HUMAN BEHAVIOR REPRESENTATION IN PHYSICAL SECURITY SYSTEMS SIMULATION

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Volkan Ustun

Certificate of Approval:

Alice E. Smith
Professor
Industrial and Systems Engineering

Jeffrey S. Smith, Chair
Professor
Industrial and Systems Engineering

Levent Yilmaz
Associate Professor
Computer Science and Software Engineering

George T. Flowers
Dean
Graduate School

HUMAN BEHAVIOR REPRESENTATION IN PHYSICAL SECURITY SYSTEMS SIMULATION

Volkan Ustun

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
May 9, 2009

HUMAN BEHAVIOR REPRESENTATION IN PHYSICAL SECURITY SYSTEMS SIMULATION

Volkan Ustun

____

Signature of Author

____

Date of Graduation

DISSERTATION ABSTRACT

HUMAN BEHAVIOR REPRESENTATION IN PHYSICAL SECURITY SYSTEMS SIMULATION

Volkan Ustun

Doctor of Philosophy, May, 9, 2009
(M.S., Middle East Technical University, July 2000)
(B.S., Middle East Technical University, July 1997)

236 Typed Pages

Directed by Jeffrey S. Smith

Physical security systems are designed to prevent access to a facility by intruders, detect the presence of intruders, or facilitate the capture of intruders once they are detected. These systems generally include a combination of physical barriers, human guards, and sensor-based detection systems such as video surveillance systems. Because of the complex interactions between guard, intruder, and neutral entities as well as the interactions between these entities and the environment, analysis of these systems is very difficult and is often limited to static "line of sight" and "field of view" models designed to help with camera placement. Existing simulation-based analysis methodologies include only crude and often hard-coded implementations of human behaviors for the guard, intruders, and neutrals. This limits the analysis capabilities of these systems. In response, this research develops a computational framework that supports realistic computer characters (or agents) that can operate within physical security system simulations. The outputs of these simulations can then be used to analyze the effectiveness of the tested physical security system configurations and to design more effective physical security systems.

The proposed computational framework is comprised of three components: a spatial model, a temporal model, and a representation of the application domain. As the spatial model, a conceptual data model named Hierarchical Graph Representation for Scenes (HIGHRES) is developed to formally represent the static features of the environment in a simulation-friendly structure. A Behavior-Intuition Framework for Realistic Agents (ABIRA) is devised as a temporal model to realistically model the decision making activities of the agents. A retail store security system is selected as the sample application domain to demonstrate the capabilities of the proposed framework and furthermore, to validate the behavior emerging from the proposed computational models. The primary contribution of this work is twofold: a generic, extensible computational framework to emulate realistic human decision making and the integrated physical security systems simulation framework. This integrated simulation framework is capable of conducting simulation experiments to analyze the effectiveness of different physical security configurations that are comprised of both the physical security measures themselves and the security policies that manage them.

Style manual or journal used: Auburn University Graduate School: Guide to preparation and submission of theses and dissertations

Computer software used: Java 5.0, Eclipse 3.1, MATLAB, Microsoft Word, Excel, PowerPoint and Visio

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Humans started to use physical security to protect their valuables from threats beginning from earlier ages. They have built fences to protect their crops from wild animals and erected scarecrows to scare away birds as primitive forms of physical security. Later on, watchtowers were constructed to keep areas under surveillance, walls were used to prevent trespassing, and locks were designed to prevent unauthorized access. While more and more technologically advanced physical security mechanisms have been introduced over the years, the main principle of applying physical security mechanisms in layers is still commonplace.

Protection of critical national infrastructure has often been taken for granted by countries despite its importance. Until recently, it had been unthinkable for anyone to purposely destroy power-plants or to contaminate water supplies (Lewis, 2006). However, the developments happening on the ever-changing world made way for the unthinkable to happen. On different parts of the world, there have recently been attacks against the different types of infrastructure such as sabotaged oil pipes or bombed subway stations. These incidents underlined the necessity for protecting the critical infrastructure. The U.S. Presidential Decision Directive of 1998 defines critical infrastructure as follows[1]:

"Critical infrastructures are those physical and cyber-based systems essential to the minimum operations of the economy and the government. They include, but are not limited to, telecommunications, energy, banking, and finance, transportation, water systems and emergency services, both governmental and private."

The 2003 U.S. National Strategy document further discusses the issues related with security of critical infrastructure within the homeland security concept[2]. This document recommends developing protection standards, guidelines, and protocols across sectors and jurisdictions and it promotes the exchange of critical infrastructure and key asset protection best practices and vulnerability assessment methodologies. It is also stated that potential options for incentives should be explored to encourage public and private sector entities to devise solutions to their unique protection impediments, which can be facilitated by conducting demonstration projects and pilot programs.

Lewis (2006) states that the vastness and the complexity of the critical infrastructure make the protection of everything almost impossible. Based on this fact, the general approach to critical infrastructure protection has involved representing the critical infrastructure as a network (Barabasi, 2002) and then to identify the critical nodes and links in this network as the candidates for allocating funds to increase security. It is important to assess the vulnerabilities of different components of the network in order to employ analytical techniques such as model-based vulnerability analysis, fault-tree

[1] www.fas.org/irp/offdocs/pdd/pdd-63.htm
[2] http://www.whitehouse.gov/news/releases/2003/02/20030214-7.html

analysis, event tree analysis etc. that are used to analyze the vulnerability of the whole network (Lewis, 2006). Outcomes of this analysis can then be used to determine the best allocation of resources to the nodes of the network. The network based approach in protection of physical critical infrastructure is very similar to computer network vulnerability analysis and this approach is widely used in the vulnerability analysis of critical physical infrastructure such as ports, water supply systems, telecommunication networks etc. as well as in the analysis of terrorist organizations and networks (Amin, 2002; Lambert and Sarda, 2005). Even though this global approach provides guidelines to determine the critical components (nodes) on the whole sector network, there is still a critical need for methods to assist the reduction of vulnerabilities at the component (node) level.

Typical components in the sector networks include fixed-site facilities such as airports, water treatment plants, dams, power plants, government facilities, etc. Physical protective systems are built at the critical fixed-site facilities to reduce the vulnerability of these critical structures. Game theoretic models (Heal and Kunreuther, 2005), optimization techniques (Candalino et al., 2004), and other analytical techniques can be used in the design of the physical protective systems. These techniques require models to evaluate the effectiveness of the security measures of concern against the vulnerabilities to which the systems are subject. However, these systems are complex and it is difficult to analytically capture the interactions between the entities of importance. A similar perspective is emphasized in 2003 U.S. National Strategy and the use of descriptive methodologies (namely modeling and simulation) is promoted for solving some of the problems in this area.

The overall goal for this research is assessing the vulnerabilities of fixed site facilities against both external and internal threats. Physical security systems are the main tools to achieve security at these facilities and it is crucial to evaluate the effectiveness and robustness of these systems against possible threat scenarios. Complex interactions between entities are inherent in these systems since security measures and threats compete with each other to achieve their individual goals: to protect the facility and to cause harm to the facility, respectively. Analytical modeling techniques appear to be difficult to use in this type of analysis, due to the complexity of these systems. Furthermore, as discussed in Section 2.2, some of the assumptions for human rationality made by classical decision theories do not hold for real life problems. Real life situations are complicated; the environments are uncertain and dynamic, problems are ill structured, there is not ample time to perform decision making activities, and multiple players exist. Therefore, simulation as a descriptive tool looks like a good candidate to perform this analysis.

In this research, modeling and simulation are used to design and analyze the physical security systems, protocols, and policies that aim to protect fixed-site facilities against intrusions by external threats, as well as unauthorized acts by insiders. However, the current understanding of physical security systems shows that realistic and credible simulations of such systems require incorporation of complex human behavior models. Therefore, it is also necessary to address human behavior representation issues for physical security simulations in fixed site facilities.

Human behavior models have long interested the researchers from several academic disciplines. Researchers in artificial intelligence are interested in development of computational models of machines or humans and they mainly focus on modeling the problem solving skills of humans. Philosophers and cognitive psychologists are interested in the relation between mind and action whereas decision theorists have worked on mathematical models of uncertainty, risk, and utility. These models provide methods to determine the optimal of the competing alternatives. Psychologists have approached the human behavior domain from the perspective of tendencies and customs humans demonstrate in different settings of real life situations and they have analyzed how these tendencies are affected by the environment.

Simulation models developed for this analysis are used to investigate the system performance in settings which resemble real life situations. This necessitates modeling the behavior of human entities (or agents) close to their real life behavior. In real life settings, humans can make mistakes, they can make awkward decisions, and they can come up with courses of action that are different from courses of action generated with extensive analysis. These tendencies are often exacerbated in the presence of stress. The assumption is that models of the human reasoning process should incorporate the shortcomings and fallacies of human reasoning as well as with its ability to generate quick solutions that are “good enough”. This assumption is crucial to correctly and completely analyze physical security systems.

Realistic and credible simulations of physical security systems require incorporation of human behavior models that involve situation awareness, cooperative team behavior, planning, and deliberative decision making processes of entities. Two interacting subsystems need to be designed in order to achieve the desired capabilities in human behavior representation (Kuehne et al., 2005; Brantingham, 2005). The first subsystem is the computational representation of the environment in which simulation entities operate and the computational models to represent the spatial interactions of entities with the environment. The second subsystem encapsulates the temporal aspects of the human representation including capabilities to generate possible courses of action and to reason about them in order to create realistic behavior. This research addresses both of these subsystems.

Human behavior models are extensively used in computer applications such as computer games. As discussed in Section 2.1, the human behavior models developed for computer games and other computer applications assume certain conditions, some of which conflict with the purposes of this research, and these models require specific knowledge of the domain. Unfortunately, there are no human behavioral models identified so far for physical security systems to protect critical infrastructures. Moreover, there is not much available data on the security of critical infrastructures and most of the existing data is classified. The lack of scenarios, models and behavioral data for critical infrastructure security systems and limitations in accessing the available information require looking at other fields to validate the developed human behavior models. One area of particular interest is criminology, which is the scientific study of crime as an individual and social phenomenon[3]. General knowledge in this domain is publicly available and various theories on the motivations and behavior of offenders both from

[3] http://en.wikipedia.org/wiki/Criminology

psychological and sociological perspectives have been developed (Cornish and Clarke, 1986; Bratingham, 1984). Computational criminology is one of the emerging fields in this domain. The general arguments of Brantingham et al. (2005) overlap with the general research directions in the physical security systems simulation. It is stated that:

> "Due to the increasing complexity and dynamics and the intricate nature of the underlying sociological systems, empirical deduction is not sufficient any more: mathematical and computational models are needed for reasoning about most likely scenarios."

The crime of shoplifting is an interesting subdivision in crime studies and countermeasures against shoplifting mainly include physical security systems (Clarke, 2003). Shoplifting is a major problem for the retail industry and annual loss of U.S. retail stores due to shoplifting is estimated to be $10.5 billion (Hollinger and Langton, 2004). What makes shoplifting interesting is that the interaction between the offenders and the security measures show significant resemblance to the general physical security structure presented by Smith et al. (1999) for the protection of fixed-site facilities. In a typical shoplifting scenario, offenders try to take merchandise from retail stores and to leave the store without getting caught by any of the security measures present in the store. Security measures used in retail stores are also similar to the security measures used in protecting the critical infrastructures such as guards, surveillance cameras, and sensors (e.g. electronic article surveillance). Field studies conducted in retail stores provide valuable information on the methods employed by shoplifters along with different classifications for shoplifting offenders (Dabney et al., 2004; Ray, 1987; Moore, 1984).

The retail store security problem is selected as a sample problem to introduce the proposed model since the problem has relatively easy to understand entity interactions. This would allow mainly focusing on the conceptual development of a human behavior model, which can later be used to analyze different environments. The hypothesis here is that if a simulation-based problem solving environment and associated decision support tools that assist the general facility and security system design problems can be developed, they can be validated to a certain extent using the retail industry's shoplifting problem as a test bed.

The main objectives of this research are first to develop computational models of human behavior that can be incorporated in general physical security systems simulation models and then to demonstrate that these simulation models can be used to analyze the effectiveness of different configurations of physical security systems. The first objective requires understanding the human behavior at a certain abstraction level. In general terms, computational human behavior models need to encapsulate both the temporal aspects of the human decision making process and the perceptions from the environment on which the entities operate. Therefore, computational representations of the environment and the interactions between the entities and the environment are integral to this research. Two conceptual models are developed in this dissertation to fulfill these requirements. First, a computational data model named Hierarchical Graph Representation for Scenes (HIGHRES) is developed to formally represent the environment. Second, a temporal conceptual model named A Behavior-Intuition Framework for Realistic Agents is developed as a model of decision making for agents in realistic environments. These models are tested and validated using the shoplifting problem in the retail industry as a sample application domain. Furthermore, retail store security domain is used to demonstrate capabilities of the developed conceptual models in the analysis of physical security system configurations.

Background on this research is presented in Chapter 2. Chapter 3 introduces the research done. Chapter 4 presents the details of this research and Chapter 5 discusses the validation of the models and the results for the sample application.

2 BACKGROUND

This section provides the background for the research conducted. It starts with the discussion of various human behavior models used in computer applications and their significance for the interests of this research. The next section introduces the perspective of various academic fields such as philosophy, cognitive psychology, artificial intelligence and economics on human rationality and human reasoning mechanisms. Based on these perspectives, agent architectures that are designed to computationally model human behaviors are identified in Section 2.3. A literature survey on shoplifting in retail industry is provided to construct the domain knowledge on the sample problem in Section 2.4. Finally, a summary of the state of the art is presented to conclude the section.

2.1 Human Behavior Models in Related Computer Applications

There are two primary uses of simulation technology in vulnerability analysis of fixed site facilities: (i) to predict the consequences of possible threats, and (ii) to assess the vulnerability of fixed-site facilities to specific threats.

Consequences of possible threats can be estimated by using simulation technology. Visual Interactive Site Analysis Code (VISAC) developed at Oak Ridge National Laboratories is an example application to predict and analyze the outcomes of accidents/incidents at various nuclear and industrial facilities[4]. Another example is the MELCOR software developed at Sandia National Laboratories that models the progression of accidents in light water reactor nuclear power plants[5]. Network oriented abstract representations of the facility and possible threats can also be used in simulations to estimate the success probabilities of threats (Jones et al., 2005). These models are capable of analyzing the consequences of several incidents at different abstraction levels but they do not interactively capture the effects of counter measures and hence they do not include representative human behavior models.

As stated in a recent National Science Foundation report, “Modeling and simulation technology can allow the decision makers to predict not only the consequences of threats but also the effects of counter measures” (Oden et al., 2006). This capability can be achieved in simulations by modeling the interactions between the threats and security measures. Discrete-event simulation has been proposed as an alternative to assess the vulnerability of fixed-site facilities by exploring the responses of a given physical protection system under various threat scenarios at a moderate cost (Jordan et al., 1998; Smith et al., 1999). The general system structure described by Smith et al. (1999) defines intruders and guards as simulation entities for physical security discrete-event simulation. Intruder entities move through the facility in order to reach or acquire a specified target and guard entities try to detect and possibly intercept the intruder entities in order to prevent them from achieving their goals. These models provide a limited level of interaction between the simulation entities. However, entities in

[4] http://visac.ornl.gov
[5] http://melcor.sandia.gov

these models do not possess any *cognitive* capabilities. Instead, entities follow predetermined routes and act according to predetermined rules of behavior. Certain events such as unauthorized access, combat etc. follow probability distributions and this structure provides the randomness in these models.

Discrete-event simulation for manufacturing systems design and operation is a well-established field with a vast number of applications (Smith, 2003). The aforementioned discrete-event models for physical security systems demonstrate modeling approaches similar to the ones that are seen in manufacturing system simulation models. Stochastic events are the main tool to represent the uncertainties in the environment and simulation entities acting in the system have very little or no decision making capabilities. This is a reasonable approach for manufacturing systems where the main simulation entities are the parts being manufactured as parts naturally do not have any cognitive capabilities including decision making, except for some basic rules such as selecting the shorter queue. However, humans are the main entities in physical security simulations and humans interact with each other and with the environment in real life situations. Certain cognitive capabilities are required to effectively model the decisions and the responses of human entities in physical security systems simulations. Modeling approaches that incorporate predetermined action plans ignore the fact that humans can deliberate based on their perception of the environment. These deliberations may result in the adoption of actions that are more suitable under current conditions towards sought goals. This factor has a vital importance in physical security systems since interaction with other entities and also with the environment can have significant effects on the courses of action of humans in real life situations.

Models of human behavior are extensively used in computer games. Game developers are primarily interested in generating non-player characters (NPCs) that are capable of integrating a wide range of functionalities including sensing the environment, reasoning about its spatial layout, planning and executing actions, as well as communicating and coordinating with other NPCs or players (Diller et al., 2004). Most of the computer games for sale today use finite state machines to enumerate possible actions or states for computer controlled characters (Bourg and Seeman, 2004) and actions of these characters are generated by scripts that scan all the predetermined rules in order to move between different states, developed at significant expense by game programmers (Lucas and Kendall, 2006). NPC models in games use rule-based applications of different techniques that can be found in the artificial intelligence literature such as perception, memory and knowledge representations, learning, and communication in addition to the models that control spatial movements.

Even though these requirements match well with the requirements of human behavior models being considered in this research, there is one definite drawback: the gaming industry’s objective is to increase the realism *perceived by the users*, not the actual realism in NPCs (Diller et al., 2004). It is a common practice in games to employ “cheating”, to collect information on human opponents without actually perceiving them in the gaming environment (Bourg and Seeman, 2004). This information can then be used by the programming tricks that are employed by NPCs in order to increase the perceived realism and the perceived intelligence (Schaeffer et al., 2008).

There are some signs of emerging trends in the gaming industry to create NPCs that are more “real”, with psychologically valid behavior models, which are more adaptive to new situations, less predictable, and more variable (Diller et al., 2004). However, these models in computer games are yet to be implemented in the majority of computer games (Schaeffer et al., 2008). Some gaming industry experts state that what users want is predictable game play and hence it is unnecessary to go after realistic NPCs. On the other hand, there are other experts and researchers, who believe that human behavior models with cognitive capabilities and more human-like intelligence with goal-directed reasoning techniques can make games more varied and more enjoyable (Lucas and Kendall, 2006).

Similar to the objectives of this research, the main challenge for computer games is to make the NPCs act realistically in addition to looking good and moving naturally. That is, they need to plan their actions, find their way around virtual worlds, and learn from their mistakes; they need to be smart (Schaeffer et al., 2008). Laird and van Lent (2001) reports developing a real time expert system –named SOAR QUAKEBOT- on SOAR architecture that has multiple goals an extensive tactics and knowledge of the game to act as a NPC in the computer game QUAKE II. As it stated by Laird and van Lent (2001):

> “While the SOAR QUAKEBOT explores a level, it creates an internal model of its world and uses this model in its tactics to collect nearby weapons and health, and set ambushes. It also tries to anticipate the actions of human players by putting itself in their shoes (creating an internal model of their situation garnered

> from its perception of the player) and projecting what it would do if it were the human player."

Another interesting application is the A.I. of the game F.E.A.R. (First Encounter Assault Recon). The designers of F.E.A.R. gave its NPCs different goals such as patrolling, killing the player's character, and taking cover to protect their lives. Furthermore, each NPC has a set of possible actions associated with the accomplishment of each of its goals. Different NPCs can seek different goals or use different actions during the game and the combination of the actions of the NPCs creates a perception that looks intelligent that was not explicitly programmed into the game at all[6]. The advantage of this approach is that it saves the developers the burden trying to specify a response to every situation that might arise (Schaeffer et al., 2008).

As stated earlier, the emphasis in computer game AI is on the illusion of humanlike behavior for limited situations. However, projections on game AI points toward more and more realistic modeling of human characters (Laird and van Lent 2001) and there is an increased interest in academic community to make research in this area (Spronck 2005; Bryant 2006; Vannakakis 2005; Togelius 2007).

Military simulations comprise another area in which human behavior models are extensively used for both combat modeling and training purposes. Pew and Mavor (1998) collected approaches and architectures that are of concern in military simulation applications. Several areas of interest including situation awareness, planning, memory, learning, and behavior moderators along with the available architectures to model

[6] http://web.media.mit.edu/~jorkin/gdc2006_orkin_jeff_fear.pdf

individual combatant are discussed in this work. Furthermore, the authors suggest that collection of human performance data and creation of accreditation procedures are vital for models of human behavior. Sokolowski's Ph.D. work (2003), where he models the decision process of a Joint Task Force Commander, and crowd behavior models developed at Virginia Modeling, Analysis & Simulation Center (Petty et al., 2004) rely heavily on the observations made in real-world settings and the experiences of real world officers. For example, the cognitive model development for crowd behavior is supported by surveys applied to active soldiers and by extensive analyses made on the video footages of incidents in 1993 at Mogadishu, Somali, which is known as “Black Hawk Down”, and in the 1999 Seattle World Trade Organization Protest. Sokolowski also validated his model with the help of an active Joint Task Force Commander.

Another interesting example in military applications is a simulation system to model few on few air combat named Brawler. As documented by Marsh (2004), Brawler models a pilot as a fully-functional decision-making entity and targets to create realistic behavior for simulated pilots. An interesting feature of Brawler as discussed by Marsh (2004) is that each pilot has its own mental status array that contains its perceived state of the system and this perceived state is not always perfect such that the simulated may be unaware of some aircraft or missiles until it perceives them in the simulation.

There is also a software tool named EXODUS that is developed to analyze building and safety designs in fire evacuations. The EXODUS uses observations from real-life situations to create realistic people-people, people-fire, and people-structure interactions and the modeled interactions are validated using the observations in real fire evacuations[7].

2.2 Human Behavior and Decision Making: A Philosophical Discussion

2.2.1 Rationality in Human Decision Making

The explanation of cognitive abilities or capacities is a central goal of contemporary science (Cummins, 1983). Among the many cognitive capacities of interest, the ability to describe people's behavior in intentional terms (e.g. belief and desire) is of specific interest for researchers from several disciplines (Stich and Nichols, 1995). Philosophers and cognitive psychologists are interested in the relation between mind and action, decision theorists have worked on mathematical models of uncertainty, risk, and utility, and researchers in artificial intelligence are interested in development of computational models that mimic human behavior. Research made in this area helped to explore the methodologies used by humans in decision making as well as to better understand the limitations in this process. One of the main points that differentiates the researches made in different fields is their approach to the question: What is required to call an action performed by a human "rational"?

Discussion on reasonableness or rationality of human decisions can be tracked back to the 17th century and it is not a coincidence that probability theory emerged around that time. The first definition for reasonableness was to choose the alternative that maximizes expected value from the letters on gambling exchanged between Blaise Pascal and Pierre Fermat (Hacking, 1975). In the 18th century, Daniel Bernoulli proposed to

[7] http://fseg.gre.ac.uk/exodus/index.html

change the definition from maximizing expected value to maximizing expected utility, which incorporates the psychological fact that money has diminishing returns. Several new definitions of expectation such as median or geometric mean as well as the issue of variability in the 19th century put a hold on the discussions of reasonableness since mathematicians decided that it is not possible to come up with a one mathematical definition for reasonableness (Gigerenzer and Selten, 2001).

After World War II, research on economical systems picked up along with changes in the global economic conjuncture. The book named “Theory of Games and Economic Behavior” written by von Neumann and Morgenstern in 1944 is accepted as the restarting point for research on rationality. von Neumann and Morgenstern discussed human choice and utility such that a person’s choices can be modeled as always favoring the alternative with the highest expected utility. This book also triggered research in psychology where researchers started to conduct empirical tests to see if people actually behaved in the manner prescribed by the expected utility theory (Goldstein and Hogarth, 1997). The second book that had a huge influence in this domain was “The Foundations of Statistics” by Leonard J. Savage in 1954. Savage stated that a person’s choices can be modeled as always favoring the alternative with the highest subjective expected utility dependent on some constraints. This opened a new door for psychological researchers since this enlarged the scope from monetary gambles to arbitrary decisions (Goldstein and Hogarth, 1997). Utility and subjective utility are still the cornerstones of decision theory taught in universities and techniques such as decision trees or optimization are applied to economical domain using these theories.

Utility and subjective utility along with a few other metrics such as addition of utilities, addition of utility differences, elimination by aspects, dominance, number of superior features, single feature superiority are used when comparing different possible alternatives for a certain task. These aspects can be collected under the general heading “rational choice theory”. The Wikipedia definition of rational choice theory[8] is:

> “Rational choice theory is a way of looking at deliberations between a number of potential courses of action, in which ‘rationality’ of one form or another is used either to decide which course of action would be the best to take, or to predict which course of action actually will be taken.”

In this definition, rationality means that according to one of the measures stated above, one alternative is superior to others and hence it is the rational choice. Rational choice theory holds that individuals must anticipate the outcomes of alternative courses of action and calculate that which will be best for them. Rational individuals choose the alternative that is likely to give them greatest satisfaction (Scott, 2000). This is perhaps the most common way to define human rationality.

However, several psychological studies reported that humans do not always behave as dictated by rational choice theory. Initial discussions for one of the two important trends of objections to description of human rationality by rational choice theory is made by Ward Edwards (1962) and he concluded that human thought, although fundamentally probabilistic, did not exactly follow the rules of probability theory. This conclusion is later supported by findings of Amos Tversky – a student of Edwards- and

[8] http://en.wikipedia.org/wiki/Rational_Choice

Daniel Kahneman who conducted inspiring empirical research on human judgment (Goldstein and Hogarth, 1997). The second important trend of objections has been stated by Herbert Simon (Simon, 1956). He underlined the cognitive limitations of humans and first coined the term "bounded rationality". Simon discussed that humans make decisions by incorporating certain thresholds (aspiration levels) that determine the acceptable outcomes and when a "satisficing" action is found, no further exploration is performed.

Amos Tversky and Daniel Kahneman have contributed to rationality discussions probably more than any other previous researchers. Empirical studies performed by Tversky and Kahneman formally introduced heuristics and biases that affect human judgment under uncertainty. They have described three heuristics that are employed by humans to assess probabilities and to predict values (Tversky and Kahneman, 1974) and another heuristic for constructing courses of action (Kahneman and Tversky, 1982). The first heuristic they have mentioned is the "representativeness heuristic", which basically states that if event <u>A</u> is highly representative of event <u>B</u>, the probability that event <u>A</u> originates from event <u>B</u> is judged to be high. The "availability heuristic" is related with memory such that perceived probability of an event is assessed by the easiness to bring instances or occurrences of this event to mind. The "anchor heuristic" explains how humans make estimates starting from an initial point and then adjust the estimates to reach a final answer. According to Kahneman and Tversky, two classes of mental operations can be used when bringing things to mind. The first operation is recall, which is the retrieval of instances, and the second operation is construction, which is the construction of examples or scenarios. The "simulation heuristic" is a process for mental construction of scenarios and it briefly explains how humans mentally simulate actions to come up with a reasonable course of action. It is important to recall that a simulation does not necessarily produce a single story. It is rather used to assess different possible outcomes and measures the propensities of one’s model to generate various outcomes given the initial conditions and operation strategies. Tversky and Kahneman later developed the “prospect theory”, which is inspired from their discussions on aforementioned heuristics. The prospect theory aims to fulfill the shortcomings of subjective expected utility theory for choices under risk. The prospect theory uses two functions –value and weight- and an overall value of an edited prospect is expressed in terms of scales of subjective value of outcomes and decision weights. This reflects the impact of probability of outcome on the overall value of the prospect. A subjective value is defined relative to a reference point and hence it is a measure of the value of deviations from the reference point. Kahneman and Tversky hypothesize that a subjective value function for changes of wealth is often concave above the reference point and often convex below it, as depicted in Figure 1. Decision weights as stated by Kahneman and Tversky are not perceived likelihood (or probabilities) of events, Instead, they measure the impact of events on the desirability of prospects.

Herbert Simon argues that there are departures in human behavior from the prescription of subjective expected utility theory even in simple choice situations and that the principal reason for this is (Simon, 1978):

> “... that human beings have neither the facts nor the consistent structure of values nor the reasoning power at their disposal that would be required,

even in these relatively simple situations, to apply subjective expected utility theory principles."

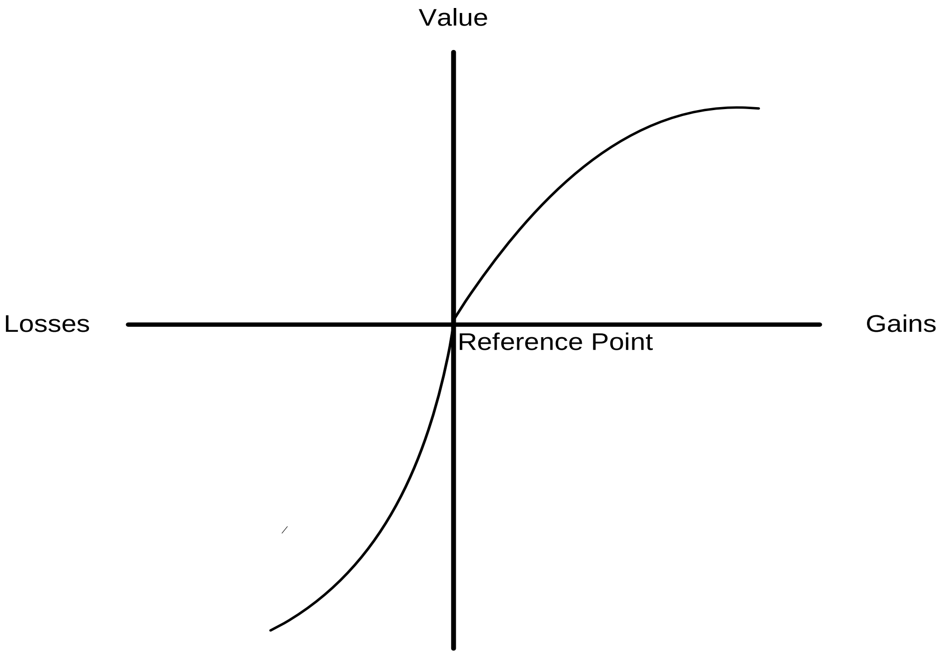


Figure 1 A Typical Value Function (Tversky and Kahneman, 1986)

Simon further discusses that a behavioral alternative to subjective expected utility theory is required to better model human behavior. He points out that even though human brainwork is sequential in nature, it can change its focus and attention as necessities emerge. This change of attention is essentially different than handling different objectives in utility theory, where concern for different objectives stays until a viable alternative is found. Humans can change attention and sometimes earlier considerations can be totally forgotten. Each time an objective is considered, there is a need for a mechanism that is capable for generating alternatives and another mechanism to acquire facts from the environment and a modest capability to draw inferences from facts. Intuitions are significant while generating alternatives and emotions play a major role while selecting objectives to consider and while choosing actions to satisfy the objectives. Furthermore, Simon states that humans do not exhaustively look for the best possible course of action, rather they try to find a “satisficing” action and implement it. This is the general framework that he has delineated as a mechanism for bounded rationality.

Arguments made by Simon, Tversky and Kahneman were significant and provided answers to the deviations in human behavior from what subjective expected utility theory dictates. Moreover, these are major remarks on why it is not possible for humans to have perfect rationality, which is to come up with a course of action that maximizes its expected utility given the information acquired from the environment at each instant, even if they have infinite processing capabilities. A behavior can be judged rational only within the frame it takes place, where the frame consists of goals, definitions of the situation, and computational resources (Tversky and Kahneman, 1986) and this is very similar to the description of bounded rationality by Simon.

Even though these views, which have arisen between the late 50s and the early 80s, are significant criticisms against rational choice theory, Elster (1990) sees them as supplements to rational choice theory. He states that the psychological theories of Simon, Tversky and Kahneman along with regret theory (Loomes and Sugden, 1982) and generalized expected utility theory (Machina, 1983) attempt to explain the observed violations of expected utility theory instead of providing a totally new alternative. Similarly, Gigerenzer and Selten (2001) state that the deviations from rational choice theory are mistakenly named as fallacies of human reasoning. Their argument is that rational choice theory does not reflect the structure and representation of information in the environment and hence it is not possible to name the deviations from what rational choice theory prescribes as irrational.

The major distinction here is what is chosen to be investigated for the rationality assumption. Rational choice theory models accept choice as rational if the outcome is rational (Zey, 1992) and they are extensively knowledge focused. They classify all behavior that does not produce a rational outcome as irrational. This might make sense for problems with limited scope (e.g. in financial domain), however, it is hard to explain or prescribe human behavior in other domains where problems lack formal objectives such as maximizing profit. However, the deviations from prescriptive models in observed human behavior do not necessarily mean that rational choice theories are completely wrong; rather, they are incomplete (Yates, 2001).

In other social sciences such as psychology and philosophy, the conceptualization of decision making is rational because of the process it employs (Zey, 1992). The emphasis in these fields is on the process by which decisions are made: rational choices are achieved by reasoning and contrasted with choices arrived at by emotion, faith, authority, or arbitrarily (Brown, 1995). This approach accepts that the outcomes of descriptive models that mimic human reasoning process are also rational even if they conflict with the outcomes of prescriptive models. Kacelnik (2006) explains the difference between the definitions of rationality in different fields by the fact that researchers guided by differing goals in various disciplines have reached different workable definitions of rationality, which have within-field consensus. Based on this observation, Kacelnik subsumes the meanings of rationality into three categories: rationality in economics (E-rationality), rationality in Philosophy and Psychology (PP-rationality), and rationality in evolutionary biology (B-rationality).

E-rationality is a conception of rationality that focuses on outcomes in the sense of maximizing of some function that is called “subjective expected utility” rather than the processes that produce them. According to this conception, only the behavior that maximizes subjective expected utility is rational and the selection of maximizing behavior alternative is restricted only by formal requirements of consistency between preferences. As discussed earlier in this section, there are several objections to this conception and there are several studies that show that human behavior is not always E-rational. Nevertheless, E-rationality is an important conception that allows the comparison of observable behavior and patterns of action based on a solid mathematical base. Furthermore, subjective expected utility is infinitely flexible and it is not tied to environments with any particular structure and it can be regarded as broadly predictive of human behavior (Hurley and Nudds, 2006).

PP rationality requires that beliefs or actions be based on reasoning according to Kacelnik’s subsumption. PP-rationality focuses on the process by which the action or belief is arrived at, in contrast to E-rationality. The assumption here is that beliefs or actions produced in appropriate ways should produce appropriate outcomes (Kacelnik, 2006). PP-rationality uses concepts from cognitive psychology and philosophy such as belief, thought, intention, reasoning etc. and a behavior is judged to be rational if it is resultant from a reasoning process that is caused by intentions and beliefs. In this sense, behaviors or actions that fail to bring about their intended outcomes for such reasons are not irrational in the conception of PP-rationality. Therefore, an individual can be rational in either PP-rationality or E-rationality while being “irrational” in the other.

The final category of Kacelnik’s subsumption is B-rationality. B-rationality is similar to E-rationality since it tries to maximize the inclusive fitness of alleles or individuals that carry these alleles, where the inclusive fitness of an allele is defined as the degree of success (growth as a proportion of the population) of individuals carrying the allele. However, B-rationality is not as flexible as E-rationality since it constrains what is to be maximized. Even though B-rationality is very specific to evolutionary biology, concepts of B-rationality are used in computational intelligence applications to generate actions that maximize certain functions.

One addition to the subsumption of Kacelnik can be “Ecological Rationality”, which is discussed by Gigerenzer and Todd (1999). Gigerenzer is one of the main critics of rational choice theory or “subjective expected utility”. He states that rational behavior is environmentally situated and the heuristics adapted to environments produce behavior with better consequences than reliance on more sophisticated and flexible but costly domain-general decision making processes. According to Gigerenzer and Todd (1999), the structure of specific environments and the information contained in specific environmental processes are highly coupled with the processes that generate rational behavior. This coupling is important in the evolution of domain specific heuristics generating rational behavior.

Hurley and Nudds (2006) specify two distinctive features for rationality. The first distinctive feature is that rationality requires some degree of ability to generalize from one context to another and it seems to involve recognition of abstract similarities or patterns. In other words, as Hurley and Nudds state:

> "The ability to generalize involves 'decentring' from me-here-now, and entertaining alternative possibilities: taking into account the past, future or counterfactual possibilities, other places, different possible actions, and other creatures' perspectives."

The second distinctive feature is the capacity to make mistakes or to recognize the fact that one can make mistakes. This implies that even in descriptive theories of rationality, there is a need for a normative aspect; possible outcomes of an action need to be assessed. This is normativity in a weak sense and it is necessary to avoid the case that anything an individual might do could count as rational.

Based on the above discussion, it is possible to define an action as rational if the individual believes that taking the action will contribute to achieving a goal that a person has. Two elements need to be underlined in this definition. First, success is not guaranteed and it is possible that the individual can make mistakes. Second, if the action does not contribute to the goal regarding the beliefs of the individual at the time, it can not be considered rational. Subsequently, the individual should be concerned with finding the best means to a given end or goal, which in turn has a logical relation with the individual's beliefs. This is similar to Bratman's definition (1987), which is formally named as the *intention-action principle:*

> “If it is rational of S to have a present-directed intention to A, and S successfully executes this intention and thereby intentionally A’s, then it is rational of S to A.”

The process of finding the best means to a given end or goal is investigated in the next section with introducing different descriptive models of human reasoning.

2.2.2 Human Reasoning

When one with an operations research background is faced with a decision making problem, the tendency is to develop a formal mathematical model of the problem that would allow reaching all possible solution alternatives. The action following the development of mathematical model is to try to find the optimal solution using one of the techniques found in operations research literature. If a mathematical model is not available, the tendency is to generate alternatives and compare them until the expected marginal cost of generating an alternative is more than the expected marginal benefit of generating the next alternative. Furthermore, the general assumption is that human reasoning works in a similar way. Bayth-Marom et al. (1991) states that:

> “… According to the most general normative model, a person facing a decision should (a) list relevant action alternatives, (b) identify possible consequences of those actions, (c) assess the probability of each consequence occurring (if each action were undertaken), (d) establish the relative importance (value or utility) of each consequence, and (e) integrate these values and probabilities to identify the most attractive cost

> of action, following a defensive decision rule. People who do so effectively are said to behave optimally.”

The statements of Bayth-Marom et al. roughly define the general boundaries of classical reasoning techniques found in the literature. Classical reasoning depends on normative processes of reasoning such as carrying out explicit probabilistic, logical, or decision theoretic inferences in order to reach decisions and judgments (Hurley and Nudds, 2006). Classical reasoning is prescriptive and it can be aimed at solving a problem, making a decision, planning a course of action, or arriving at a judgment or prediction. For example, prescriptive techniques of classical reasoning such as decision trees are exemplified by expected utility or subjective expected utility in economics. In this context, “rational choice” is the alternative that maximizes the individual’s subjective expected utility from a range of alternatives.

Tversky and Kahneman (1974) conducted research on the cases where human decision makers make decisions differently than what classical decision making process dictates. They argue that the use of heuristics in cases where there is uncertainty is the reason for this deviation since employment of heuristics creates biases in human judgment. The heuristics that Tversky and Kahneman define, such as representativeness, availability, and adjustment and anchoring heuristics, affect the evaluation of alternative courses of action. One can add the “affect heuristic” (Slovic, 2002) to the main list of heuristics that bias human judgment. The affect heuristic states that emotional responses to external stimuli might alter the human’s judgment and these emotional responses can dictate the decision. There are also other psychological heuristics defined in the literature and all of them are used to explain the cognitive biases in human decision making process for complex problems, uncertain situations and incomplete information[9] . The main argument here is that deviations from decisions that are labeled as “rational” by classical theories of reasoning can be explained by heuristics and biases. It is important to underline the fact that decisions are still considered “irrational” in this discussion if they are different than what classical decision theory dictates. However, emotions, faith, and social norms can also play a role in the decision making process (Kacelnik, 2006) and they can lead to the cases, where “rationality” in classical decision theory is systematically violated.

There have been other objections to the idea that people actually employ optimization techniques in their decision making processes. One of the stronger objections came from a group of researchers led by Gigerenzer (Gigerenzer and Selten, 2001). Their main argument is that too many assumptions have to be made in order to achieve the necessary conditions for optimization (Klein, 2001). These assumptions are mainly required to define a static search space, which is required to be able to thoroughly compare the options with each other. Gigerenzer and Selten’s approach is based on the argument that people use heuristic methods while making decisions. Their claim is that, to the contrary of arguments made by Tversky and Kahneman, employment of heuristic methods in decision making can create accurate judgments rather than producing cognitive biases. Limitations of the human mind such as limited processing power or limited memory can be overcome by the cognitive capabilities of humans according to these arguments. They propose that humans use fast and frugal heuristics in decision

[9] http://en.wikipedia.org/wiki/Heuristics

making and the success of these heuristics is dependent on their adaptation to the structure of the environment.

One of the most common examples used in describing the power of fast and frugal heuristics is the “gaze heuristic”. An example where this heuristic employed is that if a human is asked to catch a ball, he or she doesn’t perform complex mathematical calculations or solve a system of differential equations to estimate the point where the ball will land (Gigerenzer and Selten, 2001). Instead, he or she will keep the angle between the eye and the ball constant and will catch the ball while still running. Furthermore, they claim that the gaze heuristic performs better than any optimization procedure that tries to estimate the landing point of the ball since environmental conditions such as wind might not be stable and changes in these conditions results in recalculation of the ball’s landing point. Several fast and frugal heuristics are discussed such as ignorance based heuristics or take-the-best heuristics, which are sharing adaptation to the environment in common for success (Gigerenzer and Todd, 1999). It has been reported in several studies that fast and frugal heuristics give better results than optimization techniques in real world settings (Czerlinski et al., 1999).

The use of heuristics is a viable option in modeling and understanding human decision processes as demonstrated in the above discussion and by different field studies. However, discussion of the term “optimization” by some cognitive psychologists might be a little bit misleading since there are different heuristics that are proven to generate optimal solutions for certain problems. Furthermore, management science often employs heuristic algorithms to generate provably good or optimal solutions for problems where "optimality" is hard to prove. In fields like computer science or management science, heuristics are used as a methodology to solve optimization problems and as such, the terms "optimization" and "heuristic" are not mutually exclusive.

The artificial intelligence community also has a natural interest in modeling the human decision making process. Earlier studies in artificial intelligence focus on the problem solving skills of humans and these skills are modeled using extensive formal logic models that are generally applied to toy problems such as the famous prey-predator problem or to games such as chess where the search space is relatively well-structured (Russell and Norvig, 1995). However, the complexity of real life problems required a different approach in modeling the human reasoning process. In general, the artificial intelligence community assumes three stages in the human reasoning process. The first stage is situation awareness, which is described by Endsley (1997) as:

> "Situation awareness is the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future"

This is followed by coming up with a sequence of actions that will hopefully achieve a goal. This stage is known as the planning stage. Finally, a decision should be made on whether or not to implement the plan that has been constructed. This general reasoning scheme is supported by formal knowledge representation, learning, perception and learning mechanisms.

One comprehensive model that encapsulates the intuition and reasoning in the human decision making process is the two-system view that is described by Stanovich and West (2000). Human behavior scientists agree upon the characteristics that distinguish the two types of cognitive processes (Kahneman, 2002), which are labeled as System 1 and System 2, represented in Figure 2. This model also encapsulates the situation awareness, planning and decision making stages from a cognitive psychology standpoint.

The operations of System 1 in the two-system view are fast, automatic, effortless, associative, and difficult to control and modify. The operations of System 2 are slower, serial, effortful, and deliberately controlled; they are also relatively flexible and potentially rule-governed. System 1 is associated with intuition and captures the functionality of situation awareness including external stimuli processing and perception. On the other hand, System 2 is associated with judgments which are explicit and intentional and hence can be thought as the planning stage.

The two-system view model is based on the observation that complex judgments and preferences are called “intuitive” if they come to mind quickly and effortlessly. However, judgments and preferences that are normally intuitive can be modified or overridden by a deliberate mode of operation (Kahneman, 2002). The labels “System 1” and “System 2” refer to these two modes of cognitive operations. In the model presented in Figure 2, the perceptual system and the intuitive operations of System 1 generate impressions of the attributes of objects of perception and thought. System 2 is involved in all judgments and it uses these impressions while making judgments. If the judgments are direct reflections of the impressions, they are accepted as *intuitive*. Otherwise, judgments are the results of *deliberate reasoning.*

The two-system view model suggests four ways in which a judgment or choice may be made:

1. No intuitive response comes to mind, and the judgment is produced by System 2.

An intuitive judgment or intention is evoked, and

2. is endorsed by System 2;
3. serves as an anchor for adjustments that respond to other features of the situation;
4. is identified as incompatible with a subjectively valid rule, and blocked from overt expression.

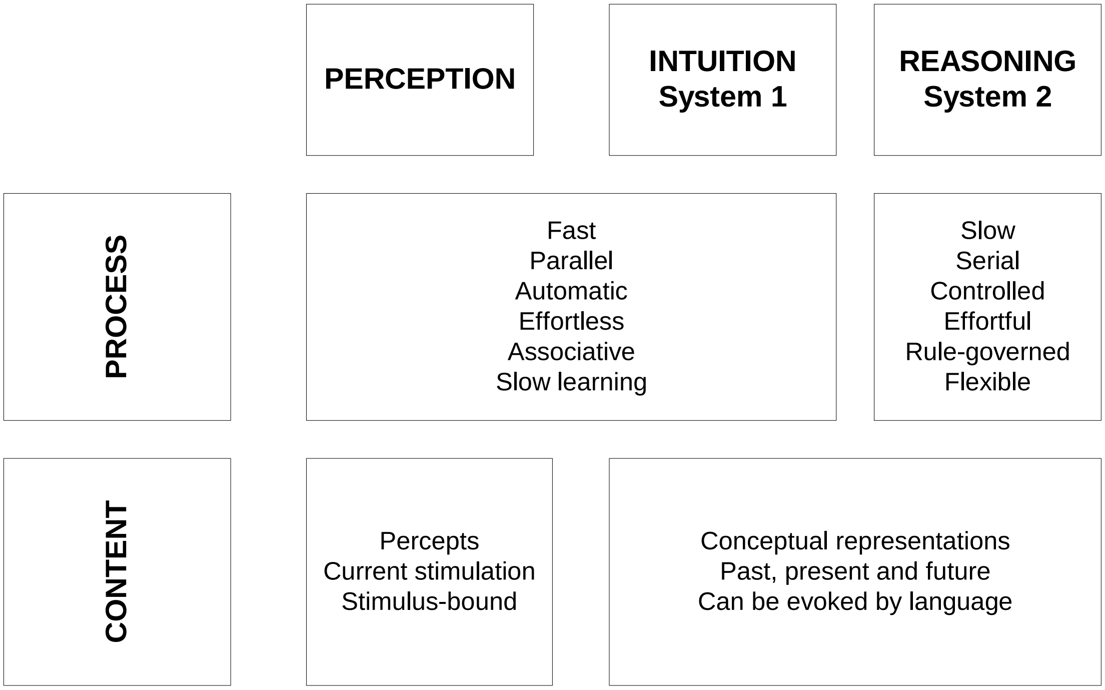


Figure 2. The two-system view (Kahneman, 2002)

This model provides a general guideline on how intuitions and judgment interact with each other from a cognitive perspective and it delineates the human reasoning system with an abstract view.

Another trend in modeling human reasoning process that emerged in the last 25 years is Naturalistic Decision Making (NDM). NDM also focuses on the use of cognitive abilities of humans in real life situations. NDM researchers stress a number of key contextual factors that affect the way real-world decision making occurs, in contrast to their counterparts in the traditional decision research paradigm (Zsambok, 1997):

1. Ill-structured problems (not artificial, well-structured problems)
2. Uncertain, dynamic environments (not static, simulated situations)
3. Shifting, ill-defined, or competing goals (not clear and stable goals)
4. Action/feedback loops (not one-shot decisions)
5. Time stress (as opposed to ample time for tasks)
6. High stakes (not situations devoid of true consequences for the decision maker)
7. Multiple players (as opposed to individual decision making)
8. Organizational goals and norms (as opposed to decision making in vacuum)

The existence of these contextual factors is the main reason why traditional optimization-based decision research fails in real world settings according to NDM researchers. NDM also focuses on the cognitive capabilities of humans and stresses the importance of experience in real world problems. Klein, who is one of the leading researchers in NDM field, uses chess grandmasters as an example to underline the factors that necessitate NDM research. He states that chess grandmasters seek to find the best move, but they do not compare options on a common set of decomposed criteria or use multi-attribute utility analysis or make probabilistic analyses (Klein, 2001). Grandmasters in chess try to see the overall lines of play and they determine a number of moves that look promising at the current state of the game using their experience. After that they progressively deepen on these moves in order to analyze possible progresses in the game. The promising moves are then evaluated based on the strategy of the grandmaster and the move that stands out among others is played.

NDM research specifically emphasizes experience and argues that experience is crucial in determining certain courses of action that probably achieve the goals and in eliminating the courses of action that are not promising. A formal definition of NDM is given by Zsambok (1997) as:

> "The study of NDM asks how experienced people, working as individuals or groups in dynamic, uncertain, and often fast-paced environments, identify and assess their situation, make decisions and take actions whose consequences are meaningful to them and to the larger organization in which they operate."

It can be said that there are certain situations where an optimization-based approach can be used in the decision making process. Examples can be purchase of real estate or economical decisions with long term implications. The common points in these examples are relatively longer times available to process the information, the relative stability of the decision maker's values, and the well-defined structure of the environment. However, in most real life situations, where there are too many uncertainties, competing goals, and limited time for decision making, there are several strong arguments for why people rely on their cognitive abilities to come up with initial plans and then deliberately search for a course of action to achieve the goals using the initial plans as anchor points.

As stated by NDM researchers, when there are multiple individuals that compete or cooperate with each other in the environment, the scope of the reasoning process changes. Classical decision theory concerns mainly “individual rationality”: what an individual should do in an environment excluding the individuals also acting in the environment. (Hurley and Nudds, 2006). However, most real life situations are more complex, where the behaviors of individuals affect other individuals and hence behavior is interactive. This social context requires the individuals to have social rationality, which is to understand that another agent may have goals and expectations different from one’s own, and the use of understanding to predict other’s behavior and to manipulate one’s own informational structure (Proust, 2006). From this perspective, game theory is a well established domain, which studies the decisions made in environments where several individuals interact. Generally, games studied in game theory are well-defined. A basic equilibrium concept is sought in game theory and this is a conception of E-rationality in a social context: one’s action is rational if it is the best reply to what others do, and others’ actions are rational if they are the best reply to what one does. Game theory can be used as a descriptive tool in predicting human’s behavior by finding the equilibria of the games or as a prescriptive tool since the equilibrium of a game constitutes one’s best response to the actions of others[10]. However, the assumption of E-rationality creates the major challenge to the possible use of game theory as a model of reasoning. As discussed in the previous section, real humans often act “irrationally” regarding E-rationality or might act to maximize the benefits of a group of people. One counter-example that is against the use of game theory as a prescriptive tool is the Prisoner’s Dilemma in which each individual pursuing their self interests leads to an equilibrium where both individuals are worse off than had they cooperated[11]. Furthermore, in certain cases, it is better to use non-equilibrium strategies if there are other individuals using non-equilibrium strategies.

Several psychologists propose simulation or the “simulation theory” in their terminology as another candidate to model the human reasoning process when social rationality is a concern (Davies and Stone, 1996). The simulation theory is based on the capability of the human brain to simulate in order to predict and understand an action by mentally processing its production in one’s own mind. Stich and Nichols (1995) state that simulation theory is one of the only two approaches of significance in description and prediction of people’s behavior. Simulation theory is based on the fact that human brains and thinking processes are similar and hence if one can make adjustments for relevant differences such as situation or emotional state then he/she can use his/her mind to simulate mental processes of others in order to describe the other’s behavior and furthermore predict these behaviors. This procedure is used cooperatively as well as

[10] http://en.wikipedia.org/wiki/Game_Theory

[11] http://en.wikipedia.org/wiki/Prisoner's_Dilemma

competitively: bridge players project themselves into their partner’s shoes whereas chess players project themselves into their competitor’s shoes. Gordon (1996) states that:

> “One source of motivation for the simulation theory starts from the recognition that, whilst the prediction of the behavior of the others may be tricky business, the prediction of our own immediate and near immediate actions is usually a simple and accurate matter.“

According to simulation theorists, using simulated practical reasoning as a predictive device is a possibility that comes via the bridge between practical reasoning and prediction. Simulating the appropriate practical reasoning would extend our capacity for self-prediction in a way that would enable us to predict our own and other’s behavior in hypothetical situations. It is important to underline the fact that putting yourself in other’s shoes is to project yourself into other’s situation but not to attempt to project yourself into the other’s mind. It is not the same as deciding what I myself would do but to try to make adjustments for relevant differences.

Simulation theory has been supported by the studies of several researchers from different fields. Tversky and Kahneman (1982) defined a heuristic named “simulation heuristic” based on their observations in laboratory settings. In this heuristic, they described how a person might build a simulation to explain how something might happen; if the simulation required too many unlikely events, the person would judge that thing to be implausible. NDM researchers also have an interest in simulation theory. George Klein, who is one of the founders of NDM research states that simulation theory or “mental simulation” as he names it, is central to decision making. He observed that people are constructing mental simulations almost the way one builds a machine (Klein and Crandall, 1995). Klein (1998) defines mental simulation as:

> "Mental simulation is the ability to imagine people and objects consciously and to transform those people and objects through several transitions, finally picturing them in a different way than at start."

An approach similar to "mental simulation", which is named "look-ahead simulation", is also applied in the real-time control and scheduling of flexible manufacturing systems. Wu and Wysk (1989) developed a discrete-event simulation based scheduling mechanism to evaluate the performance of different dispatching rules in a flexible manufacturing cell for a short planning horizon. Smith et al. (1994) applied discrete-event simulation for shop floor control for a flexible manufacturing system. In this application, discrete-event simulation models actually control the flexible manufacturing system rather than only being used as an analysis and evaluation rule. This application has an interesting resemblance to mental simulation models since it proposes discrete-event simulation as a decision making tool in real-time control of the flexible manufacturing cells. At specific intervals, real time data is collected for the shop floor and a copy of the simulation is initialized using the current physical system state. Afterwards the "look-ahead" simulation or the "copy" of the actual simulation model is run in order to evaluate the impact of different dispatching rules on the system performance. Based on this assessment, a dispatching rule is selected and the actual simulation model continues to control the flexible manufacturing cell by generating tasks for the physical equipment in the cell.

Section 2.2 discussed the views on human rationality and human reasoning in different academic fields. In general, two issues are investigated: (1) What makes the human behavior “rational”, and (2) How the reasoning process of human can be modeled. Section 2.3 discusses several computational agent architectures that can be used in modeling human behavior.

2.3 Agent Architectures

Modeling human behavior on a computer requires programming constructs that can decide for themselves what they need to do in order to satisfy their design objectives. Such constructs are generally known as *agents* and this term is shared by various disciplines even though no universally accepted definition has emerged yet. Wooldridge and Jennings (1995) define “agent” as:

> “An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives.”

Ferber (1999) describes an agent as:

> “A physical or virtual entity, which is capable of acting in an environment, which can communicate, which is driven by a set of tendencies, which possesses resource of its own, which is capable of perceiving its

environment, which has only a partial representation of this environment, and which possesses skill."

Wooldridge (1999) further distinguishes between agents and intelligent/autonomous agents. His definition of intelligent agents focuses on the flexibility in the autonomy of the agent's actions and flexibility means three things:

- *Reactivity*. Intelligent agents are able to perceive their environment, and respond in a timely manner to the changes in the environment with respect to their design objectives.
- *Pro-activeness*: Intelligent agents are able to exhibit goal directed behavior by taking the initiative in order to satisfy their design objectives.
- *Social ability*. Intelligent agents are capable of interacting with other agents in order to satisfy their design objectives.

There are several other conceptualizations of agency in various disciplines. Leaving the conceptual discussion on agency to other researchers, computational representations of humans and their behavior will be referred to as "agents" in this proposal within the general boundaries of aforementioned definitions.

An agent is characterized by its architecture. The agent architecture organizes the functions that are required by the agent to perform its actions when situated in an environment and it provides a foundation to incorporate reasoning. Here, five different architectures that are commonly used in agent systems will be discussed. These are: (1) Logic-based architectures (2) Reactive architectures (3) Belief-Desire-Intention (BDI) Architecture, (4) Recognition-Primed Decision (RPD) Model, and (5) Layered

Architectures. The first two types of architectures aim to achieve intelligent behavior without any specific emphasis on human reasoning mechanism. They are more or less based on defining a set of logical rules in order to map perceptual inputs to actions of the agent. The next two architectures have roots in philosophy and cognitive psychology. These architectures are intuitive and they provide a clear functional decomposition of reasoning into subsystems. They are more focused on replicating the human reasoning system with certain abstractions. However, the difficulty for these architectures is to know how to efficiently implement the functional decomposition in a computer program. Finally, layered architectures aim to combine reactive and pro-active behavior by creating separate subsystems to deal with these different types of behaviors.

2.3.1 Logic-based architectures

The traditional approach to building artificially intelligent systems requires having a symbolic representation of the agent's environment and its desired behavior, and syntactically manipulating this representation (Wooldridge 1999). Logic-based architectures use deductive reasoning in which the conclusion is reached from previously known facts and rules. In general, logic-based architectures use symbolic representations in the form of logical formulae, and the syntactical manipulation corresponds to logical deduction or theorem proving. An agent's decision making process is encoded as a logical theory and hence the process of selecting an action reduces to problem proof. This constitutes the sound basis of logic based architectures and the use of simple and elegant logical semantics triggered the extensive use of logic-based architectures in earlier periods of the artificial intelligence field. However, creating symbolic representations of complex, dynamic environments and constructing a logical approach on the top of these representations are extremely hard along with the problem of symbolical representation of the agent's perceptions. Furthermore, dynamic environments change during the decision making process. Therefore, an "optimal" solution for an environment state at the beginning of the decision process may be no longer optimal for the new state reflecting the changes in the environment.

One of the most well-known architectures based on symbolic representation and rules is SOAR, which is a symbolic cognitive architecture, created at Carnegie Mellon University[12]. SOAR was developed to explore the requirements for general intelligence and to demonstrate general intelligent behavior. SOAR is based on operators, which are similar to reactive plans, and states. SOAR uses explicit production rules, which are specified by a series of conditions and a set of actions, to govern its behavior[13]. A problem space is defined as a set of (possible) states and a set of operators, which individually transform a particular state within the problem space to another state in the set. The transformation on permanent basis is achieved via productions, which are defined by the pre-conditions and the set of operators to be used to reach the desired state. Problem solving can be roughly described as a search through a problem space for a goal state (or sub goal states) using operators that are distributed across productions, preferences, and memory objects within the architecture. An operator is selected when the preconditions for that operator hold and it is expected from the operator to bring the system gradually closer to its goal.

SOAR's decision cycle includes three phases:

[12] http://en.wikipedia.org/wiki/Soar_(cognitive_architecture)
[13] http://sitemaker.umich.edu/soar

1. elaboration phase involves bringing a variety of different pieces of knowledge bearing the problem to SOAR's *working memory*,
2. selection phase weighs what was found in the previous phase and assigns preferences (a preference memory is stored in SOAR in order to represent or ultimately decide the operator to be taken based on an agent's current state),
3. application phase triggers the actions that produce the post conditions of the operator (or the outputs).

The series of steps (or applied operators) from the initial state to a desired state forms the solution or behavior path. SOAR defines a sort of low-level machine for implementing algorithms. SOAR differs from rule-based systems because SOAR agents recognize available options and reason about which option to take. Furthermore, a single precondition production can pair with any number of action productions (and vice versa) in contrast to rule based systems and hence it is possible to avoid combinatorial explosion in rules (Wray and Jones, 2006). Whenever an agent comes to a decision that resolves an impasse, a new production with new preconditions and actions is generated. This methodology is referred as *chunking*, which can make the behavior generation process efficient, and it is the main learning method in SOAR. SOAR also includes modules for truth (or reason) maintenance systems for maintaining state consistency and being responsive to the environment. There are several projects ongoing in the SOAR community to expand the capabilities of the architecture such as adding episodic and semantic memories to SOAR as well as support for emotions[14]. The SOAR architecture has been successfully applied to several large-scale applications in multi-agent contexts (Georgeff et al., 1999).

[14] http://en.wikipedia.org/wiki/Soar_(cognitive_architecture)

Developers of the SOAR architecture mainly targeted the problems with logic-based architectures and rule based systems. SOAR provides an effectively working mechanism built upon encoded knowledge representations and low-level constraints on how planning can occur. Different algorithms and logical deductions can be integrated to SOAR using operators. However, the use of operators impedes explicit plan representation in SOAR. In order to represent plans in SOAR, one needs to must build them from the lower level representation of the architecture. Furthermore, it is hard to use different processes for decision making at different abstraction levels since SOAR uses only operators for deliberation. Even though it is powerful, the state based approach and defining pre conditions for rules can be cumbersome in complex environments. SOAR can be very effective for environments that are well defined since it combines different algorithms, powerful search strategy, and learning.

2.3.2 Reactive Architectures

Problems with symbolic/logical approaches led some researchers to find alternative ways to model behavior generation for agents. Two arguments that are essential in the development of reactive architectures are (Wooldridge, 1999):

- The idea that intelligent, rational behavior is seen innately linked to the environment an agent occupies – intelligent behavior is not disembodied, but is a product of interaction the agent maintains with its environment.
- The idea that intelligent behavior emerges from the interaction of various simpler behaviors.

Pure reactive architectures assume no symbolic reasoning and perceptual inputs are directly mapped to actions. One of the best known pure reactive architectures is the “subsumption architecture”, which is proposed by Brooks (1986). In the subsumption architecture, an agent’s behavior is generated by task accomplishing behaviors. Task accomplishing behaviors are individual action functions that continually take perceptual input and map them to action to perform. Each of these behaviors is expected to achieve some particular task without any complex symbolic representations. The subsumption architecture allows simultaneous occurrence of behaviors; the modules are thus arranged into a subsumption hierarchy, with the behaviors arranged in layers. Lower layers in the hierarchy are able to override higher layers. In this hierarchy, lower layers have higher priority and higher levels are supposed to represent more abstract behavior.

Reactive architectures such as subsumption architecture provide a simple, computationally tractable, and robust approach to behavior modeling of agents. However, since agents make their decisions based on local information –agents do not have a representation of the environment and hence they need to rely on their perceptions of the local environment- it is not possible to come up with behaviors with a long term impact. In fact, this is the major argument of reactive architectures – intelligent behavior emerges from simple interactions with the environment. Unfortunately, there is no principled methodology to come up with a subsumption hierarchy (Wooldridge, 1999). Developers need to create the hierarchy of task accomplishing behaviors and this process requires a laborious process of experimentation, trial, and error to engineer an agent.

2.3.3 Belief-Desire-Intention (BDI) Architecture

The BDI paradigm was originally developed by Michael Bratman (1987) and it is based on "folk psychology". The term "folk psychology" has been widely used as a label for the largely tacit psychological theory that underlies the abilities of people in describing and predicting each other's behavior. Philosophy and psychology fields use a standard assumption of normal adult beings having a rich conceptual repertoire to explain, predict and describe the actions of one another. This rich conceptual repertoire is defined as folk psychology and the conceptual repertoire constituting folk psychology includes the concepts of belief, desire and their kin such as intention, hope, fear, etc., which are called propositional attributes (Davies and Stone, 1995). In other terms, folk psychology is the body of beliefs and capacities that we use in everyday life to explain and predict our own and other people's actions (Morton, 1996).

BDI architecture used the philosophical concept of practical reasoning to model human behavior. Practical reasoning is reasoning towards actions and it states that we do make decisions moment by moment on what action to perform next in the furtherance of goals (Wooldridge 2000). Bratman (1988) defines practical reasoning as:

> "Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes."

Human practical reasoning appears to consist of at least two distinctive activities: deliberation and means end reasoning. The former one deals with the question what goals we want to achieve, and the latter one involves with how we are going to achieve these goals.

BDI framework uses belief, desire, and intention as propositional attributes to model human reasoning along with goals and plans. It is easy to understand these attributes since it is accepted that they resemble people's own descriptions of their reasoning and actions in daily life. In this context, beliefs correspond to information the agent has about the environment. Desires represent states of affairs, which are supposedly but not necessarily consistent, that the agent wishes to consider. Goals are the resultant states if the desires are achieved. Intentions represent desires that the agent has committed to achieving and intentions held should be consistent. Furthermore, an agent should believe that the goals of the intentions held are achievable for realism (Levesque and Cohen, 1990). Finally, plans can be considered as recipes for achieving intentions (Wooldridge 1999).

BDI architecture tries to achieve a balance between reactive and proactive behavior of agents. Proactive behavior is mainly achieved via the use of intention and plans. Levesque and Cohen (1990) argue that intention is a choice with a commitment. General BDI architecture allows the intentions to be present or future-directed. Furthermore, Bratman (1987) lists three types of intentions based on how they are achieved. The first type of intention is the deliberative intention, in which it is formed on the deliberation whether to commit to a choice at a certain time. Non-deliberative intentions are formed without any deliberation. For example, a future-directed intention that is formed earlier can become a present-directed intention with no deliberation even though there might be temporal updates on this intention. The third type of intention is policy based intention. Policy based intentions reflect the cases where one has a certain policy to act in certain sorts of ways in certain kinds of circumstances.

Intention can be seen as a future path that an agent chooses to follow (Rao and Georgeff, 1991). These paths are realized as plans in BDI architecture. A plan is a sequence of actions and/or sub-goals to achieve (Norning et al., 2000). Plans of action have a hierarchical structure (Bratman, 1999), where general intentions embed more specific intentions. Such hierarchically structured plans are typically partial and hence, it is not necessary to specify a complete plan before beginning to act. This means that the details of the plan will be filled as the plan progresses with sub-plans that are at least as extensive as the agent believes necessary to achieve means-end coherency (Bratman, 1988). This structure forms the backbone of BDI architecture since partial specification of the plans can later be used to constrain further practical reasoning (Pollack, 1992). For example, if one individual defines a partial plan for a meeting at a certain time, this intention and the regarding plan will constrain any reasoning that concerns the specific time frame. Based on the assumption that an agent looks for courses of action that are not necessarily optimal but “satisficing”, sticking to earlier commitments might be a valuable tool to model the bounds of cognitive resources of human reasoning.

Agent architectures using the BDI paradigm typically contain four key data structures: beliefs, goals, intentions and a plan library (d’Inverno et al. 1997). Procedural Reasoning System (PRS) (Georgeff and Lansky, 1987) and its successor dMars (d’Inverno, 1997) implements these data structures and they include predetermined plan libraries for the tasks of concern. These architectures have been deployed in many major industrial applications with success (d'Inverno et al., 1997) and they formally specify the components of a plan. Plans in dMars are composed of six components: 1) the invocation condition, 2) the pre-conditions, 3) the maintenance conditions (specify the circumstances that must remain true while the plan is executing), 4) the body of the plan (consisting sub-goals and primitive actions), 5) the internal actions if the plan succeeds, and 6) the internal actions if the plan fails. Based on this structure, when an intention is acquired, a plan from the plan library is selected to achieve the goal associated with the intention. This general process is depicted in Figure 3.

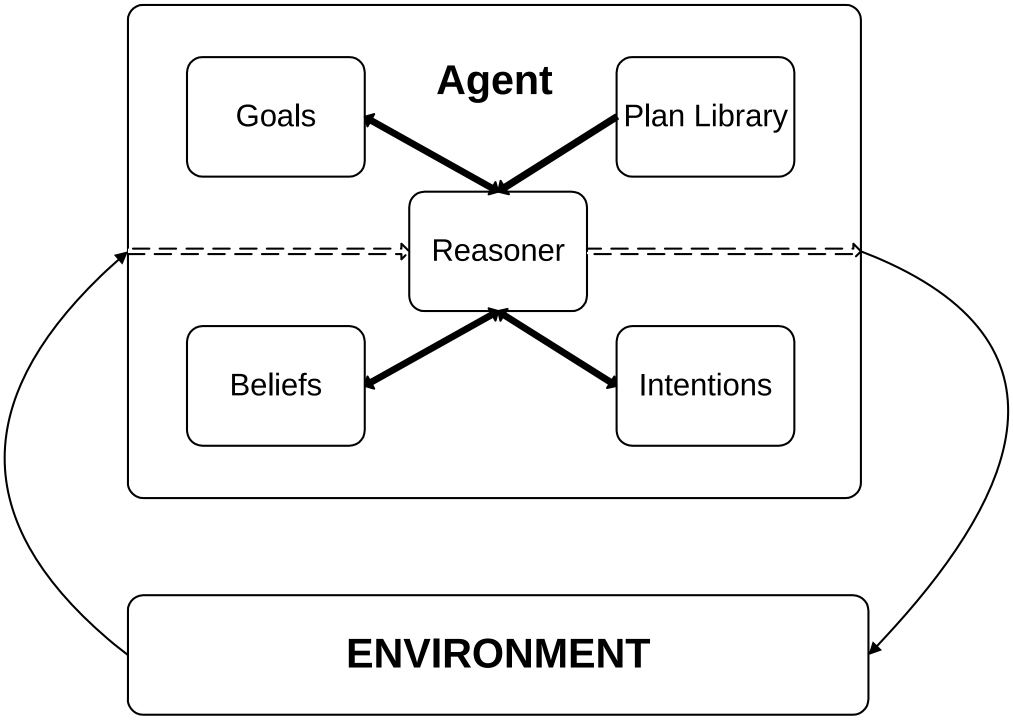


Figure 3 The Structure of a BDI Agent (Norling et al., 2000)

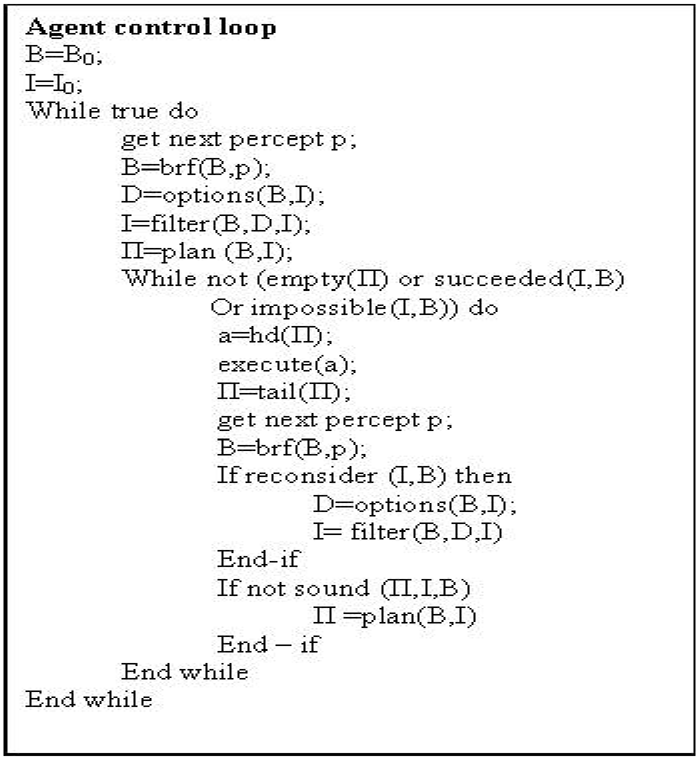
Rao and Georgeff (1991) define three commitment strategies for intentions. The first one is the blind commitment, where the agent maintains the intentions until he/she actually believes that he/she has achieved them. A basic single-minded agent maintains the intention if the intention remains as an option until the agent believes that he/she realized the intention. The third commitment strategy specifies that an agent maintains the intention if the intention stays as a goal until the intention is believed to be achieved.

So far the discussion is based on the proactive nature of the BDI architecture. However, it is necessary to reach a balance between proactive and reactive behavior. Perceptual inputs gathered from the environment are continuously used to update the beliefs of the agents. In the BDI architecture, the interaction between the updated beliefs and the current intentions is achieved via a mechanism named reconsideration. Reconsideration is basically the decision whether to reconsider a plan that is being executed. Bratman (1987) specifies three varieties of reconsideration. The first one is non-reflective reconsideration in which the decision to reconsider depends on the underlying habits, skills and dispositions of the agent. The deliberate reconsideration is the second variety where the agent deliberates in order to decide whether to reconsider.

The third variety is the policy based reconsideration in which the agent uses predetermined policies to make the reconsideration decision.

From a computational perspective, it is important to specify when to perform the deliberate reconsideration. There are three general types of agents that are defined in the BDI literature to model this phenomenon (Pollack, 1992): a “bold” agent that commits strongly to its plans; a “normal” agent somewhat more open to reconsideration; and a “cautious” agent that is prone to reconsideration. Figure 4 represents the agent loop that is defined by Wooldridge (2000).

This agent loop demonstrates the general cycle that a BDI agent goes through while performing the actions and it starts with an agent generating its options and then committing to achieve one of these options with a plan. After committing to an intention, the agent starts to perform the actions in sequential order until the goal is achieved or it is impossible to continue the plan. While performing the actions of the plan, the agent continuously updates its beliefs and it stays committed to the plan as long as the plan is sound given its beliefs. If the plan is no longer appropriate to achieve the current intention, then it engages in further means-end reasoning to find an appropriate plan. Finally, reconsideration of desires and intentions when the agent is committed to a plan is dependent on the boldness of the agent. If the agent is cautious, it will stop to reconsider its intentions before performing an action. If the agent is bold, it will stick to its current commitment until it achieves its goal or the goal becomes impossible. In this representation, B denotes beliefs; D denotes desires; I denotes intentions; Π represents plans; p denotes perceptions; brf denotes the belief revision function; a represents the primitive actions in a plan, hd function returns the first primitive action from a plan, and tail function returns the remaining part of the plan.



```
Agent control loop
B=$B_0$;
I=$I_0$;
While true do
        get next percept p;
        B=brf(B,p);
        D=options(B,I);
        I=filter(B,D,I);
        Π=plan (B,I);
        While not (empty(Π) or succeeded(I,B)
               Or impossible(I,B)) do
                a=hd(Π);
                execute(a);
                Π=tail(Π);
                get next percept p;
                B=brf(B,p);
                If reconsider (I,B) then
                        D=options(B,I);
                        I= filter(B,D,I)
                End-if
                If not sound (Π,I,B)
                        Π =plan(B,I)
                End – if
        End while
End while
```

Figure 4 The Agent Control Loop (Wooldridge, 2000)

The popularity of SOAR and BDI architectures pushed researchers to find ways to increase the interaction between separate set of researchers with similar interests. As a result of this effort, Georgeff et al. (1999) defined an abstract mapping between SOAR and BDI architectures:

- Intentions are selected operators in SOAR
- Beliefs are included in the current state of SOAR
- Desires are goals, and
- Commitment strategies are strategies for defining operator termination conditions.

Furthermore, selected operators (intentions in BDI) constrain the new operators that the agent is willing to consider by constraining the problem space. This fact is also similar to the general concept in BDI architecture, in which intentions are used as inputs in practical reasoning.

2.3.4 Recognition-Primed Decision Model

Several models of naturalistic decision making have been proposed (Lipshitz, 1993) and one of the best known models is the Recognition-Primed Decision Model (RPD) (Klein, 1998). The RPD model describes the decision process of experts operating in environments with naturalistic characteristics, which were previously discussed. The RPD model is based on the idea that individuals that are operating in the area of their expertise spend very little time on decision making; rather they focus on understanding and assessing the situation. Klein (1998) states that choosing the course of action is virtually automatic for experienced decision makers, once the situation is recognized. Studies conducted in different fields such as firefighting also support this argument. This model assumes that experts learn to recognize subtle differences in situations that suggest one course of action over others.

Sokolowski (2003) identifies three key decision making attributes that influence the use of the RPD model. These attributes are experience or expertise with the decision situation, situational awareness, and mental simulation. As depicted in Figure 5, the decision maker first comprehends the situation. This comprehension has four by products:

- Expectation of certain things to occur but not others

- Certain cues to support the diagnosis
- Plausible goals to achieve
- Actions which are likely to succeed

Once a decision maker has diagnosed the problem and generated expectancies, the decision maker uses mental simulation to sequentially evaluate the courses of action. Based on this evaluation, modifications can be applied to the courses of action in order to improve the solution or the next course of action is taken into consideration. If a certain course of action satisfies the situation, the decision maker commits to that course of action and he/she starts to implement it. Once the course of action is selected, the situation is monitored to make sure it remains as expected. If there are violations of expectancies, alternate courses of action can be considered. In other words, the fewer the number of expectancies satisfied, the less confident a decision maker would be about the accuracy of the courses of action being implemented.

Comparison of options in natural settings is not present in the RPD model. Instead, the RPD model assumes the “satisficing” principle as proposed by Simon (1957) and courses of action are evaluated on their own merits.

Norling (2000; 2004) identifies certain overlaps between the RPD model and the BDI architecture such as goal-directed behavior and commitment to a course of action. Furthermore, she defines a mapping between the RPD model and the BDI architecture where;

- Changing the context in RPD means change of beliefs in BDI,

- Situation recognition, relevant goals, and cues in RPD are plan selection in BDI
- Possible actions in RPD are represented by an applicable set of the plan library in BDI
- Expectancy violation in RPD is plan failure in BDI.

Based on this mapping, Norling (2000; 2004) argues that integrating the RPD model in BDI architecture is possible in such a way that the RPD model is used to find applicable plan(s) in the BDI architecture.

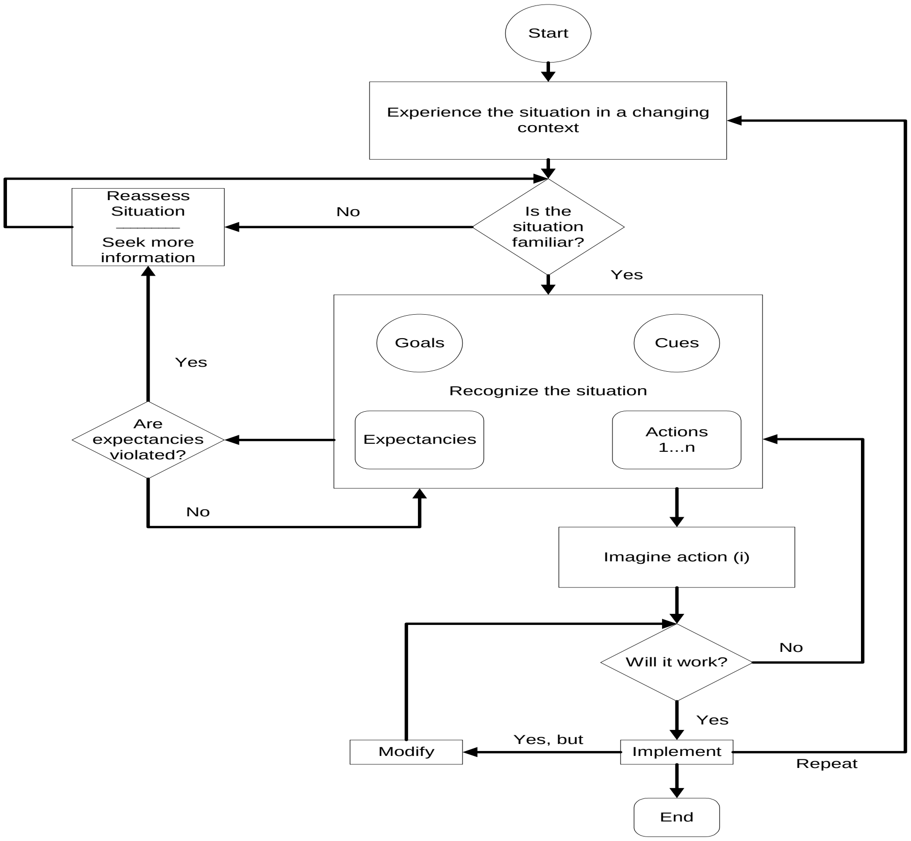


Figure 5 The Recognition-Primed Decision Making Model (Klein, 1998)

2.3.5 Layered Architectures

Reactive, proactive, and social behavior capabilities are desired in multi-agent systems. Layered architectures represent a natural decomposition of these functionalities: reactive, proactive, and social behavior can be generated by their respective layers in a layered architecture. Typically, there will be at least two layers in a layered agent architecture to deal with reactive and proactive behaviors, respectively. Furthermore, information and control flows between these layers need to be defined to handle the balance between different types of behavior. Wooldridge (2000) specified two types of control flow in layered architectures:

- *Horizontal layering:* Each software layer is directly connected to sensory input and action output. Each layer itself acts as an agent and produces suggestions on what the agent should do next. Horizontal layering is represented in Figure 6(a).
- *Vertical layering:* Information and control are passed between layers. In o*ne pass* architectures, control flows sequentially through each layer, until the final layer generates an action output. In *two pass* architectures, information flows up the architecture in the first pass, and the control flows down the architecture in the second pass. One pass and two pass architectures are shown in Figure 6(b) and 6(c), respectively.

Sycara (1998) states that most “real” layered architectures find three layers sufficient. The *lowermost layer* makes decisions on what to do based on the raw sensor input and hence, it handles the reactive behavior. The *middle layer* abstracts from the *lowermost layer* and deals with proactive behavior. The *uppermost layer* tends to deal with the social aspects of the environment. The way those three layers interact differs from architecture to architecture.

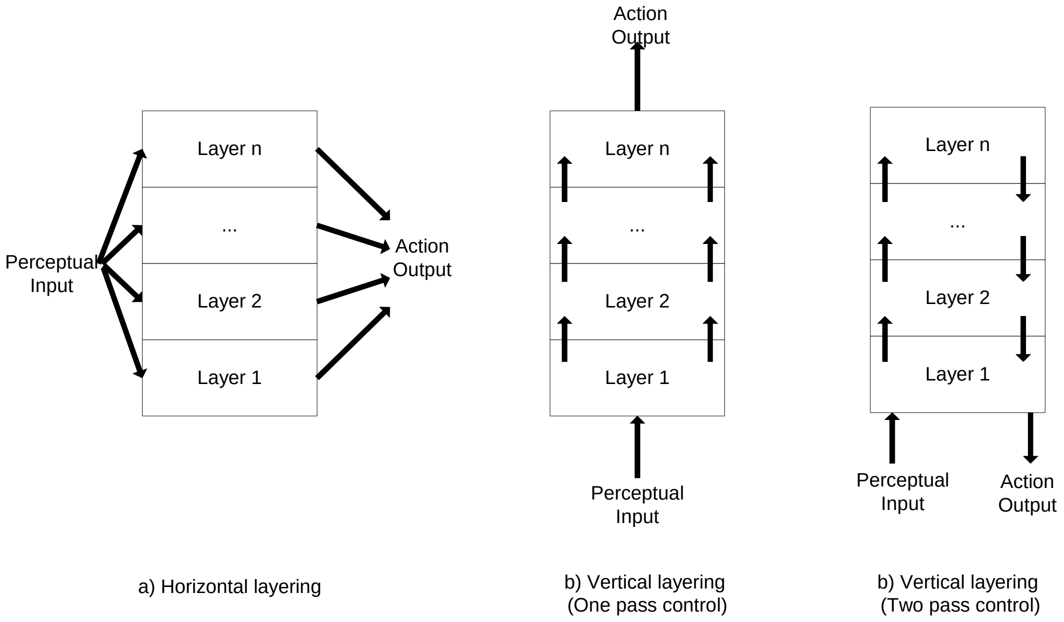


Figure 6 Information and control flows in three types of layered agent architecture (Muller, 1995)

One example for vertically layered three level architectures is INTERRAP (Fischer et al., 1996). INTERRAP is an approach to model resource-bounded, interacting agents by combining reactivity with deliberation capabilities. INTERRAP is a two-pass architecture and the three control layers along with the interactions between them are depicted in Figure 7.

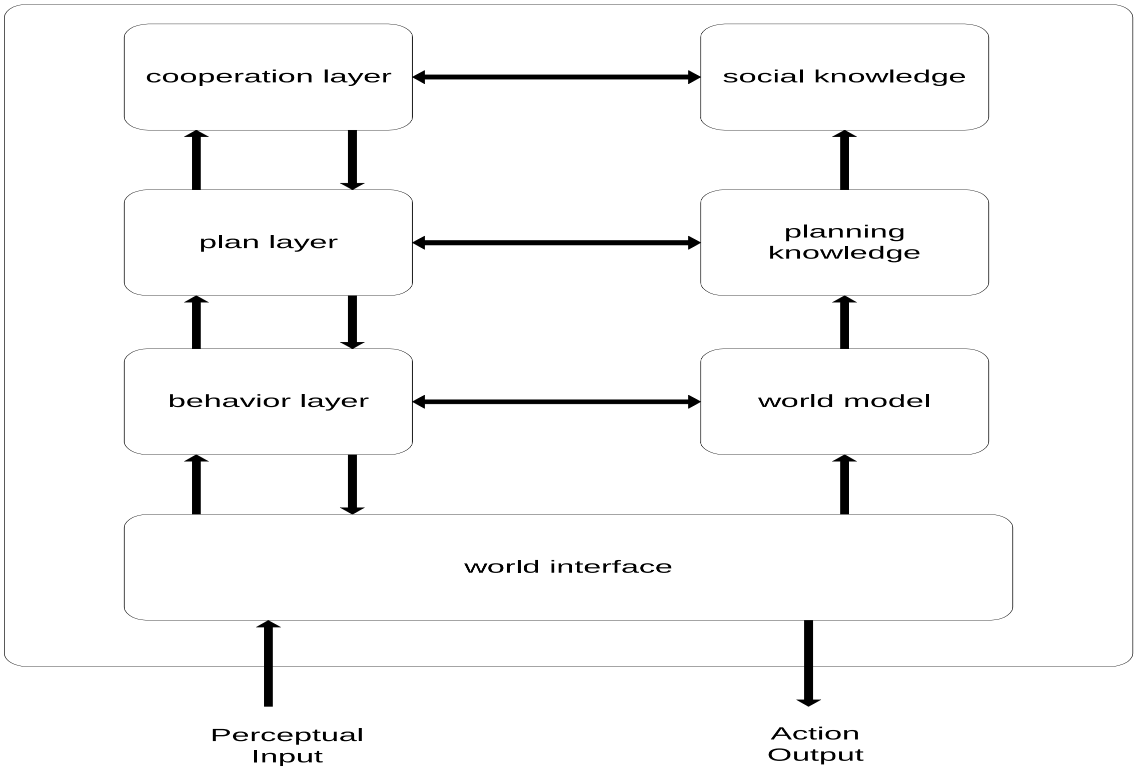


Figure 7 INTERRAP-Vertically layered two-pass agent architecture (Wooldridge, 2000)

The INTERRAP agent architecture aims at combining the advantages of BDI style architectures with those of the layered ones (Fischer et al., 1996). Each layer in the INTERRAP architecture implements two general functions. The first one is a *situation recognition* and *goal activation* function. It works similarly to the option generation function defined in the BDI paradigm such that it maps the knowledge base of the layer of concern and current goals to a new set of goals. *Planning and scheduling* function is the second general function and it is responsible for selecting which plans to execute, based on the current plans, goals, and the knowledge base of that layer.

There are two main types of interaction between layers in the INTERRAP architecture (Wooldridge, 2000). The *bottom-up activation* is triggered when a lower level passes control to a higher layer because it is not competent to deal with the current situation. The *top-down execution* occurs when a higher layer posts activation requests for patterns of behaviors to a lower layer.

Layered architectures are the most popular general class of agent architectures available (Wooldridge, 2000). However, layered architectures may not have semantically clear representation as logic-based architecture have since it is hard to achieve clear representation of the interactions between different layers.

In Section 2.3, five different agent architectures are introduced. Each one of the agent architectures approaches the computational modeling of human behavior from a different angle. The advantages/disadvantages of these architectures, as well as the similarities between different architectures, are discussed in respective subsections. The BDI architecture and the RPD model focus on the process of human reasoning and they try to emulate the actual human reasoning process, whereas logic based architectures and reactive architectures focus more on the outcomes. Layered architectures provide a mean to integrate different approaches within a formal structure.

Section 2.4 introduces the testbed shoplifting problem in retail industry and discusses several issues related with this problem.

2.4 Shoplifting

2.4.1 What Is Shoplifting?

One of the major problems of the retail industry is to detect and prevent inventory shrinkage (or inventory shortage). This shrinkage is the financial lost attributable to a combination of employee theft, shoplifting, administrative error, and vendor fraud (Hollinger and Langton, 2004). Based on the data collected in calendar years 2003 and 2004, National Retail Security Survey reports that estimated annual loss for U.S. retailers is approximately $31 billion. $14.6 billion of this figure is attributed to employee theft (47%), $10.5 billion to shoplifting (34%), $4.2 billion to administrative errors (14%), and $1.7 billion to vendor fraud (5%). 1997 National Security Survey reports an annual loss of $21 billion of which $10.5 billion is attributed to employee theft and $9.1 billion is due to shoplifting (Hollinger et al., 1997). Inventory shrinkage rate, which is the rate of inventory loss over total annual sales, was 1.54% in 2004 and 1.77% in 1997 (Hollinger and Langton, 2004). Hollinger and Langton (2004) also states that average inventory shrinkage rates have remained relatively stable in the last 13 years despite the measures taken against the inventory shrinkage problem; however, there seems to be a declining trend in the last few years. Nevertheless, it can be still said that inventory shrinkage is still a big problem for U.S. retailers and shoplifting plays a main role in this shortage even though the retail industry invests in loss prevention systems and technologies.

Shoplifting is also a problem for countries other than the U.S.A. Tonglet (2002) states that annual U.K. shoplifting losses consistently exceed $1.1 billion. Furthermore, it is estimated that U.K. retailers are spending more than $850 million annually on crime prevention. Even though it is a rough figure, annual retail theft is estimated around $700 million in Australia (Nelson and Perrone, 2000). These figures show that shoplifting constitutes a problem for the retail industry all over the world.

Shoplifting does not only hurt the retailers but it also hurts the other shoppers through the inevitable price increase. Furthermore, increases in retail security may negatively affect the shopping experience of other customers (Tonglet, 2002).

2.4.2 Shoplifters

Ray (1987) stated that one out of 12 shoppers have recently shoplifted based on the survey conducted. Dabney et al. (2004) state a similar rate (8.5%) based on their experiment performed at a drug store in Atlanta. In this research, 1243 persons are observed and 105 of them were clearly seen by the observers committing an act of merchandise theft. Farrington's research provides the figure that 4-5 % of the population up to age 40 is convicted for shoplifting in U.K (Farrington, 1999). Klemke (1992) states that 60% of consumers have shoplifted at some point in their lives and Clarke (2003) states that only one in 150 shoplifting incidents leads to offender apprehension. These figures are interesting since they show that shoplifting is a common crime within the population even though it has been perceived as a minor misdemeanor by the public (Schneider, 2003).

The first comprehensive study of shoplifting was probably done by Cameron in 1964 (Krasnovsky and Lane, 1998). Cameron classified shoplifters as *boosters* and *snitches*. In this classification, boosters referred to shoplifters who steal to sell. Snitches, whereas, do not resell the items they steal and they are otherwise respectable citizens.

Moore extended Cameron’s classification and came up with the following classification (Moore, 1984):

- *Impulse shoplifter (15.4%):* Their shoplifting had not been planned, and they typically took one inexpensive, yet tempting, item. They avoid any sort of risk and can be easily deterred by any sign of security measures.
- *Occasional shoplifter (15%)*: Shoplifting activities are not planned but can use more complicated techniques when compared to impulse shoplifters. They can take a little risk but generally risk averse.
- *Episodic shoplifter (1.7%):* This group has severe psychological problems. They can try to shoplift any item and do not use any complicated techniques. They can easily take risks.
- *Amateur shoplifter (56.4%):* They make conscious decisions to steal and are aware of its illegality. They tend to steal small items that are easy to conceal; shoplifting techniques are simple and they realistically assess relative risks and benefits. Their activities are planned, and they may try to shoplift the item more than once based on the environment situation. If they feel any risk, they tend to leave the item.
- *Semi-professional shoplifters (11.7%):* They employ more skilled techniques in their shoplifting and they target financial gain, e.g. reselling the merchandise. Their activities are planned and they may change the course of action within the store. Under risk, they may try to run away or leave the item.

Generally, there are two types of shoplifters: experts and novices. Novice shoplifters steal items when they see an opportunity and they can be quickly deterred (Carroll and Weaver, 1986; Dabney et al., 2004). Expert shoplifters demonstrate a planned behavior and they search for ways to overcome the security measures (Carroll and Weaver, 1986). Shoplifters who have psychological problems constitute only a small portion and merchandise is stolen by people from every race, sex, and age group. Some researchers state that adolescents have tendency to shoplifting (Nelson and Perrone, 2000). However, Dabney et al. (2004) state that there is no significant difference between different races or age groups based on the research they conducted in 2001.

Carroll and Weaver (1986) discuss that there is fairly a high degree of rationality in the decisions of both expert and novice shoplifters. They state that both types of shoplifters pay attention to the surrounding that may affect the consequence of their actions. However, this decision process can be labeled as a heuristic instead of normative or optimal. Shoplifters process the information they collected and evaluate the possible outcomes and risks and then they make a decision. Hence, this decision demonstrates the characteristics of PP-rationality.

Retail shoplifting experts also have observed that offenders often leave the store without making any purchase (Helena, 1996). This strategy is thought to minimize attention to them and to prevent personal contact with store employees, thereby reducing the likelihood of apprehension. Shoplifters are also thought to operate commonly in organized groups or teams (Hayes, 1993). In the 2004 National Retail Security Survey, Hollinger and Langton state that there is an increase in dollars per shoplifting case but there is a decrease in shoplifting cases. They believe that this increase is a reflection of the harm caused by organized retail crime rings (Hollinger and Langton, 2004).

2.4.3 Different Ways of Shoplifting

Most common methods for shoplifting are grabbing the item and run, concealing of goods on the person, and concealing of goods in bags, baby prams, etc. (Nelson and Perrone, 2000). Another form of shoplifting involves the taking of merchandise without *proper* payment. This can happen by means of switching price tags (Budden, 1999). Removal of packaging, which is then discarded in the store, thereby removing the tag and giving the appearance that the item is used, is also a way to defraud the store (Clarke, 2003). Retail stores with fitting rooms can suffer from shoplifters that wear the items under their current dress. It is also possible to trick the cashier by leaving some items in the shopping cart on purpose and hence not paying for them. Introduction of self check-out terminals in grocery store may lead to not scanning some items while paying for the others. Clarke also states a few ways to beat electronic article surveillance systems such as peeling off the tag, which can sometimes be done despite strong adhesives, holding the item-and the tag- tightly against the body, walking out closely behind someone, holding items outside the reach of the electronic surveillance, using products to deactivate tags, and putting items in insulated bags (Clarke, 2003).

2.4.4 Identifying and Detaining Shoplifting Suspects

Shoplifting can be defined as taking of retail merchandise without proper payment (Budden 1999). Most states’ statutes empower the merchants to detain suspects for questioning or investigating a potential intent to steal merchandise. This is called "merchant's privilege". The merchant's privilege allows detention of persons suspected of shoplifting only if there is reasonable cause to believe a person has shoplifted. When reasonable cause ceases to exist, merchant's privilege is no longer effective. Statutes of several states allow law enforcement officers, store managers, and store employees to detain suspects, whereas, few states extend this protection to third part security guards and other agent. It should be noted that reasonable force should be used in apprehension and detention of suspects and reasonable force does not usually involve threats of bodily harm since merchants do not arrest shoplifters but simply detain them for police authorities.

Detainment should be for a reasonable length of time. A few states' statutes approach the duration of a legal merchant detention in a specific manner, as does Louisiana's, which allow merchants to detain shoplifting suspects for sixty minutes.

Some states allow cursory searching of suspects, especially the items they carry such as shopping bags, handbags etc. However, statutes do not generally provide merchants the right to search inside suspect's clothing.

As an example, Alabama's law on shoplifting is presented next.

Alabama's Law on Shoplifting (AL ST § 15-10-14)

(a) A peace officer, a merchant or a merchant's employee who has probable cause for believing that goods held for sale by the merchant have been unlawfully taken by a person and that he can recover them by taking the person into custody may, for the purpose of attempting to

> *effect such recovery, take the person into custody and detain him in a reasonable manner for a reasonable length of time. Such taking into custody and detention by a peace officer, merchant or merchant's employee shall not render such police officer, merchant or merchant's employee criminally or civilly liable for false arrest, false imprisonment or unlawful detention.*
>
> *(b) Any peace officer may arrest without warrant any person he has probable cause for believing has committed larceny in retail or wholesale establishments.*
>
> *(c) A merchant or a merchant's employee who causes such arrest as provided for in subsection (a) of this section of a person fro larceny of goods held for sale shall not be criminally liable for false arrest or false imprisonment where the merchant or merchant's employee has probable cause for believing that the person arrested committed larceny of goods held for sale.*

Some state statutes make detentions a little easier to call, as they statutorily declare that concealment of goods can be considered evidence of willful concealment on the part of the customer, and stores are privileged to investigate. For example, Arkansas's state merchant protection specifies that

5-36-102. Consolidation of offenses - Shoplifting presumption - Amount of theft.

(b) The knowing concealment, upon his person or the person of another, of unpurchased goods or merchandise offered for sale by any store or other business establishment shall give rise to a presumption that the actor took goods with the purpose of depriving the owner, or another person having an interest therein.

It is important to note that merchants should have reasonable cause to detain a customer. It is not possible to detain a customer based on clothing, race, and sex. Witherspoon Security Consulting recommends that shoplifter should be observed approaching the merchandise, selecting the merchandise, concealing or carrying away the merchandise and failing to pay for the merchandise to establish the probable cause and if all of these steps are observed then the shoplifter is detained[15]. It is further recommended that suspects should be under constant and uninterrupted observation for the suggested practice. Furthermore, company or store policies may further restrict the actions to detain suspects such as limiting pursuing suspects beyond company property. These sorts of practices are important since if the merchant fails to establish the reasonable cause or does not release the suspect when reasonable cause ceases to exist or a reasonable period of time for detain expires, suspects can sue the merchant.

Dabney et al. (2004) discuss behavioral cues that are associated with shoplifting. These visible cues are (1) individual is aware of or looking for anti-shoplifting measures,

[15] www.security-expert.org/shoplift.html

(2) individual is performing repeated head and neck movements that are signs of scanning the store, or (3) individual is playing with the product packaging to reduce its size, or to remove security tags. It has been discussed that these behavioral cues are shown by the majority of the shoplifters and hence they are closer to shoplifting when compared to other customers. Closer attention to a customer who shows these behavioral cues might increase the chance to establish the reasonable cause to detain the suspects.

2.4.5 Prevention of Shoplifting

Clarke suggests that crime will be prevented by either reducing the opportunities for crime or by increasing the risks of apprehension and these approaches form the basis of the current situation crime prevention theory. Situational crime prevention departs from other crime prevention theories since it is focused on the settings for crime, rather than upon those committing criminal acts. Basically, it introduces managerial and environmental change to reduce the opportunities for crimes to occur (Clarke 1997). In a report, prepared for U.S. Department of Justice, Clark lists following precautions against shoplifting (Clarke 2003).

1. Improving store layout and displays
2. Tightening stock controls
3. Upgrading retail security
4. Posting warning notices on high risk merchandise
5. Hiring more and better trained sales staff
6. Hiring store detectives
7. Hiring security guards

8. Installing and monitoring CCTV
9. Using electronic article surveillance (EAS)
10. Attaching ink tags to merchandise

Security measures that are most used are live visible and hidden closed circuit television, digital video recording systems, secured display fixtures and shoplifting deterrence signage (Hollinger and Langton, 2004).

Security measures listed above can be taken to diminish the losses, however, some of these security measures are also costly and it is hard to estimate the returns. Clarke proposes that after the implementation of security measures, statistics should be collected to evaluate the effectiveness and modifications might be performed on the applied security measures based on the results.

2.5 Summary of the state-of-the-art

This section presented an overview of the different aspects of human behavior representation in computational environments and it portrayed shoplifting as a possible application domain. It is further discussed that shoplifting is a representative environment of physical security systems to study human behavior and complex entity interactions.

Until the late 1980’s, human decision characterization was dominated by classical decision theory, which assumes human’s decisions are E-rational. Classical decision models focus on the decision outcome itself rather than the process that generates the decision. Utility theory is incorporated in these models to represent the differences between individual risk preferences. However, classical decision theory is questioned by various researchers and it is shown that human decisions are not always E-rational. In real life settings, few people perform decision optimization calculation except for mid-term or long-term economical decisions. Furthermore, it is also hard to mathematically represent many decision episodes.

Cognitive psychologists and philosophers have a natural interest in how decisions are made by humans. These researchers are not particularly interested in the outcomes of the decisions or how “good” they are but the processes that humans incorporate to make those decisions. Researchers that focus on the process of human decision making state that intuition is an integral part of the decision making process. Naturalistic Decision Making theory especially focuses on the intuitive steps that a person follows in reaching decisions with a particular emphasis on the experience of the individual regarding similar situations. The two-system view also presents a similar approach. Intuitive courses of action (or plans) are first considered unless they are modified or deliberately overridden by the human reasoning process. In general, heuristics and mental simulation are the processes that are believed to be used by humans to evaluate the possible scenarios for the decision in question. Moreover, it is not possible to assume every single piece of information is available to individuals in real situations. Therefore, perceptions from the environment and situation awareness are major components of any decision making episode.

The Belief-Desire-Intention paradigm and the Recognition-Primed Decision Model present powerful and yet robust models to deal with the complexity of human reasoning without having to explain the mechanics of how the brain works. These models have folk psychological foundations and they are capable of representing a wide-range of complex behaviors. They are especially powerful when individuals must interact with other individuals, who can be unpredictable. However, these models are not the solution for all types of human behavior representations. BDI paradigm’s abstraction level can be too high for certain cases. Certain high-level conscious processes -such as emotions, memory and learning- require adjustments in order to incorporate them into the BDI framework. On the other hand, the RPD model does not cover teams or organizations and comparison of options in natural settings is not present in RPD.

Domain specific knowledge is important in representing human behavior. Shoplifting is a relatively well-known domain that employs physical security systems. Although there are no computational models identified for shoplifting, several papers provide descriptions for the major actors and their actions in the retail stores, where physical security systems are used. Based on the domain specific information extracted in Section 2.4, there are two main facts that make shoplifting interesting for this research. First, shoplifting is a significant issue for the retail industry and physical security systems are extensively used in retail stores. Second, there is inherent cooperation and competition between the actors. Security measures and shoplifters compete against each other and security personnel and potential shoplifters can cooperate among themselves.

3 RESEARCH PROBLEM

Garcia (2001) defines intrusion detection as the detection of a person or vehicle attempting to gain unauthorized entry into an area that is being protected by someone who is able to authorize or initiate an appropriate response. Intruder detection analysis involves assessing a facility's susceptibility to intrusion/breach by unauthorized people. Clearly, understanding a facility's susceptibility is a precursor to designing effective physical security systems to prevent unauthorized access to the facility. The ultimate goal of this research is to develop a simulation-based problem solving environment and associated decision support tools to assist with the general facility and security system design problems. The facility and security system design goal is to identify a *security configuration* that minimizes a facility's vulnerability to intrusion at minimum cost. In this context, a security configuration includes the physical structure of the facility, the set of sensors included in the facility, and the set of guards and their respective operating/patrol strategies. Modeling and simulation can be cost effective tools in the design and analysis of the physical security systems. In this research, an agent-based simulation test-bed is being developed for general physical security systems analysis and design. This test-bed will form the basis to analyze the effectiveness and cost of various forms of security policies in physical security systems. However, to improve realism, credibility, and variability in physical security systems simulation models, operating/patrol strategies need to involve situation awareness, cooperative team behavior, planning, and deliberative decision making processes of the entities. Achieving this improvement requires the development of simulation-based computational models of human behavior, which is a major component of the research being conducted.

Viewing a physical security system as a multi-agent simulation model requires explicit formal definition and specification of the environment and organizational design of the system. The development of a formal modeling system is one of the research contributions of this research. Some of the basic definitions that will be used include:

- ***Facility* –** a set of buildings and other structures (fences, walls, etc.) within a well-defined geographical region.
- ***Sensor* –** a device that can detect the presence or absence of a person, or object in a defined region.
- ***Participants* –** people that have various responsibilities and roles in the facility. Four classes of participants are defined:

  Guards – Human participants whose responsibilities involve protecting the facility by detecting and potentially intercepting intruders.

  Intruders – Human participants that are trying to gain unauthorized access to specific locations within the facility. “Access” in this context can mean a variety of things from simply reaching a location, to reaching and remaining undetected at a location for a given time period.

Workers – Human participants whose responsibilities do not involve the protection or intrusion of a facility, but who are present and working in the facility.

Neutrals – Human participants whose actions are associated with neither intrusion nor protection such as regular passengers at an airport or shoppers at a retail store.

The facility itself constitutes the interaction environment for the agents that are participating in the physical security systems simulation. There are two fundamental contributions of this research. First, formal computational models representing the environment, the spatial interactions of the agents with the environment, and the temporal aspects of the agent behavior are developed. Second, an agent simulation based analysis methodology employing the formal computational models of human behavior is designed to identify “effective” physical security configurations. The environment and the agents are further discussed in Sections 3.1 and 3.2 and this discussion will provide a conceptual basis to develop computational behavioral models for the agents of concern. However, this conceptual basis only points out the general structure of the participants of physical security systems simulation. An implementation of this conceptual basis requires domain specific information to be able to model the temporal features of agent behavior in the specific context of the implementation. Section 3.3 suggests the shoplifting problem in retail industry as a possible application domain and it provides conceptual temporal models of the shoplifting problem participants that can be used to implement the aforementioned conceptual basis.

3.1 Environment

Static aspects of the facilities (i.e. the structure and location of walls, rooms, shelves, etc.) are used by the agents mainly in two forms. First, mobile agents should consider the static features of the environment to find a path to different locations in the facility and second, vision, which is the main medium for perception, can be obstructed by static objects. These requirements necessitate a formal definition of the environment to enhance the spatial information extracted from the sketches or drawings of the environment. For this purpose, **Hi**erarchical **G**rap**h** **Re**presentation for **S**cenes (HIGHRES), which is a conceptual data model that completely and unambiguously describes the environment (or the scene), has been developed as part of this research. HIGHRES (detailed in Section 4.1) enables the user to model a physical facility at different levels of detail and to explicitly incorporate interactions among the components of the facility. Furthermore, HIGHRES facilitates the capture of the spatial features of the environment and it provides a hierarchical data model to store the scene information. This scene information is then consumed by different modules of the simulation model either directly or through the use of different graphical constructs such as Zone Movement Graphs (Section 4.1.2) and Portal Visibility Graphs (Section 4.1.3) that are interfaced with HIGHRES.

3.2 Agents

Capabilities that an agent might have include mobility, perception, autonomous behavior, and communication. Mobile agents can move within the environment and they have the capability to find paths to different locations within the facility. Immobile agents have fixed locations in the facility. Perception is mainly dependent on the vision of agents, and visual percepts form the basis for agent behavior and communication. Communication provides a medium for agents to exchange information on the state of the environment.

In a real life physical security system setting, it is possible to observe proactive and reactive behavior as well as communication and coordination between the participants. Based on the discussion in Section 2.3, layered architectures appear to be a good candidate for an agent architecture, which encapsulates different aspects of reasoning mechanisms that are employed by the participants of the agent-based simulation model. Figure 8 represents the three-layered vertical agent architecture proposed for the agents in the testbed. The bottom layer is the *reactive layer*, which provides the interface with the environment for the agent. The *deliberation layer* is responsible for generating possible courses of action and reasoning about them in order to deliberately determine the actions of the agent. The *cooperation layer* overlooks the deliberation layer and it is responsible for managing the interactions of the agent with other agents. Intruders and guards employ all three levels of the architecture, whereas neutrals, workers, and sensors operate only at the reactive layer. Intruders and guards are the only types of agents that have autonomous behavior capability based on the discussion made in Section 2.3. Therefore, they are referred to as *autonomous* agents and workers and neutrals are referred to as *simple* agents. Initial focus is given to the design of reactive and deliberation layers in this section.

The communication module in the reactive layer provides agents a message-passing medium to interact with other agents in the environment. Agents with communication capabilities can broadcast a message to all other agents or can use direct routing mechanisms to send a message to a specified agent. An example for communication is that a worker watching the video streams coming from the surveillance cameras. The worker may perceive a criminal activity in the facility. S/he then broadcasts this information as a message to the security personnel and they perform the necessary actions defined in their control module to deal with this new state. This mechanism is encapsulated in the conceptual designs of different types of agents.

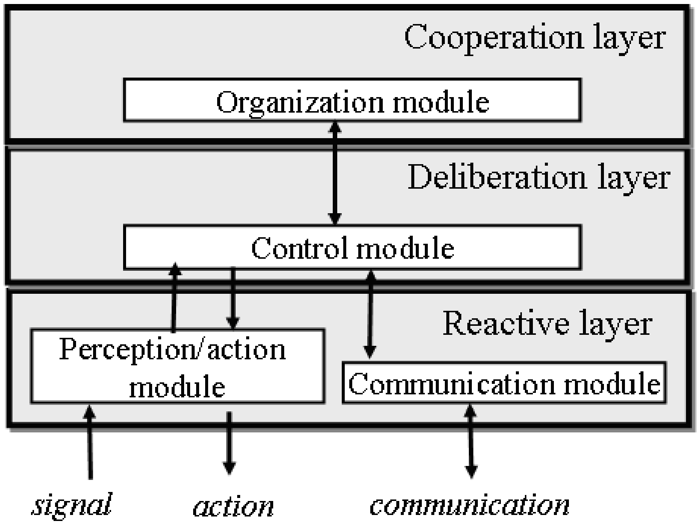


Figure 8 Agent Architecture

The perception/action module in the reactive layer captures the ability to observe the environment to facilitate gathering the requisite information used in the decision making process. Vision is the primary method that is used to perceive the environment and it is realized by performing Line of Sight (LoS) calculations between entities. Results of the LoS calculations are then fed into a cognitive recognition function in order to determine the detection. A basic cognitive recognition function takes into account only the results of the LoS calculations. However, it is also possible to define more complicated cognitive recognition functions that consider different factors present such as the illumination level in the environment or the alertness and/or stress level of the looking agent. Details of the perception mechanism and its uses in the simulation model are discussed in Section 4.1.3.

The perception/action module models the spatial interactions with the environment. The workers and the neutrals use the action module to simply follow their predetermined courses of action. In certain types of applications, it is possible to allow the workers and the neutrals to have perception and communication capabilities. Depending on the application domain, it is possible to have neutrals and workers to react to certain activities such as a worker detecting an intruder and reporting it to the security personnel and airport passengers having limited situational awareness and reporting capabilities (i.e., the ubiquitous “Please report any unusual activities or unattended bags to any airport personnel”.).

Autonomous agents can make independent decisions through the control module in the deliberation layer and hence their actions are under their own control. This autonomous decision making capability partly involves the use of “mental simulations”. Agents using the deliberation layer evaluate different courses of action by running mental simulation for likely scenarios. This process uses the agent’s perceptions on other agent and requires the agents putting themselves in other agents’ shoes. Running mental simulations in the control module requires development of simulation models on the fly and initialization of these simulation models based on the perceptions of the agent on the environment and other agents. Capability of running mental simulations is shared by different types of agents that use the control module for deliberation as further discussed in Sections 3.2.1 and 3.2.2.

At this point, it is important to underline the fact that discussions on mental simulation models in the literature are rather at a theoretical level. Implementation of mental simulation capability requires a formal framework to generate, initialize and run simulation models on the fly. This is a novel approach and the mental simulation framework introduced in Section 4.2 is one of the important contributions of this research.

The action module is responsible for performing the actions dictated by the control module if the deliberation layer exists for the agent of concern. In addition to this functionality, the action module is capable of performing movements within the facility. This capability is achieved by defining the interactions between the agents and HIGHRES. This structure is explained in Section 4.1.2.

Finally, the cooperation layer provides another medium to achieve interaction between agents. In addition to the communication ability, cooperation protocols allow agents to distribute and coordinate the performance of tasks.

Intruders and guards autonomously control their activities and they are able to perform goal-directed behavior. The deliberation layer in the agent architecture is handled separately for intruders and security personnel. Conceptual bases for the control of the activities of the intruder and the guard agents are presented in the next two subsections.

3.2.1 Intruders

The intruders will demonstrate both proactive and reactive capabilities. These capabilities are captured in the deliberation layer of the agent architecture. The proactive nature in the intruder agents' decision process is modeled using a blend of the BDI framework (Section 2.3.3) with the RPD model (Section 2.3.4) and the two-system view (Section 2.2.2) similar to the suggestion of Norling (2000; 2004). The proposed methodology uses the propositional attributes of the BDI architecture as the main data to represent the state of the mind of an agent. As recalled from Section 2.3.3, the major components of BDI architecture are beliefs, desires (or options), intentions, goals, and plans. Beliefs pertain to the information the agent has about the environment. Desires represent states of affairs that the agent wishes to consider, and goals are the resultant states if desires are achieved. Intentions represent desires that the agent has committed to achieve. A plan is a sequence of activities of which the final is the goal activity.

In the proposed methodology, aforementioned propositional attributes are first used by the control module in the deliberation layer of an intruder to pick an option (desire) that is consistent with beliefs that are updated by the perception/action module. This selection is performed via simple heuristics that define the mechanism to choose between different alternatives. Based on the discussion in Section 2.2, this is an "E-rational" behavior since the agent is only concerned with the outcome. When the agent is committed to this option, it becomes the intention of the agent, and it uses its means-end reasoning function to generate a plan for accomplishing the goal. This function will be implemented similar to the mechanisms that are discussed in the two-system view and the RPD model. Therefore, the agent will intuitively come up with an initial plan and then iteratively modify this plan throughout its execution. Each plan will be evaluated employing *mental simulations* and the agent will implement the plan if there is one that satisfies the intruder’s goal. This will make the agent “PP-rational” because of the process employed.

After committing to an intention, the agent starts to perform the actions in sequential order until the goal is achieved or until the agent determines that it is impossible to continue the plan. The actions dictated by the control module are continued to be performed by the perception/action module unless there is a need for reconsideration. Intentions will be reconsidered in two cases: 1) if there are violations of the expectancies that are generated during the mental simulation of current plan, and 2) at regular intervals defined by the agent’s *degree of boldness*. The degree of boldness specifies the maximum number of plan steps the agent executes before reconsidering its intentions. These two reconsideration mechanisms will provide the agent an ability to react to the perceived changes in the environment and hence they model the reactive nature in intruder agents’ decision making process.

Details of the deliberation layer of shoplifters are discussed in Section 4.2.

3.2.2 Guards

The guard agents in this model are reactive. They do not possess proactive skills and they simply react to the events occurring in the environment. Guard movements in the facility are specified by the *security policy*. Whenever a guard agent perceives an input that is of concern from the environment, the guard agent basically reacts to it. However, the reaction of the guard agents is modeled differently than the classical reactive behavior, in which perceptual inputs are taken continually and are mapped to actions to perform. In this model, guards can deliberate on the actions that they are going to perform based on the goals as dictated by the security policy of the facility. In this context, it can be said that the guards carry *policy based intentions*.

The reactive behavior of the guard agents can be handled directly at the reactive layer by the perception/action module or the control of the reactive behavior can be passed to the control module in the deliberation layer. Basic reactions to perceptions are directly handled at the reactive layer. One example for this can be that if a guard is following an intruder, the guard changes his/her movement reactively based on his/her perceptions about the intruder movement. However, some of the perceptions require deliberation by the guard agents and hence, they are handled by the control module in the deliberation layer. Differentiation between basic reactive behavior and deliberation based reactive behavior is domain specific and it is dictated by the security policy exercised by the guard agents.

Deliberation for the guards will be modeled similarly to the RPD model and two-system view described in Sections 2.3.4 and 2.2.2, respectively. Even though the guard agents demonstrate goal-directed behavior, their goals are not the result of a proactive deliberation process. Instead, the guard agents commit to goals (e.g. following a potential shoplifter) as a reaction to the perceived events in the environment. Therefore, the constructs of the BDI architecture can be excluded from the guard agents' decision making process. Whenever a perception from the environment makes it necessary to commit to a certain action such as chasing and following, an intuition based plan, which is comprised of simple action components, is generated. After the generation of an initial plan, subsequent plans are iteratively evaluated using mental simulation. If a “satisficing” plan is found, it is implemented. Otherwise, the current plan is modified and the new plan is evaluated using mental simulation. The details of this process are presented in Section 4.2.

3.3 Sensors

There are different types of sensors that can be used in intrusion detection inside and outside the facilities such as pressure, vibration, electromechanical sensors etc. Garcia (2001) classifies the sensors under five categories:

- Passive or active
- Covert or visible
- Line-of-sight or terrain-following
- Volumetric or line detection
- Application (i.e. buried line, fence associated, freestanding)

Passive sensors detect some type of energy that is emitted by the target of interest; whereas active sensors transmit some type of energy and detect a change in the received energy. Covert sensors are hidden from the view of the intruder and visible sensors are in plain view to the intruder. Line-of-sight sensors require a clear line-of-sight for detection. On the other hand, terrain following sensors can detect equally well on flat and irregular terrain. Volumetric sensors detect intrusion in a volume of space and line detection sensors detect along a line. Application determines how the sensors are applied in the facility. In the context of this research no differentiation is made between passive and active sensors. The remaining four characteristics need to be defined for each sensor used.

Furthermore, Garcia (2001) specifies three performance characteristics:

- Probability of Detection
- False Alarm Rate
- Vulnerability to Defeat

Probability of detection is basically the probability of detecting an activity of interest. The probability of detection can vary based on the settings and the type of the activity of interest. False alarm rate is defined as the function of the nuisance alarms over a given time period by Garcia (2001). Two general ways to defeat a sensor are: (1) Bypass and (2) Spoof. Sensors used in the sample application are further discussed in Section 4.3.2.

3.4 A Testbed Application: Shoplifting in Retail Stores

Security systems used in a typical retail store are good examples of physical security systems that one encounters everyday in daily life. There are several precautions that include hiring store detectives, using electronic article surveillance, installing and monitoring CCTV, etc. against shoplifting. These security measures can be taken to diminish the losses, however, some of these security measures are also costly, and it is often difficult to estimate the economic returns associated with the investment.

In this section, the conceptual temporal models underlying the multi-agent model of the retail security system analysis are presented. The effects of locations of security cameras, movement patterns of security personnel and security personnel with and without communication capabilities on the effectiveness of the security policy will be examined using the realization of this conceptual model in Chapter 4.

The facility used in the test-bed is a hypothetical retail store, which is comprised of aisles, shelves, fitting rooms, check-out lanes and doors. Separate doors for entry and exit are defined as well as doors to access each fitting room. The layout of this hypothetical retail store is depicted in Figure 9. The HIGHRES model for the hypothetical retail store is presented in Section 4.1.1.

Information on possible behavior patterns is extracted from the literature survey conducted on retail store security systems and shoplifter behavior. These behavioral patterns of the entities are represented using UML Activity Diagrams in the following sections. These diagrams will be used to implement the temporal models of the agents (Section 4.2), which constitute the control module in the deliberation layer. Constructs of the activity diagrams are then interpreted to simulate agent behavior. This interpretation mechanism will be explained in Section 4.3. The entities used in the testbed model along with their capabilities are shown in Table 1.

The entities in the testbed application relate to the definitions of the participants in a general physical security system. Here, the shoplifters represent the intruders, the security personnel are the guards, the regular customers are the neutrals, the surveillance camera is a sensor, and the workers are named the same.

Table 1 System Entities in the Testbed Application

	Mobility	**Perception**	**Type**	**Communication**
Regular customer	Mobile	No	Simple	Not capable
Shoplifter	Mobile	Yes	Autonomous	Not capable
Security personnel	Mobile	Yes	Autonomous	Capable
Surveillance camera	Fixed	Yes	Simple	Not capable
Worker	Fixed	Yes	Simple	Capable

3.4.1 Regular Customers, Surveillance Cameras, and Workers

Customers are passive agents that roam in the store and shop. The importance of regular customers from the model's perspective is that security personnel might be suspicious of regular customers since they can not outwardly differentiate between regular customers and shoplifters. Customers arrive at the store with a static plan that is comprised of shopping activities that will be performed by the customer (like a shopping list). Customers move in the store in order to pick up items that are on their shopping list and upon completion of the shopping list, they pay for the picked-up items and leave the store. Pick up and pay activities of regular customers might raise suspicion with some probability, which is defined as the false alarm rate, and hence it is possible for security personnel to mistakenly detain regular customers. The general action loop for regular customers is presented in Figure 10.

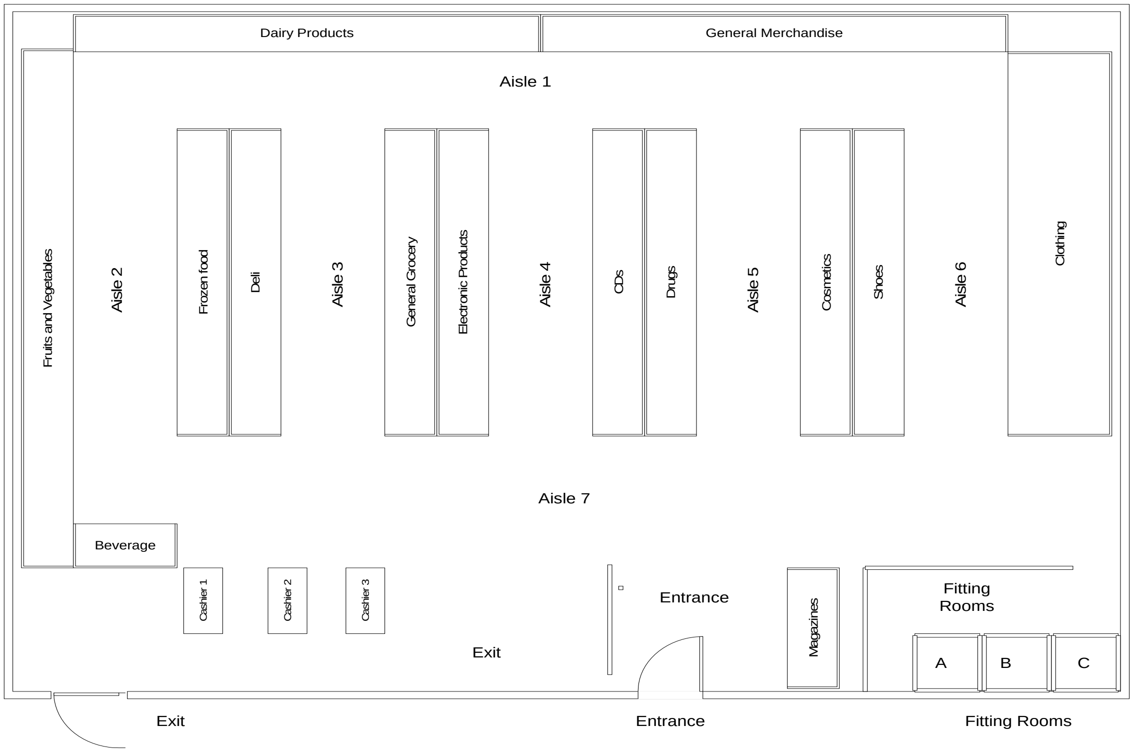


Figure 9 Hypothetical Retail Store

In real life applications, surveillance cameras continuously record the surroundings in their vision. In this conceptual model, it is assumed that only the relevant activities are captured by the computational models of the surveillance cameras. Therefore, the surveillance cameras basically check for the events related with shoplifting, namely picking up item, concealing item, and entering the exit zone with an unpaid item. Surveillance cameras are assumed to have storage capabilities and hence the cameras record these events if seen. Implementation of this conceptual model captures the interaction between the surveillance cameras and the workers as follows: when a critical event is in the view of a surveillance camera, a message is sent to the worker(s) watching the surveillance monitor in order to trigger his/her perception mechanism. The activity diagram for surveillance cameras is presented in Figure 11.

Perception mechanisms of workers are triggered by regular customers/shoplifters or surveillance cameras. Video streams of surveillance cameras can be watched by a worker and he/she broadcasts this information if he/she detects the illegal activity (e.g. concealment of an item). Workers on the retail floor broadcast information on critical activities if they are not busy and can see the occurrence of the activity. The activity diagram of a worker is shown in Figure 12.

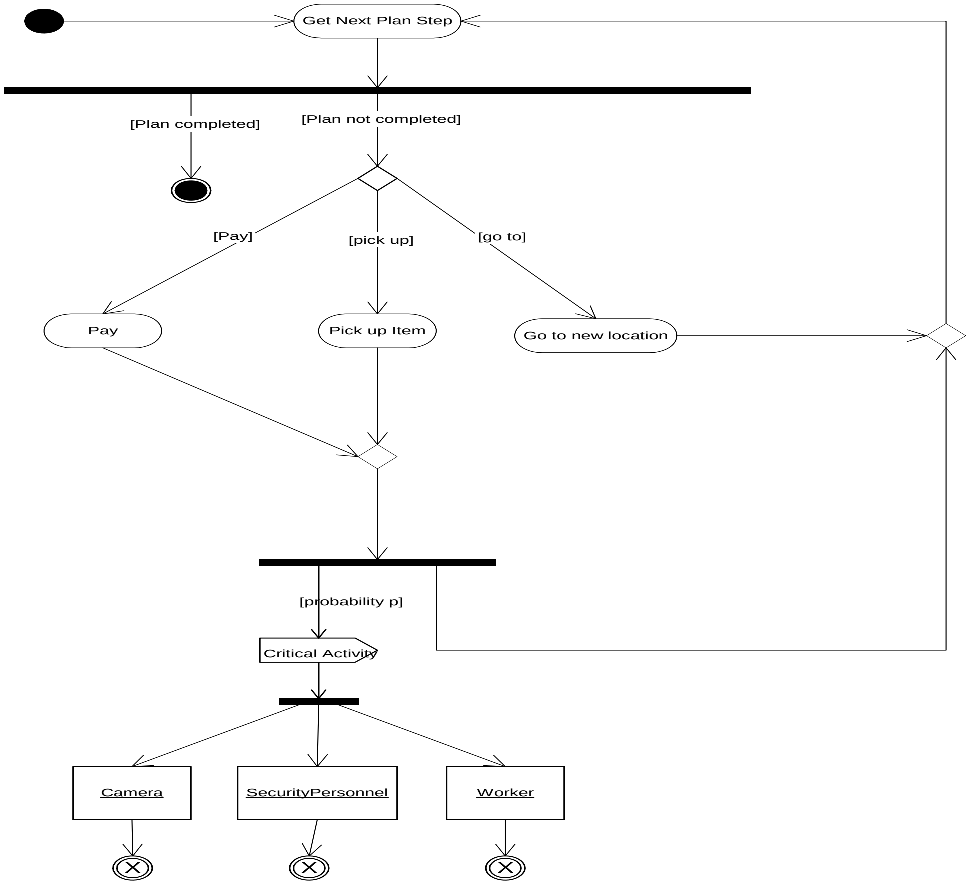


Figure 10 Activity Diagram for Customers

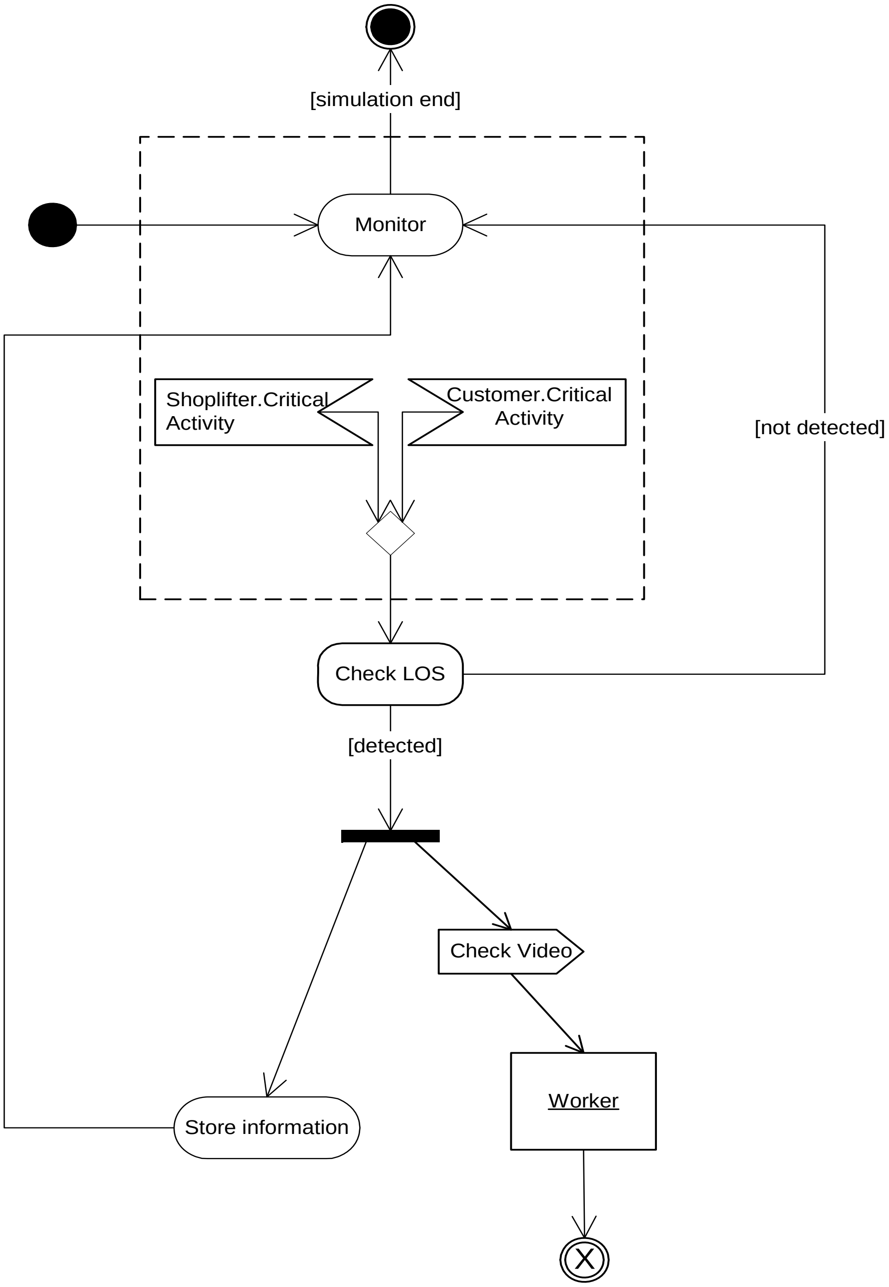


Figure 11 Activity Diagram for Surveillance Cameras

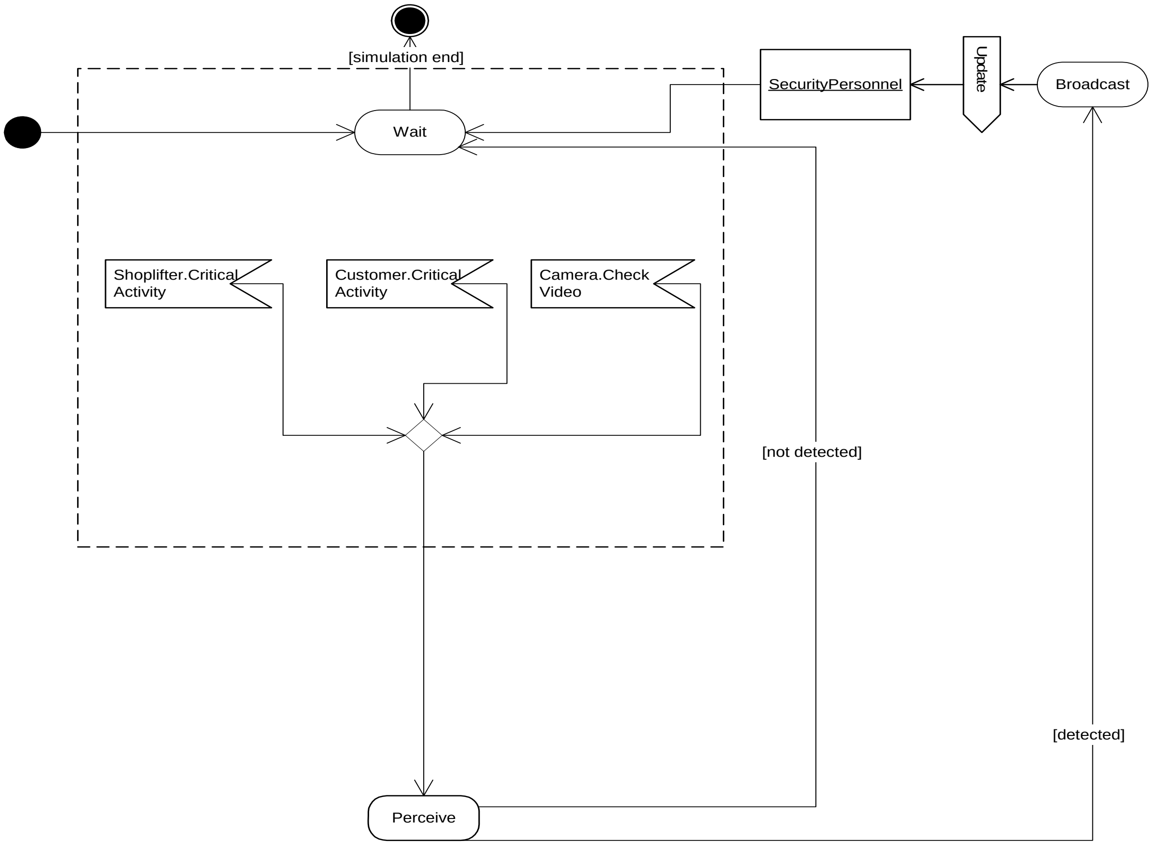


Figure 12 Activity Diagram for Workers

3.4.2 Shoplifters

Shoplifters extend the activity diagram of regular customers by adding new actions to represent the shoplifting activities. Conceal item, unconceal item, and give-up are three actions that are introduced by shoplifters, in addition to the regular customer actions. From a security perspective, three important activities that need to be tracked are picking up an item to steal, concealing that item, and leaving the store without paying for the item. In order to detain a customer, it is important to detect these activities. Hence, it is important to capture this interaction in the implementation of the conceptual data model. In this implementation, each time a shoplifter agent performs one of the three important activities; respective signals are sent to the security personnel agents, worker agents and surveillance camera objects. Receipts of these implementation specific signals trigger the respective perception mechanisms of the security agents in order to determine whether the critical activities are detected. A general agent control logic loop is shown in Figure 13.

A shoplifter's perceptions are focused on the security measures available at different parts of the retail store. For this purpose, a shoplifter agent performs line-of-sight calculations using its perception subsystem to collect information on the locations and look directions of surveillance cameras, workers and security personnel. The collected information is mapped to different areas of the retail store and allows the intruder to assess the level of security around the shelves of concern. The collected information is the basis for the beliefs of a shoplifter agent. In basic form, a shoplifter agent can believe:

- An area is permanently safe/unsafe
- An area is temporarily safe/unsafe
- S/he is being followed.

During the course of simulation, a shoplifter agent might have the following desires:

- Pickup item *i* (with shoplifting purpose) without being detected.
- Conceal the item without being detected in zone *Z*, area *A*.
- Give up (unhide the previously picked up/concealed item).

- Leave the store with shoplifted item.

As discussed in Section 3.2.1 each desire is associated with completion of the final activity of the plan, which represents the accomplishment of the goal. Whenever an agent completes the final activity of a plan, accomplishment of the goal is established. For example, an agent might have an intention to conceal the item in Aisle 6 and if the agent is in Aisle 2, its plan is comprised of one *go to* activity from Aisle 6 to Aisle 2 and one *conceal* activity. Completion of the conceal activity represents the accomplishment of the goal. The main goal for a shoplifter is to leave the store which is contingent on accomplishment of sub-goals (i.e., concealment of item *i*).

3.4.3 Security Personnel

The security personnel's goals are dictated by the security policy of the store, which basically defines certain rules such as the conditions required to commit to follow or detain a potential shoplifter in addition to the movement patterns of the guards. The security personnel's activity cycle starts with patrol activity, which continues until a specific event happens. There are two types of events that particularly interest security personnel while patrolling. The first type of event of interest is receipt of a critical activity signal originated by a shoplifter or customer activity, which might be part of a shoplifting activity sequence. A critical activity signal triggers the perception mechanism of the security personnel. Detection of a critical activity results in committing to follow the agent that sent the critical activity signal. The second type of event that interrupts the patrol activity is receipt of an update message coming either from other security personnel in the store or workers. The update message basically signals the security personnel on a critical activity that happened somewhere in the retail store. This signal may also result in committing to follow the entity that performed the critical activity. The described control flow is represented by an interruptible activity region, which represents the activities that can be interrupted by receiving a signal and is shown as a dashed box on the activity diagrams.

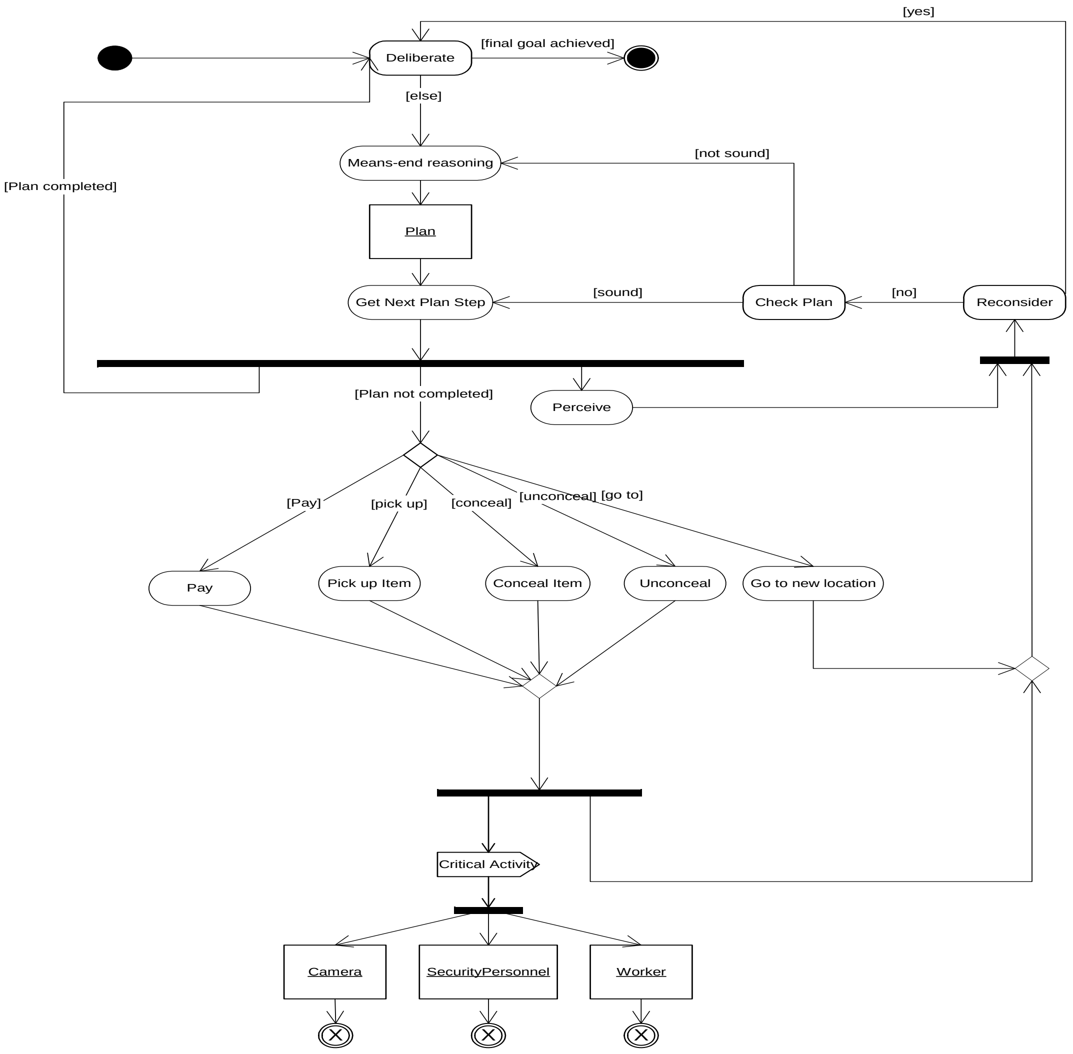


Figure 13 Activity Diagram for Shoplifters

The second interruptible activity region on the security personnel activity diagram includes the follow activity. If any of the specific events described above occur when the security personnel is following a person, a set of conditions is checked to decide whether to continue the current commitment, to change commitment, to cancel the current commitment or to detain the shoplifter (or customer). An activity diagram that shows the activities of the security personnel along with the rules that determine the control flow is presented in Figure 14.

3.5 Discussions and Validation Methodology

Simulation is an appropriate methodology to investigate general facility and security system design problems. This chapter first discussed the necessity to develop a data model that formally defines the static features of a physical security system. It then introduced the conceptual behavioral model for the participants in a general physical security systems simulation. Computational representations of the data model and the behavior model are integral to the simulation models that are used to analyze and design physical security systems. The general structure is then applied to a testbed application demonstrate facility and security system analysis and design in retail stores against the shoplifting problem using simulation.

The main participants of a physical security system simulation within the described context of this research are computational models of intruders, guards, workers and neutrals. These computational models are designed to emulate the behavior of their real life human counterparts. Consequently, effective modeling of decision making processes of the agents that participate in physical security systems simulation is critical for the credibility of simulation-based analysis.

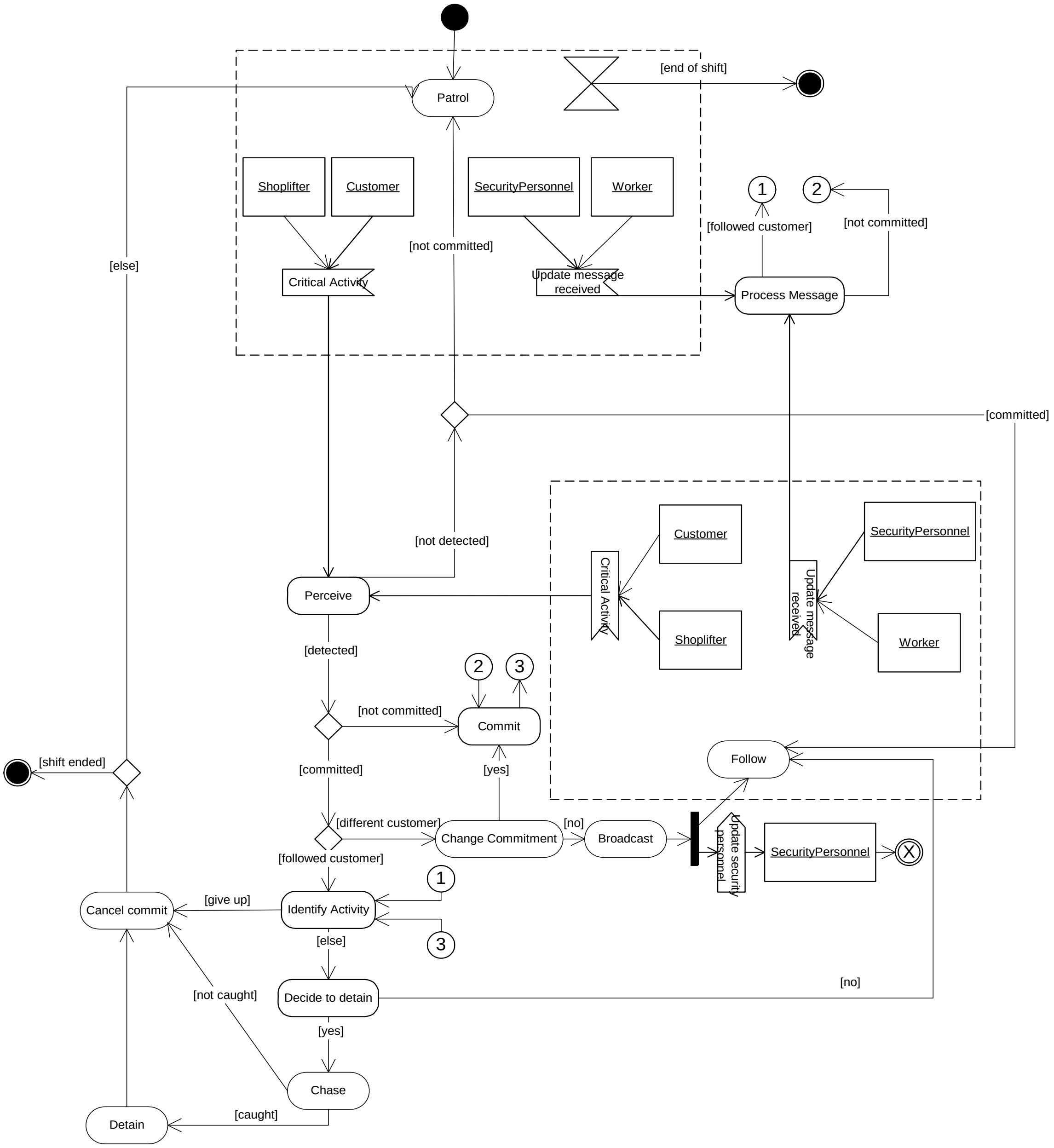


Figure 14 Activity Diagram for Security Personnel

The agent based conceptual model discussed in this research provides a conceptual basis for general physical security systems analysis and design. This model will facilitate conducting several simulation experiments to analyze the performance of different security settings and effect of factors such as communication and cooperation between both the guards and the intruders, completeness of the information that the intruders have on the facility and the operational security policies that the guards exercise on the effectiveness of the security system. Furthermore, it will be possible for the computational reasoning models of the agents to capture specific human traits like boldness and cautiousness and then to incorporate the effects of these human traits to the courses of action that are generated and executed by the intruders.

The primary deliverables of this research will be:

1. An agent based simulation testbed that can be used to analyze and design physical security systems situated at different environments.
2. Computational human behavior models that can be plugged into simulation models.

Several components of significance will be developed in this research in order to realize the aforementioned deliverables. These components are:

(1) A conceptual facility configuration meta-model (HIGHRES) that is used for flexible instantiation of environmental settings in which agents (i.e., customers, including shoplifters, workers, and security personnel) are situated,

(2) A novel line-of-sight mechanism for realizing the visual perception subsystem of agents,

(3) Models of reactive as well as deliberate decision making processes of agents,

(4) Development of mental simulation models,

(5) A mapping that will allow running agent-based simulation models on a discrete-event simulation engine.

(6) Basic cooperation models for security personnel and intruders.

These components will allow the simulation testbed to meet the goals set for this research. As it is stated earlier, it is important to use realistic human behavior models for a robust analysis of physical security systems. In real life problems, human behavior is complex and furthermore, human behavior shows significant variability. It is important to encapsulate this variability in the simulation models of concern. Different fields provide insights on this issue and the computational human behavioral models developed in this research aims to achieve a reasonable realism and variability of behavior demonstrated by the agents in the simulation testbed.

The major difficulty in validating the computational human behavioral models is that there is often no simple quantitative measure of the benefits of these models. It is essential to consider why the realistic representation is important. If it was to develop computational human behavior models that allow accomplishing given tasks in simulated environment, validation is relatively straightforward; agents can be given a set of tasks and then the accuracy of task accomplishing behaviors can be measured. However, if it was for realism, the validation is not straightforward anymore. Since the realistic behavior is a goal of this research, the validation process of realistic behavior or the assessment of the reasonableness of the agent behavior needs to include humans.

There are two commonly accepted faces of the validation process. The first one is evaluating the conceptual validity of the behavioral models. The argument that can be made regarding this face is the proposed conceptual models of human behavior are based upon the cognitive psychology models presented in the literature. Even though computational human behavior models represented in this research are novel, they are conceptually inspired from the cognitive models that are discussed in different academic fields. Furthermore, domain specific information, which is an integral part of the implementations of the proposed computational human behavior models, is extracted from the literature. The proposed models of human behavior models are conceptually valid from this perspective. Verification of the mapping of these proposed models from the conceptual domain to the implementation domain will be achieved via incorporating the techniques such as animating and tracing the behavior of the agents in the simulation environment. Therefore, the verification process will involve observing the behavior of the agents in the simulation environment and then comparing the observed behavior with the proposed conceptual models.

The second face for the validation process is the operational validity. As discussed earlier, the assessment of the operational validity of the computational behavior models is hard and it requires the involvement of humans if the desired goal is achieving realistic behavior. Here, the perspective is that if the behavior of computational agents is reasonable from a human perspective then the proposed computational behavior models are valid. It is important to emphasize the fact that the goal here is realism and variability in human behavior. Therefore, humans involved in the validation process will be asked whether the behavior of agents in the simulated environment makes sense. For this purpose, a validation protocol will be developed and this protocol will capture essentially two validation techniques that can be found in the literature. The first technique is face validity, where humans will observe the agent behavior in the simulation environment and they will provide their assessment on the observed behavior. Face validity experiment will be conducted with five to ten human participants. This group of human participants will be comprised of academicians, graduate students and security consultants. After the simulation model is introduced, human participants will follow the animations of different scenarios and they will be asked to scale the reasonableness of the behavior demonstrated by the agents at different decision points. These answers will be later used to validate the computational reasoning models of the agents.

Second technique that will be incorporated in the validation protocol is predictive validation and this experiment will involve security consultants and academicians preferably with domain knowledge. For this technique, certain decision episodes will be generated and the limitations (or the possible basic actions that can be demonstrated by the agents) will be explained to the human participants. Furthermore, the human participants will only be provided the information that the agent has about the environment. Then, the human participants will be asked to develop a certain number of courses of action regarding the information they have on the environment and the particular goals of the agents in the simulation environment. Afterwards, the courses of action developed by human participants will be compared against the courses of action taken by the agents in the simulation. If there is a certain level of match, it can be argued the proposed computational models of human behavior are valid.

4 COMPONENTS OF THE PROPOSED PHYSICAL SECURITY SYSTEMS SIMULATION MODEL

Deliverables of the proposed research will constitute the underlying structure for physical security systems analysis software that facilitates the agent simulation based problem solving environment. The general structure of this software is depicted in Figure 15. In this structure, the discrete-event simulation engine (DES) is used to run the multi-agent simulation models. The visualization tool is responsible for visually presenting the static environment and the animation of the agents. The scenario analysis tool will use the analysis and design methodology developed in this research to control the simulation experiments.

The agent model needs to address two interacting functionalities: (1) the spatial interactions with the environment and (2) temporal activities to create realistic behavior. For this purpose, a conceptual spatial data model named "**Hi**erarchical **G**rap**h** **Re**presentation for **S**cenes" (HIGHRES) is devised first and this model along with spatial interactions with the agent model is discussed in detail in Section 4.1. A general agent framework named "**A** **B**ehavior-**I**ntuition Mechanism for **R**ealistic **A**gents" (ABIRA) captures the temporal aspects of the agent's behavior model. The ABIRA framework is detailed in Section 4.2. The third component of the software that is directly related with behavior represents the characteristics of the application domain that the agents are acting upon. This general representation and its realization for the "Shoplifting in Retail Stores"example are discussed in Section 4.3. The ABIRA framework is capable of adapting the behavior of respective agents by accessing the application domain data using behavior templates, which provide formal and generic interfaces to the specifications of the domain. These templates are also provided in Section 4.3. Finally, the general software is introduced in Section 4.4.

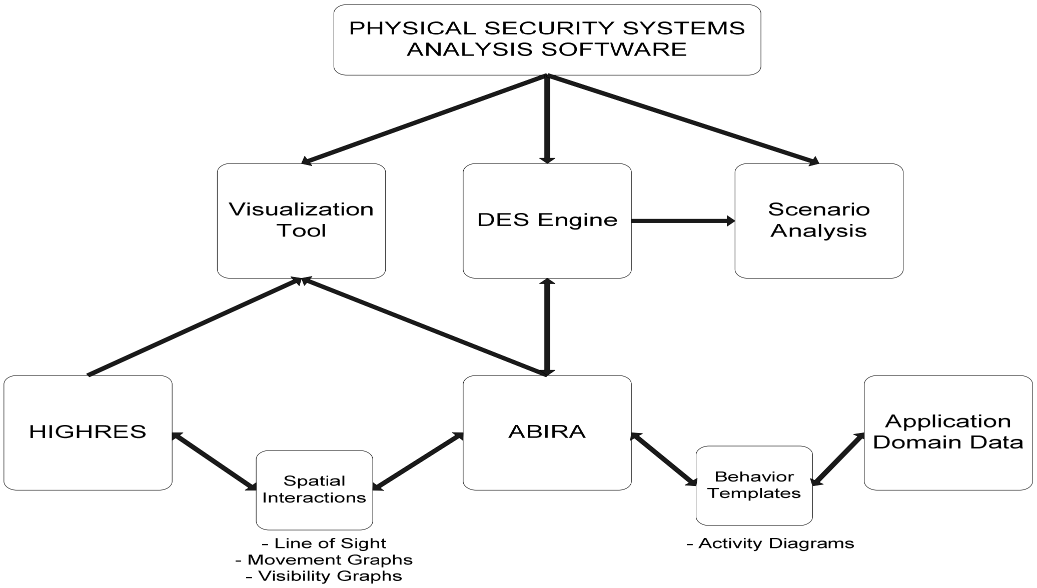


Figure 15 Physical Security Systems Analysis Software

4.1 Spatial Model- Hierarchical Graph Representation for Scenes (HIGHRES)

This section describes the conceptualization of the static aspects of a facility such as geometry, structure, etc. and defines the relations between these aspects and the active entities in the simulation of physical security systems. One inherent feature in physical security system simulation is its explicit attention to spatial features and spatial behavior. Hence, it is important to formally define a conceptual infrastructure to represent the spatial features of the facilities that would then allow modeling these aspects in an object-oriented design. Precise reflection of spatial features is of vital importance for physical security system simulations since most of the cognition and decision making activities of the entities will be based on these spatial features.

In this section, a data model is presented first to enable the user to model a physical facility at different levels of detail and interactions among the components of the facility. Second, a methodology to model human vision is suggested. This methodology constitutes the main perception mechanism for the entities in the physical security systems simulation.

4.1.1 Conceptual Data Model

Static aspects of the facilities (i.e. structure and the location of buildings, other objects present within the area, properties of the materials) will be used by several components of the simulation software and hence a common data structure should be used in order to increase the overall effectiveness of the system. This common data structure might also be thought of as a standard that dictates design rules for different components of the simulation software. The data model described in this section defines the conceptual relationships between different types of data and provides a conceptual tool to extract information from facility drawings. Moreover, it provides a representation hierarchy, which allows a way to describe the facility at different levels of abstraction.

HIGHRES is inspired from a natural hierarchical leveling such as, at the top level there is the whole facility, and as we go deeper in the hierarchy there are buildings, floors, rooms and objects in the rooms, walls, floors, etc. However, this natural representation is not sufficient to have an effective and unambiguous computational representation of the facility that is to be used by the simulation software. The HIGHRES model that we propose enhances this natural decomposition. HIGHRES defines three specific *data types* along with three specific *relation types* to represent the static features of a facility and describe the relationships between these data types. To simplify the physical modeling, all data types use an 8-point convention (e.g. cubes, cuboids, trapezoidal prisms etc.) to define the corner points of any object. The HIGHRES data types are:

- A **solid object** is any 3-dimensional solid shape.
- A **zone** is a 3-dimensional volume that is described by its bounding objects. Its distinction from a solid object is that zones allow entity movement in them. Hence, a zone is basically a hollow shape either empty or filled with something that allows entity movement (e.g. water).
- A **portal** is a 3-dimensional shape that is included in a bounding object that connects two or more zones either visually or permitting movement between zones (e.g. windows, doors etc.). Portals are initially implemented as 2D objects and the switches between zones are instantaneous.

The HIGHRES model is represented in the form of a tree-like structure; the root node (starting node) of the tree corresponds to the *highest level* of abstraction (lowest level of detail); as one goes further down the tree, the level of abstraction decreases (higher level of detail). There are several levels in the data model and these levels determine the desired precision (fidelity) in the simulation. For example, one may want to deal with a building as a solid object. A higher level of detail may be defining the object as a hollow 3-dimensional object, which is defined by a zone in this data model. The next level of detail may be partitioning the building into rooms and so on. Each structure in the facility may have different representations at different levels. For example, the whole facility may be represented as a solid cube object in level 0 and it may be represented as a hollow cube (zone) in level 1 representation.

There are primarily two reasons to have different levels of abstraction in the HIGHRES model: (1) the simulation is set to run at a certain abstraction level; for example, if the entities are not allowed to move within the buildings, a higher level of abstraction is used and hence, only the outside building structure is defined in the simulation, and (2) within the simulation run, different levels of abstractions are used to speed up the computations.

There are three types of relations that define the connections between different data types in the data hierarchy either vertically or horizontally. These relations are:

- **Bounding relation:** This relation is used to define a zone using some other objects (referred to as bounding objects) as the boundaries for the zone. This relation is a horizontal relation in the data hierarchy.
- **Inclusion relation:** Inclusion defines a different type of relation for different data types.

    - **Zone-object:** Objects can be related to zones by inclusion relation. This is a horizontal relationship. Defines the objects that are included in a volume. (e.g. desk in a room)
    - **Zone-zone:** Zones can include other zones in a horizontal relationship. The information desk area in a foyer can be an example of this relationship.
    - **Object-portal:** This relationship can only be defined for bounding objects. It connects a portal to an object. (e.g. door in a wall). This is again a horizontal relationship.
- **Parent-child relation:** This type of relation connects different data types at different hierarchical levels. Hence, this is a vertical relation. There are three different types of parent-child relationships.
    - **Zone-zone:** A zone can be represented as a combination of different zones in a lower hierarchical level. This relation defines an exact partitioning and hence the combination of zones used in the lower hierarchical level should exactly form the zone in the higher hierarchical level. Multiple rooms comprising a floor of a building can be an example of this type of relationship.
    - **Object-object:** An object can be defined as a combination of different objects in a lower hierarchical level. (e. g. at the higher level, a table can be defined as a solid cube which can be represented by 8-points. This solid cube is then represented as a combination of 5 different objects in the lower level, one top and four legs. The important point

here is any combination of these five objects does not necessarily give the cube that is used to represent the table at the higher level.)

- **Object-zone:** An object can be represented as a zone in a lower hierarchical level. An example for this type of relationship is having a building defined as an object in the higher hierarchical level and as a zone bounded by walls, a floor and a ceiling in the lower hierarchical level.

In this context, relationships between objects at different levels in the tree structure represent vertical relationships. A vertical relationship in this sense is a partitioning of an object into smaller pieces in order to obtain better fidelity. Inclusion and bounding relationships are horizontal relationships and they represent the relationships between objects at the same level in the hierarchy. In this context, the Zone-Zone parent-child relation is conceptually different from the Zone-Zone inclusion relation even though they both involve the same types of objects. The main difference is that the zones are in different hierarchical levels in the parent-child relationship while they're both in the same level in the inclusion relation. Therefore, the designer of the simulation may choose to define the whole floor as a single zone at a higher level of abstraction in which the rooms in a floor are not significant and the designer may opt to define all rooms on a floor in a lower level of abstraction using Zone-Zone parent-child relation. On the other hand, the designer of the simulation may opt to define only the room of interest at the current abstraction level ignoring the other rooms on the floor using Zone-Zone inclusion relationship. This allows the designer of the simulation the flexibility of having certain zones in certain hierarchical levels based on the precision requirements of the design of the simulation experiment.

A simple example facility is used to illustrate the conceptual data model described. This sample facility (Figure 16) is comprised of two buildings and one building is divided into two rooms. The first building has a door (D1) and a window (W1). The second building has a main entrance door (D2) and two windows (W2, W3). There is also another door that is located in the wall that separates the two rooms in this building.

This sample facility is conceptually described using the data model defined in this section. There are three hierarchical levels in this conceptual description. The first level shows the whole facility as a single solid object, which is the root of the conceptual description tree. The second level introduces three zones. The first zone (Zone 1) covers the whole facility excluding the two buildings. Zone 2 and Zone 3 present the two buildings in the facility and these zones are included in Zone 1. Each zone is bounded by solid objects. These bounding objects are simply walls for the buildings. The bounding objects for Zone 1 represent the boundaries for the facility. One of the bounding the objects of Zone 2 includes a door (D1) and another one includes a window (W1) as portals. Bounding objects for Zone 3 include a door (D2) and two windows (W2, W3) as portals. For simplicity, only one bounding object for each zone is presented in Figure 17. In this representation, zones, solid objects, and portals are depicted as cloudy objects, rectangles, and circles, respectively. Relations are represented by connections with a small rectangle in the middle. These small rectangles carry the first letter of the relation they are representing.

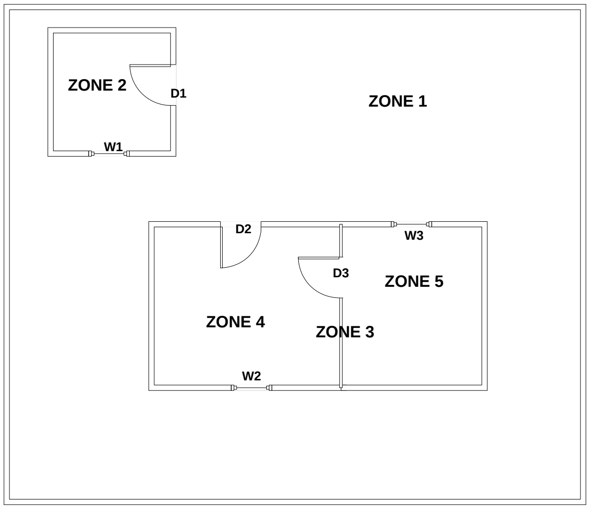


Figure 16 Sample Facility

At level 3, Zone 3 is partitioned into two zones (Zone 4 and Zone 5). Zone 4 and Zone 5 share all of the bounding objects with Zone 3 except one. Both Zone 4 and Zone 5 are also bounded by the separator wall, which includes a door (D3) as a portal.

This conceptual description facilitates running our simulations at different fidelity levels. For example, if the simulation is running at Level 1, only Zone 1 will be present in the simulation along with its bounding objects (facility borders) and two buildings (Building 1 and Building 2) that are included in Zone 1. At this level, entities in the simulation can only move within Zone 1 and buildings will be simply obstacles that impede movement and visibility. Zone 1, Zone 2, and Zone 3 will be present in the simulation along with their bounding and included objects if the simulation is running at Level 2. If we need a higher precision, simulation can be run at level 3, in which Zone 1, Zone 2, Zone 4 and Zone 5 will be present along with their associated objects. In this context, higher precision means more detail which requires more computational power. If the desired conceptual simulation model only allows entity movements outside the buildings, running the simulation at Level 1 provides sufficient level of detail. Furthermore, the simulation runs faster, since the walls, doors and windows of the buildings are excluded from computation (Only Building 1 and Building 2 solid objects are included in computation).

The hypothetical retail store (Figure 9) is represented using two hierarchical levels. In the top hierarchical level, the retail store is defined as a single solid object. The next level (Level 1) is comprised of zones, boundaries of these zones, solid objects (e.g. shelves) included in these zones and portals that allow movement or vision between zones. 16 zones are defined for the retail store. There are seven aisles, one exit area, one entrance area, one fitting room area including three fitting rooms and three cashier areas defined as zones in this layout. A basic conceptual data model, which only includes zones in the facility, is presented in Figure 18.

In Figure 18, all the zones have a parent-child relation with the Level 0 retail store object and the union of all zones in Level 1 gives the total volume of the retail store object. There are inclusion relations between the fitting room area and 3 fitting rooms as well as between the exit area and 3 cashier areas. A more detailed representation of the entrance zone is presented in Figure 19. The entrance zone is bounded by the fitting room wall, bottom wall of the facility (according to the sketch in Figure 9), entrance barrier, and the portal between the entrance zone and Aisle 7. This zone includes the magazine shelf object. Finally, the bottom of wall of the facility includes two portals, entrance door and exit door.

4.1.2 Action Model

Spatial features of the environment will affect the entities in two general ways. First, they will limit the movements of the entities and secondly, they will obstruct vision. These interactions are captured using two different graphs. The former graph represents the possible movements available to an entity while the latter (Section 4.1.3) represents the possibility of visibility allowed by the environment.

Both of these graphs can be defined by means of “portals”. For example, consider the facility in Figure 16. In this facility, there are three doors that can allow movement and visibility between zones and three windows that can allow visibility between zones depending on their states. There can be several states for a portal, some of them can be accepted as “open” states (that allow movement and/or visibility through) and some of them can be accepted as a “closed” state (movement and/or visibility is not allowed).

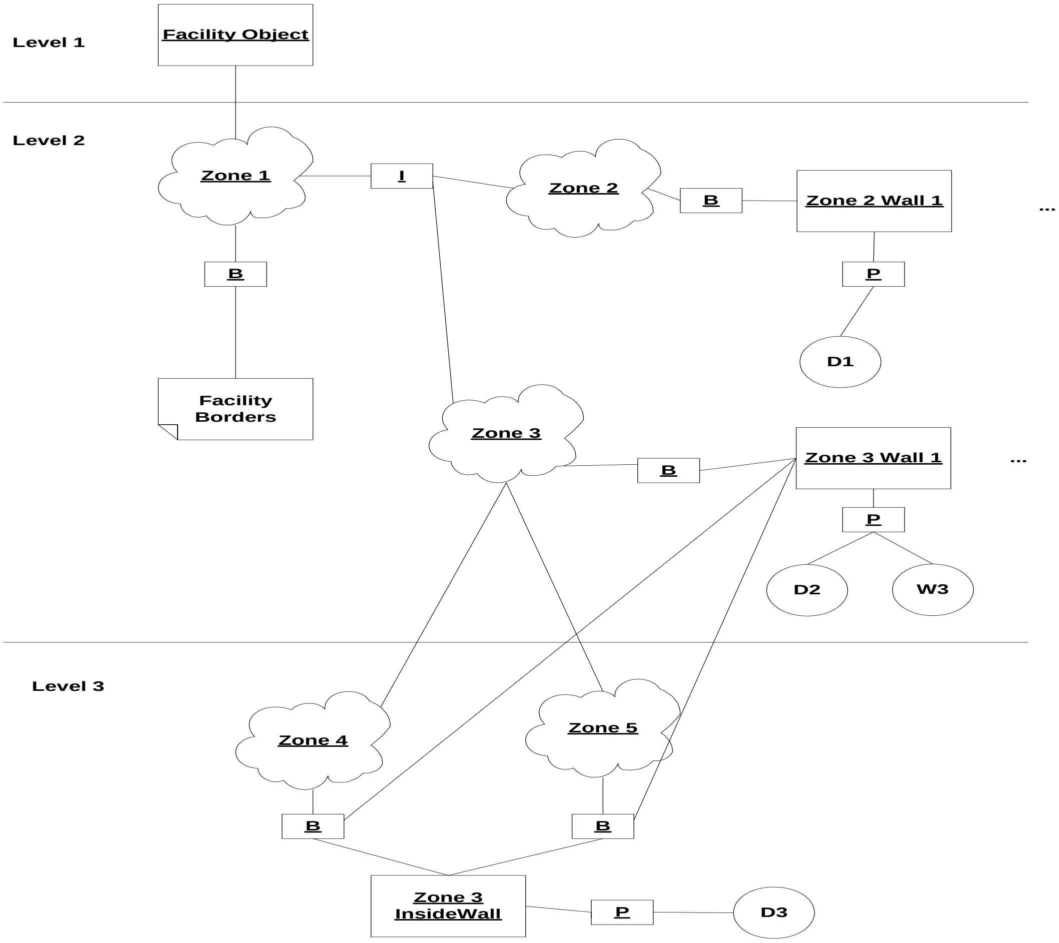


Figure 17 Sample Facility HIGHRES Model

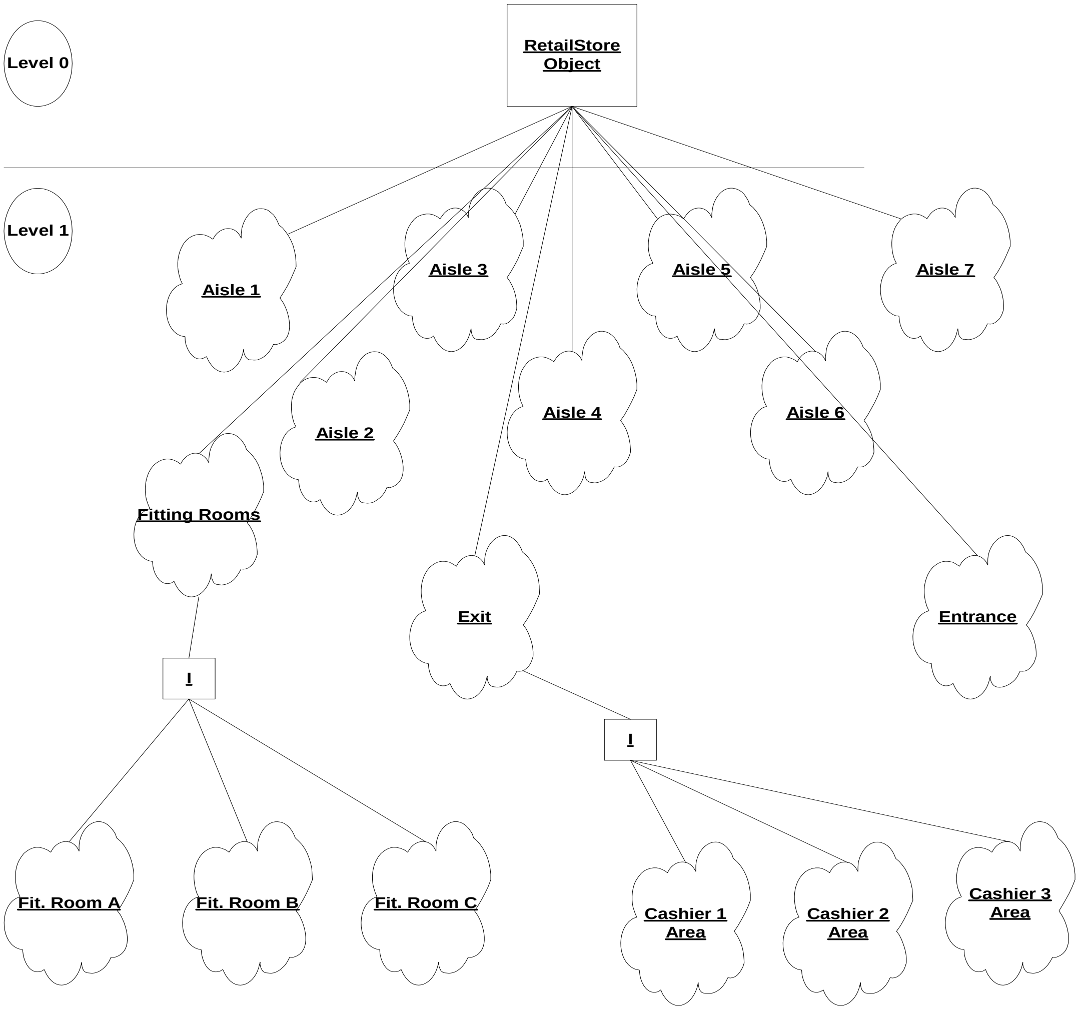


Figure 18 HIGHRES Model of the Hypothetical Retail Store

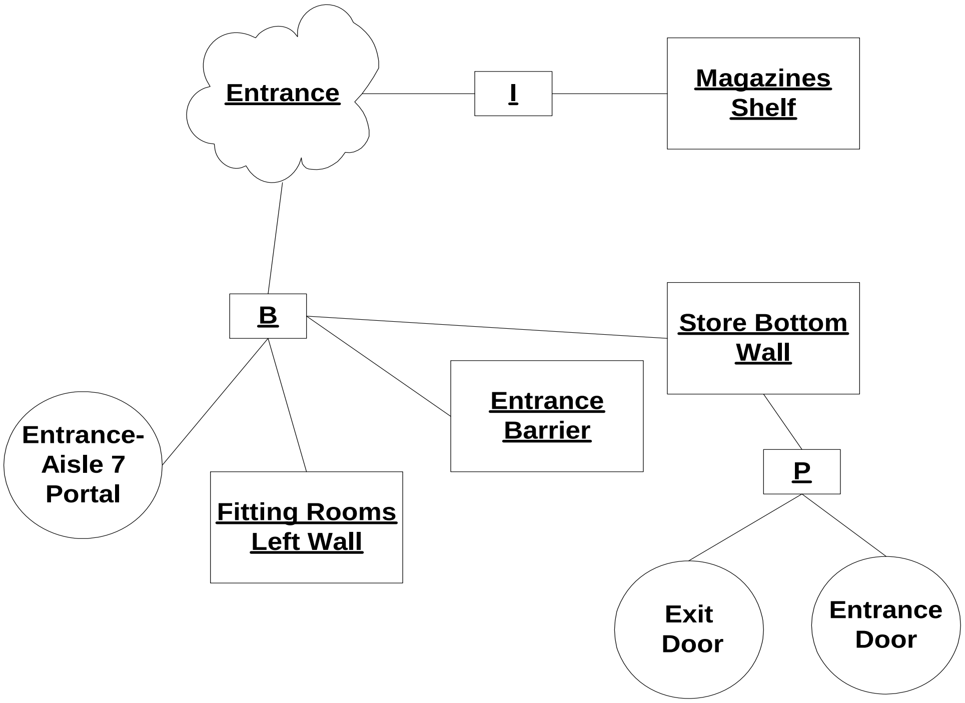


Figure 19 HIGHRES Model for the Entrance Zone

A movement module is designed and implemented to model the agent movements in the facility. The zone movement graph defines the possible movements between zones. Zones are the vertices in the zone movement graphs and there is an edge between vertices if there is a portal between two zones. Entities in the simulation can move from one zone to another zone if there is an edge between these two zones and the conditions for this edge are satisfied. Entities can move freely within the zones (not colliding with the objects that are in the zone), however, movements between zones will be controlled using the zone movement graph. The zone movement graph for example facility is shown in Figure 20.

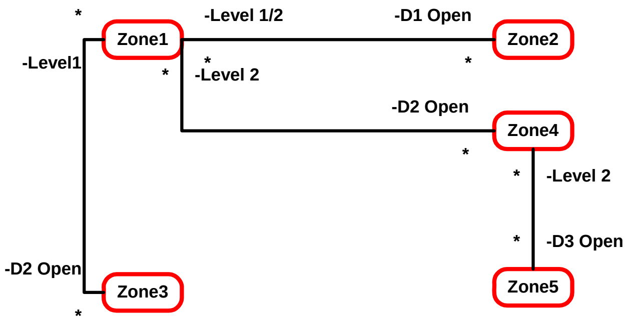


Figure 20 Zone Movement Graph

Figure 21 shows the access paths to the different zones of the facility. From Zone 1, Zones 2, 3, and 4 are accessible, and Zone 5 is accessible through Zone 4 only. It should also be noted that Zone 3 decomposes to Zone 4 and Zone 5 in a higher level of detail. Zone movement graph for the hypothetical retail store is shown in Figure 21.

Each time an agent wants to move within the facility, zones for start and destination are found. Then, a path planning algorithm can be used to find the shortest path from start to destination. One of the example path planning algorithms that can be used is the A* algorithm (Russell and Norvig 2003).

A* requires a graph that represents the environment and it keeps two lists while looking for a path: (1) the open list, and (2) the closed list. The closed list represents the positions that the algorithm has already checked and it is initially empty. A* search starts with the initial state (starting node) in the open list. For each node within the open and closed lists, A* maintains two heuristic values: *g(n)*, the best-known minimum cost, and *h(n)*, the estimate of the cost to a goal state. The best node to examine at any point in the algorithm has the lowest estimated cost: *f(n)* = *g(n)* + *h(n)*.

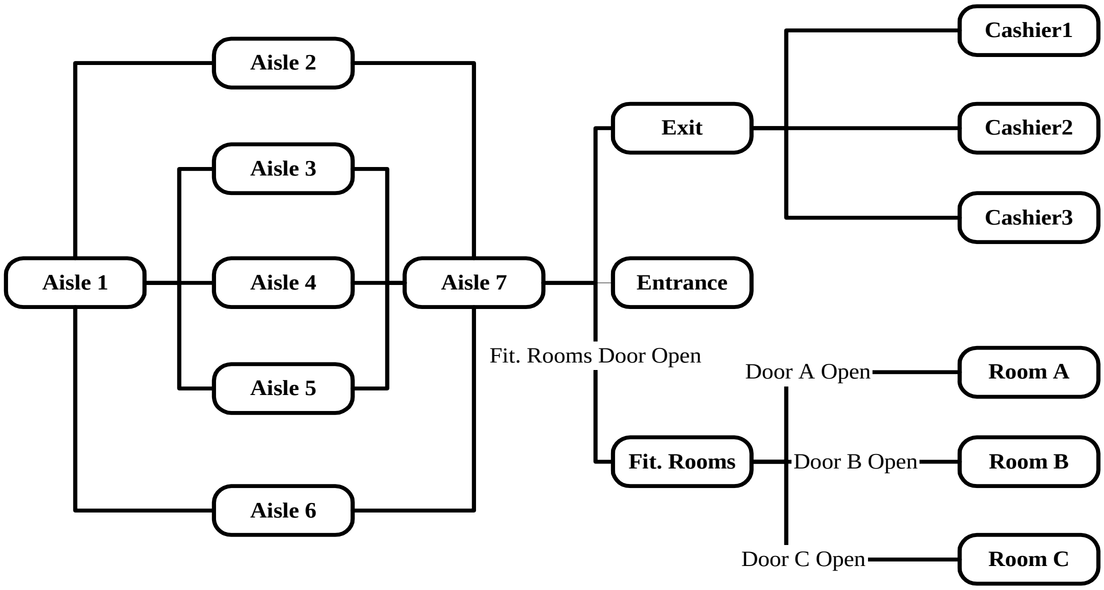


Figure 21 Zone Movement Graph for the Hypothetical Retail Store

The A* algorithm is an iterative process. In each step, A* takes the best state from the open list and moves it to the closed list. The neighbors of the best state are examined in turn. If a neighbor does not appear in either the open or closed list, then the neighbor is added to the open list. However, if the neighbor already appears in either list, it is necessary to check whether the minimum cost *g(n)* has decreased. If *g(n)* decreases, the neighbor node must be deleted from its current location and reinserted into the open list.

The A* algorithm is primarily applied using the zone movement graphs. In a basic implementation, the cost value, g(n), is the actual distance traveled so far and the heuristic value, h(n), is the linear distance between the current state and the destination. The cost metric used calculating the shortest path can also represent different things. In addition to the distance, number of surveillance cameras in a zone, number of guards in the zone can be part of the cost metric definition. Cost metrics are revisited while discussing the temporal model in Section 4.3. The heuristic, h(n), is critical for the performance of the A* algorithm. If the heuristic, h(n) never overestimates the cost of travelling to the goal state, A* is guaranteed to generate the least cost or optimal solution the first time the goal node is generated.

Agents follow the path generated by the A* algorithm to move from one zone to another through portals. By definition, zones are convex and hence agents can move within a zone without hitting any walls or in our context any bounding objects. However, obstacle avoidance algorithms are required to avoid objects inside the zones to prevent collision. For this purpose, the A* algorithm is employed once again, however, it uses a different graph this time.

A cellular decomposition approach is employed for obstacle avoidance similar to the one defined by Kuffner (2000). This approach entails representing the zones as a grid, finding the cells in the grid spanned that intersect with the objects in the zone and creating a graph using the remaining cells. This graph is referred to as the In-Zone Graph and it basically connects the cells to which the agent can freely roam. An agent is approximated by a cuboid and the dimensions of this cuboid dictate the minimum cell size in the grid. After the graph of the cells is created, the A* algorithm can be employed to find a collision-free path in the cells. The cellular decomposition of the hypothetical retail store is depicted in Figure 22. The obstructed cells are marked with red, whereas the free cells are colored as gray in this figure.

In the basic implementation of the cellular decomposition approach, the A* algorithm is run using distance for the costs. However, the security scores introduced in Section 4.3.4 can also be used in finding a path that has minimal exposure to security measures in the zone. Using bigger cells speed up the computations and hence, the cell sizes can vary for different types of agents.

The current implementation of the cellular decomposition approach ignores the low obstacles (that can be stepped on) and the overhangs (which the agent can duck under). Collisions with other agents are also ignored at this stage of the implementation. In a future research phase, the current approach may be augmented to add these capabilities to this approach.

As a summary, the A* algorithm is employed at two different levels. The implementation of the A* algorithm on the zone movement graph can either use distance as the cost metric or use the outcome of another A* algorithm run on In-Zone graphs as the cost metric. When the agents get into a zone, they use the A* algorithm on In-Zone graphs to find collision free paths in the zone.

4.1.3 Perception Model

Line of Sight (LOS) visibility computations form an important aspect of simulation models for physical security systems. The basic question is to determine all entities that a particular entity can see; indeed, in a simulation model, such visibility questions arise at every simulation step, and these need to be answered a large number of times. The answer to visibility questions determines an entity’s future behavior, and so it is important that LOS computations be as accurate and efficient as possible. As noted in

Darken (2004), LOS detection is basically a problem of visibility of an entity's surface. Furthermore, in a 3-dimensional simulation model, visibility calculations need to take into account not only the attributes of the entity (such as how far can it see), but other facility characteristics such as any *barriers* (walls for instance) that obstruct the view.

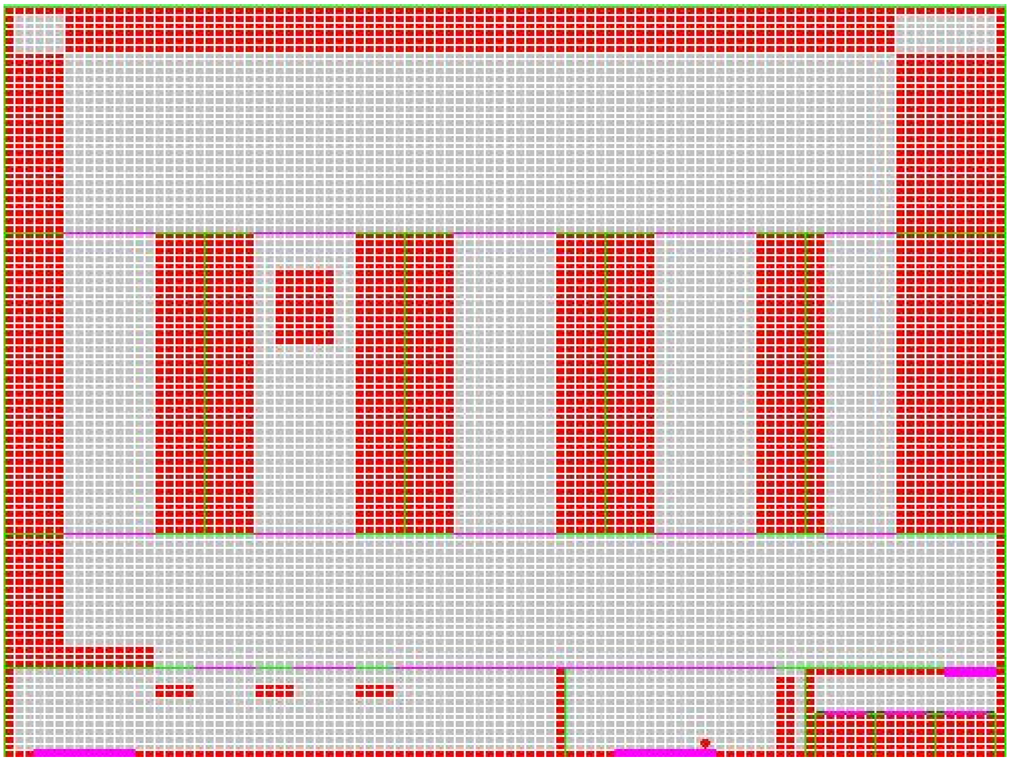


Figure 22 Cellular Decomposition

The line-of–sight module described by *Smith et al.* (1999) uses planar barriers and the entities in this module are represented by a single point. Visibility calculations in this module are performed by basic point-to-point ray casting and checking whether or not this ray intersects with any planes that are representing the barriers. As noted earlier, the models of the environment (such as buildings and rooms) described in this paper consist of 3-dimensional shapes (represented by eight points) and entities are defined by multiple points. Line-of-sight calculations discussed in this paper use these 3-dimensional definitions.

The field of vision of an entity is defined by three parameters – the looking direction, LOS range, and the cone half-angle. The LOS range determines how far the entity can see, and the looking direction is defined by a vector (in 3-dimensional space) in which the entity is looking; lastly, the actual region of view is represented as a 3-dimensional cone (with the vertex at the viewpoint of the entity) that is centered around the looking direction vector, with a half cone-angle specified by the third parameter above. Visibility detection then reduces to finding all entities that are within the field of view (the 3-dimensional cone) of an entity.

The Multiple Ray Casting Approach similar to the one discussed in Darken (2004) is followed to determine if an entity *i* can see entity *j*. Here, the visibility of multiple candidate points on entity *j*'s surface is computed by casting rays from the viewpoint of entity *i*; if all of the candidate points are not visible, then it is concluded that entity *j* is not visible, and if *at least* one candidate point is visible, it is concluded that entity *j* is visible. As per the precision level desired, a predetermined number of points representing the target entity *j* are chosen (these can be the positions of the entity's arms, feet, head, and so on). Thus, the basic assumption here is that if a sufficiently large number of points representing entity *j* are chosen, they will correctly represent the 3-D object(s) representing the entity in the model.

Now the LOS visibility algorithm is briefly described. The algorithm starts by forming a list of all possibly visible target entities; this list contains all entities that are within the cone of view of the viewing entity (the ‘source’ entity). For each entity in this list, a check is done to determine if any candidate point representing the entity is visible; a candidate point might not be visible if there is a barrier obstructing its view. This basically calls for checking if the ray cast from the source entity’s viewpoint to a candidate point of a target entity intersects any barrier in the environment. The list of possibly obstructing barriers is then constructed, mainly to speed up the overall algorithm. This is done by first translating/rotating the 8 corners of a barrier and then applying front and back clipping procedures to determine if the barrier can obstruct the view; these clipping procedures also determine the exact portions of a barrier that may possibly obstruct the view. After the list of possibly obstructing barriers is formed, the algorithm then loops through each barrier in the list and checks if the barrier obstructs any of the candidate points of the target entity; if all candidate points are obstructed by some barrier, then the target entity is not visible, otherwise the algorithm concludes that (a portion of) the target entity is visible. The computations to determine if a barrier actually obstructs vision are rather involved, and they are briefly described next.

The main idea here is to define a plane perpendicular to the ‘looking direction’ vector, and then project all barriers from the set of obstructing barriers to this plane; this plane is called the *Viewing Plane*. The candidate points of a target entity are also projected onto the viewing plane, and a check is made to see if the projected points are within the boundaries of any of the projected barriers. If all of the projected candidate points are within the boundaries of a (projected) barrier, then the barrier obstructs the view, otherwise not. Here, *Perspective Projection* is performed, since it more realistically models practical situations. It remains to determine if a candidate point lies within any projected barrier; and this is accomplished by first determining the convex hull of the projected corners of the barrier by using Graham’s Scan algorithm (Cormen et al. 1990), and then using the Crossing Number algorithm (Cormen et al. 1990) to determine if a particular candidate point is within the convex hull polygon corresponding to a barrier.

Thus, LOS computations form a major part of overall computations performed in a simulation model for physical security systems. The proposed visibility detection methodology uses techniques from computational geometry and carefully optimizes and blends these together, resulting in an efficient and effective LOS visibility algorithm.

A demonstration of vision in this context is depicted in Figure 23. Here, an agent is looking towards another agent, which is represented by 6 points, sitting on the top of a wall. The looking agent can see only 4 out of 6 points since the wall obstructs the view of two points, which are representing the feet of the sitting agent. However, seeing 4 out of 6 possible points does not necessarily mean that the looking agent has detected the sitting agent. Based on the cognitive recognition function used, the conclusion can vary. The result of a very basic cognitive reaction function that requires seeing a single point for recognition would be detection, whereas the result of another basic cognitive recognition function that requires seeing all points for recognition would not be detection.

Since LOS calculations are one of the most intensely used components of the simulation package, the execution time of the LOS visibility algorithm is of vital importance. In order to obtain a basic idea on the computational time requirement of the algorithm, the algorithm is tested using randomly generated test cases. Results are summarized in the following Table 2.

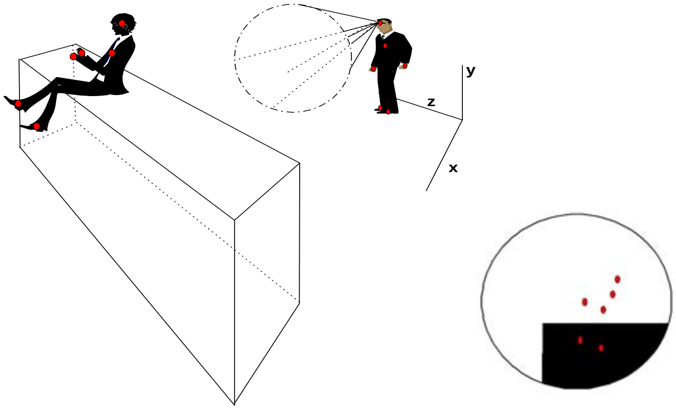


Figure 23 Perception Mechanism

In Table 2, *n* denotes the number of barriers in each test case. Note that the test problems are generated such that all the barriers created randomly within the cone of view. The LOS visibility algorithm is coded with Java and executed using a PC with a Pentium-IV 3 GHz processor and 512 MB of RAM running a Windows XP operating system.

Table 2 Computational Time

n	*Avg (in seconds)*	*CV*
10	0.0006	7.7645
20	0.0007	8.4203
30	0.0006	8.5912
50	0.0008	6.7139
75	0.0009	6.0339
100	0.0024	5.3497
150	0.0032	4.4853
200	0.0033	3.6408
250	0.0033	3.5069
300	0.0036	3.2191
400	0.0045	2.8980
500	0.0058	2.7095
600	0.0064	2.5243
800	0.0076	2.0492
1000	0.0089	2.0931

The computational time requirement changes linearly with respect to the number of barriers in the cone of view as depicted in Figure 24. The trend line fitted to the plot has an R^2 value of 96%. Even with a number of barriers of 1000, the execution of the LOS visibility algorithm does not take more than 0.01 seconds. Based on this, it can be claimed that the LOS visibility algorithm is fast enough to be accommodated within the simulation package.

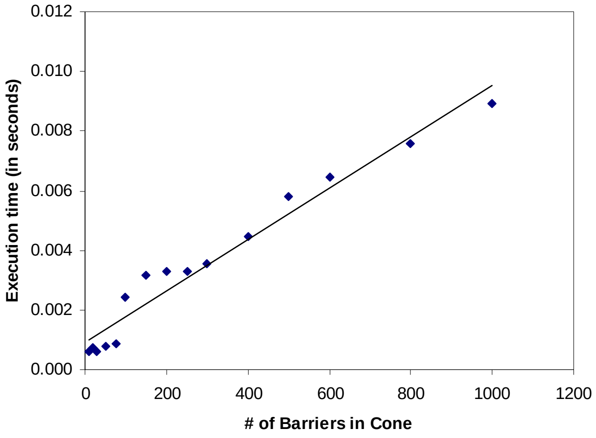


Figure 24 Average LOS Visibility Algorithm Execution Time (in seconds)

Teller (1992) introduced the concept of potentially visible sets, which try to identify the objects possibly visible before any extensive calculation, and discussed the importance of their application in virtual environments with a high degree of occlusion. Portal Visibility Graph is proposed as a similar approach and it shows which portals are visible from a certain portal. This information is important for the line of sight calculations. Portal Visibility Graph decreases the number of objects that are incorporated in visibility calculations and hence the computation required to perform line of sight calculations. Portal Visibility Graphs are constructed in the pre-processing stage by casting lines from several points in a portal to other portals. If any of these lines can reach the target portal then an edge is added to the Portal Visibility Graph between the vertices that are representing these two portals. An example Portal Visibility Graph for the sample facility is depicted in Figure 25.

Figure 25 can be read as follows: from the location of door 1, D1, there is a possibility of visibility through door D2, and windows W1 and W3. Now assume that there is an entity in the simulation which is in Zone 1 and it can only see the portal W2. Through portal W2, it may see portals W3, D2 and D3 but portals D1 and W1 are invisible to this specific entity. The first immediate conclusion is nothing inside Zone 2 is visible to this entity. The second conclusion is Zone 2 can be treated as a single object instead of using the bounding objects of Zone 2 in line of sight calculations for this entity.

The discussion on the spatial model concludes here. In this section, a formal meta-model named HIGHRES to computationally represent the environment is defined and the interaction between the agents and environment is discussed. Next section, the temporal model for the agents is introduced.

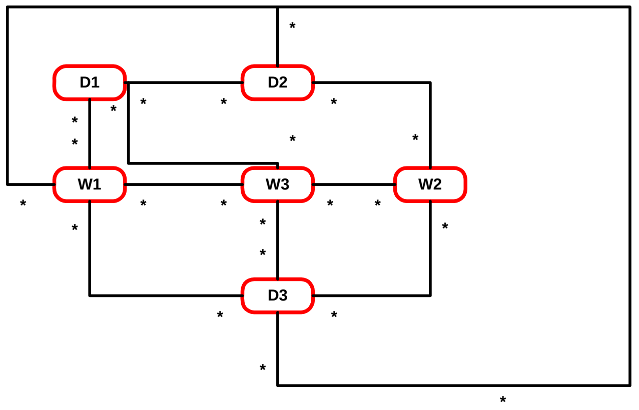


Figure 25 Portal Visibility Graph

4.2 Temporal Model: A Behavior-Intuition Framework for Realistic Agents (ABIRA)

The previous section summarized the HIGHRES model as a spatial model for the environment and it depicted the underlying mechanisms of interaction between the HIGHRES model and the agents. This section focuses on the temporal model of the agents. In other words, this section proposes a formal framework to model the decision making process of the agents. After this general framework is laid out, the details on how a mental representation of the system (environment and other participants) is constructed for each agent and how this mental representation is used in the decision making process of the agents will be provided.

As stated previously, one of the main motivations in this research is to create realistic behavior. Several researchers in cognitive psychology discuss the fact that the human decision making process is not outcome oriented in real life situations. Instead of listing and comparing relevant action alternatives, humans first rely on their intuitions and the process they employ to generate actions. The commonly accepted two-system view approach also highlights this phenomenon. As described in Section 2.2.2, the two-system defines two types of cognitive processes; intuition and reasoning. Intuition is the first cognitive process that is evoked and the reasoning cognitive process is used to either evaluate the response generated by the intuition cognitive process or to perform effortful reasoning if there is no intuitive response. The two-system view provides a general outline to model the human reasoning process rather at an abstract level. In order to capture the details of the human reasoning process, two architectures that have folk psychology roots have been benchmarked: Belief-Desire-Intention (BDI) architecture and Recognition-Primed Decision (RPD) model. The BDI architecture primarily involves the interaction between beliefs, desires and intentions, whereas the RPD model tries to encapsulate situation awareness within the reasoning process. Both architectures provide important insights and these architectures are interleaved to devise a decision framework for the purposes of this research.

The devised decision framework named ABIRA- A Behavior-Intuition Framework for Realistic Agents[16] primarily involves the methods that the humans use in the decision making process. Similar to the BDI architecture, this framework also suggests two distinctive activities: *deliberation* and *means-end reasoning*. The former is basically selection of goals to pursue and the latter generates courses of actions that supposedly achieve these goals. The deliberation process is outcome oriented; it is practically a belief-based comparison of achievability between different desires that an

[16] **Abira** is the creator god in the mythology of the Antioquia of Colombia

agent might have and committing to a one that looks promising. The proposed *means-end reasoning* process on the other hand is a blend of intuition and analysis.

The ABIRA framework is depicted in Figure 26 as an etching on the reactive and deliberation layers of the proposed agent architecture. The decision cycle for each agent starts from an initial state, which is denoted with a circle labeled “0” in the figure. The first process employed by the agents is the deliberation process, in which the agents commit to a goal. As outlined in BDI architecture, there are three inputs to the deliberation process: desires/policies, beliefs, and the goals that the agent is currently committed to. The result of the deliberation process is formation of an intention that the agent is committed to achieving. As described in Section 2.3.3, goals are the resultant states if the intention is achieved and the outcome of the deliberation process is stored as a goal by the agent. There are two different methodologies that can be employed in the deliberation process. First one is used to form *policy based intentions* and the second one is used to form *deliberative intentions.* Formation of policy based intentions is driven by a policy that an agent might have. An example for this is that security personnel in physical security systems should generally follow certain rules dictated by a governing security policy. These policies distinctively describe the goals the agent should pursue under certain specific conditions. For example, a security policy might prescribe patrolling in the facility on a predetermined route and following a person if a suspicious activity happens with the goals of finishing loop in the store and of staying within a certain distance of the suspect for a certain period of time, respectively. Formation of deliberative intentions is driven by the desires an agent might have. Desires are evaluated by employing simple rules and the agent forms intentions based on this evaluation. Deliberative intention will be revisited later in this section.

The successor of the deliberation process is the *means-end reasoning* and it is an intuition based approach as suggested by the two systems view. The Oxford English Dictionary defines intuition as “the immediate apprehension of an object by the mind without the intervention of any reasoning process”. In other words, intuition is “immediate knowledge” or “almost immediate situation understanding”[17]. Intuition is an integral part of the methodology employed by humans especially in real-life settings. Any time an individual needs to develop a course of action in daily life settings, intuition is the primary initial mechanism to suggest a feasible course of action. The ABIRA framework proposes that the primary component in modeling intuition is a set of heuristics that are separately generated (sometimes shared) for each application domain. For the retail store security problem, one can generate basic heuristics for avoiding surveillance cameras or avoiding security personnel. For example, a basic heuristic for avoiding security personnel might be always keeping a certain distance from each known security officer. Based on this proposition, ABIRA includes pattern matching functions that identify a heuristic based on the goals and the beliefs of the agent. These pattern matching functions represent the experience of the agent. This basic approach employed by ABIRA is the representation of System I of the two systems view. After the selection of the heuristic, the selected heuristic is used to generate a course of action. The generated course of action is then analyzed using mental simulations and if the result is acceptable, the agent implements the course of action. If no appropriate heuristic is found

[17] www.capyblanca.com

by experience, the agent has to go through the *reasoning* process, in which different heuristics in the agent's repertoire are tried with the hope of generating a feasible course of action for the current goal. If a feasible course of action cannot be found, the agent concludes that the goal is not achievable and drops the goal.

This general overview of the ABIRA framework leads the way to introducing the processes and the data types in more detail. This detailed introduction is categorized in five main topics: (1) Perception, Beliefs, and Information Acquisition; (2) Goals and Deliberation; (3) Intuitions, Heuristics, and Reasoning; (4) Analysis by Mental Simulation, Expectancies, and Reconsideration; and (5) Action and Communication.

4.2.1 Perception, Beliefs, and Information Acquisition

Vision is the primary perception mechanism used by the agents. The ABIRA framework models the vision mechanism by employing *visual cues*; a type of *sensory cue*. A *sensory cue* is a signal that can be extracted from the sensory input by a perceiver (http://en.wikipedia.org/wiki/Sensory_cue). In the ABIRA framework, any object or agent is defined by a set of points and the sensory input is the set of points that define the object or the agent. Line-Of-Sight calculations defined in Section 4.1.3 are then performed using these points and the points that are visible to the agent are regarded as visual cues. The agent then uses a recognition function to determine whether the agent recognizes the individual objects, the agents or the actions of the agents using the visual cues.

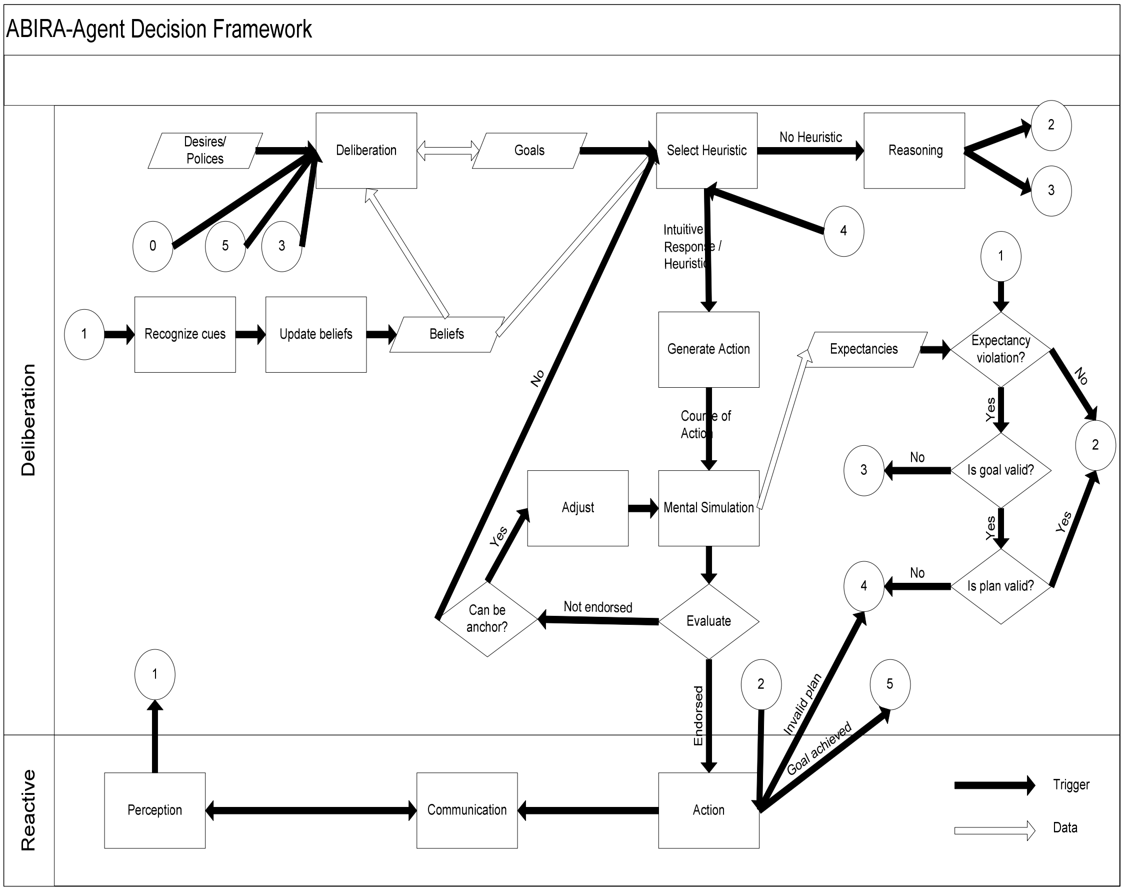


Figure 26 ABIRA-Agent Decision Framework

The sensory input for an object is comprised of the eight points that define the object in the HIGHRES model plus the center point of the object. Similarly, the sensory input for an agent is composed of six points: one for the head, one for the torso, two for the hands and two for the feet. If an object or an agent is of any interest, Line-of-Sight calculations are performed and the points that are visible from the sensory input are fed to the recognition function. Currently, two types of recognition functions are defined in the ABIRA framework (if necessary, new recognition functions can be added to the framework). Each object, agent or process that might be of interest should be associated to a recognition function. The first recognition function is based strictly on the number of visual cues – if there is certain number of visual cues then the subject is recognized. The second type of recognition function differentiates between the visual cues. For example, points representing the hands of an agent in addition to other possible points need to be seen in order to recognize the fact that the agent is concealing an item.

Results of the recognition process are then used to construct the beliefs of the agent. As it is defined earlier in Section 2.3.3., beliefs correspond to the information the agent has about the environment. In other words, beliefs are assimilated information or conclusions that the agent believes to be true. However, the beliefs can be inaccurate or partial in realistic environments. Therefore, there is a need for a mechanism that differentiates between the real world and the agent's world. Each agent stores a mental representation of the environment, which is updated each time the agent's beliefs are updated. This mental representation is then used while analyzing any possible course of action. In order to construct a realistic mental representation of the environment, an agent needs to construct beliefs on: (1) static objects in the environment, and (2) other agents in the environment.

4.2.1.1 Beliefs on Static Objects

The beliefs on the static objects in the environment are represented with the type of the objects and the coordinates of the objects. At the initialization stage, instantiation of each agent's knowledge base includes information about 4 different types of objects of the HIGHRES model: zones, portals, and solid objects. In addition to the HIGHRES objects, sensors are also of importance to the agents and hence, they need to be represented in the agents' beliefs. Each agent's knowledge base is created separately so the following discussion is applied to each agent separately.

There are three cases that are defined to describe agent's information on zones:

1. The agent does not know about the zone
2. The agent knows about the zone but doesn't have any information about the coordinates of the zone.
3. The agent has full coordinate information on the zone and the bounding objects of the zone. In this case, the agent might still have incorrect/partial information about the objects included in the zone or the portals that are connected to this zone.

Agent's information on portals and solid objects is defined in a similar fashion. However, there are five cases while defining the agent's information on portals and solid objects:

1. The agent has full coordinate information on the portal/solid object.
2. The agent assumes that there is a portal/solid object that is not actually in the environment.
3. The agent does not know about the portal/solid object.
4. The agent knows about the portal/solid object but doesn't have any information about the coordinates.
5. The agent has incorrect coordinate information on the portal / solid object.

Information on sensors is also defined in a similar fashion. There are three cases while defining the sensors:

1. The agent has full coordinate and parameter information on the sensor.
2. The agent assumes that there is sensor that is not actually in the environment.
3. The agent does not know about the sensor.

Parameters of the sensors represent the capabilities of the sensor. For example, a surveillance camera is defined by the look direction, the range and the half-cone angle in addition to the coordinates of the sensor. The inaccurate information on sensors is represented by using a sensor that is not actually in the environment. For example, if the agent has inaccurate information on the look direction of a surveillance camera, this will be represented as a non-existent surveillance camera in the agent's mental representation. This belief can later be updated when the agent discovers the existent camera.

The beliefs on static objects are captured in the agent's mental representation using two graph constructs: (1) the Zone Movement Graph and (2) the In-Zone Graphs (Cellular decomposition). Zone Movement Graphs represent the associations between the zones in the facility. Each zone has an associated In-Zone Graph and this In-Zone Graph is used in generating/analyzing movements within a zone. Zone Movement Graphs are constructed using the information on zones and portals. In-Zone Graphs are constructed using information on the zone, the solid objects and sensors included in the zone and the portals that are bounding the zone. These graphs are constructed for each agent using the cases defined above for each static object.

A few examples of the representation of beliefs on static objects are given below using the hypothetical retail store layout in Figure 9:

- Agent does not know about Aisle 1: Zone Movement Graph wouldn't include Aisle 1.
- Agent does know not the coordinates of Aisle 1: Zone Movement Graph would include Aisle 1 but no In-Zone Graph would be created for Aisle 1.
- Agent does not know about "Opening 1" (Portal): Aisle 1 and Aisle 2 wouldn't be connected in Zone Movement Graph.
- Agent does not know about Pallet 1: In-Zone Graph for Aisle 3 wouldn't include Pallet 1.
- Agent thinks that there is a portal between Entrance and Exit: A non-existent portal would be created. Entrance and Exit zones would be connected in the Zone Movement Graph.
- Agent thinks Fitting Rooms Door at a different location (still between Aisle 7 and Fitting Rooms though). Parameter "-1" would be used. Aisle 7 and Fitting Rooms would be connected in Zone Movement Graph but the paths created would be using the wrong coordinates. Agent needs to discover this information and correct its path.

4.2.1.2 Beliefs on Agents

Agents hold beliefs about other agents in the environment and these beliefs are then employed in the mental representation of the environment. There are three types of beliefs on other agents:

- Positions of other agents,
- Activities of other agents,
    - Multiple step activities such as movement,
    - Single step activities such as picking up item.
- Types of other agents.

An agent needs to know about the positions and activities of other agents in the environment in order to project the other agents' anticipated movements. In other words, the agents in the ABIRA framework should have the capability of putting themselves in other agents' shoes. The primary difference between single step and multiple step activities is that the agent needs to combine multiple perceptions in order to construct beliefs on multiple step activities of other agents. One example for multiple step activity is movement. In order to construct beliefs such as the other agent is moving in a certain direction or the other agents is patrolling in the facility on a certain route, the agent needs to recognize the agent of concern's presence multiple times at different positions.

Beliefs on types of other agents are constructed based on the other beliefs previously constructed. Four different types of agents are defined for physical security systems in Chapter 3: guards, intruders, workers, and neutrals. Observations of certain activities help the agents to construct beliefs on the types of other agents. For example, an agent can be labeled as an intruder (or shoplifter in the retail store example) if a concealment activity is observed. The details of these assessments are application specific. Detailed information for the retail store application will be provided in Section 4.3.2 on the rules and methodology to construct these beliefs.

4.2.1.3 Information Acquisition

The aforementioned beliefs are constructed by employing the perception/recognition mechanisms. Agents collect position information about other agents and objects of interest during the simulation. Furthermore, the agent activities of importance defined by the application domain are also perceived.

Information acquisition is a computationally intensive task. Therefore, it is necessary to use computationally efficient information acquisition. One way to do this is triggering the perception mechanism of the agents either when there is something interesting happening or when there is potential to discover new objects or to update beliefs that are inaccurate instead of running the perception mechanism continuously. The Portal Visibility Graph defined in Section 4.1.3 is the primary tool to limit the number of objects that need to be checked. As described in Section 4.1.3, the Portal Visibility Graph stores the information on which other portals are potentially visible from a given portal. The purpose here is to find the zones that the agent can potentially see and then run the perception mechanism only for the objects and agents that are in the potentially visible zones at the time. The basic algorithm used for this purpose is as follows:

1. Find all the portals that are bounding the zone that the agent is in,
2. Find the portals that are visible to the agent from the portal list found in Step 1 (this step incorporates the details on the current position of the agent, the agent's looking parameters that are look direction, range, and cone angle),
3. Find all the portals that are potentially visible from the portals found in Step 2,
4. Find all the zones that are bounded by any of the portals found in Step 3.

The algorithm presented above finds all the zones that are potentially visible to the agent. After finding all the potentially visible zones, it is required to find all the objects in these zones that either the agent does not know about or has inaccurate information about in the belief set. Perception/recognition process is then executed and the necessary updates are performed on the belief set for the objects recognized.

Information acquisition on the position of other agents of interest is run in a similar fashion. For the agents of interest in potentially visible zones, the perception/recognition process is executed and the belief set on the agents is updated using the results of the perception/recognition process. The question here is that which agents are of interest for the agent of concern. Even though the exact answer is dependent on the application domain and the type of the agent, a few general examples are listed below:

- The guards do not store beliefs about the workers or neutrals.
- An intruder might choose not to store beliefs about the workers or neutrals.
- Neutrals do not store any beliefs.
- Workers might store beliefs on intruders.

It is important that the type of the agent is not always revealed to other agents. For example, a guard may not differentiate between intruders and neutrals unless a significant activity is performed by the intruder. Therefore, the type check and hence belief formation is only possible if the agent knows the type of the agent of interest. This mechanism will be detailed in Section 4.3, when the details of the retail store example are presented.

The next fundamental question is how frequently is the perception/recognition process is triggered. The quick answer is each time the agent moves to a new cell in the cellular decomposition model described in Section 4.1.2. This approach will provide the necessary precision for the information acquisition process.

The information acquisition on the activities of the other agents is triggered separately. When an agent performs an interesting activity, the perception/recognition process of all the agents that are in the potentially visible zones at the time and that are interested in the perceiving this activity is triggered. Interesting activities are application specific and details on interesting activities for physical security systems will be provided Chapter 5. One caveat at this point is that the potentially visible zones should be found using all the portals that are bounding the zone of the agent performing the activity. In other words, Step 2 is skipped in the algorithm for finding potentially visible zones. The information on who is interested in which activity is again application specific and the details will be provided in Chapter 5 for the retail store application.

4.2.2 Goals and Deliberation

As it is stated in Section 2.3.3, desires are the states of affairs that the agent wishes to consider. In general life situations, humans can have a variety of desires. Deliberation is the process of selecting a desire that the agent wants to achieve. However, it is necessary to limit the number and types of desires that the agents can have for feasibility depending on the application domain. Obviously, desires can vary between different application domains. In this section, the focus is given on the physical security systems to better describe the deliberation mechanism in the ABIRA framework. Therefore, a number of desires are first introduced that are essential in physical security systems and then the deliberation mechanisms employed by different types of agents are discussed.

Another distinction has been made in the way different types of agents attain desires. The desires of the intruders are general; whereas the desires of the guards are dictated by the security policy. Therefore, it is more appropriate to label the states of affairs considered by the guards as policies.

The guards are primarily responsible from performing spatial activities as dictated by the policies. These spatial activities can be:

- Standing still;
- Patrolling;
- A combination of standing and patrolling.

In addition to these spatial activities, the guards can also be responsible from routine activities involved with physical security equipment. Monitoring the surveillance video streams is an example of this type of activity. This type of activity is combined with the spatial activities and they are dictated by the security policies. Finally, the guards react to the activities of potential intruders. Based on certain conditions specified by the security policy, the guards may follow or try to detain the potential intruders. It is important to underline the fact that the check on the conditions of the security policy is performed by the guards and hence it is based on the beliefs of the guard on other agents.

When a guard commits to any of these policies, the guard needs to generate a course of action that potentially achieves the goal, which is the resultant state if the requirements of the policy are satisfied. For example, the guard needs to generate a course of action that potentially leads the guard close to the intruder if the guard decides to detain the potential intruder. The generation of this course of action is handled by an intuition based means-end reasoning process, which is detailed in the next section.

As stated earlier, the deliberation mechanism of the guards is based on policies. The *policy based intention* mechanism for the guards requires committing to desires as prescribed by the security policy. The intruders, on the other hand, employ deliberative intention. The deliberative intention mechanism for the intruders requires committing to desires with maximum potential benefit. In other words, the intruder evaluates different desires that are held at the moment and commits to one that looks promising. In general, six different types of desires are defined for the intruders (shoplifters):

- Shoplift item I,
- Pick up item I at location X,
- Conceal item I at location X,
- Leave the store via cashiers (paying for some or all of the items in possession),
- Leave the store directly (not paying for any of the items in possession),
- Unconceal the item I at location X.

Deliberation is basically committing to one of these desires with item and location information. Clearly, current intentions also play a role in this deliberation process. For example, if the agent is not committed to shoplifting item I or has not picked up item I yet, the desire of concealing item I is irrelevant.

As discussed in Section 2.4.2., there are different classifications of shoplifters that can be found in the literature. Certain types of shoplifters predetermine the items that are interest to them and other types of shoplifters act on impulses. If there are predetermined items to shoplift, the desire to shoplift those items is immediately considered. If the shoplifter acts on impulse, beliefs should meet certain criteria to take the desire of shoplifting an item into consideration. Furthermore, some shoplifters prefer only picking up the items to shoplift and then leaving the store without paying for any item; where as some shoplifters prefer paying for some items while shoplifting others.

The decision to commit to a desire is basically handled similarly to what BDI Architecture proposes. First a filtering function is used to find a desire to evaluate and then this desire is evaluated to decide whether to commit. Filtering function is primarily responsible from prioritizing the desires that are taken into consideration and returning the one with the top priority. The evaluation function is responsible for making the decision of whether the desire is promising or not. The primary metric used by the evaluation function is security score. The security score is a number that is assigned to a certain volume in the store using the beliefs of the agent on the security measures in the facility. The most common volume unit used in this research while assigning security scores is the cell. This is the same unit employed by A* algorithm to find feasible paths. The agent using the beliefs calculates the exposure the cell has to the security measures in the facility and assigns a security score to the cell. The security scores will be further discussed in Section 4.3.4.

As an example, let’s assume that an intruder with a predetermined item to shoplift arrived at the facility. The top priority desire would be shoplifting this item and hence, it is selected out of the filtering process. The evaluation function checks whether the total security score of the zone that includes the item I is below a threshold value (the degree of risk seeking in zone assessment). If this is the case, the agent commits to the desire of shoplifting item I. If the desire to be considered is picking up item I (assuming that the agent is committed shoplifting item I), the evaluation function checks the security scores of the cells that have access to item I. If the minimum score is below the agent’s threshold value, which is determined by the risk aversion of the agent and named the degree of risk seeking in cell assessment, the agent commits to picking up the item using the cell (or location) with the minimum security score. The degrees of risk seeking values are part of the agent’s personality definition and this definition is further discussed in Sections 4.2.3 and 4.3.1.

After committing to a desire, the agent needs to find a course of action that potentially achieves the desire. This process is explained in Section 4.2.3.

The deliberation process is not used for workers or neutrals. They only perform predetermined list of activities. However, they share some of the mechanisms used to generate course of actions with the guards and the intruders.

4.2.3 Intuitions, Heuristics, and Reasoning

After committing to a goal, the question to answer is how to achieve the goal. For this purpose, the agent needs to generate a course of action that potentially achieves the goal. The proposed mechanism to model means-end reasoning process involves the methodology that the humans use in daily life. The primary argument is that humans employ different methods when they encounter different situations. The methods employed by humans are mostly simple sets of efficient rules or heuristics. Gigerenzer and Todd (1999) stated that these heuristics should be fast and frugal in order to be applicable in real life situations. The terms *heuristic* and *method* are used interchangeably to refer to the general set of methods employed by the agents in the ABIRA framework.

The means-end reasoning process in the ABIRA framework captures this methodology. Assuming there is a pool of heuristics at its disposal, the agent tries to find a heuristic after committing to a goal. The process of selecting a heuristic is affected by the personality and the experience of the agent and this process targets to find the heuristic that fits the most to the current situation as it is perceived by the agent. The selected heuristic is then used to generate a course of action and the agent analyzes the generated course of action by mental simulations. If the generated course of action is favored, it is used by the agent. If not, there are few options for the agent to consider, which will be discussed later in this section. This scheme basically represents the modeling of intuition in the ABIRA framework. This phenomenon is probably best laid out with the comment made by Cuban world class chess player Jose Raul Capablanca, who once said that: "I see only one move. Always the best one,"[18]when asked how he tries to look ahead and analyze an evolving chess game.

The selection of a heuristic for a particular situation requires the agents to understand the situation and then make a selection using the personality and the

[18] http://www.intuition-sciences.com/introduction

experience of the agent. Personality of an agent is defined by three parameters: the degree of risk seeking, the boldness, and the tendency for certain tactics. The degree of risk seeking is comprised of three values (1) the degree of risk seeking in zone assessment, (2) the degree of risk seeking in cell assessment and (3) the degree of risk seeking in course of action assessment. The first two values are used in the deliberation process as discussed in Section 4.2.2. The degree of risk seeking in course of action assessment is mainly used while deciding whether to favor a course of action generated by a certain heuristic. The boldness is mainly used in the reconsideration process. These two factors are determined by the type of the agent as classified in the application domain. The primary parameter of personality in selecting a heuristic is the tendency for certain tactics. As discussed earlier, the intruders can use different tactics while seeking to achieve their goals. The tactics of stealth, deceit, and force are associated with the personality of the agents, which are encapsulated in the types of intruders that are defined for different application areas. An agent using stealth as a tactic would try to apply heuristics that serve this purpose. Therefore, predefined heuristics should be classified based on their tactical requirements.

Experience of the agent is the other factor of importance while selecting a heuristic for a particular situation. The experience of an agent is basically represented by the success ratio of a heuristic for a certain goal under certain circumstances. There are three circumstantial factors of interest that are taken into consideration while building up experience for agents: (1) The type and number of other agents that are present in the area; (2) The activities of these agents; and (3) The static objects in the area. These success ratios can be either built up through simulation runs or can be set manually. For example, an intruder's assessment of a situation can be: (1) There are two guards, (2) Both of them are patrolling, and (3) There are 2 surveillance cameras. If the intruder is using stealth tactic, a heuristic that previously gave better results for this situation will be favored during the heuristic selection process.

Faced the same situation, two humans may or may not choose the same heuristic to generate a course of action for analysis depending on their personalities and experiences. However, even using the same heuristic does not necessarily generate the same course of action. These heuristics are applied on the current beliefs of the individual and hence it is quite possible that a different course of action is generated by the same heuristic. One final note is that the agents using the ABIRA framework are bounded by the heuristics employed by them. In other words, their solution generation capabilities are limited with the heuristics that are available to them.

If the agent finds a heuristic intuitively using the process described above, this heuristic is employed by the agent to generate a course of action. The generated course of action is then evaluated by running mental simulations, which are explained in the next section. If there is no heuristic immediately favored by the agent, the agent goes through the reasoning process. The reasoning process is similar to the System 2 of the two-system view. The reasoning process is a slow and serial process and it involves sequentially trying a set of heuristics in the agent's repertoire to generate courses of action and analyzing each course of action by mental simulations until finding a favorable course of action or concluding that it is not possible to achieve the goal. Each evaluation will require a certain amount of simulation time and hence the reasoning process will take a longer time to finish.

4.2.4 Analysis by Mental Simulations, Expectancies, and Reconsideration

The course of action devised by a heuristic is comprised of a sequence of activities that the agent is projecting to perform. In most cases seen in physical security systems, the activities only involve movements and physical activities such as picking up items, opening doors, monitoring surveillance camera streams etc. After generating a sequence of activities, the agent needs further analysis to assess the potential to achieve the goal sought. Considering the stochastic nature of some of these activities in addition to the necessity of taking into account other agents' activities, the agent generally needs to consider a *set* of possible sample paths. As discussed in Section 2.2.2., mental simulation is the process that is used by humans in performing the projection of future activities and assessing the likelihood of success in achieving the goal. Mental simulations require predicting the behavior of other agents, which requires projecting the agent into other agent's environment. For example, if an intruder uses force to open a door such that it is punctured, a guard that perceives the punctured door would likely check the situation. Therefore, mental simulations should be capable of modeling reasonable behavior on other agents' part. However, if the intruder opens the door with a key and leaves the door closed, the expected guard behavior would likely not involve checking the situation.

Mental simulations are run using the beliefs of the agent. Therefore, only the objects that are known to the agent with the known parameters are used in running the simulations. Anticipation of other agents' behavior is primarily generated by using the beliefs on other agents, which are discussed in Section 4.2.1.2. The primary output metric of the mental simulations is the proportion of the successful replications to the total number of replications in achieving the goal. Mental simulations are run and the output is used in making the decision whether to endorse the proposed course of action. The approach taken at this point is similar to the two-system view. As it is stated in Section 2.2.2., there are three cases; (1) direct endorsement of the course of action, (2) using the course of action as an anchor for adjustments, and (3) rejecting the course of action. This decision is made by comparing the agent's degree of risk seeking in course of action assessment by the result of mental simulation. The degree of risk seeking in course of action assessment, which is part of the agent personality definition, has two parameters: (1) direct endorsement and (2) accepting as an anchor. Therefore, if the probability of success, which is the result of the mental simulation run, is greater than the first parameter, the course of action is endorsed and implemented. If it is greater than the second parameter but less than the first parameter, the course of action is used as anchor to perform modifications that potentially increases the probability of success. Otherwise, the agent tries to find another heuristic that hopefully better fits the current situation.

When the agent chooses to make adjustments using the course of action in hand as an anchor, the agent first identifies the problematic activities. Based on this assessment, the agent applies an operator to the anchor course of action. Some of the example operators are:

- *Wait*: The agent waits at one of the points that is visited;
- *Hide*: The agent tries to hide behind a solid object in the facility;

- *Add, remove, change zone*: The agent changes one of the zones visited, or adds a new zone to visit, or removes a zone from the course of action.

An example for this adjustment process is given next. In this configuration, there are two surveillance cameras and a security guard in a retail store. There is one intruder at the entrance of the retail store. The goal of the intruder is to get an item from the second aisle on the left. In order to achieve the goal successfully, the intruder should not be seen while picking up this item. The intruder believes that there are two surveillance cameras in the store. In addition, the intruder believes that the security guard is patrolling in the store and the security guard is currently at the top right corner of the retail store. This situation is depicted in Figure 27.

Assume that the intruder agent developed a course of action using a heuristic that finds the shortest path to target point. This course of action is demonstrated in Figure 27. At time 1, the intruder is at the door. At time 2, the intruder gets out of the entrance area. At time 3, the intruder arrives at the lower end of the second aisle from the left and at time 4, the intruder arrives at the target point where the intruder picks up the item. Based on this course of action, the intruder agent can run mental simulations (or generate sample paths) to further investigate the situation. Based on the beliefs of the agent, the security guard is moving from right to left with a certain speed, which might differ in different replications of the simulation. One of the possible sample paths for the security guard’s movement is also depicted in Figure 27. Based on this mental simulation, the intruder is not visible to the surveillance cameras while picking up the item if the intruder’s beliefs on the surveillance camera locations are correct. However, the intruder assesses that there is a possibility that it is seen by the guard while picking up the item.

Assuming that the proportion of successful replications is less than the first degree of risk seeking parameter but greater than the second one, the agent performs a modification using the current course of action as an anchor. In this modification, the agent simple adds the fourth zone from the left to its path. This new situation is presented in Figure 28. The mental simulation process is rerun and if the proportion of successful replications is greater than the first degree of risk seeking parameter, the agent implements the modified course of action.

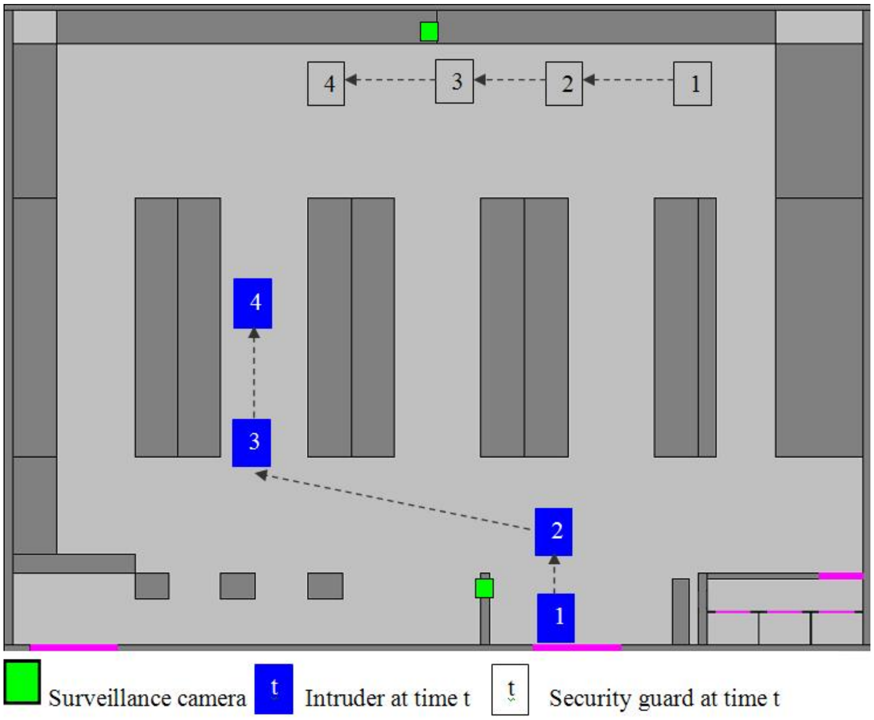


Figure 27 Initial Course of Action

Mental simulation is run similarly for the guards. The agent generates expectancies while running the mental simulations based on the successful replications. There are three types of expectancies that are generated by the agents:

- Position of the agent at time t,
- Position of other agents at time t.
- Number of times seen by security measures

The reason for expectancy generation is that the implementation of the course of action might be different than the projection. Therefore, the agent needs a mechanism to detect deviations from the situations occurring in the mental simulation. As stated in Section 4.2.3., one of the parameters that define the personality of an agent is the boldness. When a deviation from the expectancy occurs, the agent needs to make a decision on whether to reconsider the current course of action. The bolder the agent is the more deviation the agent can tolerate. If the situation requires reconsideration, the agent first checks whether the goal is still achievable. This is done by calculating the security

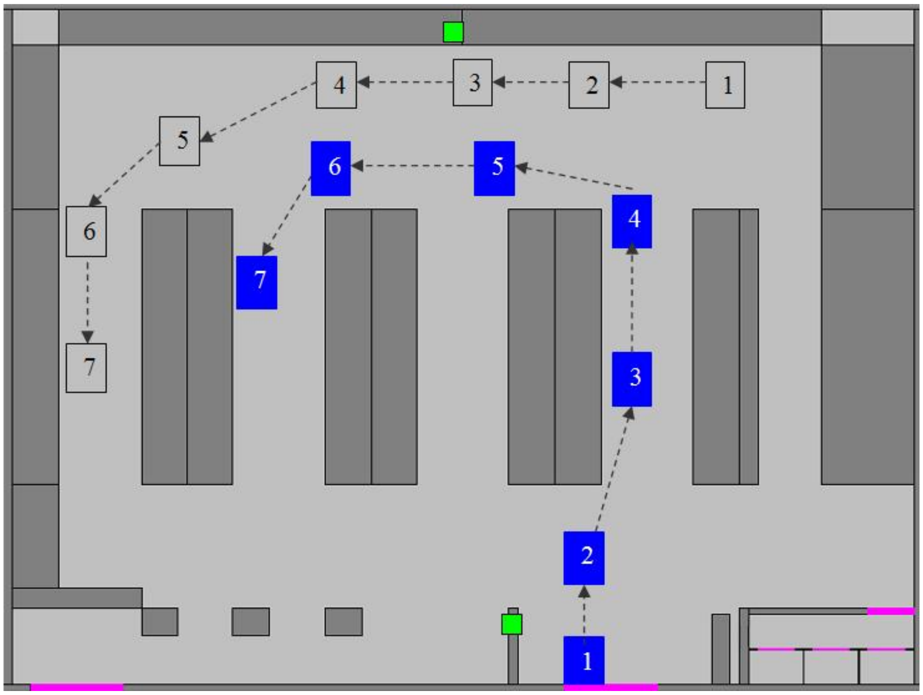


Figure 28 Course of Action Adjustment

score of the zone. The desire becomes invalid if security score exceeds the risk seeking value in zone assessment. If the goal is not achievable, the agent needs to perform a new deliberation. Otherwise, the agent updates the timing information on the current course of action and runs a certain number of mental simulation replications to check whether the course of action is still acceptable. If this is the case, the agent continues with the updated course of action. If not, the agent performs means-end reasoning starting with selection of a heuristic.

Acquisition of new information on static objects also needs to be taken into consideration. Assume that the intruder has discovered that there is a pallet in the second aisle from the left at time 6 (based on the course of action depicted in Figure 28). The mental representation of the intruder on the static objects at time 6 is presented using an In-Zone Graph construct in shown Figure 29. In this figure, the red cells represent the obstruction of movement by static objects, yellow cells are the cells seen by the surveillance cameras, and the gray cells are the cells that the intruder can go based on the intruder's beliefs on the object locations and the look direction of the surveillance cameras. The pallet blocks the original path devised for moving between locations at time 6 and at time 7. Therefore, the agent needs generate a new path that turns the corner around the pallet. This new course of action needs to be evaluated again by running mental simulations.

4.2.5 Action and Communication

The action module is primarily responsible for implementing the course of action that is generated in the deliberation layer of the ABIRA framework. However, the generated course of action is rather a sketch of what the agent anticipates as happening. Therefore, the action module needs to check the feasibility of the activities being performed as dictated by the course of action and to react to the changes in the environment by modifying the movements of the agent if necessary. Furthermore, the action module is also responsible for performing the activities such as opening a door such that the time required to perform these activities is also handled by the action module.

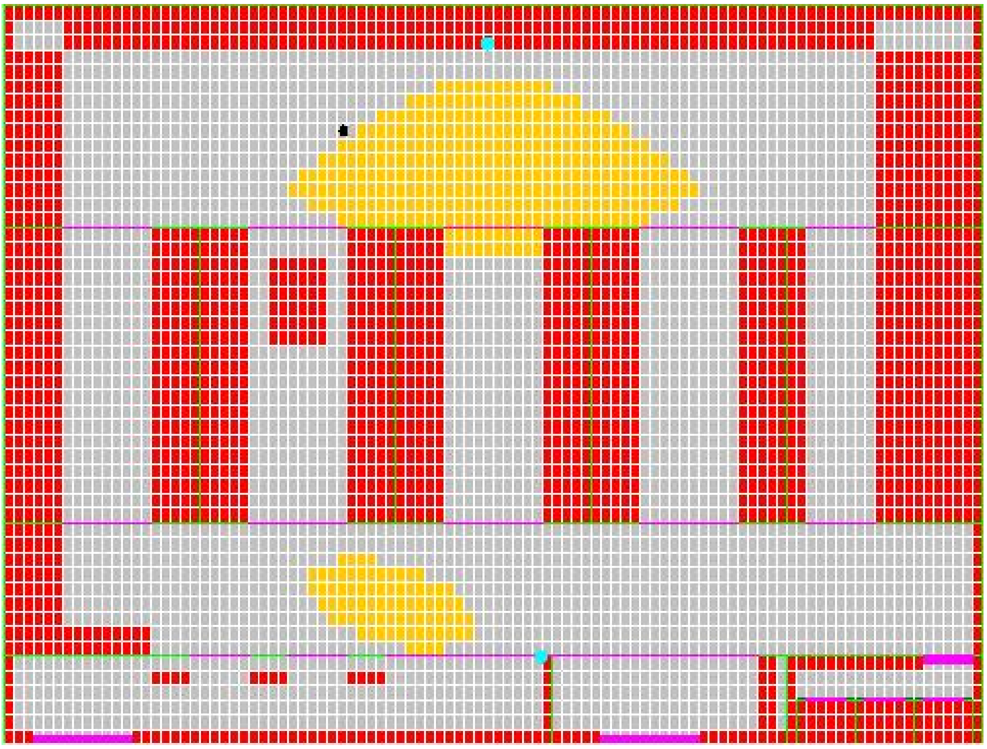


Figure 29 Intruder's Mental Representation at Time 6

The activities dictated by the course of action are developed using the beliefs of the agent. It is possible that the actual environment is different than the agent's mental representation of the environment. It is possible that there is solid object that the agent was not aware of that potentially blocks the movement of the agent or there is a possibility of colliding with other agents. If the next point in the movement sequence is inaccessible, the action module checks the remainder of the sequence for an accessible point. When such point is found, the action module runs the A* heuristic to update the path of the agent and to make it feasible. If no such point is found, the plan is assumed to be invalid and the means-end reasoning process restarts.

Communication between agents simply serves the purpose of belief update in the current ABIRA framework. Agents can update the other agents' beliefs using the communication mechanism. Communication between guards and between guards and workers is dictated by the security policy. Communication between intruders may take place if the intruders work in teams. Teams are not currently present in the ABIRA framework.

4.3 Application Domain Representation

The spatial and temporal models for the agents are captured in HIGHRES model and ABIRA framework, respectively. The final element of importance in agent's behavior is the application domain specific information. In this section, "Shoplifting in Retail Stores" example is used to demonstrate the application domain representation. The application domain representation is comprised of four parts: (1) Behavior characteristics for the participants and (2) Operational characteristics of the sensors including the guards, (3) The recognition functions for the agents, and (4) .The security scores as metrics for security assessment.

4.3.1 Participants

4.3.1.1 Intruders

Definition of intruders specific to the application domain is a two step process. First, the desires and their respective goals (goal is a resultant state of a desire; showing that the desire is achieved) and the activities that are used to accomplish the goals need to be defined. Second, the types of intruders that are acting in the environment need to be formally introduced.

The retail store security model is used to demonstrate the intruder definition process. Desires of the shoplifters were introduced in Section 4.2.2, while discussing goals and deliberation. Each desire is defined with the goal state (the desire is accomplished) and the interaction with other desires:

- *Shoplift item I:* This desire represents the agent’s commitment to shoplift item I. The goal is to leave the store with item I without being detained by the guards. This would at least require the agent to pick up item I and leave the store.
- *Pick up item I at location X:* This desire is handled in two different ways; if the agent does not have a desire to shoplift item I, the goal is simply going to the location X and picking up the item I. If the agent does have a desire to shoplift item I, the goal is going to the location X and picking up the item I without being detected by a security measure.

- *Conceal item I at location X:* This desire is considered if and only if the agent has a desire to shoplift item I and already picked up item I. The goal is going to location X and concealing item I without being detected by a security measure.
- *Unconceal item I at location X:* This desire is considered when the agent has concealed item I but dropped the commitment to desire “shoplift item I” for some reason. The goal can be either going to location X and unconceal item I without being detected by a security measure or the goal can simply be unconcealing the item I. This preference is dependent on the agent type.
- *Leave the store via cashiers:* The goal is going to the cashiers and paying for the items in possession except for the items that the agent has commitments to shoplift. The agent commits to this desire if there are no desired items to pick or conceal or unconceal.
- *Leave the store directly:* This desire is handled in two different ways; if the agent does not have a desire to shoplift any item, the goal is simply going to the exit door. If the agent does have a desire to shoplift any item, the goal is getting to the exit door without being detained.

The agent needs to perform activities to accomplish the goals that they have. The general flow of the activities for the intruders is depicted in Figure 13 using an activity diagram. Each activity requires some time to perform and this duration can vary. As stated in Section 2.4.4, some of the activities are critical in the sense that the guards need to detect these activities to confidently detain a shoplifter. The activities defined for the intruder are going to a location, picking up an item, concealing an item and unconcealing an item. The activities are denoted as critical if they are associated by shoplifting and hence they are candidates for detection by the security measures in the store. The conditions for being critical for each type of activity are listed as:

- *Picking up an item:* If the item is picked up while a desire for shoplifting the item is present, the activity is accepted as critical. The rationale behind this argument is that humans demonstrate some behavioral cues while picking up items for shoplifting as stated by Dabney (2004).
- *Concealing an item:* Concealing an item is definitely a visual cue if detected by the security measures in the store.
- *Unconcealing an item:* If the agent drops the goal of shoplifting the item, the agent can unconceal an item that is concealed. If possible, this needs to be caught by the security measures to prevent false detainments and hence avoiding the costs associated with false detainments.
- *Paying:* The activity of paying for the items is important from the security perspective, if the agent is paying for some items and not for the others. Therefore, it is important for the security measures to actually detect the agent not paying for the item. This also aims preventing false detainments.
- *Going to the exit area:* If a shoplifter has possession of item(s) and the shoplifter is getting into the exit area, the guards should approach the shoplifter and detain the shoplifter before the shoplifter leaves the store. Therefore, it is important to detect the suspects approaching the exit area.

The conditions for detection by the detection mechanisms in the store are discussed in Section 4.3.2. The mechanism to handle the interaction between the sensors and critical activities is explained in detail while discussing the simulation model in Section 4.4.

Second step in defining the intruders is classification of the intruders for the particular application domain, which, in this case, are the shoplifters. Before introducing the types of shoplifters used, one more definition needs to be done: *the security score.* As stated in Section 4.2.2, the security score is the perceived security of a certain area in the facility. Security scores can be assigned to zones or the cells, which are defined in Section 4.1.2. The security score for a cell is a function of number of cameras visible from the cell, number of guards becomes visible from the cell in X time units, and number of workers becomes visible from the cell in X time units. The security score for a zone is a function of the total security score of the cells in the zone and the average number of regular shoppers in the zone over X time units.

Definition of each type of intruder includes personality, desires and experience. An overall layout of the personality definition is as follows:

- *The degree of risk seeking*
    - *The degree of risk seeking in zone assessment*
    - *The degree of risk seeking in cell assessment*
    - *The degree of risk seeking in course of action assessment*
        - *Direct endorsement*: A value between 0 and 1.

        - *Using as an anchor*: A value between 0 and 1. Less than direct endorsement.

- *The boldness*: A value between 0 and 1. Five levels are defined:
    - Extremely bold- 0.9
    - Very bold- 0.7
    - Bold – 0.5
    - Cautious – 0.3
    - Very cautious – 0.1
- *The tendency for certain tactics*:
    - Stealth: Concealing of goods on the person including wearing the items under the current dress, pick up and conceal location are different. May or may not pay for other items.
    - Deceit: Switching price tags or removal of packaging. Conceal activity is used to model these activities and they happen at pick up location. Pay for other items.
    - Force: Simply grabbing the item and running away. Does not get any other items and hence does not pay for anything.

Ray (1987) and Dabney (2004) stated that approximately 8.5% of shoppers are shoplifter. Furthermore, Moore (1984) provided a classification of shoplifters as discussed in Section 2.4.2. Based on that discussion, different personalities are assigned to the shoplifter types in Moore's classification. This assignment is presented in Table 3. In this table, degree of risk seeking is abbreviated as DRS, zone assessment as ZA, cell assessment as CA, and course of action assessment as CAA. The first personality parameter “impulse” determines whether the shoplifter is an impulse shoplifter or has a predetermined item(s) before arriving at the store. The column “Activity Duration” represent how much time the shoplifters spend while performing activities. As discussed in Section 4.3.2, longer activity duration times increase the probability of detection. The column “Replan” determines whether the agent will try to shoplift again, if for some reason the agent drops the initial shoplifting intention.

Table 3 Intruder classification

Type\Personality	**Impulse**	**Tactics**	**Boldness**	**DRS -ZA**	**DRS -CA**	**DRS-CAA1**	**DRS-CAA2**	**Activity Duration**	**Re-plan**
Impulse	Yes	Deceit	Very Cautious	N/A	0.2	0.1	0.2	Long	No
Occasional	No	Stealth	Cautious	0.8	0.2	0.15	0.2	Medium	No
Episodic	Yes	Force	Extremely Bold	N/A	0.9	N/A	N/A	Short	No
Amateur	No	Stealth, Deceit	Bold	0.6	0.4	0.2	0.4	Short	Yes
Semi-pro	No	Stealth, Deceit, Force	Very Bold	0.5	0.4	0.3	0.4	Very Short	Yes

Different types of heuristics can be employed by the shoplifters: Some examples are:

- Minimum distance: Force, Later stages (while leaving the store) of deceit and stealth.
- Minimum security score: Stealth
- Avoid the guards: Force and stealth
- Avoid the cameras: Deceit

Heuristics defined for the developed scenarios are further discussed in Chapter 5.

As discussed earlier, experience plays a role in the selection process of the heuristics. Different mechanisms for learning such as reinforcement learning can be used to build up experience for the agents. However, experience models have not been implemented within this research. A further discussion on experience is made in Chapter 6, where future work is discussed.

4.3.1.2 Guards

A general outline for guard activities is presented in Figure 14 and further discussed in Section 4.2.2. There are four types of activities present in this discussion:

- *Regular Activities:* Patrolling, standing still, combination of patrolling and standing still, and watching surveillance video streams.
- *Sensory Activities:* Perceive
- *Decision Activities:* Decision to commit (state of being suspicious of a customer), decision to changing/dropping a commitment, and decision to detain.

- *Planned Activities:* Follow and Chase (and detain if caught the suspect)
- *Communication Activities:* Broadcast

Security policy dictates the rules for activities of type: (1) Regular, (2) Decisions, and (3) Communication; whereas the guard is solely responsible from sensory and planned activities as well as implementing the activities dictated by the security policy.

In general terms, a security policy should specifically define the rules for aforementioned three activity types. In addition, a guard can be wearing a uniform (and hence visible to shoppers) or disguised as a regular shopper (and hence covert). Each guard working in the facility should be using a security policy and there can be more than one security policy for different guards.

For sensory activities, the guards use their perception. This activity is further detailed in Section 4.3.2.

For planned activities, the guards use means-end reasoning.

An example security policy is:

- *Regular:* Continuously patrol in the store for two hours.
- *Decision:*
  - Commit when a critical activity is detected or broadcasted if not committed already.
  - Drop commitment if expectancies (developed in mental simulations) are violated.
  - Detain if the committed suspect reaches the exit area.

- *Communication:* When a critical activity is detected, broadcast this information to all guards in the store.

The goal in chase is catching the suspect before the suspect reaches exit. The goal in follow is to have the suspect visible all the time.

4.3.1.3 Workers

Workers' role defined in this research is primarily standing still at various locations in the retail stores (as if they are helping customers or working as cashiers). Workers may detect critical activities of potential shoplifters and let the guards know via communication. The detailed role descriptions for workers in the developed scenarios are provided in Chapter 5.

4.3.1.4 Neutrals

Neutrals are regular shoppers in the retail store security application. They try to shop for the items that are in their shopping list, which is randomly generated. The primary role of regular shoppers from the physical security perspective is that they can create false alarms. In other words, some pick up activities of regular shoppers are probabilistically labeled as critical activities and if these activities are recognized by the security measures, the guards act under the assumption that the regular shopper is a potential shoplifter unless something conflicting with this assumption is recognized.

4.3.2 Sensors

A classification on sensors of interest was presented in Section 3.3. The sensors used in the application are defined following this sensor classification. As it was stated, there are two ways to defeat a sensor: (1) Spoofing, and (2) Bypassing. No spoofing techniques are defined for the sensors in current state of this research. Bypassing is possible and this behavior needs to be generated by the intruders as described in the ABIRA framework and in Section 4.3.1.1.

Detection in the retail store security model is a little different than the other intrusion detection systems. First of all, the guards need to collect evidence in order to successfully detain the suspects. Therefore, the detection in the retail store security context refers to detecting critical activities which create suspicion. There are two types of sensors and both of them involve guards. Both types are responsible for detecting suspicious activities; the first one is the guards on the retail floor and the second one is the guards watching video camera surveillance stream. The next two sections detail the descriptions of these types of sensors.

4.3.2.1 Guards on the retail floor

As discussed earlier, there can be security guards in the retail store. These guards work as sensors and they try to detect suspicious activity. They are visible to other agents; however, not all the security guards wear uniforms. Therefore, security guards without a uniform can be accepted as covert. They primarily use vision as the detection mechanism and as discussed in Section 4.1.3., they use Line-Of-Sight for vision calculations. A range for vision and visibility cone half angle should be specified for these agents to perform volumetric Line-Of-Sight calculations.

Probability of detection for the guards on the retail store varies between different activities and depends on a certain number of factors. As stated in Section 4.1.3, there are two general types of recognition functions: (1) based on number of visual cues and (2) based on the type of visual cues. However, the critical activities that are defined in Section 4.1.1.1 require human comprehension to correctly interpret the activities happening in the environment. Tickner and Poulton (1973) discuss the factors that affect the detection probability of humans on certain activities from surveillance video streams. Important factors in detecting an activity are:

- Distance of the location of the activity to the camera (or looking point), closer to the activity better detection
- Duration of the activity, longer the activity better detection
- Hours worked by the guard, less hours worked by the guard better detection

These factors are encapsulated in the probability of detection model for guards on the retail floor. Each critical activity that is defined for the intruders is associated with these factors:

- *Pick-up (with the purpose of shoplifting):* Hands need to be seen. Distance, duration, and hours worked are important.
- *Conceal:* Hands need to be seen. Distance, duration, and hours worked are important.
- *Unconceal:* Hands need to be seen. Distance, duration, and hours worked are important.
- *Pay:* Only the number of visual cues is important.
- *Reach exit area:* Only the number of visual cues is important.

False alarms can be caused by regular customers (neutrals). The only activity that can cause a false alarm is the pick-up activity. Some of the pick-up activities of regular customers are candidates for a false alarm. If the hands of a regular customer performing the candidate pick-up activity are visible to the guard, this activity can raise a false alarm with a probability that is dependent on the distance, duration and hours worked (i.e. the farther the location of the activity, the higher the probability of a false alarm).

4.3.2.2 Guards watching the surveillance camera streams

Surveillance cameras located in the retail store have two functionalities: (1) Record the (critical) activities in the store for later review and (2) Stream the videos to the monitors that are watched by security guards. For the second case, security guards use the surveillance camera streams to perform detections. The guards watching the surveillance camera streams have lower probability of detection compared to the guards on the retail floor. The study performed by Tickner and Poulton (1973) reports that a group of people watching 16 monitors was able to detect the critical activities 59% of the time in an experimental study. They also state that decreasing the number of monitors can increase the percent detections such that the same group watching 8 monitors this time was able to detect the critical activities around 80% of the time.

The guards watching the surveillance camera systems are less likely to detect the critical activities when compared with the guards on the retail floor since it is harder to perform assertions from a motion picture. Detection probability model for guards watching the surveillance camera stream use the same factors that are introduced for guards on the retail floor. However, the number of monitors being watched is added as a factor such that the more monitors being watched the less probable is the detection.

4.3.3 Recognition

Agents use perception mechanisms to develop beliefs on other agents. The perception mechanism involves two consecutive activities: vision and recognition. Each agent in ABIRA framework is represented by a set of points and the Line-of-Sight calculations used to model the vision find how many points of the set representing the other agent are visible to the looking agent. This information is then fed to the recognition function to determine whether the looking agent recognizes the other agent (i.e., a recognition has occurred). Successful recognitions are used to update the beliefs of the looking agent.

As we have discussed in Section 4.2.1.2, there are 3 types of beliefs constructed on other agents: positions, types, and activities. Activities are further classified as either single step or multiple step activities. The agents use these beliefs for deductions on other agents' goals and to generate anticipations on agents' upcoming activities.

In the ABIRA framework, factors introduced in Section 4.3.2 are used to estimate the likelihood of recognition, which can be interpreted as probability of recognition. This probability is then used to determine whether recognition has occurred; a uniform (0, 1) random variate is generated and compared with the likelihood of recognition. The estimation process to determine the likelihood of recognition is discussed next. Note that the estimation process is specific to the problem domain and so is specific to the retail store security application for this dissertation. It is necessary to develop different mechanisms for the likelihood of recognition estimations for different application domains.

4.3.3.1 Positions of the agents

Recognition of the position of an agent involves basically detecting the presence of an agent at a certain location. Position recognition does not imply the type of the agent is detected; it is merely the construction of the belief on an agent's presence. However, if the type of the agent is known – this belief can be incorrect-, position information is associated with the agent type.

Likelihood of recognition for the position of the agent is calculated as:

$$\mathrm{k}_1 * \left(\frac{\#\text{ of points visible}}{\text{Total }\#\text{ of points representing the agent}}\right) + \mathrm{k}_2 * \left(\frac{(\text{Range} - \text{Distance})}{\text{Range}}\right)$$

(note that this likelihood is only valid when the agents are closer than the range

(Distance < Range).

For the guards watching surveillance camera streams, number of monitors being watched is also important. As it is stated by Tickner and Poulton (1973), the likelihood of recognition decreases when the number of video streams being watched is increased. This is modeled by defining coefficient based on the maximum number of cameras can be monitored and the number of cameras being monitored. If the maximum number of cameras that can be monitored is 16, the likelihood of recognition function is defined as:

$$\left(\frac{\text{\# of points visible}}{\text{Total \# of points representing the agent}}\right)$$

$$* \left(log_{16+2}\left((16+2) - \ number\ of\ monitors\ being\ watched\right)\right)$$

Assuming all points of the agent are visible to the camera, the likelihood of recognition as a function of number of videos monitors is plotted in Figure 30 for the case where the maximum number of video streams can be monitored is 16.

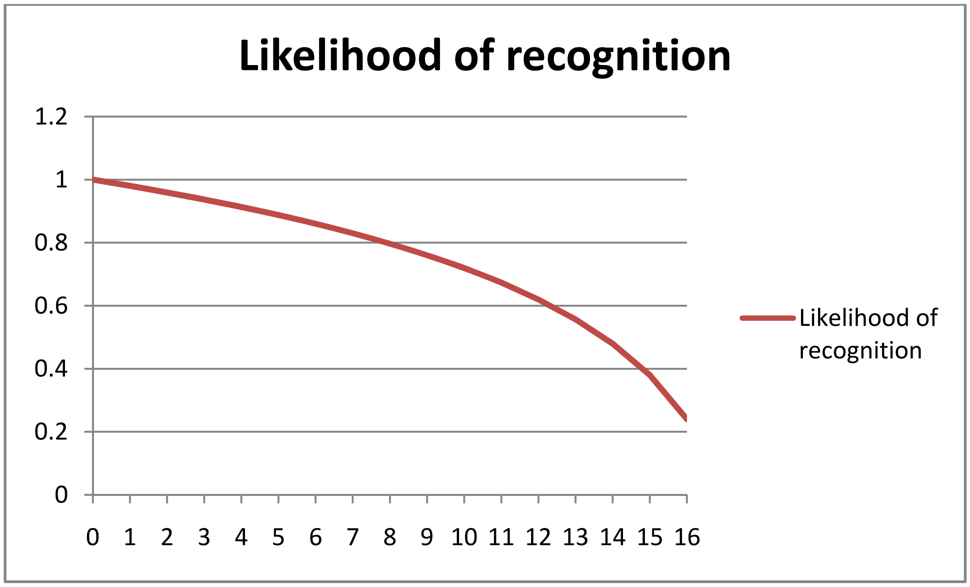


Figure 30 Likelihood of recognition as a function of number of video streams monitored

Guards collect position information only on potential shoplifters. Workers do not check shoplifter positions unless such a request is broadcasted by the store security personnel. Regular shoppers do not collect position information. Shoplifters check the positions of all other agents in the store.

4.3.3.2 Types of the agents

Recognition of the agent type is dependent on recognition of activities or the presence of other agents and hence it can be stated that recognition of the type is a by-product of other recognitions. Nevertheless, it is extremely important to correctly recognize the types of other agents within the context of the retail store application.

In the retail store security application, guards, workers, and regular shoppers need to recognize the shoplifters. For workers and regular shoppers, this does not mean tracking the shoplifters but informing the store security on recognized activities. The guards, on the other hand, should be able to label agents as potential shoplifter based on observed activities. Initial labeling of agents as potential shoplifter would only mean that these agents need to be tracked. Decisions on detain will be made based by assessing the activities observed by the agents labeled as potential shoplifters.

The shoplifters also need to differentiate between guards, regular shoppers, and workers. Even though the presence of workers or regular shoppers around would deter the shoplifters, the shoplifters are mostly interested in which agents are guards. As stated before, guards can be in uniform or not and this implies recognition of different things to recognize the type of the agent as guard. In parallel to the general approach in the ABIRA framework, the type of the agent is not always correctly recognized and the beliefs constructed using agent type recognitions effect further mental processes.

Table 4 summarizes what triggers the recognition of which type of agent and who can be mistakenly recognized as a different type of agent. As it is stated, in Section 4.3.1.1, it is assumed that guards can recognize shoplifters while they are picking-up items even though it is possible to recognize regular shoppers as shoplifters. However, this is only limited to the guards; workers or regular shoppers need to detect a conceal activity to recognize agents as shoplifters.

Table 4 Agent type recognition

Detecting Agent	**Type Of Agent**	**Required recognitions**	**False recognitions**
Shoplifter	Guard in uniform	Presence	None
Shoplifter	Guard without uniform	Follow activity	Workers and regular shoppers
Shoplifter	Worker	Presence	None
Shoplifter	Regular shopper	Presence	None
Guard	Shoplifter	Pick-up or conceal activities	Regular shoppers
Workers	Shoplifter	Conceal activity	None
Regular shoppers	Shoplifter	Conceal activity	None

4.3.3.3 Activities of the agents

As it is stated in Section 4.2.1.3, two types of activities are specified: single step and multiple steps. In single step activities, the belief is constructed only by recognizing the activity itself, whereas in multiple step activities, the agent needs to infer a belief from multiple recognitions of the presence of other agents. Furthermore, the belief is continuously updated in multiple step activities as new recognitions of the presence occur. All critical activities are assumed to be single step activities; picking up item, concealing an item, unconcealing an item, paying for items and going to the exit area are all single step activities. Example multiple step activities are patrolling, following, and chasing.

For the single step case, the objective is to determine whether the activity is recognized anytime during its duration. Important parameters in recognizing a single step activity are the duration of the activity, number of points seen on the subject agent, and distance to the subject agent. For guards, fatigue level also plays a role in recognition. For surveillance stream monitoring guards, the number of video streams being watched also plays a role in recognition.

The approach used in modeling the likelihood of recognition for single step activities is that there will be a number of check points during the duration of the activity. At each check point, the looking agent generates a value for likelihood of recognition, which also takes into account earlier checks. In general, the longer is the duration of the activity, the higher is the number of check points. As it is going to be demonstrated later, increase in the number of checkpoints also increases the likelihood of recognition when everything else remains constant. It is assumed that for each activity, there is a fixed number of check points and this is dictated by the type of the activity.

For guards there are three components of the likelihood of recognition function: the vision component, the distance component, and the fatigue component. If the guard is watching a surveillance camera video stream, the vision component is updated in a way that the number of surveillance video monitors is taken into account. For shoplifters, workers and neutrals, the fatigue component is dropped out of the likelihood of recognition function. Final value for the likelihood of recognition is a weighted average of the components used.

The vision component of the overall likelihood of recognition function is defined as:

$$\frac{\left(\sum_{i}^{number\ of\ possible\ checks} \frac{number\ of\ points\ seen\ in\ check\ i}{total\ number\ of\ points\ representing\ the\ agent}\right)}{number\ of\ possible\ checks}$$

In order to visually demonstrate this function, assume that an activity defines ten check points. The subject agent is represented by 6 points and the looking agent sees only 1 point of the subject agent in the first 20 % of the checks, 2 points in the second 20% of the checks, 2 points in the third 20% of the checks, 4 points in the fourth 20% of the checks, and 5 points in the final 20% of the checks. The likelihood of recognition based on only the vision component is depicted in Figure 31.

As stated in Section 4.2.1, a differentiation is made between the visual cues in recognition. For some activities, only the number of visual cues is important for recognition, whereas for other activities, certain visual cues need to be seen for recognition. At each check point, recognition is possible if the agent sees at least one visual cue. If the agent needs to see a certain set of points (e.g. hands) for recognition then at least one of the visible points should be from the set of points that is associated with the activity.

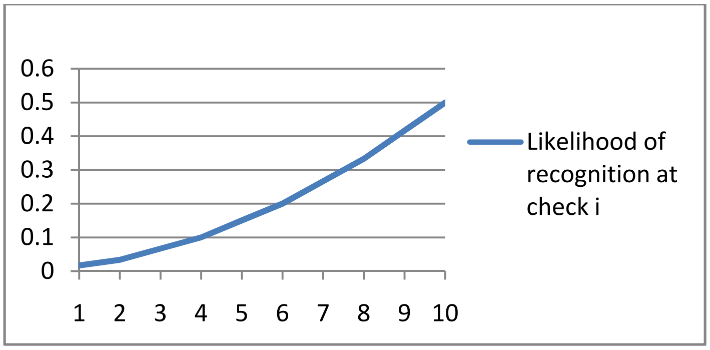


Figure 31 Likelihood of recognition at check i only using the vision component

The cumulative likelihood of recognition is shown in Figure 32. Based on the sequence of number of points seen of the subject agent, there is an approximate 7% chance that the activity of the subject agent will not be recognized by the looking agent.

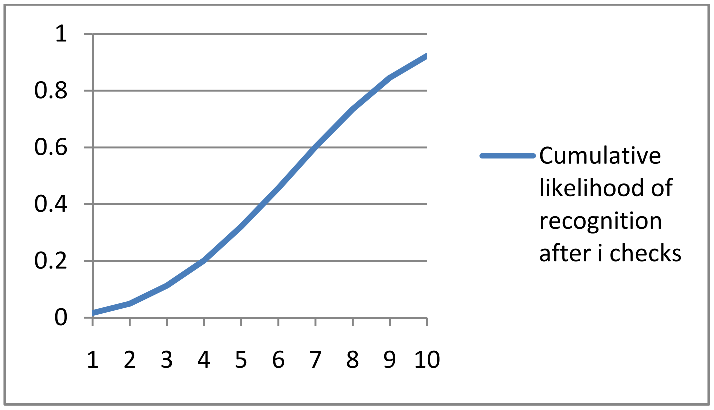


Figure 32 Cumulative probability of recognition after i checks only using the vision component

If the number of checks is increased to 30 given that the sequence of number of points seen is same, the chance of the activity being unrecognized drops to 0.1%. The cumulative likelihood of recognition for both cases is demonstrated in Figure 33.

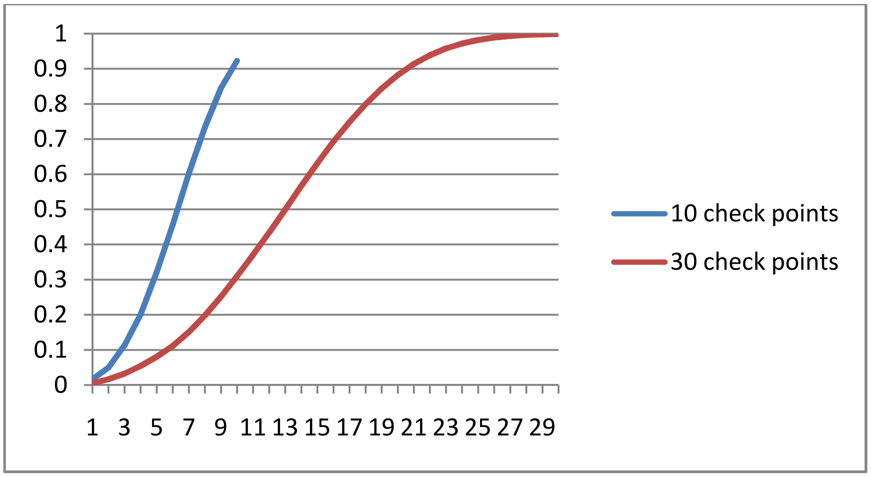


Figure 33 The cumulative likelihood of recognition for 10 and 30 check points after i checks only using the vision component

If the guard is not on the retail floor but watching surveillance video stream, the vision component of the likelihood of recognition function needs to be updated. The vision component for this case is defined as:

$$\frac{\left(\sum_{i}^{number\ of\ possible\ checks} \frac{number\ of\ points\ seen\ in\ check\ i}{total\ number\ of\ points\ representing\ the\ agent}\right)/number\ of\ possible\ checks}{number\ of\ video\ streams\ watched+1}$$

Assume, for example that the activity defines 30 check points and all six points of the subject agent are visible to the camera all the time. The cumulative likelihood functions of recognition for the vision component in cases of different numbers of surveillance video streams being monitored are depicted in Figure 34. As it can be read from the figure, the likelihood of recognizing the activity for the eight monitors is approximately 83% and for the 16 monitors is 60%. These numbers are close to the results of the experimentation of Tickner and Poulton (1973) as discussed in Section 4.3.2.2.

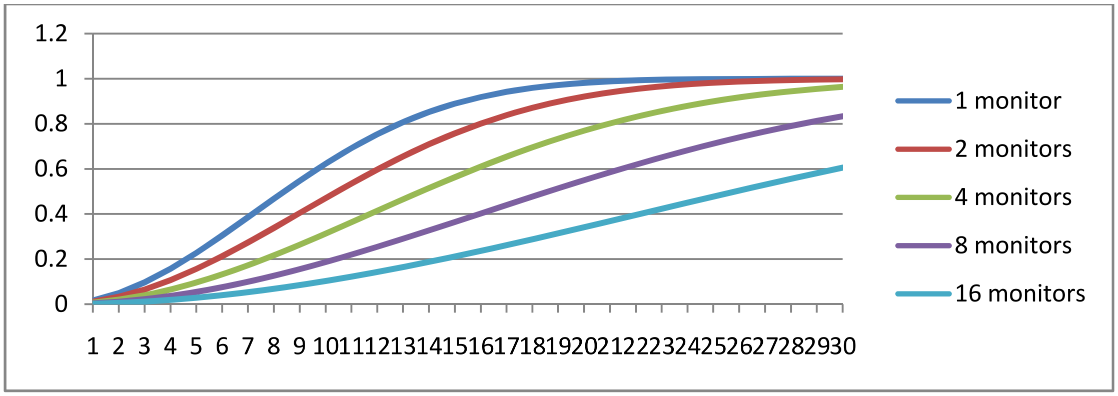


Figure 34 The cumulative likelihood of recognition for 1,2,4,8,16 monitors cases using only the vision component

The distance component of the overall likelihood of recognition function is defined as:

$$(Range - Distance)/Range$$

where range is the maximum distance that can be effectively seen by an agent.

The fatigue component of the overall likelihood recognition function is defined as:

$$(Shift\ length - Hours\ already\ worked)/Shift\ length$$

The overall likelihood of recognition function is then:

$$k_1 * Vision\ Component + k_2 * Distance\ Component + \ k_3 * Fatigue\ Component$$

Where $k_1+k_2+k_3=1$, $k_2=0$ for guards watching the surveillance video stream, and $k_3=0$ for shoplifters, workers, and regular shoppers.

For multi step activities, two different approaches are employed. First one follows a similar logic as the single step activities; a number of check points are created for the activity and if the activity is recognized then the agent recognized the activity correctly as a whole. The second approach requires on the agent part the ability to develop an understanding of other agents' actions. For example, chase and follow activities are recognized if a shoplifter recognizes the presence of a guard for a certain number of consecutive checks. Another example is devising a patrolling path for the guard based on the locations of the guard agent, where its presence is recognized.

4.3.4 Security Scores

Security scores are used for intruders (shoplifters for the retail store application) in the ABIRA framework and they represent the perceived security of a certain area in the facility. Higher security scores represent a high level of perceived security and the intruders try to avoid high security areas in general.

The security score for any area is dependent on the number of guards, surveillance cameras, workers and neutrals around that area for a specified time interval. In the ABIRA framework, security scores vary in the range of zero to 1 and are comprised of four components; guard security score, surveillance camera security score, worker security score, and neutral security score. The overall security score is a weighted average of these individual security scores.

The retail store application defines two types of areas that security scores are calculated for: zones and cells. Cells are the result of the cellular decomposition applied to areas as discussed in Section 4.1.2. Basically, a zone is comprised of a number of cells. Each cell is a cuboid defined by 8 points. In the discussion of security score, the center of the cell is also used and hence, there are 9 points of concern for each cell. The security score for a zone is calculated by averaging the security scores of randomly selected cells that are in the zone. In the current application, the percentage of cells used in zone security score calculation is around 20%.

Security scores are based on the exposure of the cells to the security measure. Assuming that the primary perception mechanism is vision, the security score of a cell is a reflection of the intersection of the cell with the cones of view of security measures. The visibility of the cell’s 9 points to the security measures is used in calculating security scores as an approximation of volumetric intersections in the retail store security application.

Time plays an important role in security score calculations. The shoplifters are more interested in the exposure of the cells to the security measures during the planned time interval of the activity. In the retail store security application, only the surveillance cameras have fixed locations but even the surveillance camera’s look direction can change in the case of a scanning surveillance camera. Therefore, the security score of a cell can vary by time depending on the positions, look directions and movements of other agents and surveillance cameras.

The security score of a cell is calculated for a certain integer time interval and this time interval is approximately double the time required to perform the activity. The security score calculation basically involves checking the exposure of the cell at discrete instants during the time interval and it uses the beliefs of the agent to anticipate the locations of other agents. The security score of a cell for a single security measure (e.g. guard) is calculated as:

$$\log_{\text{time interval}}\left(1 + \sum_{i=0}^{time\ interval} log_{number\ of\ guards}\left(1 + \sum_{j=1}^{number\ of\ guards} log_{10}(1 + number\ of\ cell\ points\ visible\ at\ time\ i\ by\ guard\ j\right)\right)$$

For a single time unit evaluation with one guard, the change of security score as a function number of points visible is depicted in Figure 35. As it can be seen from the figure, this is a concave function and marginal contribution to the security score is bigger for smaller number of points visible.

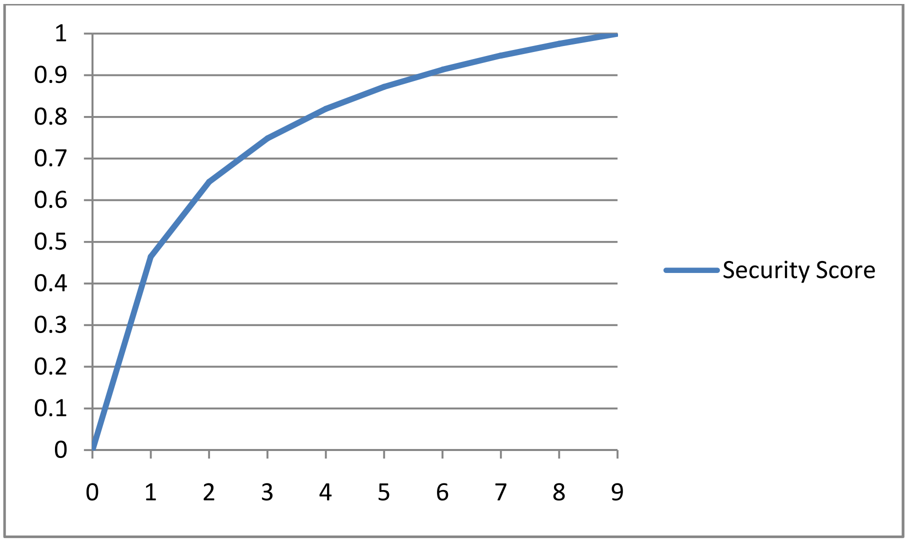


Figure 35 Security score as a function of points visible – single guard, one time unit case

For a single time unit evaluation with two guards, the change of security score as a function number of points visible by each guard is demonstrated in Figure 36.

Overall security score for a cell can then be calculated as:

$$k_1 * guard\ security\ score + k_2 * surveillance\ camera\ security\ score + k_3 * worker\ security\ score + k_4 * neutral\ security\ score$$

Where $k_1+k_2+k_3+k_4=1$ and $k_1>k_2>k_3>k_4$

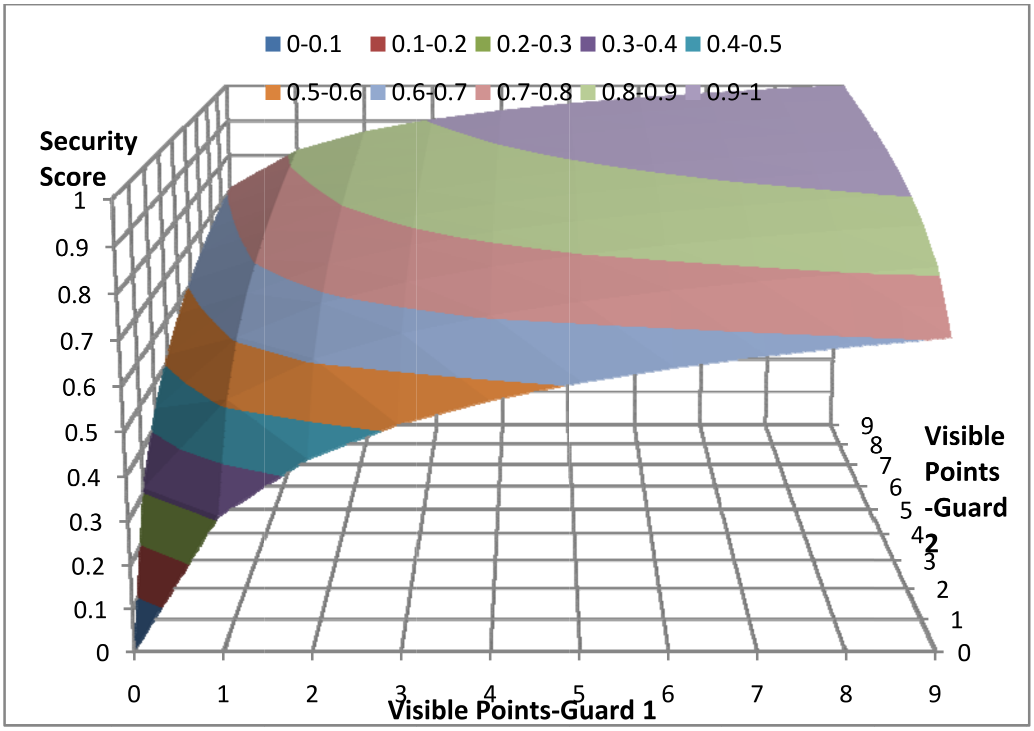


Figure 36 Security score as a function of points visible – two guards, one time unit case

4.3.5 Security Policy

The security policy defined for the hypothetical retail store example follows the general discussions made in Sections 3.4.1 and 3.4.3. Roles of guards, surveillance cameras, workers and regular shoppers are all specified in the security policy of the hypothetical retail store. Table 5 summarizes the responses of agents except guards to different types of critical activities. As it can be guessed, the response of the surveillance camera is initiated by the guard if he/she recognizes the critical activity. The activity of inform for the regular shopper involves either one of the following: (1) Go to a worker and tell or (2) Call a guard and tell. The regular shopper prefers to call a guard if there is a guard (with uniform) visible immediately after the recognition. If there is no guard around, the regular shopper checks for a worker. If there is one visible, the regular shopper informs the working by going near the worker and telling. If no workers are visible, the regular shopper continues his/her regular path until he/she sees a guard or a worker. The action is taken based on who is seen first. If the regular shopper informs a worker, this information is broadcasted by the worker.

The security guards are primarily responsible for the follow and detain activities and they recognize all types of critical activities. The detain activity includes chasing the potential shoplifter and detaining him/her if caught. If a security is not already committed to a following a potential shoplifter, recognition of any type of critical activity results in the security guards committing to following the potential shoplifter. If there are multiple not committed security guards (in the case of critical activity is broadcasted), the security guard closer to the location of the critical activity commits to following the potential shoplifter. If all the security guards are committed, it is possible for one security guard to change commitment. In this case, the following priority criteria are used in order to change commitment.

1. Go to exit area
2. Pay (recognition of not paying for certain items)
3. Conceal
4. Pick-up

For example, if a security guard is following a potential shoplifter of whose only recognized activity is *pick-up*, recognition of an activity *conceal* overrides the initial commitment (if there are no other security guards available).

Table 5 Responses to critical activities

Critical Activity/ Type of Agent	**Surveillance Camera**	**Worker**	**Regular Shopper**
Pick-up	Broadcast	No	No
Conceal	Broadcast	Broadcast	Inform
Unconceal	Broadcast	Broadcast	Inform, if detected conceal
Pay	Broadcast	Broadcast, if notified	No
Go to exit area	Broadcast	Broadcast, if notified	No

Detain is possible if and only if the potential shoplifter is headed to exit (in the exit zone). Basis of this argument was laid out in Section 2.4.4. The security guards detain potential shoplifter in cases of following sequences of critical activities:

- Pick-up- Go to exit area;
- Pick-up- Conceal- Go to exit area;
- Pick-up-Pay- Go to exit area;
- Pick-up- Conceal- Pay- Go to exit area;

- Conceal- Go to exit area;
- Conceal- Pay- Go to exit area.

Recognition of *Pay* activity in this context actually means that the security measures recognize that the potential shoplifter is not paying for some items while paying for the others.

Commitment to following a potential shoplifter is dropped if an unconceal activity is recognized (for the item of concern).

One addition to the defined security policy is having the option to check the records of surveillance cameras for earlier activities of potential shoplifter. For example, the security guards watching the surveillance video streams can check the recorded video streams for earlier pick-up activities of a potential shoplifter if he/she is headed to the exit area. However, this functionality is not incorporated in the security policy of the current scenarios.

This section defines the security policy that is used for sample scenarios defined in Chapter 5 for the hypothetical retail store example. It is definitely possible to define different types of security policies but for demonstration purposes, the defined security policy is sufficient.

4.4 Simulation Model

The general simulation and the mental simulations run on a Discrete Event Simulation (DES) engine that is developed based on the descriptions by Schriber and Brunner (1997) and Law and Kelton (2000). Agents move in the facility on paths which are discretized by points. Heuristics employed to generate paths use the In-Zone Graph representation and the generated path is comprised of the center points of the visited cells. In the DES representation, arrival to each of these points is an event and hence, the discrete event approach can be used for simulating the movements of all of the agents.

The activities of the agents are modeled with two instantaneous events; the start of the activity and the end of the activity. Before the agent starts an activity, the start and the end of the activity events are scheduled and pushed into the Future Event List (FEL). The critical component in the discrete event representation is the recognition component. As it is stated in Section 4.3.3, the looking agents perform multiple observations during the duration of the activity and the results of these are observation are fed to the recognition function. This phenomenon is modeled by generating observation events for the agents that are potentially interested in recognizing the activity. For example, before a shoplifter starts a critical activity (e.g. conceal), multiple events are scheduled for the sensors in the facility (e.g. guards and surveillance cameras) throughout the duration of the critical activity. These sensors then perform Line-Of-Sight calculations and the results are fed to the respective recognition functions. Similarly, the observations on movements are performed at discrete points arrived along the path of the subject agent.

As described previously, the primary activities for agents are go to a location, pick-up, conceal, unconceal, pay, go to exit, follow, chase, inform, and standing still. The *go to location* activity is comprised of mainly a sequence of points which constitutes the path of the agent. Each arrival to a point is modeled as an event and movement of the agent occurs on discrete steps.

Pick-up, conceal, unconceal, and pay activities are similar from an event oriented perspective. Each activity is modeled with a starting and an ending event. These activities are all critical activities and hence, recognition events are created for other agents that are interested in these activities. These recognition events are created for the duration of activity and the number of recognition events created is dependent on the type of activity.

Inform activity of the regular shoppers is modeled in two different ways depending on who is being informed. If it is a security guard being informed, the regular shopper waits for the security guard to come near and informs the security guard. If it is a worker, the regular shopper goes near to the worker and informs the worker.

Standing still is an activity in which the agents wait for a certain period of time. Standing still activities are used in inform and follow activities as well as in go to location activities. Even though the agents stand still while performing critical activities, standing still activity has a different meaning for intruders; they can use this activity as a mean to avoid being exposed to security measures. Therefore, a *go to location* activity can include several standing still activities.

Follow and chase activities are composed of several *arrive location* events. In addition, a follow activity can include *stand still* activity. Both of these activities are planned primarily based on the anticipated movements of other agents and heuristics are employed to generate the events to model these activities.

4.5 Chapter Summary

This chapter formally defines the spatial and the temporal models for the agents. The HIGHRES model is developed as a formal computational representation of the facility and the ABIRA framework suggests a formal framework to generate realistic agent behavior. Finally, the application domain information is defined for the hypothetical retail store security application. These three components together are required to run simulation experiments for assessing the vulnerability of a facility, which in this case the hypothetical retail store. In Chapter 5, multiple scenarios are developed and run to demonstrate the capabilities of the proposed model and to validate the generated agent behavior.

5 EXPLORATORY RESULTS AND VALIDATION

The HIGHRES model, the ABIRA framework and the hypothetical retail store security application were conceptually defined in Chapter 4. These conceptual definitions are used to develop the physical security systems analysis software. A number of scenarios have been developed in order to validate and demonstrate the conceptual models discussed. The exploratory results generated from sample runs of the scenarios are presented and a brief discussion of the effects of security policies on agent behavior is provided in Section 5.2. The exploratory results along with the execution of the scenarios are used to demonstrate the achieved variety in agents' behavior. Finally, validation of the computational framework is presented using the validation scheme discussed in Section 3.5. Animations of interesting interactions in-between agents and between the agents and the environment are captured as videos from various simulation runs and they are used to support various discussions in this chapter. These videos are presented with the attached digital media or can be obtained from http://www.volkanustun.com.

5.1 Scenarios

A pool for security system measures is generated first. In this pool, patrolling paths for two security guards, locations for eight surveillance cameras, and locations for three workers are defined. Each scenario developed uses a combination of the security measures defined in this pool. Locations for the cameras, locations of the workers, and the patrolling paths for the security guards are shown in Figures 37, 38, 39, and 40. One addition to the visual representation of guard patrolling paths is that the guards stop and look towards the mid-aisles while passing by during their patrols. Table 6 presents the five different security configurations developed for testing and experimentation purposes. All surveillance cameras are mounted to the ceiling and are all visible to the shoppers in the store. The “look direction” of cameras 1, 2, 5, 6, 7, and 8 is downwards. The camera 3 looks towards the top aisle and the camera 4 looks towards the dressing rooms.

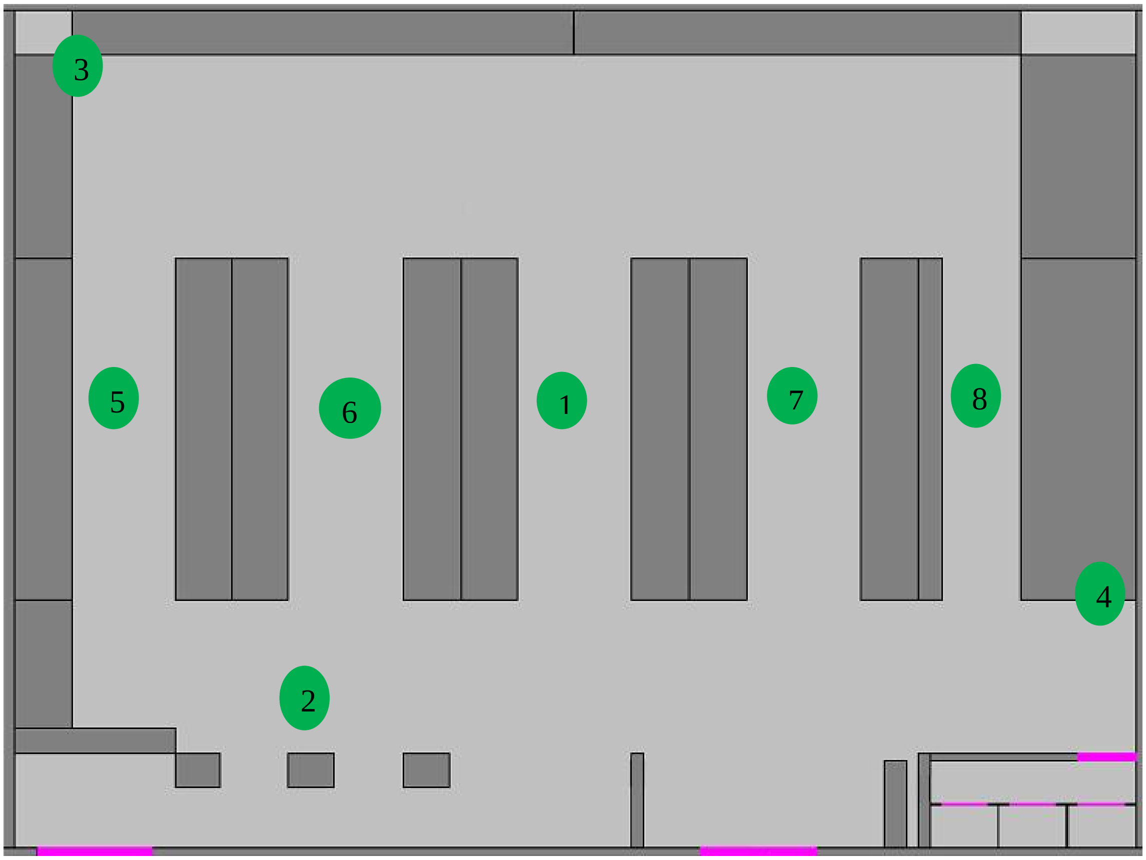


Figure 37 Surveillance camera locations

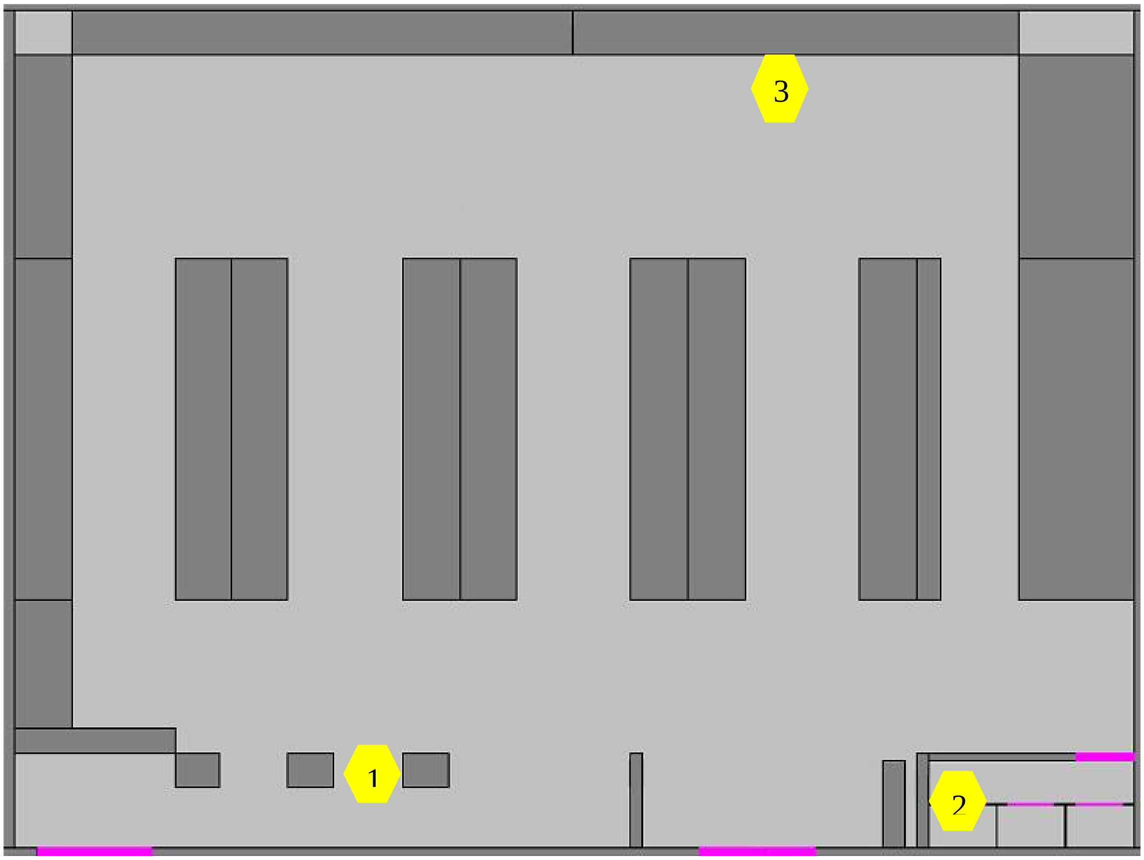


Figure 38 Worker Locations

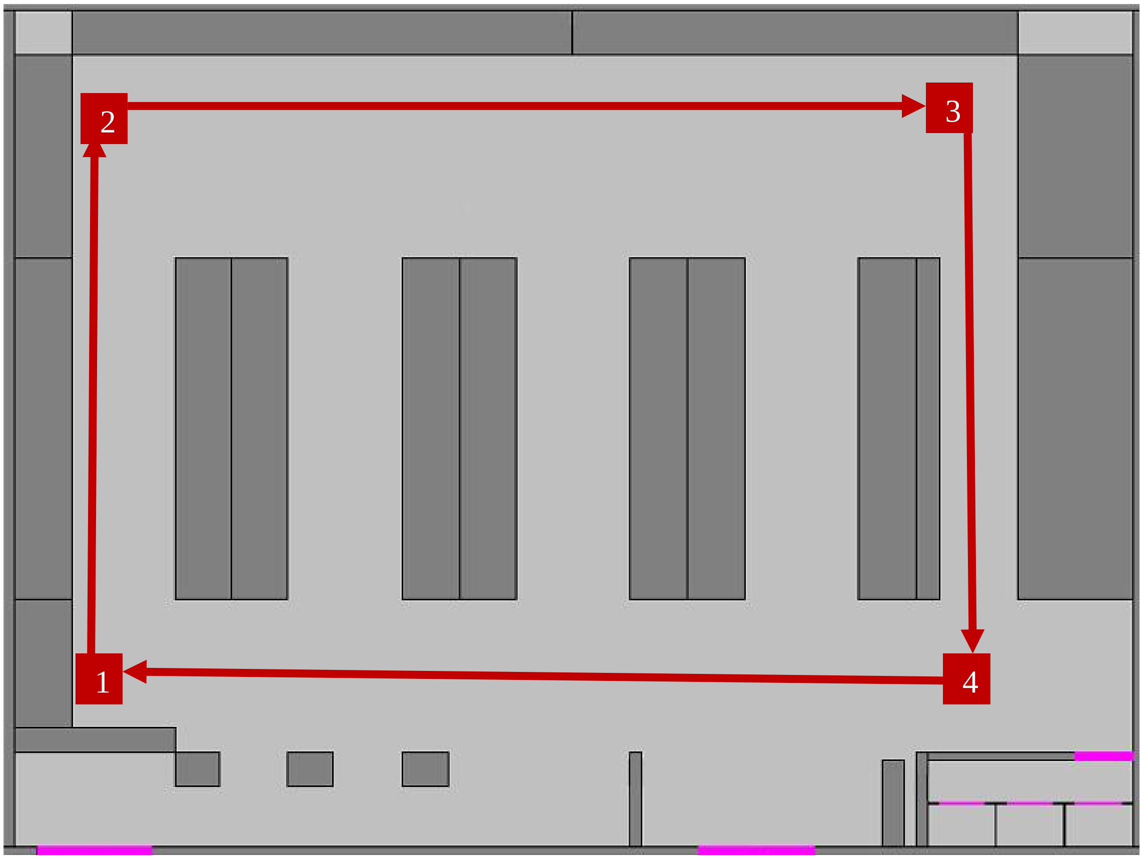


Figure 39 Patrolling Path of Guard 1

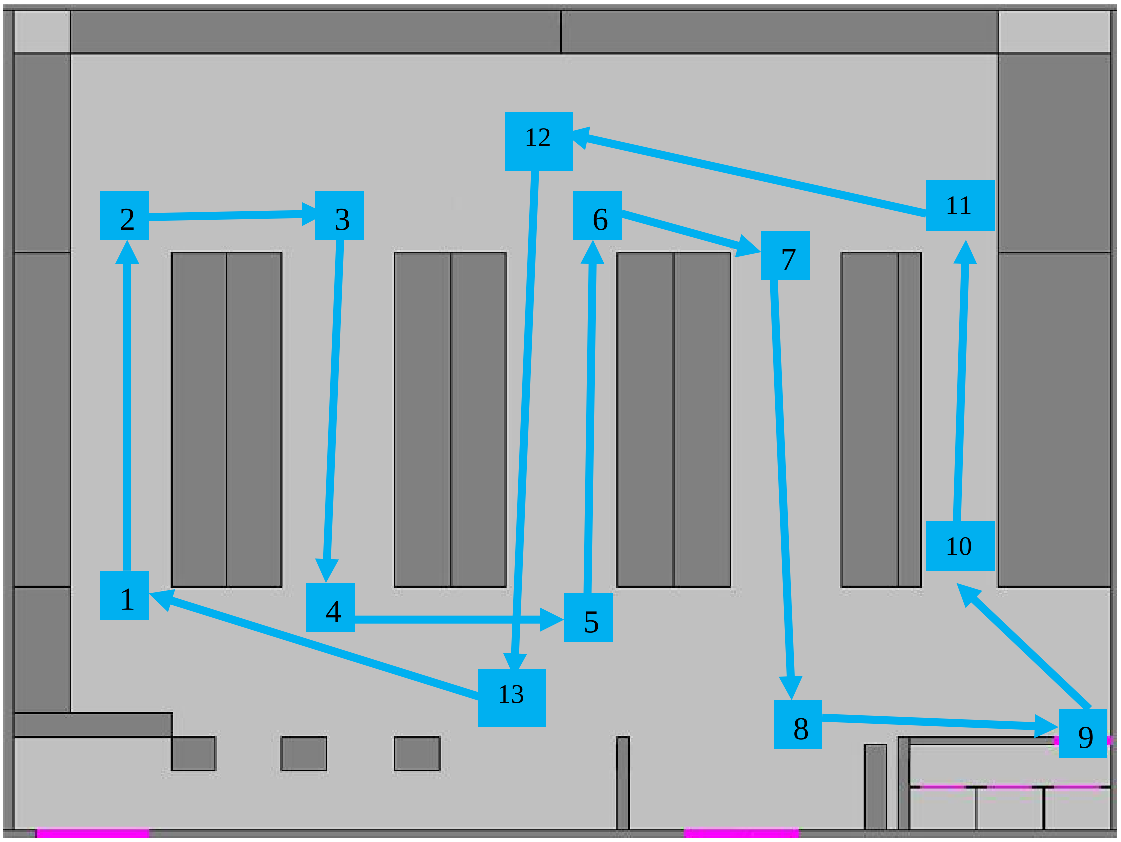


Figure 40 Patrolling Path of Guard 2

Table 6 Security System Configurations

	Guards Used	**Surveillance Cameras**	**Workers**	**Shoppers**
Scenario 1	1	1,2,3,4	None	No false alarms
Scenario 2	2	1,2,3,4	1	No false alarms
Scenario 3	1	1,2,3,4	1	False alarms
Scenario 4	1-2	1,2,3,4	All	False alarms
Scenario 5	1-2	All	All	False alarms

In recognizing positions of the other agents, all types of agents use 0.8 as the value of k_1 (associated with number of points visible) and 0.2 as the value of k_2

(associated with distance to the target agent). The coefficients used in activity recognition are listed in Table 7.

The coefficients that shoplifters use for calculating security scores are tabulated in Table 8.

Table 7 Coefficients for Activity Recognition

Type of Agent/ Coefficients	**Guard/Worker**	**Stream Watching Guard**	**Shoplifter/Shopper**
k_1-Vision	0.8	0.8	0.8
k_2-Distance	0.15	0.0	0.2
k_3-Fatigue	0.05	0.2	0.0

Table 8 Coefficients for security scores

Shoplifter type/ Coefficients	**Impulse**	**Occasional**	**Episodic**	**Amateur**	**Semi-pro**
k_1-guard	0.4	0.4	1.0	0.5	0.35
k_2-surveillance camera	0.3	0.4	0.0	0.4	0.6
k_3-worker	0.2	0.15	0.0	0.1	0.05
k_4-neutral	0.1	0.05	0.0	0.0	0.0

A number of heuristics have been defined for the use of different types of agents in these scenarios. All agents use A* algorithm while creating movement paths. However, the cost structure used by different types of agents can be different based on the goals and the tactics of the agent. Guards and regular shoppers employ the A* algorithm with physical distance as the cost metric. The shoplifters' cost metric also includes exposure to surveillance cameras, workers, and neutrals in addition to the physical distance component. The shoplifters committed to shoplifting and using the stealth tactic use the updated version of cost metric while running A* algorithm.

Shoplifters that are impulsive (impulse and episodic shoplifters) use the following basic heuristic while deciding whether to shoplift an item:

- If the activities of all detected guards are recognized (only for impulse shoplifters, episodic shoplifters do not perform this check),
    - After picking up an item, look around.
    - If there is no guard visible and the security score is below the degree of risk seeking, commit to shoplifting the item.

Guard agents' follow heuristic is defined as:

- If target is visible
    - If the target is more than 20 distance units away, move to 20 units distance.
    - If not, look towards the target
- If not
    - Move to the last known location of the target
    - If still not visible
        - Look around

        - If still not visible, stop the “follow” activity

Guard agents’ chase heuristic is as follows:

- Run immediately to the exit door
    - If the potential shoplifter has not arrived yet, wait for the potential shoplifter to arrive and detain.
    - If the potential shoplifter already left, stop the “chase” activity.

The locations of different items in the hypothetical retails store were depicted in Figure 9. These items and their hypothetical average dollar values are presented in Table 9. The scenarios defined in this section are used to get some exploratory results in the next section. Furthermore, they form the basis for the validation discussion in Section 5.3.

Table 9 Hypothetical Retail Store Items and Their Dollar Values

Item	**Dollar Value**		**Item**	**Dollar Value**
Beverage	$2		Drug	$15
Fruit	$1		CD	$15
Dairy	$5		Electronic	$50
General	$10		Grocery	$4
Clothing	$15		Deli	$3
Shoe	$25		Frozen	$4
Cosmetic	$20		Magazine	$5

5.2 Exploratory Results

Following output metrics are defined for the hypothetical retail store security problem:

A. Number of shoplifters that successfully shoplifted an item,

B. Number of shoplifters detained,

C. Number of false detains,

D. Number of potential shoplifters deterred,

E. Total value of shoplifted items.

F. Percentage of successful shoplifters

All scenarios are run for a 2-hour shift for 5 replications. Inter-arrival times of customers are exponentially distributed with a mean of 3 minutes. Arrivals are cut at 1 hour of simulation time and 2 hours of run time was enough for all agents to leave the store. Each customer's shopping list size is uniformly distributed between 1 and 15 except for episodic shoplifters, who have shopping list size of 1 (they grab an item and leave the store without paying for it). 30% percent of customers are potential shoplifters. These scenarios are run with only impulse and episodic shoplifters and 80 % of potential shoplifters are impulse shoplifters, whereas the rest are episodic shoplifters. There is a 0.05 chance for each pick-up activity belonging to a regular customer to be considered as critical and hence, the recognition functions of the security measures are triggered for these activities. There is a 75% chance that a shoplifter knows about the location of a surveillance camera and the surveillance cameras are not visible from the retail floor. The average results of 5 replications and the half width of the 95% confidence interval on the mean are presented in Table 10.

Table 10 Results of exploratory simulation runs

Avg. Metric Values/ Scenarios	**A**	**B**	**C**	**D**	**E**	**F**
Scenario 1	1.4±1.0	1.2±1.14	1.2±0.96	2.2±1.14	20.4±22.81	0.22±0.12
Scenario 2	3.2±1.69	1.8±1.69	1.8±0.73	1.4±1.0	58.8±52.36	0.39±0.15
Scenario 3	2.6±0.78	2.0±1.07	1.2±0.39	1.8±1.14	33.6±18.60	0.37±0.14
Scenario 4	1.8±1.14	1.6±1.71	2.6±1.0	2.4±1.59	17.0±27.00	0.23±0.17
Scenario 5	1.2±0.96	2.2±1.14	2.0±0.88	1.0±0.62	16.4±21.9	0.16±0.13

The main assumption made in these scenarios is that the recognition of “PayAllItems” or “PaySomeItemsDoNotPayOthers” activities requires recognition of at least 75% of the all pick up activities. As can be seen from the results, the lowest percentage of successful shoplifters seems to be with Scenario 5 (more replications needed to show statistical difference). Overall, episodic shoplifters seem more successful than the impulse shoplifters: over all scenarios and replications 72% of the episodic shoplifters were successful, whereas only 44% of the impulse shoplifters were successful.

The relatively high number of false detain is mainly associated with the assumption made with the recognition of “PayAllItems” or “PaySomeItemsDoNotPayOthers”. With this assumption, the security policy dictates to detain the potential shoplifters if there is suspicion. A more lenient security policy can be not detaining the potential shoplifters if a pay activity is recognized, but whether the potential shoplifter is paying for all the items picked up cannot be determined, unless there is a recognized “conceal” activity of the potential shoplifter. 5 replications for scenario 5 are run with this more lenient security policy and as expected, no false detains occurred. However, the percentage of successful shoplifters appears to increase in this case; the average is 0.33 and the half width of the 95% confidence interval on the mean is 0.07. The software tool provides the capability for discussions of this kind and supports the analysis of different security system configurations.

The significant point here is not the results themselves but the demonstrated variety of agent behaviors. This variety is not merely the result of different random numbers used; it is the result of the deliberate course of action generation process used by the agents. Agents use their intentions and beliefs while generating courses of actions that potentially achieve their goals. The beliefs about the environment and other agents are constructed by recognizing the environment. Different courses of actions are generated by applying heuristics on the perceived environment (the mental representation), which creates interactions and behaviors that are difficult to anticipate and impossible to enumerate in advance. Therefore, even with a limited number of heuristics, it was possible to observe a wide variety of potential activity sequences and interactions between agents that cannot be easily foreseen.

Two animations captured while running replications for Scenario 3 are provided (Scenario3_1.avi and Scenario3_2.avi) to support these arguments. In both cases, black solid circles represent guards (one is patrolling and one is monitoring the surveillance streams), blue solid circles represent the impulse shoplifters, the green solid circles represent the episodic shoplifters, and the white solid circles represent the regular shoppers. The light blue solid circles that are not moving represent the locations of the surveillance cameras in the retail store. The impulse shoplifters (blue solid circles) change color when they consider shoplifting an item after pick up; they either turn yellow if they commit to shoplifting the item or turn purple if the perceived security is higher than their risk seeking parameters. Furthermore, if an expectancy violation occurs, the impulse shoplifters turn purple from yellow representing the fact that they have given up on shoplifting the item. Immediately after this color change, the impulse shoplifters perform the “unconceal” activity. Guards go to the exit door if they have decided to detain a potential shoplifter; if they go to the exit door for a purple or white color agent, these are false detains. If they go to the exit door for a yellow or green color agent, this is a detain activity that is right on the target. If purple color agents leave the store without being detained, they are .considered as deterred shoplifters.

In Scenario3_1.avi, the single patrolling guard follows impulse shoplifters and mostly deters them from committing to shoplifting an item. There is only one impulse shoplifter that commits to shoplifting at animation time 1:10. The conceal activity of this impulse shoplifter is not recognized (the location of conceal is outside the area covered by surveillance cameras). When this impulse shoplifter is about the leave the store at animation time 1:43, this activity is recognized by the guard that is monitoring the surveillance stream and a message is broadcasted. Per the original security policy, the patrolling guard heads to the exit door but the patrolling guard cannot get to the exit door before the impulse shoplifter leaves the store. There is also a false detain that happens at animation time 1:56.

In Scenario3_2, false alarms mostly keep the patrolling guard busy; the pickup activity of a regular shopper at animation time 0:51 falsely recognized as a critical activity by the guard monitoring the surveillance stream. While the patrolling agent is involved with following the regular shoppers, an episodic and an impulse shoplifter leaves the store unrecognized with stolen items.

As shown in these sample animations, there is a significant variety in behaviors exhibited by the agents. This clearly demonstrates that the computational framework introduced in this dissertation is capable of creating different and reasonable behaviors as required by the agents' perceptions of the environment. As a result, the differences in agent behaviors are caused not only by the randomness typical in most simulation models, but also by the computational behavior models included in the ABIRA framework.

5.3 Validation

The hypothetical retail store security application has provided a realistic platform to validate the computational human decision making framework. As discussed in the validation scheme, a number of interesting interactions between agents are captured as animations and they are used to validate the computational decision making framework. This is done by first asking human subjects whether the actions in the captured videos are reasonable given the goals and knowledge of the decision making agent and the heuristics available to them (Face Validation). Secondly, human subjects are asked what they would do anything different if they were in the same situation as the decision making agent was (Predictive Validation). Ten human subjects were shown the captured animations. The actions in these animations are found to be reasonable by all the human subjects. Furthermore, no human subject could think of a course of action different than the one taken by the agents in the animations. Brief descriptions are provided next for the captured animations that are used in validation.

5.3.1 Camera Discovery

This case can be seen in cameras.avi. Light blue circles represent the surveillance cameras. The agent is at the entrance of the retail store at the beginning of the video and the agent is trying to go to the top aisle with minimum exposure to the surveillance cameras. The agent has information on a single surveillance camera and the initial plan is made with this information. However, as the agent tries to move according to the plan, the agent recognizes other cameras and the plan changes each time a new camera is discovered (cameras are discovered at animation times 0:02, 0:04, 0:08, 0:12, and.0:14). Finally, using the A* algorithm with a camera exposure factor included in the metric, the agent selects the path with minimal expected exposure to the surveillance cameras to reach the destination point.

5.3.2 Zone Discovery

This case can be seen in NoZoneInfo.avi. The agent has coordinate information on the destination point but the agent does not know the existence of the zone that the destination point is located. The agent selects the zone from its knowledge base that is closest to the destination point. While going to the selected zone, the agent discovers the zone that includes the target point (at animation time 0:03). Agent changes paths as it moves along since the agents try to minimize its exposure to surveillance cameras.

5.3.3 Object Information Update

This case can be seen in WrongInfoSolidObject.avi. The agent has incorrect information on the size of the object that is located in Aisle 3 (second aisle from the left in Figure 9) such that the agent believes that the object blocks Aisle 3. Therefore, when it is planning the path to reach the destination point in the top aisle, the agent avoids Aisle 3. When the agent reaches its destination point, it realizes that the object actually does not block Aisle 3 (at animation time 0:10) and hence, on its way back to the entrance area, it uses Aisle 3 since the path through Aisle 3 is shorter than the path used to get to the point. A similar case of object information update would be when an agent plans a path to a target and does not know about an object blocking the aisle (e.g., a store clerk unloading a pallet and blocking the aisle). In this case, the agent would re-plan the path after seeing the object and incorporating the newly seen object into its mental representation of the environment.

5.3.4 Guard Following Potential Shoplifter

This case can be seen in guardfollow.avi. Here, the blue circle represents the impulse shoplifter, the white circle represents the regular shopper, and the black circle represents the guard. The impulse shoplifter uses the heuristic defined in Section 5.1 to decide whether to shoplift an item. The guard recognizes the pick-up activity of the potential impulse shoplifter (at animation time 0:16). The guards finds this pick-up activity suspicious and starts following the potential shoplifter. The potential shoplifter probabilistically decides to look around after some pick-up activities (and at each time the potential impulse shoplifter sees the guard and hence, it cannot commit to shoplifting any item (the color of the agent turns to purple if the potential shoplifter considers shoplifting but he/she can not commit to shoplifting the item after looking around). The potential shoplifter leaves the store without any attempt to shoplift an item when the shopping list is completed. The guard returns back to patrol activity after the potential shoplifter leaves.

5.3.5 Guard Detains Shoplifter

This case can be seen in detain.avi. The blue circle again represents the impulse shoplifter. The potential shoplifter checks several times to shoplift an item. Eventually at animation time 0:43, the potential shoplifter perceives an opportunity (the agent looks around and perceives no security measures). While evaluating this opportunity, the potential shoplifter runs a mental simulation (detainMentalSimulation.avi) for the expected duration of the conceal activity using the anticipated movements of the guard (the recognized patrol activity). The potential shoplifter decides to commit to the shoplifting activity (the color of the agent turns yellow) and starts the activity of concealing the item. However, the guard recognizes the conceal activity and starts following the impulse shoplifter. When the guard recognizes that the impulse shoplifter is not paying the full amount and is headed to the exit door, the guard runs to the exit door and the guard detains the impulse shoplifter before the shoplifter leaves the store. After the detain activity, the guard continues to patrol the store.

5.3.6 Shoplifter Shoplifts Item

This case can be seen in shoplift.avi. The potential shoplifter picks up several items and probabilistically decides to try shoplifting the last item in the shopping list. The potential shoplifter looks around after picking up the item and recognizes no security measures. The agent then runs a mental simulation (shopliftMentalSim.avi) at animation time 0:50 and assesses that the guard wouldn't be able to recognize the conceal activity (The "conceal" activity for an impulse shoplifter involves changing the price tags of the item or removing the package of the item). The impulse shoplifter conceals (e.g. changes the prices tags) the item, goes to the cashiers, pays for the other items in the shopping list (and pays less for the shoplifted item. Recall that the impulse shoplifter uses the deceit tactic and it involves changing the tags for the shoplifted. So practically the impulse shoplifter pays less for the shoplifted item and leaves the store. The security system in the store was unable to detain the impulse shoplifter in this case.

5.3.7 Expectancy Violation for Conceal Activity

This case can be seen in expectancyviolation.avi. In this case, the impulse shoplifter does not know the presence of Surveillance Camera 3 and at animation time 1:13, the impulse shoplifter commits to shoplifting an item that is just picked. After the commitment, the first activity performed is the "conceal" activity and the expectancy for the impulse shoplifter is not seeing any guards while performing the "conceal" activity (based on the mental simulations run). However, the "conceal" activity is recognized by the guard monitoring surveillance stream and this information is broadcasted as a message to the patrolling guard. The patrolling guard goes to the "conceal" location to check the situation. The "concealing" impulse shoplifter recognizes the patrolling guard (between animation times 1:18 and 1:22) and since this conflicts the expectancies previously constructed, the impulse shoplifter gives up on shoplifting and performs the "unconceal" activity (and hence its color turns purple). This is an example of a highly complex behavior (determining expectations, observing reality, identifying the conflicts between expectation and reality, and replanning), that seems relatively simple.

5.3.8 Reconsideration of Commitment to Shoplifting an Item

This case can be seen in reconsideration.avi. An impulse shoplifter decides to “conceal” an item at animation time 0:06. The “conceal” activity is recognized by the surveillance camera monitoring guard and this information is broadcasted. The patrolling guard goes to the “conceal” location to check the situation. The guard agent arrives at the “conceal” location towards the end of the “conceal” activity and even though the “concealing” impulse shoplifter recognizes the guard, this does not violate the expectancies (guard recognized for only short amount of time during conceal and the impulse shoplifter can tolerate that). However, the impulse shoplifter later recognizes that the guard is following (at animation time 0:17). Reconsidering the situation, the impulse shoplifter decides to give up the intention of shoplifting and performs the “unconceal” activity (and hence its color turns purple).

5.4 Chapter Summary

In this chapter, different security configurations have been defined for the hypothetical retail store application and these configurations are used to collect evidence to validate the computational human decision making framework. As suggested by the human subjects, the computational human decision framework is capable of emulating the human decision making process and the generated course of actions are realistic and reasonable. Furthermore, agents exhibit a variety of different behaviors when they face different situations as discussed in Section 5.2. None of the responses in these demonstrated behaviors are preprogrammed; they are the results of the computational agent decision making process that uses the agents' beliefs and intentions. All these evidences point in the direction that the developed computational human decision making framework is valid. Furthermore, it is also capable of creating different courses of actions that actually reflect the current mental representation of the environment by the agent.

6 CONCLUSIONS AND FUTURE RESEARCH

Two recent articles published in mainstream newspapers[19,20] report increased shoplifting activities for the retail store amid the current recession. These recent articles further discuss that retail stores are not only investing in technologies to prevent shoplifting but also investing in development of knowhow and better tactics to deal with the increased shoplifting activities. One of the main reasons for this change is the formation of shoplifting crime rings and the perception of shoplifting becoming more of a professional activity than just being crime of the adolescents and drug addicts. These crime rings work in teams and their activities are planned better than random acts of shoplifting.

The author of this dissertation believes that developed computational models of human behavior and devised model to represent retail store security application domain can be valuable for the retail stores to analyze their current and planned physical security systems. Furthermore, the computational models in this dissertation can be utilized to design more effective physical security configurations. These security configurations of concern are not merely comprised of physical security tools and technology but also tactics and policies that are integral to the overall physical security of the retail stores. In addition, the proposed methodology provides a medium to the retail stores to better

[19] http://www.nydailynews.com/news/us_world/2008/08/24/2008-08-24_targets_target_theft_as_economy_sinks_an.html

[20] http://www.washingtonpost.com/wp-dyn/content/article/2008/12/23/AR2008122302585.html?sub=new

estimate the returns on investment of physical security systems tools and technology. For example, it is possible to estimate the indirect benefits of security system configurations in deterring potential shoplifters. Therefore, the author believes that the proposed methodology can be employed by the retail stores if the general idea and structure of the proposed methodology can be conveyed to the security system managers of these retail stores.

The success of the proposed methodology depends on the realism and the variety of behavior that is generated by the ABIRA framework. The ABIRA framework is extendable in this sense since it uses heuristics to model human intuition. Introduction of different heuristics directly relates to the emerging behavior. In addition, applying these heuristics on the perceived environment (the mental representation) creates interactions and behaviors that are difficult to anticipate in advance. Therefore, even with a limited number of heuristics, it is possible to observe a wide variety of potential activity sequences and interactions between agents that cannot be easily foreseen. The deliberate choice of modeling intuition by heuristics relies on the findings in the literature as well on the way the author visualizes his own planning activities in real life situations. In most real life cases, the author believes that he uses simple heuristics to generate plans for his daily activities and then he uses reasoning (via mental simulation) to assess whether these plans can work within the perceived environment.

The HIGHRES model and the interaction between the HIGHRES model and the ABIRA framework capture the visual cognition of the agents. The central role of visual cognition on agent behavior and the difficulty of explicitly incorporating this visual cognition in simulation models is a major cause of the limited use of complex and realistic behavior in simulation models. The difficulty of representing the spatial features of the environment is a significant component in the difficulty of modeling visual cognition. The HIGHRES model was an important milestone for this dissertation since it allows to formally define the environment and to effectively incorporate the spatial features of the environment for visual cognition. Performance tests conducted for the line-of-sight algorithm generated good results and this algorithm is used extensively for visibility calculations in physical security system simulations. Vision is used as the primary perception mechanism by the agents. Recognition is dependent on vision and is modeled as a probabilistic function in which recognition probability is dependent on certain factors such as the percentage of visible volume on the target entity and number of successful observations made over the duration of the activity, the fatigue level of the looking agent etc. Recognitions are then used to construct and to update the internal representation of the environment and other agents and they use this model while making decisions.

Physical security systems used in banks, retail stores, and buildings are possible application domains for implementation of the research and tools described in this dissertation. However, the described methodology is quite generic and it can be plugged into a wide variety of simulation models such as simulation-based games and evacuation models that use human entities and are particularly concerned with the interactions of these entities. Application of computational models to a different domain also provides an interesting opportunity to test the adaptability of the proposed models to different application domains. Furthermore, it provides a chance to develop better formalisms to define the application domain. Current application domain descriptions need refining to make them more accommodating.

As in retail stores, several physical security systems face intruders that work as teams and they also include guard team that collaborate effectively. Current models in this dissertation use communication as a medium to support collaboration but integration of formal collaboration frameworks would contribute to the value of the proposed computational models.

Experience is another area that the proposed models need to incorporate. Different learning techniques can be employed to build up experience for the agents and then agents can use their experience in effectively finding heuristics that are more appropriate to the perceived situation.

REFERENCES

Aminadn M. (2002). Toward *Secure and Resilient Interdependent Infrastructures*. Journal of Infrastructure Systems 8(3): pp: 67-75

Barabasi AL (2002). *Linked: The New Science of Networks.* Perseus: Cambridge, MA.

Beyth-Marom R, Fischoff B, Quadrel MJ, and Furby L (1991). *Teaching Decision Making to Adolescents: A Critical Review*. In Teaching Decision Making to Adolescents, eds: J. Baron and R.V. Brown. pp: 19-60. Lawrence Erlbaum Associates: Hillsdale, NJ.

Bourg D.M. and Seaman G. (2004). *AI for Game Developers.* O'Reilly Media.

Bratingham P.J. and Bratingham P.L. (1984). *Patterns in Crime.* McMillan Publishing Company.

Bratingham P.J., Bratingham P.L., and Glasser U. (2005). *Computer Simulation as a Research Tool in Criminology and Criminal Justice.* Criminal Justice Matters. 58.

Bratman M.E. (1987). *Intention, Plans, and Practical Reason.* Harvard University Press.

Bratman M.E., Israel D.J., and Pollack M.E. (1988). *Plans and Resource-Bounded Practical Reasoning.* Computational Intelligence. 4(4). pp: 349-355

Bratman M.E. (1999). *Faces of Intention: Selected Essays on Intention and Agency.* Cambridge University Press.

Brooks R.A. (1986). *A robust layered control system for a mobile robot*. IEEE Journal of Robotics and Automation. 2(1). pp:14-23.

Brown H.I. (1995). *Rationality*. In The Oxford Companion to Philosophy, ed: Ted Honderich. Oxford University Press.

Bryant B.D. (2006). *Evolving Visibly Intelligent Behavior for Embedded Game Agents.* Technical Report AI-06-334. Artificial Intelligence Laboratory, University of Texas at Austin.

Budden M.C. (1999) *Preventing Shoplifting without Being Sued: Practical Advice for Retail Executives.* Quorum Books.

Candalino T.J., Kobza J.E., and Jacobson S.H. (2004). *Designing Optimal Aviation Baggage Screening Strategies Using Simulated Annealing.* Computers & Operations Research. 31. pp: 1753-1767.

Carroll J., and Weaver F. (1986). *Shoplifters' Perceptions of Crime Opportunities: A Process-Tracing Study.* In *The Reasoning Criminal: Rational Choice Perspectives on Offending.* eds: Derek B. Cornish and Ronald V. Clarke. Springer-Verlag.

Clarke R.V. (1997). *Situational Crime Prevention: Successful Case Studies.* ed: Ronald V. Clarke. pp:1-43, Harrow and Heston, 2nd edition.

Clarke R.V. (2003). *Shoplifting.* Problem-Oriented Guides for Police: Problem-Specific Series No.11. Office of Community Oriented Policing Services, U.S. Department of Justice.

Cohen P.R. and Levesque H.J. (1990). *Intention is Choice with Commitment.* Artificial Intelligence. 42. pp: 213-261.

Cormen T.H., Leiserson C.E., Rivest. R.R. (1990). *Introduction to Algorithms.* MIT Press: Cambridge, MA.

Cornish D. and Clarke R. (1986). *The Reasoning Criminal: Rational Choice Perspectives on Offending*. Springer-Verlag.

Cummins, R. (1983). *The Nature of Psychological Explanation*. MIT Press.

Czerlinski J., Gigerenzer G., and Goldstein D.G. (1999). *How Good Are Simple Heuristics?*. . In Simple Heuristics That Make Us Smart, eds: Gerd Gigerenzer and Peter M. Todd. Oxford University Press.

Dabney D.A., Hollinger R.C., and Dugan L. (2004). *Who Actually Steals? A Study of Covertly Observed Shoplifters*. Justice Quarterly. 21(4).

Darken C. (2004). *Visibility and Concealment Algorithms for 3D Simulations.* In Proceedings of the 2004 Conference on Behavior Representation in Modeling and Simulation (BRIMS)

Davies M. and Stone T. (1995). *Folk Psychology: The Theory of Mind Debate.* Blackwell Publishers.

Davies M. and Stone T. (1996). *Folk Psychology as Simulation.* In Mental Simulation: Evaluation and Applications. Blackwell Publishers.

Diller D.E., Ferguson W., Leung A.M., Benyo B., and Foley D. (2004). *Behavior Modeling in Commercial Games.* In Proceedings of the 2004 Conference on Behavior Representation in Modeling and Simulation (BRIMS).

d'Inverno M., Kinny D., Luck M., and Wooldridge M. (1997). *A formal specification of dMars.* In Intelligent Agents IV: Agent Theories, Architectures, and Languages. eds: Munindar P. Singh, Anand Rao, and Michael Wooldridge. pp: 155-176.

Edwards W. (1962). *Subjective probabilities inferred from decisions*. Psychological Review. 69. pp: 109-135.

Endsley M.R. (1997). *The Role of Situational Awareness in Naturalistic Decision Making.* In Naturalistic Decision Making, eds: Zsambok C.E. and Klein G. Lawrence Erlbaum Associates.

Elster J. (1990). *When Rationality Fails.* In The Limits of Rationality, eds: Karen S. Cook and Margaret Levi. The University of Chicago Press.

Farrington. D. P. (1999). *Measuring, explaining and preventing shoplifting: A review of British research*. Security Journal. 12. pp: 9-27

Ferber J. (1999). Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence. Addison-Wesley.

Fischer K., Muller J.P., and Pischel M. (1996). A Pragmatic BDI Architecture. In Intelligent Agents II (LNAI Volume 1037). eds: M. Wooldridge, J. P. Muller, and M. Tambe. pp: 203-218. Springer-Verlag

Garcia M.L. (2001). The Design and Evaluation of Physical Protection Systems. Elsevier: Burlington,MA.

Georgeff M.P. and Lansky A.L. (1987). *Reactive Reasoning and Planning: An Experiment with a Mobile Robot.* In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-1987). pp 677-682.

Georgeff M.P., Pell P., Pollack M., Tambe M., and Wooldridge M. (1999). *The belief-desire-intention model of agency*. In Proceedings of the. 5th International Workshop Intelligent Agents. eds: J. Muller, M. P. Singh, and A. S. Rao. pp. 1-10.

Gigerenzer G. and Todd P.M. (1999). *Fast and Frugal Heuristics: The Adaptive Toolbox.* In Simple Heuristics That Make Us Smart. eds: Gerd Gigerenzer and Peter M. Todd. Oxford University Press.

Gigerenzer G. and Selten R. (2001). *Rethinking Rationality.* In Bounded Rationality. eds: Gerd Gigerenzer and Reinhard Selten. The MIT Press.

Goldstein W.M. and Hogarth R.M. (1997). *Judgment and Decision Research: Some Historical Context.* In Research on Judgment and Decision Making: Currents. Connections, and Controversies, eds: William M. Goldstein and Robin M. Hogarth. Cambridge University Press.

Gordon R.M. (1996). *Folk Psychology as Simulation.* In Mental Simulation: Evaluation and Applications eds: Martin Davies and Tony Stone. Blackwell Publishers.

Hacking I. (1975). *The Emergence of Probability*. Cambridge University Press.

Hayes, R. (1993). *Shoplifting control*. Prevention Press

Helena. R. W. (1996). *Steal me blind: The complete guide to shoplifting and retail theft*. Bluelight Publishing.

Hollinger, R. C., Dabney D.A., Lee G., and Hayes R. (1997). *1997 national retail security survey: Final report*. University of Florida.

Hollinger, R. C. and Langton L. (2004). *2004 national retail security survey: Final report*. University of Florida.

Hurley S. and Nudds M. (2006). *The Questions of Animal Rationality: Theory and Evidence*. In Rational Animals?. eds: Susan Hurley and Mathew Nudds. Oxford University Press.

Jones D.A., Turnquist M.A., Kozick L.K. (2005). *Simulation of Imperfect Information in Vulnerability Modeling for Infrastructure Facilities*. In the Proceedings of the 2005 Winter Simulation Conference, eds: M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines. pp:965-972.

Jordan S.E., M.K. Snell, M.M. Madsen, J.S. Smith, and B.A. Peters. (1998). *Discrete-Event Simulation for the Design and Evaluation of Physical Protection Systems*. In the Proceedings of the 1998 Winter Simulation Conference.

Kacelnik A. (2006). *Meanings of Rationality*. In Rational Animals? eds: Susan Hurley and Mathew Nudds. Oxford University Press.

Kahneman D. and Tversky A. (1982). *The Simulation Heuristic*. In Judgment under Uncertainty: Heuristics and Biases. eds: Daniel Kahneman, Paul Slovic, and Amos Tversky. Cambridge University Press.

Kahneman D. (2002). *Maps of Bounded Rationality: A Perspective on Intuitive Judgment and Choice*. Nobel Prize Lecture.

Klein G. and Crandall B.W. (1995). *The Role of Mental Simulation in Problem Solving and Decision Making.* In Local Applications of the Ecological Approach to Human Machine Systems. eds: P. Hancock, J. Flach, J. Caird, and K. Vincente. pp: 325-358. Lawrence Erlbaum Associates.

Klein G. (1998). *Sources of Power*. The MIT Press.

Klein G. (2001). *The Fiction of Optimization.* In Bounded Rationality. eds: Gerd Gigerenzer and Reinhard Selten. The MIT Press.

Klemke. L. W. (1992). *The sociology of shoplifting: Boosters and snitches today*. Praeger.

Krasnovsky, T. and Lane, R. C. (1998). *Shoplifting: A review of the literature*. Aggression and Violent Behavior. 3. pp: 219-235.

Kuehne R., Wille C., and Dumke R. (2005). *Software Agents Using Simulation for Decision Making.* ACM SIGSOFT Software Engineering Notes. 30(1).

Laird J.E. and van Lent M. (2001). *Human-Level AI's Killer Application: Interactive Computer Games.* AI Magazine. Summer 2001. pp: 15-25

Lambert J.H. and Sarda P. (2005). *Terrorism Scenario Identification by Superposition of Infrastructure Networks.* Journal of Infrastructure Systems. 11(4). pp: 211-221.

Law A.M. and Kelton D.W. (2000). Simulation Modeling and Analysis, 3rd edition. McGraw-Hill: Singapore.

Lewis G.L. (2006). *Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation.* Wiley-Interscience.

Lipshitz R. (1993). *Converging themes in the study of decision making in realistic settings.* In Decision Making in Action: Models and Methods, eds: Gary Klein, Judith Orasanu, Roberta Calderwood, and Caroline E. Zsambok. pp: 103-137. Abbex Publishing Corporation.

Loomes G. and Sugden R. (1982). *Regret Theory: An Alternative Theory of Rational Choice under Uncertainty*. Economic Journal. 92. pp: 805-824.

Lucas S.M. and Kendall G. (2006). *Evolutionary Computation and Games*. IEEE Computational Intelligence Magazine. 1(1). pp: 10-19.

Machina M. (1983). *Generalized Expected Utility Analysis and the Nature of Observed Violations of the Independence Axiom*. In Foundations of Utility and Risk Theory with Applications. eds: B.T. Stigum and F. Wenstop. Reidel.

Marsh W.E. (2007). *An Initial Methodology for the Definition and Implementation of Unmanned Aerial Vehicle Agent Behaviors*. M.Sc. Thesis. Wright State University.

Moore. R. H. (1984). *Shoplifting in Middle America: Patterns and Motivational Correlates*. International Journal of Offender Therapy and Comparative Criminology. 28. pp: 53-64.

Morton A. (1996). *Game Theory and Knowledge by Simulation*. In Mental Simulation: Evaluation and Applications eds: Martin Davies and Tony Stone. Blackwell Publishers.

Muller J.P., Pischel M., and Thiel M. (1995). *Modelling reactive behaviour in vertically layered agent architectures*. In Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890). pp: 261-276. Springer-Verlag

Nelson D., and Perrone S. (2000). *Understanding and Controlling Retail Theft*. Trends & Issues. May (152): 1-6.

Norling E., Sonenberg L., and Rönnquist R. (2000). *Enhancing Multi-Agent Based Simulation with Human-Like Decision Making Strategies*. In Proceedings of Multi-Agent Based Simulation, Second International Workshop, eds: Scott Moss and Paul Davidsson. Boston, MA.

Norling E. (2004). *Folk Psychology for Human Modelling: Extending the BDI Paradigm.* In Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems.

Oden J.T., Belytschko T., Fish J., Hughes T.J.R., Johnson C., and Keyes D. (2006). *Revolutionizing Engineering Science through Simulation.* Report of the National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science.

Petty M.D., McKenzie F.D., Gaskins R.C., and Weisel E. (2004). *Developing a Crowd Federate for Military Simulation.* In the Proceedings of the Spring 2004 Simulation Interoperability Workshop. pp: 484-493.

Pew R.W. and Mavor A.S. (1998). *Modeling Human and Organizational Behavior: Application to Military Simulations.* National Academy Press.

Pollack M.E. (1992). *The Uses of Plans.* Artificial Intelligence. 57(1). pp: 43-68.

Proust J. (2006). *Rationality and Metacognition in Non-Human Animals.* In Rational Animals?. eds: Susan Hurley and Mathew Nudds. Oxford University Press.

Rao A.S. and Georgeff M.P. (1991). *Modeling Rational Agents within a BDI-Architecture.* In Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). pp: 473-484.

Ray, J. (1987). *Every twelfth shopper: Who shoplifts and why*. Social Casework. 68. pp: 234-239.

Russell S. and Norvig P. (2003). *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Savage L.J. (1954). *The Foundation of Statistics.* Wiley.

Schaeffer J., Bulitki V., and Buro M. (2008). *Bots Get Smart: Can video games breathe new life into AI research?*. IEEE Spectrum. 45(12). pp: 48-56.

Schneider J.L. (2003). *Prolific Burglars and the Role of Shoplifting.* Security Journal. 16. pp: 49-59.

Schriber T.J. and Brunner D.T. (1997) *Inside DiscreteEvent Simulation Software" How it Works and Why It Matters.* In Proceedings of the 1997 Winter Simulation Conference.

Scott J. (2000). *Rational Choice Theory.* In Understanding Contemporary Society: Theories of the Present, eds: G. Browning, A. Halcli, and F. Webster. Sage Publications.

Simon H.A. (1956). *Rational Choice and the Structure of the Environment.* Psychological Review. 63. pp: 129-138

Simon H.A. (1957). *Models of man: Social and rational.* Wiley.

Simon H.A. (1978). *Rational Decision-Making in Business Organizations.* Nobel Memorial Lecture.

Slovic P., Finucane M., Peters E. and MacGregor D.G. (2002). *The Affect Heuristic.* In Heuristics and Biases. eds: T. Gilovich, D. Griffin, and D. Kahneman. pp: 397-420. Cambridge University Press.

Smith J.S., Wysk R.A., Sturrock D.T., Ramaswamy S.E., Smith G.D., and Joshi S.B. (1994). *Discrete-Event Simulation for Shop Floor Control.* In Proceedings of the 1994 Winter Simulation Conference.

Smith J.S., Peters B.A., Jordan S.E., and Snell M.K.. (1999) *Distributed Real-Time Simulation for Intruder Detection System Analysis.* In Proceedings of the 1999 Winter Simulation Conference.

Smith J.S. (2003). *Survey on the Use of Simulation for Manufacturing Systems Design and Operation.* Journal of Manufacturing Systems. 22(2). pp: 157-171.

Sokolowski J.A. (2003). *Modeling the Decision Process of a Joint Task Force Commander.* Ph.D. Dissertation. Old Dominion University.

Spronck P.H.M. (2005). *Adaptive Game AI.* Ph.D. Dissertation. Maastricht University

Stanovich K.E. and West R.F. (2000). *Individual Differences in Reasoning: Implications for the Rationality Debate.* Behavioral and Brain Sciences. 23. pp: 645-665.

Stich S. and Nichols S. (1995). *Simulation or Tacit Theory*. In Folk Psychology: The Theory of Mind Debate. eds: Martin Davies and Tony Stone. Blackwell Publishers.

Sycara K.P. (1998). *Multiagent Systems.* AI Magazine. pp: 79-92.

Teller S.J. (1992). *Visibility Computations in Densely Occluded Polyhedral Environments.* PH.D. Dissertation. University of California at Berkeley.

Tickner A.H. and E.C. Poulton. (1973). *Monitoring up to 16 Synthetic Television Pictures Showing a Great Deal of Movement*. Ergonomics 16. pp:381-401.

Togelius J. (2007). *Optimization, Imitation, and Innovation: Computational Intelligence and Games.* Ph.D. Dissertation. University of Essex.

Tonglet. M. (2002). *Consumer misbehaviour: An exploratory study of shoplifting.* Journal of Consumer Behaviour. 1(4). pp: 336-354.

Tversky A. and Kahneman D. (1974). *Judgment under Uncertainty: Heuristics and Biases.* Science. 185(4157). pp: 1124-1131.

Tversky A. and Kahneman D. (1986). *Rational Choice and the Framing of Decisions.* Journal of Business. 59(4).

von Neumann J. and Morgenstern O. (1944). *Theory of Games and Economic Behavior.* Princeton University Press.

Wooldridge M. and Jennings N.R. (1995). *Intelligent agents: Theory and Practice.* The Knowledge Engineering Review. 10(2). pp: 115-152.

Wooldridge M. (1999). *Intelligent Agents.* In Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. ed: Gerhard Weiss. The MIT Press.

Wooldridge M. (2000). *Reasoning about Rational Agents*. The MIT Press.

Wray R.E. and Jones R.M. (2006). *Considering Soar as an Agent Architecture.* In Cognition and Multi-Agent Interaction. ed: Ron Sun. Cambridge University Press.

Wu D.S. and Wysk R.A. (1989). *An Application of Discrete-Event Simulation to On-line Control and Scheduling in Flexible Manufacturing.* International Journal of Production Research. 27. pp: 1603-1623.

Yannakakis G.N. (2005). *AI in Computer Games: Generating Interactive Interesting Opponents by the Use of Evolutionary Computation.* Ph.D. Dissertation. University of Edinburgh.

Yates J.F. (2001). *"Outsider:" Impressions of Naturalistic Decision Making.* In Linking Expertise and Naturalistic Decision Making, eds: Eduardo Salas and Gary Klein. Lawrence Erlbaum Associates.

Zey M. (1992). *Criticisms of Rational Choice Models.* In Decision Making: Alternatives to Rational Choice Models. ed: Mary Zey. Sage Publications.

Zsambok C. and Klein G. (1997). *Naturalistic decision making.* Lawrence Erlbaum Associates.