in a position where no single move may rescue it from being in check, then a checkmate has occurred, and the game is over. Thus, it is easy to see that the capture of the opposing king is a player's main objective of the game aside from protecting his own.

Additionally, a draw (tie) may occur in numerous situations. One such scenario is when both players are left with only their king or one player a knight and his king. Another is when a player who is not in check cannot make a move that does not put his king in check. Both of these conditions are called stalemate, in which case the game is declared a tie.

5.1.2 Chess Piece Behavior

There are six different types of pieces that each player commands. Each type has a point value that can be considered its worth relative to other piece types (with the king being the most valuable). Their descriptions and limitations of movement are now described:

- *Pawn*: Each player controls eight. ♙
    - Located at squares a→h2 for white, or a→h7 for black.
    - The least valued piece of the game, and also the most expendable.
    - May move one square forward or has the option of moving two squares if it has not yet moved.
    - May only attack into its two forward diagonal squares.
    - Carries a value of one point.

- *Knight*: Each player controls two. ♘
    - Located at b1 & g1 for white and at b8 & g8 for black.
    - The only piece that can move directly to a square without a path.

    - May move two squares forward, backward, left, or right followed by one square perpendicular to the previous motion. In other words, its move is 'L' shaped.
    - Carries a value of three points.

- *Bishop*: Each player controls two. ♗
    - Located at c1 & f1 for white and at c8 & f8 for black.
    - Movement is limited to squares of the same color.
    - May move diagonally in any direction any number of squares
    - Carries a value of three points

- *Rook or Castle*: Each player controls two. ♖
    - Located at a1 & h1 for white and at a8 & h8 for black.
    - May move forward, backward, or side to side any number of squares.
    - Carries a value of five points.

- *Queen*: Each player controls one. ♕
    - Located at d1 for white and at d8 for black.
    - May move any one direction for any number of squares.
    - Carries a value of nine points.

- *King*: Each player controls one. ♔
    - Located at e1 for white and at e8 for black.
    - May move any one direction for one square.
    - Carries a value of ten points.

The piece values should provide the reader a ranking of relative importance among the various pieces. They are the key to deciding whether or not to trade captures with the opposition. For example, a player would not wish to exchange his queen for an enemy pawn, because the queen is far more valuable. A piece's assigned number is usually a reflection of its degree of mobility, with the pawn's being the least and the queen's being near limitless. The only exception to this rule is the king, whose value trumps all others. This is because of the pieces immense importance to its controlling player. Should that piece fall, the game will end, and thus its safety is critical.

5.1.3 Special Movements & Rules

While the game operates largely from the rules already set forth, there are certain special moves and rules that may also come into play. These items are listed below:

5.1.3.1 *En Passant*

French for "In passing", this is an optional move involving a pawn during its first movement. If a player opts to move a pawn to squares forward on its first turn, and the opposing player has a piece that may attack the skipped square, then that player has the option to capture the pawn at said skipped square. Figure 4 shows an example of a white pawn performing en passant on a black pawn.

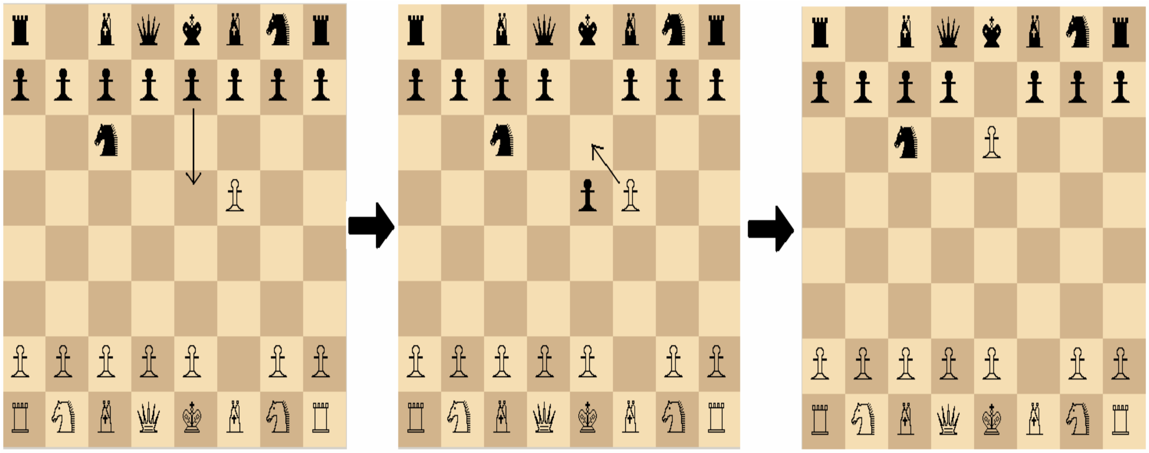


Figure 10: Illustration of en passant

5.1.3.2 *Castling*

Castling is a move involving the player's king and either rook. Neither the king nor the involved rook may have previously moved. Additionally, there may be no pieces in between these two, and the king may not castle to get out of check. An example of a castling move is provided below to provide the reader with a visual understanding.

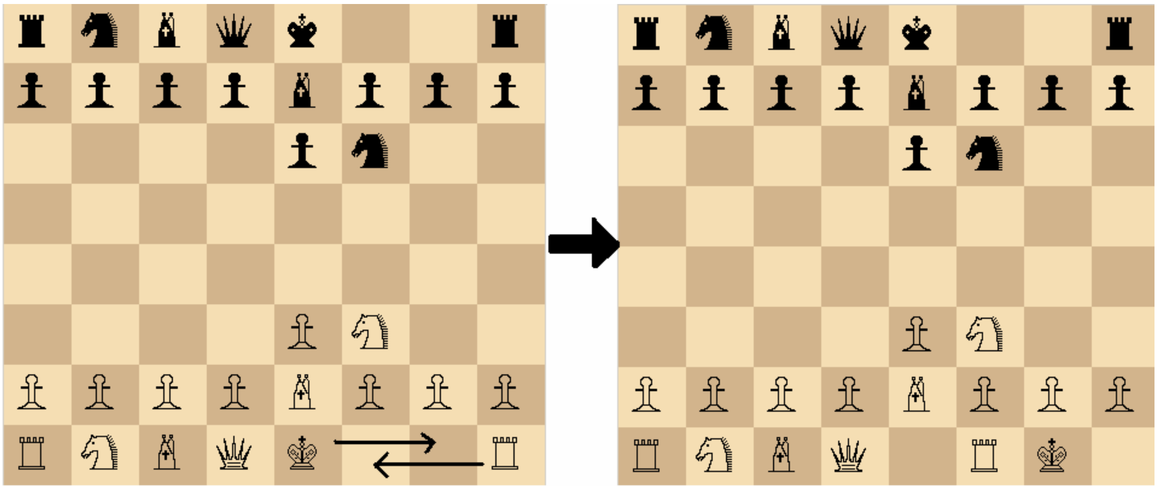


Figure 11: Illustration of a kingside castle

Note that between the two states, two pieces have moved. A castling maneuver is the only kind that allows two pieces to move during one turn. The player's rook slides two spaces towards the king, after which the king is tucked behind the rook. This technique also can be mirrored with the rook on the Queen's side, but can only be done once per player per game.

5.2 Chess Simulator

The components of the chess simulation are now described. To make it easier to understand, the description is decomposed into three sections: user input, processing, and output.

5.2.1 User Input

The player interacts by moving a single chess piece from one square to another according to the rules of chess. When the piece is released onto its destination square, the simulator verifies its legality. It first determines the piece type, the initial square, and the final square, and uses that to determine if the move matches the proper behavior of the moved piece. For example, if the player attempts to move a bishop in an orthogonal fashion, the game rejects the move outright as an invalid move. If the move proves valid, the next phase of testing for the move's legality begins. Here, the game makes the move virtually, and assesses the resultant board state for any rule violations, such as the player putting its own king in check or not resolving an existing check situation. If this aspect of the move proves legal, the game then performs a scan for the endgame criteria of a checkmate, and failing that, a stalemate. If either checkmate or stalemate proves true, the game is declared over. Otherwise, control is turned over to the opposite player, and play resumes.

5.2.2 Legal Move Processing

At the outset of the game, and after each move is made, the chess component runs an algorithm in the background that finds every legal move available to the active player and compiles them into a list. This list is in turn sent to the intention recognition component of the system, which is used to perform the future state searches. A legal move consists of the following pieces of information:

Starting square – The square that the piece moves from

Destination square – The square that the piece moves to

Involved piece – The piece being moved from starting square to destination square

Victim piece **–** The piece at the destination square being captured; does not always contain a piece, in which case this is null.

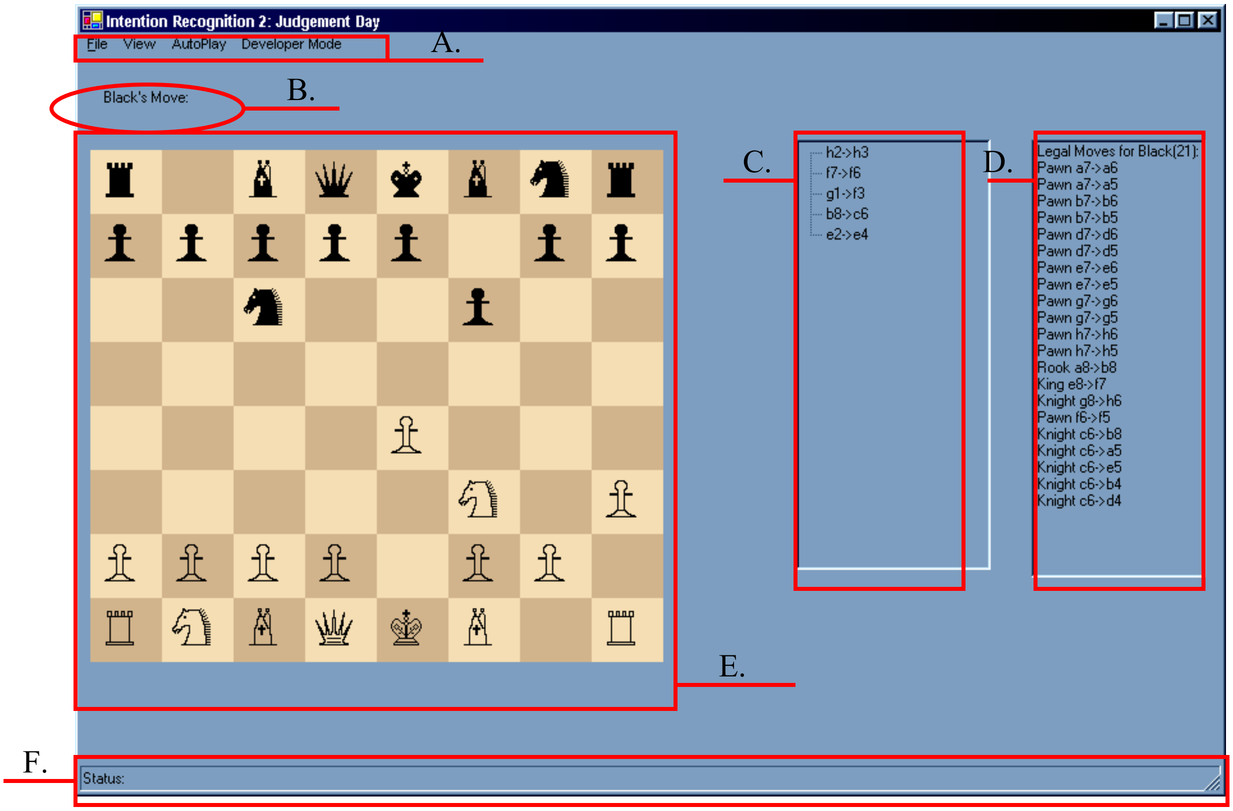
Additionally, the legal move compilation algorithm operates on a separate thread, which keeps the main thread free for all user interaction. The thread runs a function that finds every legal move via the following algorithm:

```
FOR every piece (Pc) that the active player controls and its square (Sq1)
        FOR every square the piece can move to (Sq2)
                IF (Pc) move from (Sq1) → (Sq2) proves legal
                        Create and append legal move to list
                END IF
        END FOR
END FOR
```

When a move is made by the player, the thread is killed and the legal move list is cleared. The process is then restarted for the new board state.

5.2.3 Output

The chess game simulator also provides the user with useful messages and game data relevant to the chess environment (i.e. whose turn it is), though not necessarily with any form of intelligence. Nevertheless, they are mentioned here for the sake of a complete description of the testbed. The figure below labels each item, which are then identified.



```
h2->h3
f7->f6
g1->f3
b8->c6
e2->e4
```

```
Legal Moves for Black(21):
Pawn a7->a6
Pawn a7->a5
Pawn b7->b6
Pawn b7->b5
Pawn d7->d6
Pawn d7->d5
Pawn e7->e6
Pawn e7->e5
Pawn g7->g6
Pawn g7->g5
Pawn h7->h6
Pawn h7->h5
Rook a8->b8
King e8->f7
Knight g8->h6
Pawn f6->f5
Knight c6->b8
Knight c6->a5
Knight c6->e5
Knight c6->b4
Knight c6->d4
```

Figure 12: User Interface

A: *Toolbar* – Provides the user with access to an array of basic functions provided by the chess simulator

- File: Contains basic operations
  - *New* – Resets the board to its initial state and makes white the active player.
  - *Save* – Creates a save file for the current game state that records the current board state and the active player.
  - *Load* – Allows the user to load a previously saved board state.
  - *Exit* – Exits the simulator

- View: Contains binary operations
  - *Game Messages* – hide/show the message box (B).
  - *Legal Moves* – hide/show legal moves list (D).
  - *Status Bar* – hide/show status bar (F).
- AutoPlay: Contains features that allow the two sides to move autonomously. The moves are made by simply selecting a legal move completely at random.
  - *Enable/Disable White* – Controls whether the simulator controls white
  - *Enable/Disable Black* – Controls whether the simulator controls black
- Developer Mode – Allows for any piece on the board to be moved to any square. This option is primarily used to arrange custom scenarios for testing purposes.

B: *Message Box* – Displays game related messages, such as which player is active. It also warns when a player is in check, and acknowledges a checkmate or stalemate.

C: *Previous Moves List* – Displays the moves that have been made thus far in the current game.

D: *Legal Moves List* – Displays every move that can be made by the active player for the current board state. Note that the box remains empty until the list is fully compiled.

E: *Game Board* – The simulated chess board. It contains sixty-four painted squares in a top down point of view.

F: *Status Bar* – Displays any messages related to the simulator but not the game itself. These include verification messages for the saving and loading of game states.

5.3 Template Implementation Introduction

The various templates to be included in the system are now presented to familiarize the reader with the actual contents of the library itself. While some templates are individually and explicitly defined, others are abstract, and can be duplicated many times with the implementation of their abstract components being what makes them unique. All of them are representative of tactics that the observed agent is implementing.

5.3.1 Board Control Templates

The attributes of these templates represent squares can be attacked by the agent's pieces. A square is said to be under attack if a piece can legally move to it within one turn or if a friendly piece resides at the square. For every square in the predefined group that meets this criterion, the template gains confidence. Thus, control over a certain area of the board can be defined as a combination of having pieces on squares in the area and being able to attack (or move to) squares in the area. These tactics represent analogies for the holding of (and attack on) territory during war. Additionally, these tactics can be implemented at any point in the game, thought they are more meaningful towards the beginning, since establishing control of the board is prevalent in most tactics (Esterin & Panov, 1980). Below are individual templates that fit this category and figures that illustrate an example of their proper execution and recognition by the IRS. A complete listing of their respective attributes can be found in Appendix D.

5.3.1.1 *Lower Left Board Control*

The agent creates piece formations with the objective of controlling the lower left of the chess board. Note in Figure 10 below how the queen and two pawns of the white player are positioned in the lower left quadrant of the chess board. The queen can attack many of the squares that are defined as being in the lower left area of the board (the highlighted area in Figure 10), and the adjacent pawns can attack the squares in front of them, which are also considered part of the lower left area.

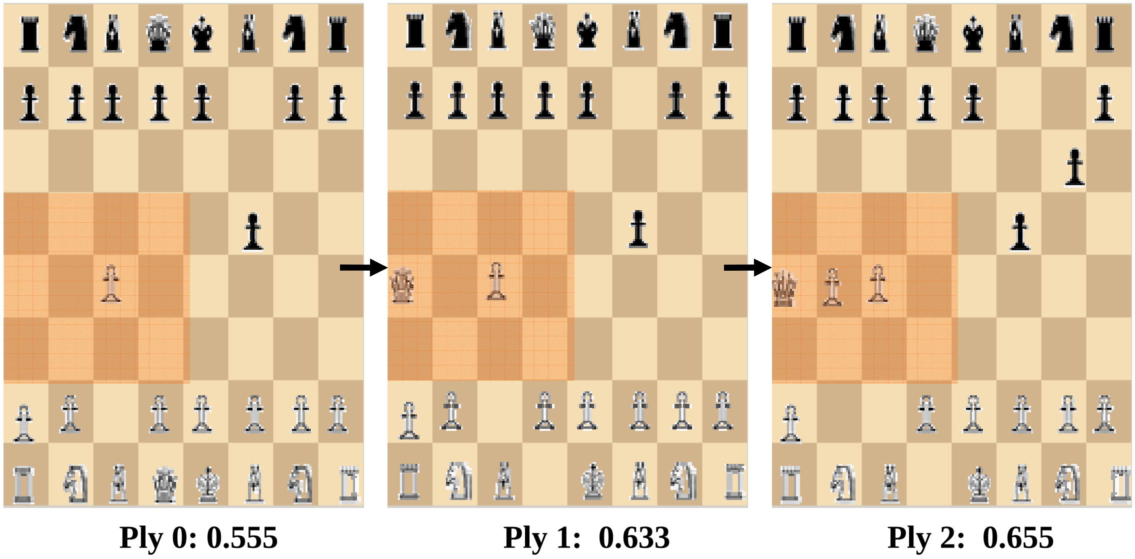


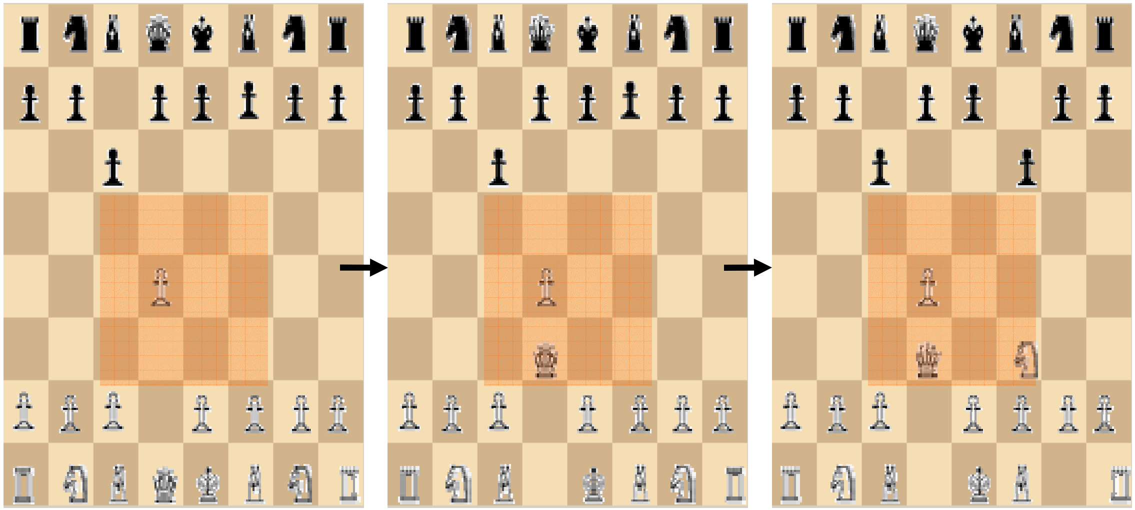
Figure 13: Execution of Lower Left Board Control for ply 0 (Left), ply 1 (Center), and ply 2 (Right), with corresponding ATBI readings

For this example board state in Figure 10, the IRS does not detect the tactic by analysis of the current state (ply 0). However, it does break the confidence threshold when the first future ply is added. Additionally, the example shows the following ply of states (ply 2) adding more confidence to the template.

5.3.1.2 *Lower Center Board Control*

The agent creates piece formations with the intention of controlling the lower center of the chess board. This template gains confidence if the squares near the lower center area of the board are occupied or can be attacked (moved to) by friendly pieces. The area covered by this template is highlighted in Figure 11 below.

In the example below, the white player has its queen and its king's knight available to move to several of the squares in the region. The pawn within the area also has two squares that it could attack.



Ply 0: 0.550 **Ply 1: 0.633** **Ply 2: 0.655**

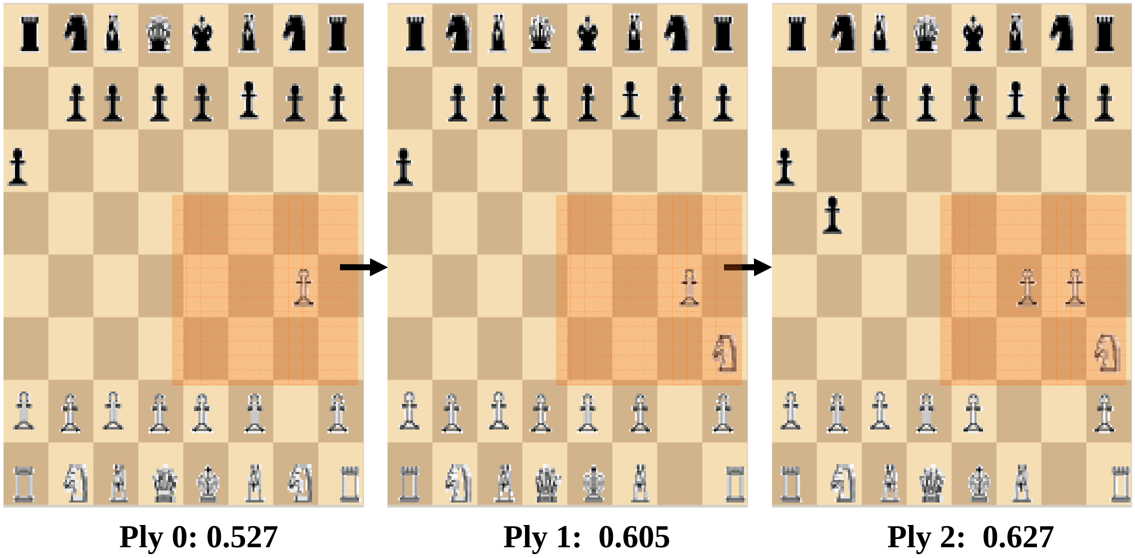
Figure 14: Execution of Lower Center Board Control for ply 0 (Left), ply 1 (Center), and ply 2 (Right), with corresponding ATBI confidence readings

Just like the example for Lower Left Board Control, The IRS finds the template breaking the confidence threshold at ply 1, and increasing it in ply 2. There is also a showing of Lower Left Board Control in ply 2, which is understandable since the two regions overlap somewhat.

5.3.1.3 *Lower Right Board Control*

The agent moves his pieces into a formation that suggests an intention to gain control of the right side of the board. This template is a mirror of the Lower Left Board Control template, with its region being defined on the opposite end of the board. This region is highlighted in Figure 12 below.

In the example board state in Figure 12, the white player has its king's knight out in the template's zone, along with a pawn.



Ply 0: 0.527 **Ply 1: 0.605** **Ply 2: 0.627**

Figure 15: Execution of Lower Right Board Control for ply 0 (Left), ply 1 (Center), and ply 2 (Right), with corresponding ATBI confidence readings

Since the board in Figure 12 has not been opened up greatly (there are not many moves available to the agent), the IRS barely recognizes the tactic in ply 1. The confidence again increases slightly in ply 2.

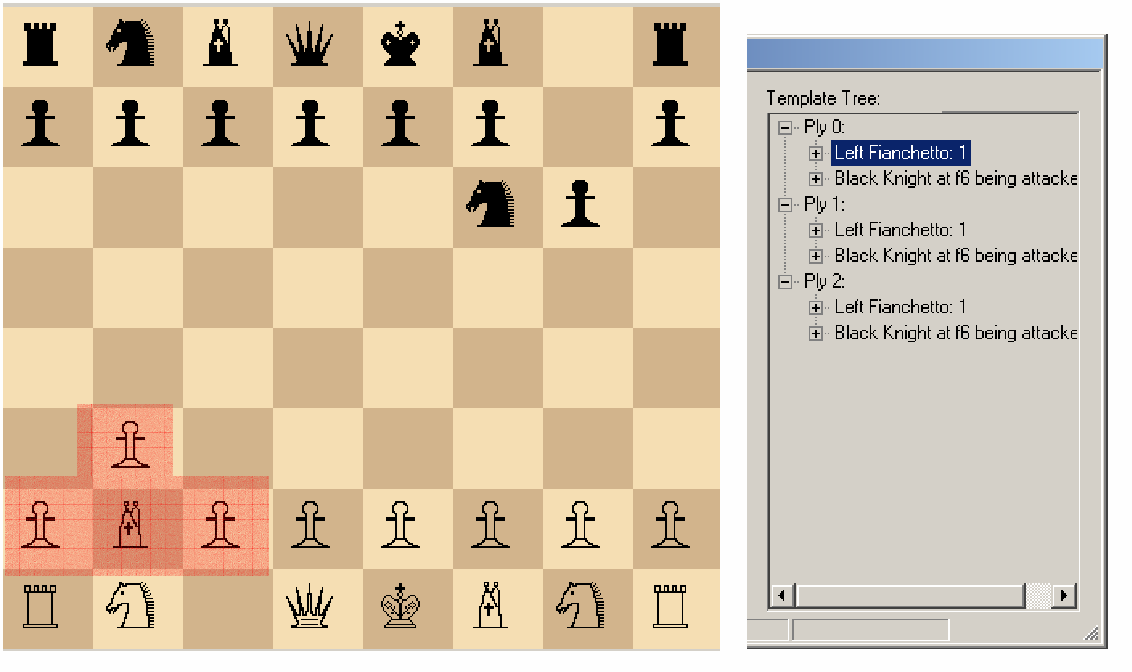
5.3.2 Fianchetto Templates

Used in Akridge (2005), the Fianchetto is a fairly rigid tactic that relies on a player's bishop controlling the longest diagonal from its respective corner of the board. These tactics are traditionally implemented at the outset of the game to gain control of the diagonals (Esterin & Panov, 1980). There are two types of Fianchetto available, both of which are discussed next.

5.3.2.1 *Left Fianchetto*

The requirement for implementing the Left Fianchetto is to simply have the queen's bishop positioned in the b2 square for the white player, or the b7 square for black. An illustration of this tactic is shown in Figure 13 below.

The example board state in Figure 13 shows a perfect Left Fianchetto formation, with the queen's bishop in the designated position and three pawns surrounding it.



Template Tree:

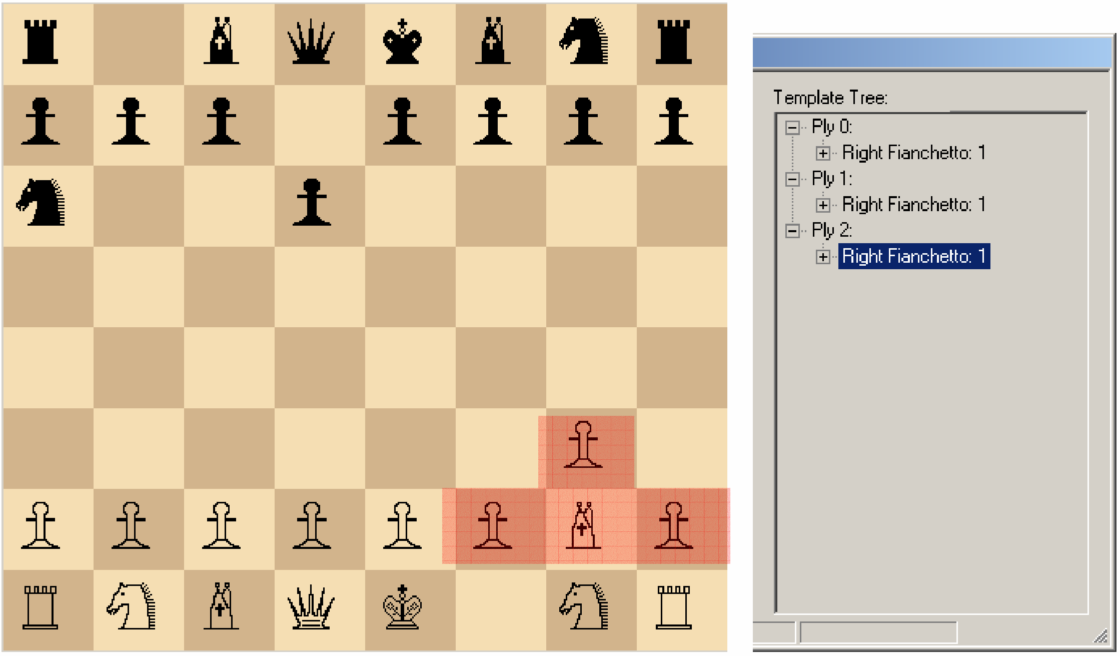
- Ply 0:
  - Left Fianchetto: 1
  - Black Knight at f6 being attacke
- Ply 1:
  - Left Fianchetto: 1
  - Black Knight at f6 being attacke
- Ply 2:
  - Left Fianchetto: 1
  - Black Knight at f6 being attacke

Figure 16: Execution of Left Fianchetto by white player (Left) with corresponding ATBI reading (Right)

5.3.2.2 *Right Fianchetto*

Mirroring the Left Fianchetto, the Right Fianchetto requires that the king's bishop reside in square g2 for white and g7 for black. The attributes are structured the same, but are shown for the sake of completeness.

The example board state in Figure 14, the white king's bishop is in the designated position, with three pawns around it. The highlighted region defines the squares used by the template.



Template Tree:

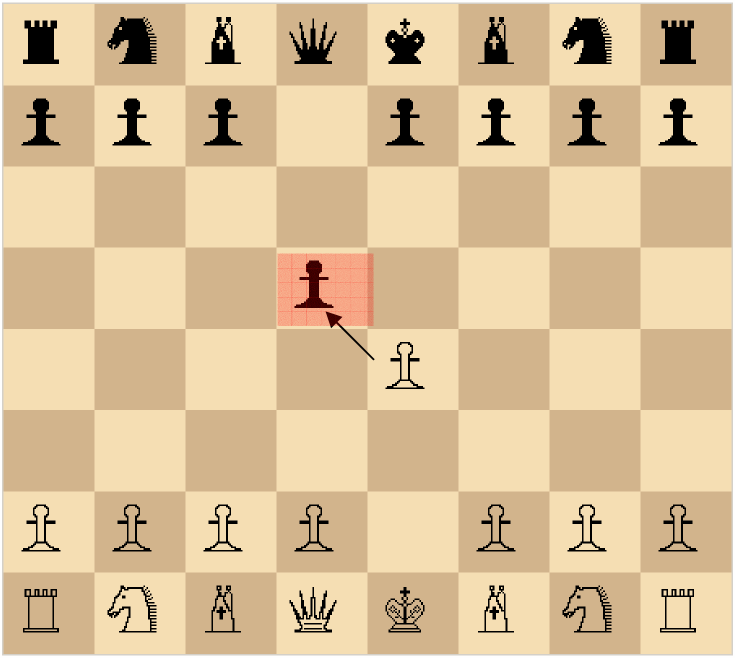
```
⊟ Ply 0:
    ⊞ Right Fianchetto: 1
⊟ Ply 1:
    ⊞ Right Fianchetto: 1
⊟ Ply 2:
    ⊞ Right Fianchetto: 1
```

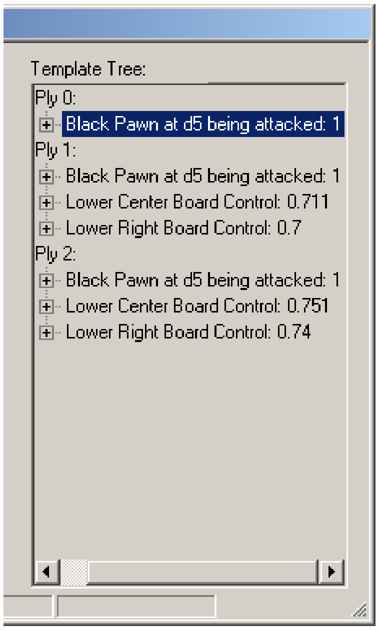
Figure 17: Execution of Left Fianchetto by white (Left) and corresponding ATBI reading (right)

5.3.3 Piece Attack Templates

There is also a series of 64 templates, each watching a square for a piece being under attack by an opposing piece. While each of these templates has a relatively simple composition (only one attribute in each), they serve an important role in the IRS. Collectively, they detect any likely attack against a friendly piece. The attribute only activates if the method finds an attack that the opponent would find beneficial. In other words, the attack must be either against an unguarded piece or one of greater value than the attacking piece. If there is no relative value to be gained in the attack (i.e. trading a pawn for a knight would be favorable), then the attribute remains deactivated. Below is one example of an attack that the IRS recognizes.

In the example board state in Figure 15 below, white's pawn is in position to attack black's pawn. The highlighted area indicates square under attack.





Template Tree:

Ply 0:
⊞ Black Pawn at d5 being attacked: 1
Ply 1:
⊞ Black Pawn at d5 being attacked: 1
⊞ Lower Center Board Control: 0.711
⊞ Lower Right Board Control: 0.7
Ply 2:
⊞ Black Pawn at d5 being attacked: 1
⊞ Lower Center Board Control: 0.751
⊞ Lower Right Board Control: 0.74

Figure 18: Example of a simple attack (Left) and corresponding reading from ATBI (Right)

5.4 Summary

At this point, the prototype has been explained in detail, including the testbed, in which the IRS operates, the templates representing agent intent, and the TBI Moderator that governs them. Now that the solution has been completed, it may now be tested for its validity as an IRS. In the next chapter, the system is put through a series of experiments designed to test the hypothesis of this thesis.

CHAPTER 6: PROTOTYPE TESTING

Since the primary contribution of this thesis is the extension of the TBI paradigm into ATBI, the testing process must evaluate the reliability of the advanced detection of agent intent made by the prototype IRS. The test process for the prototype will consist of three phases of increasing difficulty. The dampening effect discussed in Chapter 4 serves as a method of controlling the impact that the future ply have on the templates of the system, and can be scaled during the testing phase. By duplicating a group of tests with different values, the dampener can be used as a means of determining their optimal value for the ATBI system in this application. This variable can be perceived as a layer of fog, blanketing the state space and reducing the visibility of those more temporally distant from the watchful eye that is the recognition system. The following sections describe the different testing phases including their purpose and actual results. First, however, the specific tests for each phase are discussed.

6.1 Testing Method

As a control to the experiment, duplicates of the data for the series of tests from each subject (human player) in the first phase is sent to the other phases. Each phase runs the same set of tests with the same input, but with a different dampener value. This action is taken to ensure that the results that are produced by each phase accurately reflect the change in dampener value and nothing else. Thus, the same set of test data from each subject is recorded and repeatedly simulated again with varying dampener values.

6.1.1 Test Subjects

None of the test subjects in this section are recognized as “ranked” players, but expertise with the game of chess is not required. As long as the subjects understand the game enough to implement basic attacks and board control tactics, domain expertise is not a requisite. They were chosen primarily because of their willingness to participate without compensation. All four of them, however, have a sufficient understanding of the domain to qualify them as test subjects. They are each instructed to control the white pieces and implement a tactic. The type of tactic that the subject executes is defined in the following sections, in which each test is explained. The black player’s moves are automatically made through a simple random selection of any legal move available to it.

6.1.2 Test #1: Simple Attack

The test subject is asked to perform a straightforward attack against an opposing piece. The attack must be one where it either makes a clean pick, stands to make a favorable trade (it captures an enemy piece of great value and loses one of lesser value), or an even exchange. None of the pieces involved in the attack are to be more than one move away from actually making the capture. The end criterion of the test is met when the test subject claims that the attack is one move away from completion, meaning the attacking piece can make its capture on its next move. This test was also used in Akridge (2005) for the same purpose, which is to establish that the IRS is capable of recognizing simple straightforward attacks.

6.1.3 Test #2: Hidden / Complex Attack

This test is aimed at addressing a shortcoming recognized by Akridge (2005). The tactic to be implemented here is similar to that of a simple attack, but with one exception. The attacking piece cannot be able to attack the target in the current state (ply zero). In other words, the test subject must set up a situation in which one move is required for the attack to be readily apparent. The goal is for the system to detect the attack in advance and warn the player controlling the attacked piece of the impending assault. The criteria for this test are again similar to that of test #1. The test subject must declare that the attack against the intended target with the chosen attacker has been set up. The game is then allowed to progress one more full turn to allow for the capture to take place. This way it will be clear whether the IRS was able to detect the attack. This test was used in Akridge (2005) to test its limits. Since his system did not account for future states, however, none of the tests succeeded.

6.1.4 Test #3: Board Control

The final test is a departure from basic attack recognition. The test subject is instructed to choose an area of the board on which to focus his attention and move his pieces such that they control that section. The strategy here is that the target area of the board is either controlled or being attacked by the player. Relevant templates for this test are all of those mentioned in Chapter 5 involving control over the left, right, and center areas of the board. The test subject should seek to position his pieces such that their coverage of his predetermined "zone" is maximized. In other words, his pieces should be able to attack as many of the squares in the zone as possible. The end criteria for this test are less concrete than those of the previous two tests, thereby requiring more moves to complete than the other tests. In this case, the test subject himself should decide whether he has made enough moves to set up his strategy, at which time, the test ends. The purpose of this test is to force the IRS to choose from out of 3 "zones" which one the subject is really interested in controlling. These tactics also take time to set up since multiple pieces have to be involved, which means increased complexity.

6.1.5 Summary of Test Method

Each test subject is to determine a strategy before the outset of each test, and follow it without any deviation. Test subject input is only required for one phase of testing, after which the same input set can be re-run for each subsequent phase. The dampener value for the first phase is large, so that the system will not see the intended strategy for many moves. The effect is that tests in subsequent phases (with lower dampener values) will fail if the tactics implemented in them are not found within the existing input set. This is because lower dampener values allow more future confidence and should therefore find the same tactic sooner.

6.2 Test Runs

In this section, the actual games played out by the subjects are described in detail. These data are given its own section for ease of reference in the later sections when discussing the effects of varying dampener values. As mentioned before, for each dampener value, the same subject input is re-run through the system to see the impact it has on the IRS's ability to properly recognize future intent. Therefore, only one set of test data per human test subject is required, since the same data can be processed again for all phases.

6.2.1 Subject #1

The following subsections discuss the moves from the three tests run on the system involving subject #1.

6.2.1.1 *Test #1*

Table 1: Subject #1, Test #1

Move	Description
1. e2->e4, g8->f6	White pushes king's pawn twice. Black brings its king's knight to f6.
2. e4->e5, h8->g8	White pushes the same pawn. Black slides its king's rook over one square (a move of no significance). At this point, white is in position to capture black's knight.
3. e5xf6	White captures the knight on f6 with the pawn from e5
Stopping Criteria Met	

6.2.1.2 *Test #2*

Table 2: Subject #1, Test #2

Move	Description
1. c2->c4, e7->e5	White pushes its queen's bishop's pawn twice. Black pushes its king's pawn twice.
2. g2->g3, d8->f6	White pushes its king's knight's pawn once. Black brings its queen out to f6.
3. d2->d3, b7->b6	White pushes its queen's pawn once, giving the queen's bishop some mobility. Black pushes its queen's knight's pawn once.
4. h2->h4, e5->e4	White pushes its king's rook's pawn two squares, creating some pressure on the right side of the board. Black advances its king's pawn.
5. c1->g5, c8->a6	White brings its queen's bishop out to attack Black's queen from g5. Black bring its queen's bishop down to the left.
6. d3xe4, f8xe7	White pawn captures black pawn. Black brings its king's bishop down slightly
7. f1->g2, h7->h6	White Fianchettos its king's bishop. Black pushes its king's rooks pawn one square
8. e4->e5, f6->g6	White pushes its queen's pawn (now in king's file) into an attack against Black's queen. Black's queen slides to the right to avoid capture
9. g2xa8	White captures Black's rook at a8 with its bishop at g2.
Stopping Criteria met	

6.2.1.3 *Test #3*

Table 3: Subject #1, Test #3

Move	Description
1. g2->g3, h7->h6	White pushes its king's knight's pawn once. Black pushes its king's rook's pawn once.
2. e2->e4, f7->f6	White pushes its king's pawn twice, giving its queen mobility towards the right. Black pushes its king's bishops pawn once
3. d2->d3, c7->c5	White pushes its queen's pawn once, giving its queen's bishop mobility to the right. Black pushes its queen's bishop's pawn once.
4. f1->g2, a7->a6	White Fianchettos its king's bishop. Black pushes its queen's rook's pawn once.
5. d1->f3, b8->c6	White brings its queen out to the right. Black brings its queen's knight out.
6. c1->f4, d8->c7	White brings its queens bishop out just above its queen. Black responds by bringing its queen out in line of sight of the bishop.
7. e4->e5, e7->e6	White advances its king's pawn, blocking lie of sight between the conflicting queen and bishop. Black brings its king's pawn out to meet the pawn
Stopping Criteria Met	

6.2.2 Subject #2

The following subsections discuss the moves from the three tests run on the system involving subject #2.

6.2.2.1 *Test #1*

Table 4: Subject #2, Test #1

Move	Description
1. g2->g3, e7->e6	White pushes its king's knight's pawn once. Black pushes its king's pawn once
2. g3->g4, b7->b5	White advances the same pawn again. Black pushes its queen's knight's pawn twice
3. f1->h3, b5->b4	White moves its king's bishop behind the king's knight's pawn. Black advances its queen's knights pawn.
4. c2->c3, b8->c6	White pushes its queen's bishop's pawn once. It is now in position to attack the pawn Black has been advancing. Black brings out its queen's knight.
Stopping Criteria Met	

6.2.2.2 *Test #2*

Table 5: Subject #2, Test #2

Move	Description
1. g1->f3, f7->f6	White brings out its king's knight. Black pushes its king's bishop's pawn once.
2. d2->d3, a7->a5	White pushes its queen's pawn once. Black pushes its queen's rook's pawn twice.
3. c1->f4, f6->f5	White brings its queen's bishop out to the right side of the board. Black advances its king's bishop's pawn.
4. e2->e4, c7->c6	White pushes its king's pawn twice. Its is now in position to attack Black's pawn at f5. Black pushes its queen's bishop's pawn once.
Stopping Criteria Met	

6.2.2.3 *Test #3*

Table 6: Subject #2, Test #3

Move	Description
1. g1->f3, d7->d5	White brings out its king's knight to the left. Black pushes its queen's pawn twice.
2. b1->c3, g8->f6	White brings out its queen's knight to the right. Black brings out its king's knight to the left.
3. e2->e4, h7->h6	White pushes its king's pawn twice. It is now in position to capture Black's pawn at d5. Black pushes its king's rook's pawn once.
4. f1->c4, h6->h5	White brings its king's bishop out to the left. It is now in position to capture Black's pawn at d5. Black advances its king's rook's pawn.
5. d2->d4, h8->h6	White pushes its queen's pawn twice. Black brings its kings rook down twice.
Stopping Criteria Met	

6.2.3 Subject #3

The following subsections discuss the moves from the three tests run on the system involving subject #3.

6.2.3.1 *Test #1*

Table 7: Subject #3, Test #1

Move	Description
1. e2->e4, g8->f6	White pushes its king's pawn twice. Black brings out its king's knight to the left.
2. d2->d4, c7->c5	White pushes its queen's pawn twice. Black pushes its queen's bishop's pawn twice. It is now in position to capture White's pawn at d4.
3. c1->e3, h7->h6	White brings its queen's bishop out to the center. Black pushes its king's rook's pawn once.
Stopping Criteria Met	

6.2.3.2 *Test #2*

Table 8: Subject #3, Test #2

Move	Description
1. e2->e4, b8->c6	White pushes its king's pawn twice. Black brings out its queen's knight to the right.
2. d1->h5, a7->a6	White brings its queen up to the right side of the board. Black pushes its queen's rook's pawn once.
3. f1->c4, c6->e5	White brings its king's bishop up to the left. Black moves its queen's knight to a position that threatens White's bishop at c4.
4. h5xe5, a8xa7	White's queen captures the offending Black knight. Black brings its queen's rook down once.
Stopping Criteria Met	

6.2.3.3 *Test #3*

Table 9: Subject #3, Test #3

Move	Description
1. e2->e4, h7->h6	White pushes its king's pawn twice. Black pushes its king's rook's pawn once.
2. d1->h5, c7->c6	White moves its queen up to the right side of the board. Black pushes its queen's bishop's pawn once.
3. d2->d4, b7->b6	White pushes its queen's pawn twice. Black pushes its queen's knight's pawn once.
4. c1->f4	White moves its queen's bishop up to the right/center.
Stopping Criteria Met	

6.2.4 Subject #4

The following subsections discuss the moves from the three tests run on the system involving subject #4.

6.2.4.1 *Test #1*

Table 10: Subject #4, Test #1

Move	Description
1. f2->f4, b8->c6	White pushes its king's bishop's pawn twice. Black brings its queen's knight out to the right.
2. h2->h4, c6->e5	White pushes its king's rook's pawn twice. Black moves its queen's knight to e5, in range of capture by White's pawn at f4.
Stopping Criteria Met	

6.2.4.2 *Test #2*

Table 11: Subject #4, Test #2

Move	Description
1. h2->h4, g8->f6	White pushes its king's rook's pawn twice. Black brings out its king's knight to the left.
2. h1->h3, b8->c6	White brings its king's rook up twice. Black brings its queen's knight out to the right.
3. h3->f3, f6->h5	White moves its king's rook left twice, into the line of sight of Black's knight at f6. Black moves the knight to h5.
4. f3->f5, c6->b8	White moves the rook up to f5, threatening Black's knight to its right. Black moves its queen's knight back to its starting location.
5. e2->e4	White pushes its king's pawn twice
Stopping Criteria Met	

6.2.4.3 *Test #3*

Table 12: Subject #4, Test #3

Move	Description
1. g2->g4, c7->c5	White pushes its king's knight's pawn twice. Black pushes its queen's bishop's pawn twice.
2. g1->f3, b7->b5	White brings out its king's knight to the left. Black pushes its queen's knight's pawn twice.
3. f1->h3, c5->c4	White brings its king's bishop up to the right side of the board. Black advances its queen's bishop's pawn.
4. f3->g5, g8->f6	White brings its kings knight up to the right, in range of Black's front row of pawns. Black brings its king's knight out to the left.
5. f2->f4, b8->c6	White pushes its king's bishop's pawn twice. Black brings out its queen's knight to the right.
6. e2->e4, e7->e6	White pushes its king's pawn twice. Black pushes its king's pawn once.
7. d1->f3, f8->d6	White brings its queen up to the right, supporting the row of pawns above it. Black moves its king's bishop out, down and to the left.
8. e1->f2, d6->f8	White moves its king up and to the right. Black reverses its previous move.
Stopping Criteria Met	

6.2.5 Criteria for Success/Failure

Now that the individual test runs for each subject have been presented, the input can be run back through the system with he dampener set at various levels. Since the test subjects provided the stopping points for all of the tests, the method of defining a success versus a failure is simple. If the system detects the subject's intentions before the game is halted, then the test is scored a success. Otherwise, it is scored as a failure. Note that recognition of agent intention also means that there cannot be a superfluous amount of false positives (wrong template candidates) mixed with the correct one.

6.3 Phase #1 – Dampener 0.80

The purpose of this test is to find the region of dampening values that reduces the future ply confidence contributions to the point that they simply make no difference. At this level of dampening, the system's performance may resemble that of the IRS of Akridge (2005), because the majority of the confidence received by the templates occurs at the current state, or ply zero. At this time, the system should fail to see almost every hidden / complex attack. It is hoped that the board control tests will succeed, though the IRS is may take some time to spot the subject's intention. The first test, which relies only on the current game state (ply 0), should succeed every time.

6.3.1 Subject #1 Test Data

The first test had positive results. The simple recognition of an attack in the current state (ply 0) was immediately recognized. The subject simply pushed up a pawn to meet with an enemy knight that was pulled out in the previous move. The high value of the dampener had no effect on the IRS's recognition of the straightforward attack.

For the second test, the subject set up an attack for a bishop to capture an opposing rook with a pawn obstructing the bishop's path. The IRS did not detect the future attack, however, and did not recognize that the rook was in danger until the pawn moved. At that point the attack was reduced to a simple one like in test #1, which is virtually guaranteed to be recognized.

For the last test, the IRS detected the subject's lower right surprisingly well, considering the high value of the dampener. The template crossed the 70% confidence threshold after three moves.

Table 13: Phase #1 Results for Subject #1

Subject #1		
Test	**Strategy**	**Result**
#1 Simple Attack	Attack enemy knight at f6	Success
#2 Hidden/Complex Attack	Hidden attack with Bishop at g2 attacking black rook at a8.	Failed
#3 Board Control	Lower Right Board Control	Success (after 3 moves)

The first test succeeded because the attack was a simple one, meaning the target piece was one move away from being captured. This intention is picked up in the current state (ply 0) which does not rely on the dampener at all. The second test failed because the IRS never gave any advanced warning of the attack. The third test took three moves before the IRS detected any board control tactic, though it detected the correct one.

6.3.2 Subject #2 Test Data

The simple attack succeeded again, with one white pawn moving up to meet a black pawn. The intended hidden attack involving a knight and a pawn was completely missed, however. This missed detection could be attributed to the high value of the dampener, though. The third test was recognized just in time, barely breaking the threshold on the last recorded move.

Table 14: Phase #1 Results for Subject #2

Subject #2		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	pawn (b4) vs. pawn (c3)	Success
#2 Hidden/Complex Attack	knight (f4) could attack pawn (f5) after one move	Failed
#3 Board Control	Lower Center Board Control	Success (after 5 moves)

The first test succeeded again because the attack was a simple one. Again the second test failed. The white player's (subject's) knight was destined to attack one of black's pawns in two moves, and the dampener decreased the corresponding template's confidence to the point below the threshold. The tactic implemented in the third test required five moves of development before the IRS detected it. This result concurs with the hypothesis that the increased dampener reduces confidence to the point that many of the template's attributes must be active for its confidence to meet or exceed the threshold.

6.3.3 Subject #3 Test Data

Just as with the first two subjects, the first test succeeded, while the intention in the second test is never acknowledged. The third test again yielded success, this time after only two moves.

Table 15: Phase #1 Results for Subject #3

Subject #3		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	White Pawn (d4) vs. Black Pawn (c5)	Success
#2 Hidden/Complex Attack	White Bishop (c4) vs. Black Knight (c6)	Failed
#3 Board Control	Lower Center Board Control	Success (after 2 moves)

Again, the first test succeeded. The attack in second test again eluded the IRS, most likely because of the high value of the dampener. The future attack of white's bishop against black's knight was not found. The third test succeeded after a curiously small number of moves. The test subject brought white's queen out into the open on the second move, allowing it to attack many of the squares in the area defined by the template. This move thus explains the result.

6.3.4 Subject #4 Test Data

This test yielded the same results as previous ones in this phase, with one notable difference. The third test required six moves for the subject's zone control tactic to be recognized. The change of the required number of moves in the next phase should be a barometer of how much the dampener affects the proper detection of agent intentions.

Table 16: Phase #1 Results for Subject #4

Subject #4		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	White Pawn (f4) vs. Black Knight (e5)	Success
#2 Hidden/Complex Attack	White Rook (f3) vs. Black Knight (h5)	Failed
#3 Board Control	Lower Right Board Control	Success (after 6 moves)

The first test was again recognized as was hypothesized. The IRS again failed to detect the complex attack. White's rook was one move away from being in position to attack black's knight. The third test again required several moves before the subject's intention to control the lower right side of the board was recognized. The IRS did not have any readings up to the point where last move was made.

6.3.5 Summary of Results for Phase #1

Some of the strategies implemented during this phase were either recognized late or not at all. While the simple attacks were spotted, they did not require future state confidence to be found. The hidden attacks for the most part went completely unnoticed. No doubt the lack of much IRS activity in this phase can be attributed to the high value of the dampener, which acts to suppress future confidence.

Next, the dampener keeps a low value, allowing much future confidence through.

6.4 Phase #2 – Dampener 0.20

The purpose of this test to prove that relying too heavily on future ply is folly. If the dampener is set too low, the confidence all of the possible future strategies blend in with those that may actually be candidates. Since the dampener is low, any template that decides to "speak up" as a candidate based on a state far into the future (ply two) reflects the confidence of one that has already "spoken up" in a state nearer the present (ply zero). This effect could conceivably confuse the system into presenting a tactic that has not yet been realized over one that is already in motion. This phase will show that the dampener must have some minimum value if the IRS is to avoid a large amount of false positives.

6.4.1 Subject #1 Test Data

The first test was unaffected by the change in future confidence contribution, though two other templates did create some "noise" by showing themselves as future intentions that clearly have not yet been implemented. The second test performed similarly, with the hidden move being detected in the first future generation (ply 1) along with Right Fianchetto and a board control template. Interestingly, the board control template was detected after 2 moves, although up to that point it was being drowned out by false positives for piece attacks. It stood alone in ply 0, but when ply 1 and ply 2 finished, it was beaten by a false reading for lower center board control.

Table 17: Phase #2 Results for Subject #1

Subject #1		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	Attack enemy knight at f6	Succeeded
#2 Hidden/Complex Attack	Hidden attack with Bishop at g2 attacking black rook at a8.	Succeeded
#3 Board Control	Lower Right Board Control	Succeeded (after 3 moves)

The first test succeeded again as intended. The IRS picked up the basic attack with no problems. The second test succeeded, though the IRS picked up some intentions that may not have been accurate along the way. After the first move, the system believed the subject was attempting to control the lower left side of the board and also projected a hidden attack not intended by the subject. The second move set up a Right Fianchetto to be satisfied in one move, which caused the IRS to present that template in ply 1. Additionally, all three board control templates were viewed as candidates in ply 2. On the fourth move, one of black's pawns was under direct attack, which made it the top candidate. This misread could be attributed to black's random movements, however and can be discarded. Skipping ahead to the eighth move, white finally puts its bishop in place to attack black's rook, with white's pawn blocking the direct line of attack (thus making the attack hidden). The IRS picks up the attack with high confidence, but it is not the top candidate. A Right Fianchetto and all three board control templates that were recognized during the game now had higher confidence levels than the attack. Also, other possible future attacks were now detected by the IRS, making the intended attack one of eight possible attacks with equal confidence. However, none of the templates with greater confidence were piece attack templates, so the test can therefore be graded as a success on the basis that the intention was found and not outranked by another template of the same type. For the third test, the IRS detects a Right Fianchetto as the subject's future intent after the first move. This happened because the dampener was so low that it allowed the template to break the threshold, which did not occur in the first phase. The IRS actually detected the subject's Lower Right Board Control tactic in ply 2, though it more strongly believed in the Fianchetto. After the third move, however, the intended tactic is recognized over the Fianchetto as the top candidate.

6.4.2 Subject #2 Test Data

The board control strategy was found quickly enough, though future confidence levels proceeded to drown its relative candidacy with a flood of false positives, one which received a higher confidence level than it. The template for lower center control overtook it with greater confidence in the 1st and 2nd ply, however. The future attack missed in phase one was recognized, though it was less confident than a lower right board control template. It also shared the same weight as three other equally confident piece attack templates. It is becoming evident that it is difficult to isolate single strategies when the dampener is low. There appears to always be at least a couple false positives mixed in with the correct recognition.

Table 18: Phase #2 Results for Subject #2

Subject #2		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	pawn (b4) vs. pawn (c3)	Succeeded
#2 Hidden/Complex Attack	knight (f4) could attack pawn (f5) after one move	Succeeded
#3 Board Control	Lower Center Board Control	Failed

The first test again succeeded. The simple attack of pawn against pawn was the unanimous belief of the IRS. For the second test, after the first two moves, the IRS falsely recognized a Lower Right Board Control and an attack against a black pawn in ply 1, and even more in ply 2. After the fourth move however, the hidden attack was correctly recognized along with a few others, just as with subject #1. Lower Right Board Control was the only candidate template that was not for a piece attack. The rest were random attacks against other pawns. For the third test, the IRS found the intended Lower Center Board Control tactic, after the first move in ply 2, though it was second in candidacy to a possible hidden attack against a black pawn. After the second move, however, the tactic disappeared from recognition completely, replaced by Lower Right and Lower Left Board Control templates. This test is thus scored as a failure.

6.4.3 Subject #3 Test Data

The IRS actually foresaw the simple attack one move in advance, which makes sense. Even though the attack is not purposefully hidden, it can still be recognized ahead of time. The IRS actually detected the hidden attack after the 2nd move. As soon as the king's pawn moved from the bishop's path, the system found the path to the capture.

For the board control test, the IRS eventually did select the desired template, but not before choosing an incorrect one. So this test was unsuccessful.

Table 19: Phase #2 Results for Subject #3

Subject #3		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	White Pawn (d4) vs. Black Pawn (c5)	Succeeded
#2 Hidden/Complex Attack	White Bishop (c4) vs. Black Knight (c6)	Succeeded
#3 Board Control	Lower Center Board Control	Failed

The simple attack again succeeded. The simple attack was found by the IRS with no problems after the third move in the game. For the second test, the subject intended to attack black’s knight with his bishop. After only one move, the IRS detected it as the number one candidate in ply 1, over Lower Right and Lower Center Board Control templates; though in ply 2 Lower Right Board Control overtakes it. Still, the ability for the IRS to detect the tactic so quickly earns a success for the test. For the third test, the subject’s intended tactic (Lower Center Board Control) was found after the first move in ply 1 and ply 2, but it was of lower confidence than Lower Right Board Control. The next move (bringing the queen out into the open) created many false piece attack templates that are blended in below the same two, and with Lower Left Board Control at the bottom of the list. The third move (pushing a white pawn up the center) pushed the Lower Center Board Control template to the top of the list, but the wrong template had been at the top for too long. This test is scored as a failure.

6.4.4 Subject #4 Test Data

For the first test, the simple attack is detected successfully and, more importantly, alone. Looking at the board though, it is easy to see why. The pieces that were moved did not open up the board much for other pieces, thereby limiting the amount of future states and their respective false positives. The hidden attack was also successfully recognized under the same circumstance. The test subject’s intended attack was the only one available for at least two moves, and thus received the highest confidence by default. Ply one saw the recognition of a board control template, and in ply 2 that template gained enough confidence to bump the template for the intended attack beneath it. However, the correct attack was found first.

Amazingly, the IRS found the subject's intention for controlling the right side of the board after the first move, although it was found in the second ply of search, and after an attack had been spotted in ply 1. However, it was properly identified as the most probable "zone" that was being controlled.

Table 20: Phase #2 Results for Subject #4

Subject #4		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	White Pawn (f4) vs. Black Knight (e5)	Succeeded.
#2 Hidden/Complex Attack	White Rook (f3) vs. Black Knight (h5)	Succeeded
#3 Board Control	Lower Right Board Control	Succeeded (after 1 move)

Again, the simple attack was recognized after two moves with no difficulty. The second test also succeeded with the recognition of white's rook future attack against black's knight. The only misread by the IRS during this test was after the second move in ply 2 with a Lower Right Board Control. After the third move, the system detected the subject's intent as the top candidate in ply 1, and as the second highest (after Lower Right Board Control) in ply 2. It therefore scores a success. The third test scored a success after the first move. One template (Right Fianchetto) had a higher confidence in ply 1, but was ignored since the desired template passed it in ply 2.

6.4.5 Summary of Results for Phase #2

Some tests worked well with this low dampener setting and appeared alone as the representative of agent intent. The majority of the test cases, however, showed the subjects' intentions mixed in with many others that were not close. The original suspicion that this setting is to low to allow for the IRS to consistently tell the actual strategy apart from all others is validated. When mixed with that many erroneously confident templates, the subjects' choices did not stand out well.

6.5 Phase #3 – Dampener 0.60

The aim of this phase is to test a dampening value that better combines the template confidence levels of the current state with those of future ply. The idea is to find the right balance so that the system neither relies too heavily on the future states, nor ignores them completely. The proper dampener setting should allow the contribution of future template confidence to boost strategies that are nearly implemented, but that are lacking one or two moves before they are clearly visible.

6.5.1 Subject #1

The simple attack was successfully recognized succeeded with minimal background noise beforehand. The only falsely recognized template was actually the one that the subject intended to implement in a later test, so it is quite possible that it was not noise at all, but rather a true intention that was not specifically requested of the test subject. In any case it did not interfere with test #1, so the test is a success.

Unfortunately, the dampener was still too high to successfully recognize the hidden attack in test #2. Based on results from phase #2, it appears that it is near impossible to get only one future attack visible, simply because they are all sitting at approximately the same confidence level, and when one jumps over into candidacy, many tend to follow.

The number of moves required before the IRS recognized the board control strategy agrees with those of phases one and two for this subject. The lower dampener setting required only 1 move to recognize it, whereas the higher value required 3 moves. This case fits in between them in required moves and has the middle dampener value to match. It would appear then that there is some correlation between the dampener and number of moves required for strategy recognition, at least for the zone control templates.

Table 21: Phase #3 Results for Subject #4

Subject #1		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	Attack enemy knight at f6	Succeeded
#2 Hidden/Complex Attack	Hidden attack with Bishop at g2 attacking black rook at a8.	Failed
#3 Board Control	Lower Right Board Control	Succeeded (after 2 moves)

The simple attack in first test was successfully recognized after the third move. The second test did not produce any IRS readings until after the fourth move, when black's pawn was under attack by one of white's pawns. In ply 2 it also saw Lower Center and

Lower Right Board Control tactics. None of these were intended though, so they are ignored. After the sixth move, the IRS detects Lower Left Board Control in ply 0, and all three Bard Control templates in ply 1. After the eighth move, these candidates remain along with a strong belief for Right Fianchetto, but the intended hidden attack (and all others for that matter) remained undetected. This test is scored as a failure. The third test was recognized in a very straightforward manner. No templates were presented after the first move, but the Lower Right Board Control template was presented as the top candidate along with the Lower Center Board Control template in ply 1 after the second move was made. This test was therefore scored as a success.

6.5.2 Subject #2

The simple attack is detected with minimal interference again, providing more evidence that the dampener level truly does help to reduce the number of false positives.

The test for the hidden attack however falls slightly short of being recognized once more. The confidence threshold and dampener combined seem to be equally suppressing any piece attack template more than one move away from being realized.

The intended board control template went unnoticed until the end of the 5th move, and the IRS presented two other probable zones before the one intended by the test subject. Since it was not even present as a candidate when the incorrect ones were presented, this test must be scored as a failure.

Table 22: Phase #3 Results for Subject #2

Subject #2		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	pawn (b4) vs. pawn (c3)	Success
#2 Hidden/Complex Attack	knight (f4) could attack pawn (f5) after one move	Failed
#3 Board Control	Lower Center Board Control	Failed

The attack from the first test was again the only tactic recognized by the IRS. It was hoped that the complex attack of the second test would be recognized, but the tactic was a weak candidate since the dampener is at a moderate level. The IRS saw no candidate templates until after the 3rd move, when it detected Lower Right Board Control in ply 0 and Lower Center Board Control in ply 1. No hidden attack was detected however, so this test has failed. For the third test, the expectation was that the Board Control tactic would be found somewhere between the time required for phase 3 and phase 1. The first candidate of the test, a simple attack against a black pawn, was found after the third move. After the fourth move, Lower Left Board Control was added to the candidate list at ply 1 under the attack from the previous move. The desired tactic was finally added as a candidate after the fifth and final move of the input set, though it appears under the existing piece attack candidate recognized earlier. Since the IRS presented this tactic after having first presented an incorrect Board Control template, this test is scored as a failure.

6.5.3 Subject #3

The first test again succeeds without interference. The second test fails again, as did the previous two. The third test produced similar results as the first phase. The lower center area that was intended by the test subject was not detected until a couple moves after the IRS presented the lower right zone as its belief. However, the formation of the pieces at the time of the reading justifies the IRS's decision, since there is much presence on the right side for one turn before it is shifted to the center. This test therefore is successful.

Table 23: Phase #3 Results for Subject #3

Subject #3		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	White Pawn (d4) vs. Black Pawn (c5)	Success
#2 Hidden/Complex Attack	White Bishop (c4) vs. Black Knight (c6)	Failed
#3 Board Control	Lower Center Board Control	Successful (after 3 moves)

The first test succeeds after the third move. For the second test, the IRS showed Lower Right and Lower Center Board Control as candidates at ply 0 after two moves, with Lower Left Board Control added at ply 1. At that point the IRS should have noticed a future attack on black's knight with the white queen, but it showed no piece attack candidates. After the following move, the IRS showed a black knight as being attacked directly by white's queen, which is one move too late to be correctly spotted as a future intention. This test therefore fails. For the third test, the expectation was for the system to detect the subject's Board Control strategy somewhere between the time required for phases 1 and 2. The IRS showed Lower Right Board Control above Lower Center Board Control as two candidates at ply 0 after the second move. Lower Left Board Control was added to the list under them both at ply 1. After the third move, the top two candidates swap places, and the subject's chosen tactic is properly chosen. Even though an incorrect template was initially chosen over the correct one, this mistake was rectified before the subject had finished implementing his strategy. This test therefore scores a success.

6.5.4 Subject #4

The first two tests again confirm the results of the other test subjects in this phase, thereby showing consistency in the operation of the IRS with certain values for the dampener.

The third test, whose quality of results seem dependent on how well the player executes his intended strategy, still seems to be fairly consistent, with the majority of the intended templates being recognized by the IRS after two moves.

Table 24: Phase #3 Results for Subject #4

Subject #4		
Test	**Strategy**	**Success/Failure**
#1 Simple Attack	White Pawn (f4) vs. Black Knight (e5)	Success
#2 Hidden/Complex Attack	White Rook (f3) vs. Black Knight (h5)	Failed
#3 Board Control	Lower Right Board Control	Successful (after 2 moves)

The subject's simple attack was detected after two moves as the sole candidate. The expectation for the second test is for the subject's complex attack to be recognized as a future intention. The only candidate found during this test was a Lower Right Board Control template in ply 2 after the third move. At this point the IRS was supposed to have spotted the future attack, so this test failed. For the third test, the expectation was for the IRS to recognize the subject's Board Control strategy (preferably between the times required for the third test of phases 1 and 2). The IRS successfully detected the subject's strategy at ply 2 after the second move. It was the first and only candidate presented during the test, which makes this test a success.

6.5.5 Summary of Results for Phase #3

This phase of testing definitely shows the usefulness of the dampener as a means of fine tuning the PTBI engine to use a proper portion of future state confidence. The test subjects' intentions were either both promptly detected and presented with high confidence, or they were not found at all. This phase of testing shows that while the existing system may not be flawless in its detection of agent intentions, it does show that there is promise for ATBI if the dampener is properly calibrated.

6.6 Overall Test Results & Conclusions

Having completely tested the system in all phases, some conclusions regarding overall performance may be drawn.

First, a small dampener, while it does allow the correct strategy to be seen almost every time, also allows many incorrect ones to be recognized with almost as much confidence.

Second, a large dampener defeats the purpose of PTBI altogether, since the results of phase one included several failed tests. A high dampener value simply will not let enough future confidence in to make a difference.

Third, a moderate dampener value tends to let a moderate amount of confidence from future states be considered, though it tends to be hit or miss depending on the behavior of the test subject.

In conclusion, no single dampener value appears to simultaneously satisfy all of the expectations of the IRS. It is either too high to allow the complex / hidden attacks to weight in, or too low to single out the subject's actual intentions from all of the incorrect ones.

The table below summarizes all of the test results by phase and test number. The numbers below each phase indicates the respective dampener value of that phase.

Table 25: Summary of Test Results

Summary of test Results					
Phase 1: **0.80**	**Test #**	**Subject #1**	**Subject #2**	**Subject #3**	**Subject #4**
	1	Success	Success	Success	Success
	2	Fail	Fail	Fail	Fail
	3	Success	Success	Success	Success
Phase II: **0.20**	**Test #**	**Subject #1**	**Subject #2**	**Subject #3**	**Subject #4**
	1	Success	Success	Success	Success
	2	Success	Fail	Fail	Success
	3	Success	Fail	Fail	Success
Phase III: **0.60**	**Test #**	**Subject #1**	**Subject #2**	**Subject #3**	**Subject #4**
	1	Success	Success	Success	Success
	2	Fail	Fail	Fail	Fail
	3	Success	Fail	Success	Success

CHAPTER 7: SUMMARY, CONCLUSION, & FUTURE WORK

This chapter summarizes the research done in this thesis and draws conclusions from the experimental results. These conclusions include what was accomplished as well as the weaknesses discovered in the system. From there, a final list of contributions that summarizes what this research has made is presented. Finally, future work that can be done to further address the problems is identified.

7.1 Summary

A summary of this thesis is now presented. First, the initial overall goal is restated, followed by an approximation of how well the IRS validates the initial hypothesis.

7.1.1 Initial Expectation

The aim of this thesis was to expand on the paradigm of Intention Recognition as applied in a strategic environment by Gross (1991) and Akridge (2005). The goal was to enhance Akridge's IRS from a temporally static state space to a temporally dynamic one. Doing so required the expansion of the TBI paradigm of Drewes (1997) and Gerber (2001) to enable a template to consider multiple states of the environment instead of just one.

However, by simply generating a few ply of future states to be considered along with the current state (ply 0), the template moderator could easily be confused with what the agent intends to do. Therefore, some additional measure had to be taken to chronologically separate the states. The dampener was thus implemented as a means of limiting the maximum amount of confidence that a template could gain from the analysis of a future state. This value is then raised by order of the ply number that the state is in, which allows even less confidence to be contributed from any state too distant from the current state.

Based on this method of future state management, the templates were hoped to gain enough confidence from future states to push those almost above the TBI moderator's threshold over into candidacy.

7.1.2 Testing Results

The testing phase sought to prove the influence of the dampener as a means of establishing a relative priority for attributes activated at different times. Those activated at later states provided a lower fraction of their weight to their respective templates than those activated in states nearer in time to the present. The proper value for the dampener to allow for an optimal percentage of the maximum weight through (if any) had to be found through testing.

The three tests were designed so that they would have increasing complexity (test #1 being the easiest). The first test was a complete success, as it also was the first test for Akridge (2005) and his temporally static IRS. The test achieved a 100% success rate for all three phases because the intended plan did not require any future states for making its decision. Still, this test was vital in its role of establishing this thesis's enhanced IRS as being at least as competent as Akridge's (2005). The second test addressed the shortcoming that Akridge (2005) identifies as future work in his thesis. For this thesis, however, the change of making a simple move become hidden produced interesting results. The test results showed success rates of 0%, 50%, and 0% for phases 1, 2, and 3 respectively. In the first phase, high dampeners blocked all recognition, and in the second phase low dampener values made some correct recognition. The third phase, however, helped conclude that a template with only one attribute is difficult to detect because the entire confidence is dependent on a single condition being met. This theory is validated with test # 3 which tests templates that are complex. Each one has twelve attributes, thus allowing for the template to still be recognized even if several are not activated. The success ratings for all three phases were 100%, 50%, and 75% for phases 1, 2, and 3 respectively. The lower performance in phase 2 can be accredited to the agent's intentions being secondary to others with erroneously high confidence ratings.

7.2 Conclusions

The tests overall produced a mixed bag of results. With a perfect score, the first test proved that this system's IRS is at least as functional as that of Akridge (2005).

7.2.1 Interpretation of Results

The results from the third test validated the hypothesis by proving that the analysis of future ply do serve to increase the confidence of templates to the point of recognition. Additionally, those that were recognized in ply 0 or ply 1 usually had a confidence boost by ply 2. The performance numbers mentioned in the previous section also support the theory that the dampener has a strong role in dictating how many tactics can be seen as candidates by the PTBI moderator.

The second test, however, test exposes the weakness of this system. The strong performance of the third test is proof that simple templates do not fare well when facing the dampening effect on their attribute activating in a future state. Because a simple attack cannot be broken down into several attributes, it is impossible for the system to reliably detect it apart from the other possible attacks with the current setup.

Overall, the IRS presented in this thesis has demonstrated that it can perform as well as was hypothesized earlier. However, there is now a new issue that needs to be addressed. While it has been established that more attributes generally make a template more easily and accurately recognized, there are some templates that cannot be broken down beyond the satisfaction of a single condition, such as one piece attacking another.

7.2.2 Applicability of Advanced Template-based Interpretation

It is believed that the chess environment used to test the enhancement of TBI could serve as an analog to environments such as actual battlefields, where there are no artificial boundaries for movement, and the capabilities of individual soldiers are much broader than those of chess pieces. Still the concept of individual units taking certain roles remains constant. For instance, medics, tank drivers, riflemen and the like typically act in certain distinct ways, just like how pawns, knights, and rooks all behave differently.

7.3 Future Work

Although it may be a domain issue, the issue of a template having trouble overcoming the confidence threshold because of the dampener effect could still pose a problem in a conflict outside the realm of a chess game.

7.3.1 Template Scope Dampener

One possible solution to this issue, as briefly discussed in Chapter 6, could be a new redesign of the dampener mechanism to better support the needs of individual templates. On top of a global dampener enforced upon the entire template library, there could be some other local "suppression" value associated with individual templates as they are instantiated. Templates such as those that recognize attacks could be given a smaller value for its "suppressor," which would be added on to the global dampener to assess the weight for an attribute when it is activated. If the global dampener is kept low, (just high enough to keep the relative weight contributions of the current state above that of any future states) then the simpler templates with fewer attributes could be properly recognized of their "suppressor" value is low enough. The goal of this alteration of dampening is to give templates the power of determining their own "fair level" of weight reduction versus a single global value being enforced on all templates.

This change could be likened to the queue for a roller coaster at a theme park, with a simple template with a low "suppressor" being viewed as a handicapped guest. The majority of the people in the line would then be like templates with normal (high) "suppressor" values since they have ample attributes and do not need the extra help. The handicapped person is then naturally allowed on the attraction first, since it has a higher priority by comparison. Similarly, the template with a low suppressor is biased with a larger relative weight percentage than the other templates because it would have a tough time being recognized otherwise, just like how a handicapped person would have a difficult time maneuvering through the normal queue for the theme park attraction.

7.3.2 Attribute Scope Dampener

Another possibility for improving the system could be introducing a dampener at the level of individual attributes within each template. Similar to the addition of the "suppressor" at the template scope of the TBI library, the attribute dampener would provide another level in the flexibility of the IRS's method for future weight reduction. The only flaw with this design would be that attributes might have to be proprietary to their respective templates (one attribute cannot be shared with multiple templates), since its activation for one template could be relatively more important than its activation for another. The decision of how to value each attribute's dampener could also conceivably be a difficult one. Careful consideration would have to be taken in not overpowering an attribute nor or underestimating its importance for that matter.

APPENDIX A: TEST DATA

PHASE # 1		
TEST # 1		
SUBJECT #	INTENTION	RESULT
1	Attack enemy knight at f6	Success
2	pawn (b4) vs. pawn (c3)	Success
3	White Pawn (d4) vs. Black Pawn (c5)	Success
4	White Pawn (f4) vs. Black Knight (e5)	Success
TEST # 2		
SUBJECT #	INTENTION	RESULT
1	Hidden attack with Bishop at g2 attacking black rook at a8.	Failed
2	knight (f4) could attack pawn (f5) after one move	Failed
3	White Bishop (c4) vs. Black Knight (c6)	Failed
4	White Rook (f3) vs. Black Knight (h5)	Failed
TEST # 3		
SUBJECT #	INTENTION	RESULT
1	Lower Right Board Control	Success (after 3 moves)
2	Lower Center Board Control	Success (after 5 moves)
3	Lower Center Board Control	Success (after 2 moves)
4	Lower Right Board Control	Success (after 6 moves)

PHASE # 2		
TEST # 1		
SUBJECT #	INTENTION	RESULT
1	Attack enemy knight at f6	Succeeded
2	pawn (b4) vs. pawn (c3)	Succeeded
3	White Pawn (d4) vs. Black Pawn (c5)	Succeeded
4	White Pawn (f4) vs. Black Knight (e5)	Succeeded

Test # 2		
Subject #	Intention	Result
1	Hidden attack with Bishop at g2 attacking black rook at a8.	Succeeded
2	knight (f4) could attack pawn (f5) after one move	Failed (too many false positives)
3	White Bishop (c4) vs. Black Knight (c6)	Failed (too many false positives)
4	White Rook (f3) vs. Black Knight (h5)	Succeeded
Test # 3		
Subject #	Intention	Result
1	Lower Right Board Control	Succeeded (after 3 moves)
2	Lower Center Board Control	Failed
3	Lower Center Board Control	Failed
4	Lower Right Board Control	Succeeded (after 1 move)

PHASE # 3		
Test # 1		
Subject #	Intention	Result
1	Attack enemy knight at f6	Succeeded
2	pawn (b4) vs. pawn (c3)	Succeeded
3	White Pawn (d4) vs. Black Pawn (c5)	Succeeded
4	White Pawn (f4) vs. Black Knight (e5)	Succeeded
Test # 2		
Subject #	Intention	Result
1	Hidden attack with Bishop at g2 attacking black rook at a8.	Failed
2	knight (f4) could attack pawn (f5) after one move	Failed
3	White Bishop (c4) vs. Black Knight (c6)	Failed
4	White Rook (f3) vs. Black	Failed

	Knight (h5)	
Test # 3		
Subject #	Intention	Result
1	Lower Right Board Control	Succeeded (after 2 moves)
2	Lower Center Board Control	Failed
3	Lower Center Board Control	Succeeded (after 3 moves)
4	Lower Right Board Control	Succeeded (after 2 moves)

APPENDIX B: TABLE OF DAMPENERS

Maximum Weight per Ply (%)				
Base Dampener (%)	Ply 0	Ply 1	Ply 2	Ply 3
0.05	1	0.95	0.9025	0.857375
0.1	1	0.9	0.81	0.729
0.15	1	0.85	0.7225	0.614125
0.2	1	0.8	0.64	0.512
0.25	1	0.75	0.5625	0.421875
0.3	1	0.7	0.49	0.343
0.35	1	0.65	0.4225	0.274625
0.4	1	0.6	0.36	0.216
0.45	1	0.55	0.3025	0.166375
0.5	1	0.5	0.25	0.125
0.55	1	0.45	0.2025	0.091125
0.6	1	0.4	0.16	0.064
0.65	1	0.35	0.1225	0.042875
0.7	1	0.3	0.09	0.027
0.75	1	0.25	0.0625	0.015625
0.8	1	0.2	0.04	0.008
0.85	1	0.15	0.0225	0.003375
0.9	1	0.1	0.01	0.001
0.95	1	0.05	0.0025	0.000125

APPENDIX C: TEMPLATE ATTRIBUTES

Lower Left Board Control Template Attributes		
Attribute Name	Weight (one attacker)	Weight (two attackers)
Attacking e5	1 / 18	1 / 36
Attacking f5	1 / 18	1 / 36
Attacking g5	1 / 18	1 / 36
Attacking h5	1 / 18	1 / 36
Attacking e4	1 / 18	1 / 36
Attacking f4	1 / 18	1 / 36
Attacking g4	1 / 18	1 / 36
Attacking h4	1 / 18	1 / 36
Attacking e3	1 / 18	1 / 36
Attacking f3	1 / 18	1 / 36
Attacking g3	1 / 18	1 / 36
Attacking h3	1 / 18	1 / 36

Lower Center Board Control Template Attributes		
Attribute Name	Weight (one attacker)	Weight (two attackers)
Attacking c5	1 / 18	1 / 36
Attacking d5	1 / 18	1 / 36
Attacking e5	1 / 18	1 / 36
Attacking f5	1 / 18	1 / 36
Attacking c4	1 / 18	1 / 36
Attacking d4	1 / 18	1 / 36
Attacking e4	1 / 18	1 / 36
Attacking f4	1 / 18	1 / 36
Attacking c3	1 / 18	1 / 36
Attacking d3	1 / 18	1 / 36
Attacking e3	1 / 18	1 / 36
Attacking f3	1 / 18	1 / 36

Lower Right Board Control Template Attributes		
Attribute Name	Weight (one attacker)	Weight (two attackers)
Attacking a5	1 / 18	1 / 36
Attacking b5	1 / 18	1 / 36
Attacking c5	1 / 18	1 / 36
Attacking d5	1 / 18	1 / 36
Attacking a4	1 / 18	1 / 36
Attacking b4	1 / 18	1 / 36
Attacking c4	1 / 18	1 / 36
Attacking d4	1 / 18	1 / 36
Attacking a3	1 / 18	1 / 36
Attacking b3	1 / 18	1 / 36
Attacking c3	1 / 18	1 / 36
Attacking d3	1 / 18	1 / 36

Left Fianchetto Template Attributes	
Attribute Name	Max Weight
Pawn resides at f2	1 / 9
Pawn resides at g3	1 / 9
Pawn resides at h2	1 / 9
Bishop resides at g2	2 / 3

Right Fianchetto Template Attributes	
Attribute Name	Max Weight
Pawn resides at a2	1 / 9
Pawn resides at b3	1 / 9
Pawn resides at c2	1 / 9
Bishop resides at b2	2 / 3

Square Attack (1 → 64)	
Attribute Name	Max Weight
Piece at square is under attack	1

LIST OF REFERENCES

1.) Akridge, Cameron. "Intention Recognition in a Strategic Environment." *University of Central Florida*. 2005.

2.) Atkinson, G. Chess and Machine Intuition. Exeter, England and Intellect Books, 1998.

3.) Calistri, R.J. "A Modified A* Algorithm for Robust Plan Recognition." *IEEE International Workshop on Tools for Artificial Intelligence, 1989. Architectures, Languages and Algorithms.* Fairfax, VA, USA. Oct 1989. p. 36-42.

4.) Camilleri, G. "A Generic Formal Plan Recognition Theory." *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on.* Bethesda, MD, USA. 1999. p. 540-547.

5.) Campbell, M. "Knowledge Discovery in Deep Blue." *Communications of the ACM*. v. 42 no. 11. November 1999. p. 65-67.

6.) Coursey, D. "Deep Thought is a Chess-Playing Computer." MisWeek. April 2, 1990, Vol. 14, p 33.

7.) Drewes, P. *Automated Student Performance Monitoring In Training Simulation*. University of Central Florida. 1997.

8.) Esterin, Y. & Panov, V. N. Comprehensive Chess Openings. v. 1. Pergamon Press Inc. Elmsford, NY, U.S.A. 1980.

9.) Esterin, Y. & Panov, V. N. Comprehensive Chess Openings. v. 3. Pergamon Press Inc. Elmsford, NY, U.S.A. 1980.

10.) Fogel, D. B. "Evolving a Checkers Player Without Relying on Human Expertise." *Intelligence.* v. 11 no. 2 (2000). p 20-27.

11.) Gerber, W. J. *Real-Time Syncronization of Behavioral Models with Human Performance in a Simulation*. University of Central Florida. 2001.

12.) Gobet, F. "A Pattern Recognition Theory of Search in Expert Problem Solving." *Thinking and Reasoning.* November 1997, Vol. 3 Issue 4, p291, 23p

13.) Gonzalez, A. J. & DeMara, R. F. & Gerber, W. J. & Georgiopoulos, M. "Context-driven Near-term Intention Recognition." *Journal of Defense Modeling and Simulation*. August 2004, Vol. 2 Issue. 3, p153-170.

14.) Goodman, B. A. & Litman, D. J. "Plan Recognition for Intelligent Interfaces." *Sixth Conference on Artificial Intelligence Applications*. Santa Barbara, CA, USA 1990. p. 297-303 vol.1.

15.) Gross, D. L. *Intelligent Learning in a Game Based Simulation Learning Enviornment.* University of Central Florida. 1991.

16.) Gu, Wen-Xiang, & Cao, Chun-Jing, & Guan, Wei-Zhou. "Hostile Plan Recognition and Opposition." *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*. Changchun, China. Aug. 2005. Vol. 7. p. 4420 – 4426.

17.) Huff, Karen E. & Lesser, Victor R. "A Plan-based Intelligent Assistant that Supports the Software Development Process." *IEEE Transactions on Reliability*. Mar 2000. p. 26-36. vol. 49. Issue 1.

18.) Imura, A. & Takagi, T. & Yamaguchi, T. "Intention Recognition using Conceptual Fuzzy Sets." Second IEEE International Conference on Fuzzy Systems. San Francisco, CA, USA 1993. p. 762-767 vol.2.

19.) Kalles, D. & Kanellopoulos, P. "On Verifying Game Designs and Playing Strategies Using Reinforcement Learning." *Symposium on Applied Computing Proceedings of the 2001 ACM symposium on Applied Computing.* Las Vegas, United States. 2001. p. 6-11.

20.) Litman, Diane J. & Allen, James F. "A Plan Recognition Model for Clarification Sub Dialogues." *Annual Meeting of the ACL*. Stanford, California. 1984. p. 302 - 311

21.) Mao, W. & Gratch, J. "A Utility-Based Approach to Intention Recognition." *Proceedings of the AAMAS Workshop on Agent Tracking: Modeling Other Agents from Observation.* July, 2004, New York City.

22.) Qin, X. & Lee, W. "Attack Plan Recognition Using Causal Networks." *Annual Computer Security Applications Conference*. December 2004, Tucson, AZ.

23.) Rosin, C. D. & Belew, R. K. "A competitive approach to game learning." *Annual Workshop on Computational Learning Theory Proceedings of the ninth annual conference on Computational learning theory.* Desenzano del Garda, Italy. 1996. p. 292-302.

24.) Samuel, A. L. "Some Studies in Machine Learning using the Game of Checkers." *IBM Journal of Research and Development.* 1959 p. 210-229.

25.) Schmidt, C. F. & Sridharan, N. S. & Goodson, J. L. "The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence." Artificial Intelligence: An International Journal. Aug. 1978. Vol. 11. p.45-84.

26.) Sukthankar, G., & Sycara, K. "A cost minimization approach to human behavior recognition. In Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)." *Proceedings of Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems.* July, 2005.

27.) Suzić, R. "A Generic Model of Tactical Plan Recognition for Threat Assessment." *Swedish Defense Research Agency*. Stockholm, Sweden. 2005.

28.) Suzić, R. "A Generic Model of Plan Recognition Using Embedded Simulations, Microeconomics, and Behavioral Models." *Swedish Defense Research Agency*. Stockholm, Sweden. 2006.

29.) Tahboub, K.A. "Intention Recognition of a Human Commanding a Mobile Robot." Cybernetics *and Intelligent Systems, 2004 IEEE Conference on*. Dec. 2004. Freiburg, Germany. Vol. 2. p. 896 – 901.

30.) Yin, Ming-Hao, & Gu , Wen-Xiang, & Lu, Ying-Hua. "Incorporating goal recognition into human-machine collaboration." *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on.* Changchun, China. Aug. 2004. Vol. 3. p. 1429 – 1434.