Iowa State University
Digital Repository

2008

Construction dispute mitigation through multi-agent based simulation and risk management modeling

Islam Hassan El-adaway
Iowa State University

Recommended Citation

Construction dispute mitigation through multi-agent based simulation and risk management modeling

by

Islam Hassan El-adaway

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Civil Engineering (Construction Engineering and Management)

Program of Study Committee:
Amr Kandil, Major Professor
Christopher Williams
Edward Jaselskis
Kelly Strong
Mark Power

Iowa State University

Ames, Iowa

2008

DEDICATIONS

I would like to dedicate my doctorate research to *my parents* for their absolute love and care, to *my fiancée and future wife Dina El-Wassief* for her full understanding and support during the past months, to *Mr. Erminio Granata Managing Director of Nile Aster International* for his genuine and continuous fatherly support, to *Dr. Janet Yates Professor of Construction Engineering and Management at Ohio University* for believing in me and fully standing by my side both personally and professionally, to *Ms. Marla McIntyre Executive Director of the Surety Information Office* for uniquely and genuinely helping me in getting data for this research with all her network connections, to *Dr. Ezzat Fahmy Professor of Construction Engineering and Associate Dean for School of Sciences and Engineering at the American University in Cairo* for his perennial personal and academic advices, to *Dr. Safwan Khedr Professor of Construction Engineering at the American University in Cairo* for foreseeing my possible success in the area of construction dispute resolution; to *Dr. Ali Hadi Professor of Mathematics and Vice Provost at the American University in Cairo* for creatively teaching me principles of probability and statistics; and to *Dr. Samer Ezeldin Professor of Construction Engineering at the American University in Cairo* for being a true source of assistance and guidance since my undergraduate studies and to-date.

For all of you

I hope to be always up to your expectations both personally and professionally

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

ABSTRACT

The construction industry is regarded not only as a backbone of the nation's economy but also as an integral indicator of its efficiency and effectiveness. However, as a result of the risks and complexities that are naturally inherent with construction projects as well as the diverging interests of the parties involved, claims and disputes could be considered an unavoidable consequence of the construction processes. In fact, over the years, the frequency and severity of claims and disputes have significantly increased to the extent that the estimated total annual cost of construction conflicts and disputes in the U.S. is $5 billion. That said, the main goal of this dissertation was to develop an integrated and coherent methodology for mitigation of construction disputes through both, multi agent based simulation concepts and risk management modeling principles. In this regard, the associated work carried out under this research has: (1) developed an innovative method for using logical induction decision support in construction claims and disputes; (2) created a multi agent system for construction dispute resolution (MAS-COR) that will simulate legal discourse in construction disputes; (3) developed a new method for addressing the issue of risks in the construction industry using portfolio insurance; and (4) created an innovative way for mitigating negative effects of contractor's construction claims and disputes using a risk retention approach. It is conjectured that the attainment of the aforementioned objectives, as detailed under this dissertation, would mitigate the negative effects of claims and disputes in the construction industry and thus, have a positive impact on the contracting parties, their projects, the construction industry as a whole, and consequently, the nation's economy.

1.0 INTRODUCTION

1.1 Research Topic

The U.S. construction industry employs 7.5 million full and part time employees and contributes nearly US $1.2 trillion to the U.S. economy making it the largest single production sector (BEA 2007). Moreover, construction works represents approximately 4.6% of the nation's Gross Domestic Product (BEA 2007). In addition, it is estimated that the world construction market has reached US $5.5 trillion at the end of 2007 (Harmon 2003). Thus, the construction industry is regarded not only as a backbone of the nation's economy but also as an integral indicator of its efficiency and effectiveness. This industry is characterized by the close integration of many different trades in construction projects. In fact, this integration has intensified in the past decades with the transformation from simple, straightforward small projects to complex, sophisticated large-scale projects. A single construction project would require interaction between dissimilar, yet contractually integrated parties, including inter alia, owners, designers, contractors, sub-contractors, suppliers, manufacturers, and others. As a result, construction is described as a collaborative teamwork process where parties with different interests, functions, and objectives, share a common goal, which is the successful completion of a project (Harmon 2003). The above described interdependencies and diverging interests in the construction process make construction projects described as complex adaptive systems (Anumba *et al* 2005). That said, risks and uncertainties are an integral part of the construction process.

A contract risk can be defined as the element that would push-off the parties to the contract from the required service or quality standards (Black 2005). Construction risks can be categorized in a number of ways based on the source of risk, impact of risk, or by project phase (Klemetti 2006). In this regard, Finnerty (1996) describes nine types of risk that include supply, technological, completion, economic, financial, currency, political, environmental, and force majeure risks. Another study divided construction risks according to their impact and by where their control lies, to include business, insurable risks, external risks, and internal risks (Turner 1999). Moreover, Miller and Lessard (2001) studied large engineering projects and categorized risks according to their source and categorized risks under market, completion and institutional risks. According to their definition, market risk is

mainly caused by the demand uncertainty, completion risks refer to technical risks during and after the completion of a project, and institutional risks are related to the political uncertainties in a specific situation. Furthermore, Artto *at al* (2003), as quoted in (Klemetti 2006), presents a four-level risk categorization which is divided into pure risks such as hazards and weather conditions; financial risks including cash flow or credit risk; business risks that confines almost anything that can happen in a project; and political risks which are caused mostly by extreme conditions such as wars. Most recently, Brown (2004) indicated that construction risks would include design, construction, site, economic, political, environmental, and human risks. Regardless of the categories, risks remain a fact that negatively affects the construction process. In fact, as a result of the risks and complexities that are naturally inherent with construction projects as well as the diverging interests of the parties involved, claims could be considered an unavoidable consequence of the construction processes (Levin 1998).

The American Institute of Architects, as quoted in Levin (1998), defines a claim as "*a demand or assertion by one of the contracting parties seeking, as a matter of right, adjustment or interpretation of the contract terms, payment of money, extension of time, or other relief with respect to the terms of a contract*". In the last 30 years, the volume of claims in the construction industry has substantially increased. Construction claims mainly arise due to organizational, planning, and contractual problems (Mitropoulos and Howell 2001). Moreover, many claims are a direct result of: (1) the complexity of the projects being undertaken; (2) an inflexible price structure that does not permit the absorption of unanticipated additional costs by the contractor; and (3) the contractual approach taken by most owners whereby once a contract is let and price is determined, all financial risks or exposure should fall on the contractor (Levin 1998).

In this connection, the statistics for the U.S. and Canada alone are staggering where: (1) 50% of claims constituted an additional 30% of the original contract price; (2) 33% of claims amounted to at least 60% of the original contract price; and (3) in some cases the claim values were as high as the original contract price (Cheeks 2003). Also, as noted by Ren *et al* (2001): (1) 52% of all U.K. construction projects will involve a claim of some type; (2) ₤ 1.2

billion could be the subject of construction claims or disputes at any point in time; and (3) 83% of contractors requested at least one extension during the course of their projects.

A construction claim is the main tool that: (1) contractor should use to recuperate the unlawful extra costs incurred during any construction project or (2) owner should utilize to recover the extra costs due to the poor execution of contractor. Management of construction claims is one of the biggest challenges that are facing contractors and owners in today's vacillating business environment (Kalulanga *et al* 2001 and Ren *et al* 2001). The basic procedures for handling claims include identification, notification, documentation, delay analysis, cost analysis, pricing, presentation, and negotiation for amicable settlement (Levin 1998). Accordingly, it is not strange that the estimated total annual cost of construction conflicts and disputes in the U.S. is $5 billion (Peña-Mora *et al* 2002). In all cases, and despite a well-defined process, most claims are not settled within the construction period, transforming instead into conflicts and disputes to be resolved through the wide spectrum of resolution mechanisms.

The goal of any dispute resolution mechanism is to insure that the duties under the contract are fulfilled and to provide compensation for any breaches of those duties (Spurin 2003). Most standard contracts provide either a specific mode of dispute resolution or a range of options for resolution. Dispute settlement mechanisms include traditional and alternative dispute resolution (ADR) techniques.

Traditional dispute settlement mechanisms include litigation and arbitration. Litigation is played out in the court system and therefore can be a time-consuming process. The number of construction disputes that have been in court three or more years doubled between 1984 and 1992 (Treacy 1995). In addition, initial court decisions are not final but subject to review by courts of appeal and cessation if one of the parties is not satisfied with the award. The longer a dispute lasts, the greater the expense for the parties. The failure to resolve problems quickly means that the parties have to allocate both time and money to the resolution process (Harmon 2003). Arbitration is the principal alternative to litigation for the settlement of both domestic and international construction disputes. In many ways the arbitration process resembles litigation. The arbitral tribunal will make determinations of facts, contract, and law, and apply those in order to quantify losses, determine damages, and make awards

(Spurin 2003). Arbitration is more flexible than litigation and can be tailored to a specific case. However, it is still, in most cases, a post-project completion resolution mechanism that in some way or another intensifies the negative effects of the dispute on the parties (Colledge *et al* 2000).

Non-traditional dispute settlement mechanisms, known as alternative dispute resolution (ADR), comprise a wide range of procedures that includes:

- mediation—a non-binding, consensual process of resolving conflicts through settlement conferences expedited by an impartial third party who facilitates negotiations between the disputants (Treacy 1995);

- med/arb— mediation followed, if necessary, by arbitration (Harmon 2003);

- mini-trial—a two-day voluntary procedure during which disputing parties each present their case to a qualified neutral panel that, in turn, renders a non-binding decision and then mediates to settle the dispute (Harmon 2003);

- early neutral evaluation (ENE)—a confidential conference during which the parties to the dispute present the factual and legal basis of their case to an experienced, impartial attorney who issues a non-binding determination (Treacy 1995);

- adjudication—a British method of achieving a quick decision using an inquisitorial approach with limited hearings; the ruling is immediately binding but does not necessarily constitute a final resolution because the parties can subsequently review or appeal or *de novo* the award by arbitration or litigation, usually after completion of the project (Groton *et al* 2001); and

- dispute review boards (DRB)—the U.S. version of adjudication where a board of impartial professionals formed at the beginning of the project to follow construction progress resolve disputes for the duration of the project (Harmon 2003).

ADR techniques have several advantages, including reduced time to disposition, less costly discovery of facts, more effective case management, increased confidentiality, facilitation of direct communication, preservation of ongoing party relations, and provision of qualified neutral experts (Cheung *et al* 2002, Peña-Mora *et al* 2002). However, even the least disruptive ADRs still impact construction projects in negative ways (El-adaway and Ezeldin 2007). That said, the negative effects of claims and disputes have had a serious

negative impact on the contracting parties, their projects, the construction industry as a whole, and consequently, the nation's economy.

1.2 Problem Statement

In light of the above, industry professionals and researchers thought of ways to mitigate the negative effects of construction disputes. Those can be highlighted as follows:

1) In a construction dispute, attorneys and judges use precedent cases to reason analogically. Rule predicates are simply not sufficiently well-defined for the inference of correct decisions deductively (Ashley and Rissland 1988). Legal experts make competing arguments, pitting conflicting interpretations of cases and facts against each other (Ashley and Rissland 1988). Thus, recent jurisprudential theory has focused on law as a process of argument between two parties (Ozorhon *et al* 2006). In construction claims preparation and dispute resolution, a vast amount of information needs to be gathered, analyzed, and presented by lawyers. Accordingly, various AI models were developed in order to support and aid in saving time and cost during construction claims preparation and dispute resolution. All of these effective systems models use various rule-based reasoning techniques such as game theories, Bayesian-Nash equilibrium, and behavioral-orientated approaches (Peña-Mora and Wang 1998 and Ren *et al* 2001). *Nevertheless, none of the above mentioned projects has attempted to utilize the powerful abilities of logical induction as a decision support for construction disputes.*

2) Multi agent systems (MAS) heavily impacted various construction-related applications such as material management, engineering design, supply chain management, scheduling and control, and negotiations in dispute resolution. Those studies include agent-based integrated building design environment (Fenves *et al* 1994), agent-based collaborative design for construction (Heckel 1996), agent-based design changes system (Chiou and Logcher 1996), agent-based resource management system (Oliviera 1997), CONVINCER mediating agent-based model (Peña-Mora and Wang 1998), agent based tendering system (Denk and Schnellenbach-Held 2002), MASCOT negotiating agent-based model (Ren *et al* 2003), and model for situational simulations in construction industry (Rojas and Mukherjee 2006). Of the aforementioned construction MAS tools, CONVINCER and MASCOT are of significant importance to the topic of this dissertation as they deal with

the issue of resolution of construction disputes through negotiation. CONVINCER is a collaborative negotiation methodology and computer agent that is used to facilitate the negotiation of conflicts in the construction industry. This simulation model is primarily based on game and negotiation theories realizing that the construction industry is naturally a non-zero sum game where all participants can share profits (Peña-Mora and Wang 1998). On the other hand, MASCOT is multi-agent system for construction claim negotiations where autonomous agents act on behalf of a project participant and can directly negotiate with each other to resolve construction claims. MASCOT deals with the comprehensive view of the interactive negotiation process where the attained settlement is based on Bayesian-Nash equilibrium and behaviour-oriented theories (Ren *et al* 2003). The above-mentioned models have concentrated on conducting effective negotiations between the conflicting parties. *Nevertheless, none of these models has actually attempted to develop a multi agent system that will reason the claims of the conflicting parties in light of the previous similar cases, and thereby simulate legal discourse.*

3) Construction researchers and practitioners thought of ways to allocate risks and thus, reduce negative effect of construction risks through development of organizational strategies for projects. Organizational strategies for any construction project would decide upon the optimal combination between the appropriate procurement method as well as the suitable contract type. Despite all these efforts, risk continued to negatively affect the contracting parties, their projects, and the construction industry as a whole. Thus, many research projects have been initiated to focus on the issues of risk in construction industry. These studies would detain studies pertaining to: pricing construction risk using fuzzy set application (Paek *et al* 1993), predicting contractor failure using stochastic dynamics of economic and financial variables (Russell and Zhai 1996), assessing corporate risk using historical cost control data (Minato and Ashley 1998), studying contingency and assumption of risk in small to medium contractors (Smith and Bohn 1999), developing integrated methodology for project risk management (Del Cano and Cruz 2002), evaluating investments in emerging A/E/C technologies under uncertainty (Ho and Liu 2003), evaluating and investing in construction projects under

uncertainty (A.M and Liu 2003), quantifying risks in construction works (Janadi and Al Mishari 2003), implementing relational contracting and joint risk management (Rahman *et al* 2004), using of owner-controlled insurance programs by transportation agencies (Schexnayder *et al* 2004), risk management on large capital projects (Turnbaugh 2005), and partnering mechanism in construction (Tang *et al* 2006). *Nevertheless, none of these research projects has explored: (1) the possibility of insuring against construction risks that are not only beyond the control of contractors, not covered under surety policies, and which result in construction conflicts, claims, and disputes; (2) whether it is better to insure against single risks or bundle of risks; and (3) the concept of risk retention groups towards the mitigation of negative effects of contractors construction claims.*

In light of the aforementioned, this research has been initiated in order to fill the above mentioned research gaps in relation to the utilization of multi-agent based simulation concepts and risk management modeling techniques towards construction dispute mitigation.

1.3 Goals and Objectives

In general, the main goal of this dissertation is to develop an integrated and coherent methodology for mitigation of construction disputes through both, multi-agent based simulation concepts and risk management modeling techniques. In the attempt for accomplishing the same, the said research would have four (4) specific objectives detailed hereunder:

1) Develop an innovative method for decision support in construction claims and disputes using a logical induction approach. *In this regard, the following research questions would be answered:* how can construction cases be analyzed as a set of factors? How can relevance be drawn between those factors and disputing parties? How are rules generated out of such factors? How are factors analyzed towards categorization of construction cases? How to build interrelated logical arguments that put hands on similarities, differences, strengths, and weaknesses between current and precedent construction disputes?

2) Create a multi agent system for construction dispute resolution (MAS-COR) that will simulate legal discourse in construction disputes. *In this regard, the following research questions would be answered:* What is the best way to decompose and model dispute

resolution? What is the optimum breakdown and allocation of subtasks to agents? How will partial results be synthesized? How will the distributed perceptual information be handled? How will agents be enabled to maintain consistent sharing of their environment? How will decentralized control be implemented and how will efficient coordination mechanisms among agents are built? What is the best way to design efficient multi-agent planning and learning algorithms? How will knowledge be represented? How will agents be enabled to reason about the actions, plans, and knowledge of other agents? What are the best communication languages and protocols? What, when, and with whom should an agent communicate? How will agents be enabled to form organizational structures like teams or coalitions? How will roles be assigned to agents? How will coherent and stable system behavior be insured? How will the proposed system be coded? What is the optimum MAS supporting platform to use in the process? What is the best way to test, debug and validate the proposed system?

3) Develop a new method for addressing the issue of risks in the construction industry using principle of risk management using bundled insurance. *In this regard, the following research questions would be answered:* Is there a possibility for insuring against construction risks that are beyond the control of contractors, not covered by surety risk transfer policies, and negatively affect the financial and economic standing the contractors resulting in construction conflicts, claims, and disputes? What are bootstrapping techniques? Can we price insurance using the real options theory? What are the principles of Monte Carlo simulation in insurance pricing? Is it better to insure construction risks on single or portfolio basis? Can this study open horizons for new kind of insurance markets that contribute to the effectiveness and efficiency of risk management process in the construction industry?

4) Create an innovative way for mitigating negative effects of contractor's construction claims and disputes using a risk retention approach. *In this regard, the following research questions would be answered:* What are risk retention groups? How are they utilized in the construction industry? Can we insure against construction claims through a risk retention group? How can we design a risk retention group for construction claims? What are the performance indicators for such risk retention group? How can we modify or alter

the created risk retention group? What are the benefits of such a risk retention group? Would this study put basis for a leading risk management technique that could be extended for the benefit of relieving the negative consequences associated with lengthy claim and dispute resolution in the construction industry?

The aforementioned four (4) objectives would contribute towards more effective and efficient employment of distributed artificial intelligent concepts and risk management techniques towards mitigation of negative effect of claims and disputes in the construction industry.

1.4 Organization

This dissertation is organized as follows:

1) Chapter 2, titled "Literature Review", would provide an extensive backgrounds to issues interrelated with topics of construction disputes, multi-agent based simulation, and risk management modeling.

2) Chapter 3, titled "Logical Induction in Construction Claims and Disputes", would highlight backgrounds related to rule based reasoning, arguments and law, case based reasoning, adversarial case based reasoning, and logical induction. Moreover, it would detail the methodology utilized for using logical induction for decision support that will simulate legal discourse in construction disputes. Lastly, it presents a detailed worked example where the developed approach was utilized to generate argument pertaining to 5 previously arbitrated construction cases.

3) Chapter 4, titled "Multi Agent System for Construction Dispute Resolution (MAS-COR)", would emphasize backgrounds in pertaining to distributed artificial intelligence, agents and multi agent systems, and multi agent systems in the construction industry. Furthermore, it would outline the methodology utilized for developing MAS-COR through building of agent-based role model, creation of communication protocol, implementation, and testing/validation of the developed system using various test cases.

4) Chapter 5, titled "Construction Risks Insurance", would highlight backgrounds related to risk management basics and bootstrapping techniques. Moreover, it would detail the methodology employed for insuring against construction risks that are beyond the control of contractors, not covered by surety risk transfer policies, and negatively affect the

financial and economic standing of contractors. Lastly, it shows a worked example that tests and verifies the developed methodology.

5) Chapter 6, titled "Risk Retention Group for Contractors Claims", would emphasize backgrounds related to risk retention groups and their application in the construction industry. Furthermore, it would outline the methodology for mitigating negative effects of contractor's construction claims and disputes using a risk retention group that will function according to insurance principles and will confine set of contractors who in return for a certain paid premium would be indemnified against the risk of any legitimate construction claim. Lastly, it shows the prosperous results of applying the newly developed concept on a set of 10,193 construction projects spanning over 12 different California districts.

6) Chapter 7, titled "Conclusions and Future Work", would draw conclusions in relation to the developed integrated and coherent methodology for mitigation of construction disputes through both, distributed artificial intelligent concepts and risk management techniques. Moreover, it will highlight the future prospects for interrelated research works.

2.0 LITERATURE REVIEW

2.1 Construction Disputes

2.1.1 Changes in the Construction Industry

In the construction industry, changes after the work starts are not unusual, and in some types of underground construction, they are the standard. The owner usually needs to have the flexibility under the contract to accommodate unanticipated conditions affecting the project such as changes to technology, errors in the plans and specifications, and the like (Harmon 2003). In this regard, a change order is a written direction to the contractor, signed by the owner or its authorized representative, issued during contract execution, authorizing a change to the original scope of work, which might entail an increase or decrease in the contract time or cost (Groton *et al* 2001). Normally, these official changes do not lead to claims or disputes. Nevertheless, the main threat to the construction process stems from directed or forced changes. Directed change orders are those that the owner does not believe changes the work beyond the contemplation of the parties prior to the time the contract was signed, while the contractor believes that current conditions warrant a change to the contract time and/or money and is unwilling to proceed without an agreement to the increased contract duration and/or costs (Harmon 2003). In this instance, while the owner directs the contractor to perform the work in question, the disputed work will then be claimed under the contract's claim resolution process.

2.1.2 Claims in the Construction Industry

In today's construction industry, claims are simply unavoidable because they represent the administrative process required to handle the results and implications of design changes, defective specification, quantity variations, delays, disruptions, and accelerations (Levin 1998). Construction claims are caused by size and duration of the project, complexity of contract documents, poor communication, limited resources, financial issues, inadequate design, labor issues, and force majeure events (Harmon 2003). Moreover, many claims are a direct result of: (1) the complexity of the projects being undertaken; (2) an inflexible price structure that does not permit the absorption of unanticipated additional costs by the contractor; and (3) the contractual approach taken by most owners whereby once a contract is let and price is determined, all financial risks or exposure should fall on the contractor (Levin

1998). On the other hand, Fenn (2002) asserts that all reasons listed in literature for claims occurrence are mere predictor variables lacking explicit cause and effect relationships.

2.1.3 Contractual Determination of Construction Claims

Construction contracts often provide an expert determinator, which is a resident engineer or architect, to certify questions of fact governing construction claims related to design changes, defective specification, quantity variations, delays, disruptions, and accelerations. In fact, under the Institution of Civil Engineers (ICE) and the Federation Internationale Des Ingenieurs-Conseils (FIDIC), there is the resident engineer that is an impartial employee of the employer who administers significant aspects of the project that comprise certifying work, extensions of time, and other contractual aspects. In other contracts, such as the Joint Council of Tribunals (JCT) and American Institute of Architects (AIA) range of contracts, the same certification role is carried out by architects.

Regardless of contract type, the appointed expert will issue interim decision pertaining to claims and if one of the parties is not satisfied with the same, he/she can contest such a decision under a wide range of dispute resolution mechanisms.

2.1.4 Construction Dispute Resolution Mechanisms

When claims are not settled within the construction period, they are transformed into conflicts and disputes to be resolved through the wide spectrum of resolution mechanisms. Dispute resolution mechanisms would comprise traditional and alternative ways (ADRs). Traditional procedures would normally embrace litigation and arbitration. On the other hand, ADRs would include mediation, med/arb, mini trials, early neutral evaluation, adjudication, and dispute review boards (DRBs). The ideal dispute resolution process is the one that minimizes aggravation and can potentially assist in producing an enforceable settlement, quickly at minimum cost and without disruption for the parties to the project (El-adaway and Ezeldin 2007). Despite the availability of wide-ranging dispute resolution techniques, the construction industry still loses time, cost, and effort in resolving its disputes (Mitropoulos and Howell 2001, Ren *et al* 2001, Cheeks 2003).

Litigation

The courts provide the principal mechanism for settling construction disputes. The principal disadvantage of using the courts lies in the fact that it may take a long time to get a

court hearing. Treacy (1995) stated that number of construction disputes in court, that were more than 3 years old, has doubled between 1984 and 1992. Moreover, even these long-time taking awards, are not final but subject to review by courts of appeal and cessation if one of the parties is not satisfied with the award. The longer a dispute drags on, the greater the expense for the parties. The failure to resolve problems quickly means that the parties have to allocate both time and money to the resolution process.

Arbitration

Arbitration is the principal alternative to litigation for the settlement of both domestic and international construction disputes. In many ways the arbitration process resembles litigation. The arbitral tribunal acts as a private as opposed to a state appointed judge. In addition, and as outlined by Spurin (2003), the arbitral tribunal will make determinations of both facts of contract and law, and apply these in order to issue an award. This award must bear legal and contractual responsibility for losses arising out of a breach of duty as governed by the terms of contract. Moreover, having apportioned liability, arbitrator will quantify the loss, determine damages, and award costs. Colledge *et al* (2000) outline that the main advantage of arbitration is that it offers a dispute resolution system that can be tailored to a particular dispute to an extent which litigation cannot accommodate. Moreover, Spurin (2003) indicates the express advantages of arbitration over litigation comprise: (1) speed to get to the process and often quicker proceedings; (2) the cost of arbitration is often less than the cost of litigation; (3) less formal than the courts in the sense that the parties often have control over the process, which is not prescribed by rules of court; (4) choice of venue and potentially more convenient to the parties and witnesses; (6) specialist arbitrators with industry experience and knowledge; (7) international awards are globally enforceable by virtue of the New York convention on the enforcement of arbitral awards; and (8) more amenable than courts to choices of law and jurisdiction. Despite all of the above mentioned advantages, Spurin (2003) indicated that the most crucial disadvantage of arbitration is that it is still in most cases a post project completion resolution mechanism. Thus, the duration taken by the arbitral tribunal to issue its award, is in some way or another intensifying the negative effects of the dispute on the parties.

Mediation

Mediation is a non-binding, consensual process of resolving conflicts through settlement conferences expedited by an impartial third party who facilitates negotiations between the disputants. In terms of the process, mediation is a structured negotiation where the mediator will establish ground rules and acts as a referee who facilitates communications between the parties. Treacy (1995) highlighted that the mediator, unlike a judge or arbitrator, has no power to impose a solution on the parties. The mediator's sole function is to help disputants resolve their problems as he would identify and narrow the issues focusing on each side's underlying interests, convey messages between the parties, explore areas of agreement, and point out the consequences of not settling. Compared to traditional methods, mediation has proved to be a faster, less expensive, more confidential, and more satisfactory way to resolve disputes. Moreover, recognizing the value and importance of mediation in resolving construction disputes in the US, the American Institute of Architects has added an interim mediation step, prior to binding arbitration, in the disputes clauses of its widely used General Conditions of the Construction Contract A-201 (Harmon 2003).

Nevertheless, despite mediation's advantages, it is still in most cases a post project completion resolution mechanism that does only provide a non binding recommendation. Thus, it can be used by some parties as a way to lengthen the dispute duration before resorting to any other binding mechanism.

Mediation/Arbitration (Med/Arb)

Med/Arb is an amalgam of mediation and arbitration that was developed by Associated Soil and Foundation Engineers (ASFE) in the 1970s (Harmon 2003). It uses mediation first, and disputes that are not resolved via mediation are then arbitrated. Thus, some would call it a binding mediation. The process would involve the selection of a neutral mediator by the parties at the start of construction to make unbinding decisions under mediation pertaining to arising disputes, and if parties are not satisfied, he would make binding decisions under the arbitration. However, the main danger in this process is that it combines mediation that is a conciliatory process, with arbitration that is an adversarial process where it is unlikely that the appointed neutral can remain unbiased during the arbitration proceeding after the failure of mediation (Treacy 1995). Thus, majority of the industry professional call that a mediator

for an unsettled mediation should under no circumstances, whatsoever, serve as an arbitrator for the same dispute (Harmon 2003).

Mini Trials

Mini trial is a voluntary, confidential, non-binding settlement procedure in which attorneys, from each disputant side, present their best-case position in summary fashion to the opposing party, its attorneys, and a qualified neutral or to a panel of top management representatives who are not involved in the dispute (Harmon 2003). The hearing process usually takes between 1 and 2 days. Afterwards, the neutral party may, at the parties' request, render a non binding decision and then mediate to settle the dispute. Being the case, a mini trial aims to mimic or anticipate the outcome of litigation or arbitration. This is believed to enable the parties to come to a business decision to resolve their dispute before resorting to arbitration or litigation. To date, use of mini trials in resolution of construction disputes is not widespread (Harmon 2003).

Early Neutral Evaluation (ENE)

Early neutral evaluation (ENE) is a system, started in 1990, that offers a confidential non binding conference where the parties and their counsels present the factual and legal basis of their case to one another and to an experienced impartial expert with deep knowledge in the subject matter of the case (Treacy 1995). That said, ENE provides a cost-effective mechanism to determine the relative merits of the case while maintaining confidentiality. Nevertheless, and as quoted in Harmon (2003), surveys shows that ENE was not voluntarily chosen by disputant parties but was forced by courts or arbitral tribunals. Thus, it is not surprising to note that out of 94 US state courts; only 14 courts make use of the ENE process.

Adjudication

Adjudication is a UK based dispute resolution mechanism that has started in the 70s of the previous century. Groton *et al* (2001) defines adjudication is being the method of achieving a quick decision, using an inquisitorial approach with limited hearings, which is immediately binding upon the parties but is not a final resolution of the dispute because the parties can subsequently review or appeal or *de novo* the said award by arbitration or litigation, usually after completion of the project. In 1996, the UK Parliament enacted The Housing Grant, Construction and Regeneration Act that mandated the use of statutory

adjudication on all commercial construction projects in England, Scotland and Wales. Nevertheless, and despite these express advantages, it is worth noting that adjudication still comprises two main disadvantages. One disadvantage, pointed out by Colledge *et al* (2000), is that pursuant to the Act of 1996, the claimant can prepare for his case in numerous months, issue a notice to the adjudicator, and within a period of only 7 days both the claimant and respondent should submit their substantiating documents. This status entails express unfairness towards the respondent. Another disadvantage, highlighted by Colledge *et al* (2000), is that adjudication is carried out by one person that is only appointed by invitation and not at the commencement of the construction process. This would highlight the absence of the exchange of opinions as well as the lack of the ongoing familiarity with the activities of the project and the associated parties.

Dispute Review Boards (DRBs)

Vorster (1993) defines a dispute review boards (DRB) as: "*A small group of independent, knowledgeable and respected individuals selected by the owner and contractor and appointed under the contract to review and make recommendations on disputes that arise on a project. The board is appointed at the onset of a project. The board visits the project regularly and has a long-term perspective on any issue it is asked to address. The recommendation it makes is the first step in the dispute resolution process. The recommendations are not binding and they do not preclude either party to the contract from exercising any of the other mechanisms for resolving claims*". Thus, DRBs allow claims to be settled on a weekly, biweekly, or monthly basis during construction rather than at the end of construction. Meanwhile, DRB does not supplant existing dispute settlement methods, DRB could be used earlier to augment the dispute resolution process and it is a non-binding, intermediate step directed at avoiding the need to resort to other more expensive, time-consuming, and less satisfactory procedures (Matyas *et al 1995*). There is a Construction Dispute Review Board Manual that may be used as a guide to setting up the mechanisms needed to implement DRBs. The manual includes a checklist that may be used to assess compliance with the methodology developed by the American Society of Civil Engineers. The manual also includes information on sample contact specifications, DRB practices, case histories, and a table that describes projects that have used DRBs.

In light of the above, ADR techniques have several advantages, including reduced time to disposition, less costly discovery of facts, more effective case management, increased confidentiality, facilitation of direct communication, preservation of ongoing party relations, and provision of qualified neutral experts (Cheung *et al* 2002, Peña-Mora *et al* 2002). Nevertheless, even the least disruptive ADRs, still impact construction projects in negative ways (El-adaway and Ezeldin 2007). That said, it is worth to investigate how multi-agent based simulation concepts could be used as supportive tools in mitigation of construction disputes.

2.2 Multi-Agent Based Simulation
2.2.1 Distributed Artificial Intelligence

Distributed artificial intelligence (DAI) is concerned with systems that consist of various interacting agents that are usually able to perceive their environment and also act so as to change that environment according to some kind of set goals (Taylor 2003). In DAI, social interaction is the mechanism of coordinating these various components and their activities and achieving a useful outcome on the system level. Pursuant to Edmonds (1998), this stands in contrast to earlier classical research in Artificial Intelligence (AI) that did not account for the social aspect and where intelligence is attributed to the abilities and outcomes of the single unified system. Moreover, in DAI, the individual components are viewed in the context of their relationship to the other parts of the system, and intelligence is best viewed in terms of the macro-level behavior of the system. Accordingly, and as mentioned by Moss (2001), functionality is highly interrelated with how coordination of the separate components is achieved. In this connection, it is crucial to understand how models are designed under DAI systems.

In this regard, Taylor (2003) indicates that DAI models are characterized by a 'bottom-up' system design. In such systems, component rules of interaction and behavior at lower levels are specified firstly, and then higher level or aggregate layers are built upon the lower ones. In 1978, Thomas Schelling was among the first people to use this design concept in his well known demo, which serves as the basis of many of the simulation models in today's world. In addition, Moss (2001) states that while there is no central control in the 'bottom-

up' approach, control of the system is intended to emerge from the specification of interaction processes amongst the component parts. As a result of the influence of these interaction mechanisms, individual components will tend to behave in a regulated way. According to Taylor (2003), the systems studied within DAI are typically very complex in the sense that the causal relationships and the behaviors that underlie them cannot be identified merely by inspecting the macro behavior of the system.

DAI Modeling and Simulation

Researchers develop models in order to understand or illustrate some aspect of the target system of interest. Gilbert and Troitzsch (1999) define a model as follows: "*A model is a simplification – smaller, less detailed, less complex, or all of these together – of some other structure or system*". Accordingly, models should be less complex than the real system, and lead to some improved understanding of how the real system functions or might function. Taylor (2003) states that developing a model corresponds to a process of abstraction from the target system, a process which must be made as transparent as possible so that the nature of the mapping between the system and its model is clearly understood.

Over the years, a large number of different approaches to modeling have been developed, assessed, and tailored. In this regard, research carried out in the area of modeling often involved mathematical approach where relationships amongst dependent variables are described by sets of non-linear differential equations (Edmunds 1998). Difficulties associated with mathematical modeling are that it mainly requires a high level of ability in mathematics to manipulate the equations.

That said, Taylor (2003) asserts that the most current used technique for modeling is computer simulation. In this connection, computers allow fast numerical solutions to be generated for equations of models that contain many parameters. According to Moss (2001), computer simulation enjoys set of advantages over mathematical modeling that include: (1) computer program code is more expressive than systems of equations; (2) programs are better suited to modeling parallel processes; (3) programs are (or can be) designed modularly and hence are easily modified; and (4) programs have flexibility to model heterogeneity amongst social actors.

Accordingly, as quoted in Taylor (2003), computer-based approaches to simulation models can help researchers achieve a variety of different objectives that confine: (1) obtaining a better understanding of social processes, especially dynamic ones; (2) making predictions about the occurrence of certain social events; (3) simulating human abilities by modeling knowledge with expert systems; and (4) developing theories through series of theory formalization and simulation.

2.2.2 Multi Agent Systems (MAS)

Multi-agent systems (MAS) are a fast developing information technology where a number of intelligent systems representing real world entities co-operate or compete to reach the desired objectives of their owners. Anumba *et al* (2005) asserts that the increasing interest in MAS is because of its ability to provide robustness and efficiency to allow inter-operation of existing legacy systems and solve problems in which data, expertise, or control are distributed.

The term agent was first proposed at MIT in the 1950s. As described by Anumba *et al* (2005), an agent is usually understood as a system or device acting on behalf of another one. Moreover, agents are expected to operate continuously over relatively long periods and to behave at least in a semi-autonomous ways. Also, agents do not wait for instructions; they initiate actions and guide the interactions. Being the case, Pena-Mora and Wang (1998) stated that agents perform different actions like querying, asking, and sensing to find the information they need as well as undertaking actions to change the environment or communicate with other agents. According to Edmonds (1998), a multi agent system (MAS) can be defined as a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver. These problem solvers, often called agents, are autonomous and can be heterogeneous in nature.

Rojas and Mukherjee (2006) highlight that the general goal of MAS is to create systems that interconnect separately developed agents and thus enabling the ensemble to function beyond the abilities of any singular agent in the set-up. Accordingly, and pursuant to Anumba *et al* (2005), if a problem domain is particularly complex, large, or dynamic (i.e. such as the construction industry) then the only way it can be reasonably addressed is to

develop of agents that specialize in solving a certain problem. This decomposition allows each agent to use its best knowledge for solving such a problem. Thus, when independent problems arise, the agents need to co-ordinate and collaborate with one another to ensure the interdependencies are properly managed from all perspectives.

The characteristics of multi-agent systems are that: (1) each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint; (2) there is no system global control; (3) data are decentralized; and (4) computation is asynchronous. In a more comprehensive overview, and pursuant to Moss (2001), it is perceived that multi-agent systems can be utilized to: (1) solve problems that are too large for a centralized agent to solve; (2) allow for the interconnection and interoperation of multiple existing legacy systems; (3) present solutions to problems that can naturally be regarded as a society of autonomous interacting components-agents; (4) provide solutions that efficiently use information sources that are spatially distributed; (5) come up with solutions in situations where expertise is distributed; (6) enhance performance along the dimensions of computational efficiency, reliability, robustness, maintainability, responsiveness, flexibility, and reuse.

According to Taylor (2003), there are two venues in all researches confining multi-agent systems. While, the first technique aims to control the scope of the interactions between agents in order to create systems that have predictable and designed behavior; the second strand places no such restriction on the design of interaction mechanisms. In this connection, Edmonds (1998) asserts that while the former method is mainly used in engineering applications, the former one is employed in social works.

In general, the rules governing the agents' interaction can become quite complex. Nevertheless, engineers usually spend lots of time and efforts in order to correctly set the same. This is a direct result of the fact that in engineering applications, simulators and designers would like to work in a controlled environment where the possibility of any unpredictable emergent phenomena is nearly eliminated. In relation to this matter, Edmonds (1998) and Moss (2001) argue that engineering simulations must be able to operate predictably and efficiently in situations where environmental factors are circumscribed and are well known in advance, and system behavior is bounded.

Making decisions and being autonomous, in a way or another, requires from agents to understand the situation. Therefore, every agent should possess some ability to learn, adapt, or change the behavior. Anumba *et al* (2005) stated that depending on the application, number of agents, environment, function, and other factors, agents of different complexity exist. In this connection, it is not strange that some agents are simple and work as components of large systems and others are complex trying to come up with intelligent behavior using advanced methods of learning and reasoning.

In light of the above, and as asserted by Pena-Mora and Wang (1998) as well as Anumba *et al* (2005), coordination is central to agent based systems to ensure a community of agents that are acting in a coherent manner. The subject of agent coordination has been studied by many researchers in fields of organizational theory, economics, and anthropology. Typical ways for coordinating between agents would include: (1) organizational structure; (2) contract net protocols (CNP); (3) multi-agent planning; and (4) peer-to-peer negotiation. Talking about negotiation and as stated by Anumba *et al* (2007), it should be highlighted that negotiation is a main method for interaction that enables group of agents to arrive at a mutual agreement regarding belief, goal, or plan. There are various negotiation models that would confine: (1) contract-based negotiation; (2) plan-based negotiation; (3) market-based negotiation; (4) game-theory based negotiation; (5) AI-based negotiation such case based reasoning and k-b approach; and (6) psychology-based negotiation.

In the same regard, and when working with MAS, it is hard to predict or foresee all potential situations an agent would encounter in such open, complex, and dynamic environments. Accordingly, there has been considerable research in the area of learning for MAS. Tesfatsion (2007) has highlighted that the major elements for agent learning are expectations, feedback and evaluation criteria. In this connection, and as highlighted by Anumba *et al* (2005) and Tesfatsion (2007), the most famous methods for learning would compromise: (1) reactive-reinforcement learning whether deterministic or stochastic; (2) belief-based learning; (3) anticipatory learning such as Q-learning; (4) evolutionary learning such as genetic algorithms; and (5) connectionist learning such as neural networks.

Being the case, in the past years, MAS has been one of the most popular scientific approaches for studying real life applications in economics, sociology, biology, engineering,

and other various disciplines. It is expected that such dispersed employment of MAS would be more intensified in the coming years.

Agent-Based Modeling

Agent-based modeling aims to employ agents as the core component units that compose the model. In this sense, and according to Moss (2001), agents can be thought of as intelligent and autonomous programs that interact with other components of the system and their environment in order to affect a certain set of programmed goals. In this regard, Edmonds (1998) asserts that more sophisticated kinds of agents have the facility to construct their own internal representations of the agent-world, to form expectations about events, and to exhibit limited kinds of learning behavior. Therefore, agent-based modeling is clearly distinguished from other kinds of modeling research by this focus on the concept of agents.

It is perceived that agent-based modeling is a relatively new research field. As such, there is a lack of established theory and research methodology underpinning the design of models, standards for programming platforms, verification and validation of models, techniques for comparison of models and for establishing the generality of models. However, and pursuant to Taylor (2003), agent-based modeling is now rapidly gaining attention in many different areas due to its interdisciplinary appeal. This appeal can be attributed to the flexibility in the construction of agent architectures and the fact that some quite complex and interesting model behavior can be generated with this framework once the basic concepts have been understood.

In a construction dispute, the modeler is faced with various interrelated social factors. According to Taylor (2003), it is usually difficult to model system that confines social factors. In fact, the complexity of social systems has been one of the greatest challenges for different modeling approaches as the representation of social actors and their relationships is very problematic because social factors are heterogeneous, operate in very large systems, are not rational, and are subject to social influence (Moss 2001). This complexity problem has greatly limited the success of efforts aimed at modeling systems of the sort typically found in the real world. Accordingly, Gilbert and Troitzsch (1999) assert that there is much use of 'toy models' in multi-agent simulations. Toy models are simple systems which bear little resemblance to reality because they are based on assumptions derived from more general

theory. This type of model may therefore be classified as exploratory and focused on theory building. However, Taylor (2003) stresses that such situation can become misleading where people have built toy models and tried to generalize them to observed systems, and have done so by distorting the abstraction process. One has to be very careful not to infer from toy models things about the real world without substantiated reasoning. In other words, simplifications must not undermine any arguments that are being made about their level of realism.

Edmonds (1998) states that the choice of using agents as a core part of model specification extends upon the role that objects play in many of today's computer applications through object-oriented programming. In fact, and pursuant to Taylor (2003), object oriented programming has become prevalent in many areas of computer science because its properties of encapsulation, i.e. programming rules that manipulate that data are treated as a unit, easy modification through public/private levels of access to data inside classes, inheritance, and others. The same can be said of agent-based programming. In this connection, it is worth mentioning that currently there are various robust agent-based tool kits that include NetLogo, Repast, Ascape, Swarm, Mason, and others. The said toolkits are widely used to graphically represent interaction of agents in variety of applications where agents do series of interconnected activities.

As previously mentioned, the aim of agent-based simulation and modeling is to improve scientific understanding and explanations of aspects of complex systems that are composed of many interacting components. Accordingly, it is of significant importance to understand the methodological strands confining the same. As quoted in Taylor (2003), the said methodological process is more or less the same as any scientific research approach and can be broken down into several steps that include: (1) abstraction of the studied system to a conceptual model; (2) formalization of the said abstraction using computer modeling; (3) model execution and behavior exploration; (4) integration and analysis of results; and (5) carrying out substantiated conclusions on the original target system.

As a basic premise for the modeling process, Edmonds (1998) asserts that that it its more appropriate to model an agent as part of the total system of agents as opposed to modeling a single agent that is interacting with an essentially unitary environment. Consequently, this

implies that agent cognition and behavior can be understood only in the context of interactions with other agents. In this framework, and pursuant to Moss (2001), developed multi-agent systems will always incorporate some specifications of social interaction such as imitation, communication, persuasion, trading, bargaining, and others as an integral part of the model design.

Nevertheless, and as quoted in Taylor (2003), such modeling should be carried out bearing in mind some limitations. The first is bounded rationality which expresses the idea that social agents have limited computational ability in terms of processing information to inform decisions as to how they should rationally act in situations. The second is that agents have imperfect information about the environment in which they operate, often being only familiar with events that are locally significant. For agent research, this implies that systems should be designed that do not attribute unrealistic levels of ability to agents. Systems that are designed in such a way will not only be based upon questionable assumptions but will also be poor in terms of system speed and flexibility pertaining to operations.

According to Moss (2001), there is always a trade-off on a more practical level in the design of agent-based models that is made between agent functionality and system scale. Taylor (2003) highlights that functionality of an agent is the scope of the behavior available to it in forming internal models representing aspects of the system, and upon various actions undertaken that might affect that system. On the other hand, the scale is measured in terms of the number of agents active in the model. Being the case, some models are composed of simple agents that are completely specified in a small number of behavioral rules, whereas some contain agents that have vast numbers of such rules and procedures and involve much information processing. Accordingly, Moss (2001) states that the trade-off exists between the complexity of the agents, the scale of the system, and the computational resources required for simulation. Nevertheless, Taylor (2003) states that as computers become more powerful, the main barrier remaining will be that of system complexity, which is exponential in the number of agents composing the system and the number of inputs to those agents.

Testing and Validation

Usually, model design and experimental scenarios are well defined in the planning stages before the beginning of model implementation. However, and according to Taylor (2003), it

is quite possible that further questions may arise during implementation that demand clarification of issues relating to the model design or that analysis of simulation results may provoke the design of further experiments. On the other hand, researchers might simply be interested in some aspect of model behavior that is poorly understood. This latter is what we call exploration with the model and is a distinct activity from experimentation because it does not aim to test some predefined hypothesis.

Edmonds (1998) asserts that experiments can range in sophistication from the variation of one key parameter along one dimension up to a detailed exploration over many different model assumptions and inputs. Scenario analysis involves stipulating plausible alternatives of the model specification and drawing conclusions about the consequences by inspection of the results. This involves the comparison of different agent interaction mechanisms, learning mechanisms, parametric values, environmental constraints, or exogenous inputs to the system in terms of their effects upon model outcomes.

Validation of agent-based models is a crucial step that builds trust in outcomes of the same. The validation process usually focuses on qualitative and quantitative approaches (Taylor 2003). On one hand, the qualitative part consists of comparison of behavior of the model with empirical observations or descriptive accounts drawn from relevant literature and case studies. On the other hand, quantitative validation is provided by comparing the statistical analyses across time-series outputs from the model with empirical data. It is worth mentioning that these two validation methods are complimentary as they assess the micro and macro levels of the model (Edmonds 1998).

In light of the above, multi-agent based simulation concepts could be used to automate the process of administering construction disputes using logical induction through a multi agent system for construction dispute resolution (MAS-COR). Along the same line, it is worth investigating how risk management modeling principles could be used as supportive tools in mitigation of construction disputes.

2.3 Risk Management Modeling

2.3.1 Risk Management

Risk management is a pre-loss planning for post-loss delivery that aims to cost-effectively controlling and financing the risks an organization faces (Harrington and Niehaus 2004). The traditional discipline of risk management focused only on managing pure risks. Pure risks are those that involve the potential for loss without a corresponding possibility of gain (Chance and Brooks 2007). Over the years, the emphasis on only pure risk has changed with the adoption of "enterprise risk management" by large corporations, which seeks to integrate the management of all risks that the corporation faces, including investment and other business risks. The goal of any risk management plan is to: (1) protect the assets and financial viability of the organization and (2) minimize the cost of risk (Ling *et al* 1999). As previously mentioned, risks constitute a substantial element in the cost of construction projects. Thus, a contractor can gain a significant competitive advantage by reducing cost of risk below the industry average

Any risk management program is a combination of risk control and risk financing techniques. On one hand, risk control techniques, also commonly referred to as loss control techniques, are designed to reduce the frequency (i.e. loss prevention) and severity (i.e. loss reduction) of losses (Harrington and Niehaus 2004). On the other hand, risk financing involves arranging for funds to be available to pay losses that occur. The primary types of risk financing are risk retention and risk transfer. Losses can be retained through a number of methods including loss sensitive rating plans, qualified self-insurance, captive insurance subsidiaries, and risk retention groups (Harrington and Niehaus 2004). Meanwhile risk retention involves the contractor funding the losses from internal means; risk transfer involves shifting the financial burden of losses to another party. The two prevalent risk transfer techniques used by contractors are contractual risk transfer and the purchase of insurance. Contractual risk transfers are implemented by indemnity provisions and insurance requirements within construction contracts that shift the financial consequences of losses that may affect the project to third parties (i.e. sureties) that are not related to the contract (Chance and Brooks 2007). On the other hand, insurance involves buying coverage from a professional risk bearer, an insurance company.

2.3.2 Principle of Insurance

Insurance is defined as the equitable transfer of the risk of a loss, from one entity to another, in exchange for an indemnity rate called premium (Tsanakas and Desli 2005). According to Raviv (1979), the optimal insurance premium should be based on a meaningful historic data of around 3000 points. That said the accuracy of information is the most important element for the success of any insurance business (Gogol 1993). There are two issues that negatively affect any insurance policy, namely adverse selection and moral hazard. On one hand, adverse selection is the situation where the insured pool should not be mostly composed of high risk beneficiaries and thus, premiums are kept at a fair level (Janssen and Karamychev 2005). On the other hand, moral hazard is the setting where the insured pool should always realize that a loss will always be a misfortune and thus, the existence of the insurance will not change the behavior or due diligence of the insured party (Lee and Ligon 2001, Breuer 2005, and Doherty and Smetters 2005).

According to Jaffee and Russell (1997) as well as Harrington and Niehaus (2004), an insurable risk should have: (1) large number of homogeneous exposure units that allows insurers to benefit from the "law of large numbers," which in effect states that as the number of exposure units increases, the actual results are increasingly likely to become close to expected results; (2) definite loss where the event that gives rise to the loss that is subject to insurance should, at least in principle, take place at a known time, in a known place, and from a known cause; (3) accidental loss such that the event that constitutes the trigger of a claim should be least outside the control of the beneficiary of the insurance; (4) large loss where the size of the claim must be meaningful from the perspective of the insured because insurance premiums need to cover both the expected cost of losses plus around 30% to 50% of this cost to account for issuing and administering the policy (Brocket *et al* 1986), adjusting losses, and supplying the capital needed to reasonably assure that the insurer will be able to pay claims; (5) measurable loss where the frequency and severity of losses are easy to be calculated in a designated pool; and (6) affordable premium in the sense that the premium cannot be so large that there is not a reasonable chance of a significant loss to the insurer.

2.3.3 Perspectives of Insurance Pricing

The specification of appropriate functional forms for premium calculation principles has been the subject of much discussion in the insurance and actuarial communities (Harrington and Niehaus 2004). Traditionally, the fair premium in insurance pricing is equal to the expected loss resulting from the underwritten risk. Accordingly, a premium calculation principle is a function that takes as an argument (i.e. the probability distribution) of a risk and returns its fair premium. In this regard, the properties of a premium principle should reflect the characteristics of the actual prices charged in insurance market (Tsanakas and Desli 2005). Thus, the premium calculation principle is used to produce an actuarial benchmark, which is consistent with market prices. That said, an insurance premium is simply calculated from the average payout in cases of losses (Brocket *et al* 1986). Interestingly, this is the very same concept behind the pricing of a "put" or "call" warrants under the options theory (Hart *et al* 2006 and Mun 2006).

2.3.4 Theory of Options Pricing

An option is a security giving the right to buy or sell an asset or any financial instrument, subject to certain conditions, within a specified period of time (Chance and Brooks 2007). The price that is paid for the asset when the option is exercised is called the "exercise price" or "striking price." The last day on which the option may be exercised is called the "expiration date" or "maturity date".

Mun (2006) asserts that an option exists when a decision maker has the right, but not the obligation, to perform an act. Being the case, real options refer to the fact that firms have similar rights with regard to non-financial assets. Options add value as they provide opportunities to take advantage of an uncertain situation as the uncertainty resolves itself over time.

There are various types of options with different rules and methods of handling. The most well known options are European and American options. On the one hand, a European option is an option that gives the owner the right to buy/sell a financial asset at a predetermined future time and price. On the other hand, an American option would give the owner the right to buy/sell the asset at a predetermined future price at any time over the life of the option. In general, and as quoted in Mun (2006), the process of real options valuation

confines: (1) identification of the traditional NPV model; (2) determination of the key sources of uncertainty and putting hands on the appropriate parameters for these uncertain variables; (3) recognition of the opportunities that management have to respond to these key uncertainties; (4) calculating the real options estimate value using Black Scholes, Exotic Options, or Monte Carlo approaches; and (5) adding the real options value estimate to the NPV.

Black-Scholes Options Pricing Theory

The Black-Scholes model developed in 1973 a model for the valuation of European call options. It showed that it is not necessary to use any risk premium when valuing an option. According to Mun (2006), the model is based on set of variables that include stock price, strike price, time until expiry, interest rate, volatility, and yield. These can be detailed as follows:

1) Stock price is the market price of the underlying asset on the valuation date;
2) Strike price is the price level at which the option holder has the right to buy or sell the underlying asset;
3) Time until expiry is the time (in years) until the option expires and the holder is no longer entitled to exercise the option;
4) Risk free interest rate is the risk free interest rate for the period until the option expires;
5) Volatility is the variable implied by the market price of traded options;
6) Yield is the average yield generated by the underlying asset for the life of the option. This can be either a dividend (for a stock or stock index) or the income generated by a commodity.

In this regard, the following equations presents the price of a call option with exercise price K on a stock currently trading at price S, i.e., the right to buy a share of the stock at price K after T years. The constant interest rate is r, and the constant stock volatility is σ:

$$C(S,T) = S\Phi(d_1) - Ke^{-rT}\Phi(d_2) \qquad (1)$$

Where:

$$d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \qquad (2)$$

And

$$d_2 = \frac{\ln(S/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T}.$$

(3)

Moreover, Φ is the standard normal cumulative distribution function. Thus, the price of a put option may be computed from this by put-call parity and simplifies to:

$$P(S,T) = Ke^{-rT}\Phi(-d_2) - S\Phi(-d_1).$$

(4)

It should be noted that the Black-Scholes model is based on certain assumptions that would confine: (1) stock pays no dividends during the option's life; (2) European exercise terms are used; (3) markets are efficient; (4) no commissions are charged; (5) interest rates remain constant and known; and (6) returns are lognormally distributed. Thus, returns that follow all other distribution follow the general theory of exotic option pricing.

Exotic Option Pricing Theory

The whole theory of exotic option pricing is referred to Samuelson (1965) where : (1) a stock price is a definite probability distribution, $P(X,x;T)$, with constant mean expected growth per unit time $\alpha \geq 0$ and (2) the warrant's price, derivable from the stock price, must earn a constant mean expected growth per unit time $\beta \geq \alpha \geq 0$. Once the axioms, the numbers α, β, and the form of $P(X,x;T)$ are given, the rational warrant price functions $Y_t(T_t) = F(X_t, T_t)$ as follows:

$$F(x, T) = e^{-\beta T}\int_1^{x_a}(X - 1)dP(X,x;T) = e^{-\beta T}e^{\alpha T}x - e^{-\beta T} + e^{-\beta T}\theta_2\theta_2$$

(5)

Options pricing theories, especially those for exotic options such as Samuelson (1965), are very complex and thus, are simplified using the Monte Carlo method (Glasserman 2003).

Monte Carlo Analysis

In finance, the Monte Carlo method is used to simulate the various sources of uncertainty that affect the value of the instrument, portfolio or investment in question, and to then calculate a representative value given these possible values of the underlying inputs (Harrington and Niehaus 2004). In recent years, the complexity of exotic options outlined by Samuelson (1965) has tremendously increased, putting more demand on computational speed and efficiency. Thus, Boyle (1977) using the standard Brownian motion theorem was able to deduce that exotic options can be priced using Monte Carlo method as follows: (1) simulate the dynamics of the underlying asset using the Euler scheme; (2) calculate the payoff of

derivative security on each path; (3) discount payoff at risk-free rate; and (4) calculate average over path. Since then, interest in use of Monte Carlo methods for option pricing has been increasing because of the flexibility of the method in handling complex financial instruments.

2.3.5 Construction Industry and Insurance

Contractors usually purchase a portfolio of insurance policies that include: (1) workers compensation and employers liability insurance; (2) commercial general liability insurance; (3) umbrella or excess liability insurance; (4) contractors equipment insurance; and (5) property insurance covering the contractor's real and personal property.

In light of the above, insurance modeling principles could be utilized to: (1) develop an insurance policy that would cover the risk that are beyond control of contractors, not covered by surety risk transfer policies, and negatively affect the financial and economic standing the contractors such as environmental, economic, political, and site risks; and (2) create a risk retention group for contractors claims.

3.0 LOGICAL INDUCTION IN CONSTRUCTION DISPUTES

3.1 Background Information

3.1.1 Rule Based Reasoning

Rule based reasoning can be modeled using various forms of logic including logic of responsibilities and obligations (Allen *et al* 2000). However this type of reasoning has two problems when applied to the legal domain. First, logic will always miss an important part of argumentation, which is its power of persuasion (Ashley 1990). A second problem is the non-monotonicity of legal rules where rules both appear to be equally valid and yet lead to opposite conclusions (Alevan 1997). These two problems are extremely likely to occur in rule based reasoning where opposing sides attempt to draw opposite conclusions from the same set of facts. Therefore, legal arguments are not only necessarily concerned with presentation of facts but are also related to persuasion.

3.1.2 Arguments and Law

An important concept of common law is that of *stare decisis*, that is like cases should be decided alike (Ashley and Rissland 1988). This is the law of precedent where the outcome of a case is decided by its similarity to prior cases. However it is unlikely that we will have cases that are identical. There will, most likely, be various factual differences between a current dispute and any precedent. It is these differences that professionals argue about during construction disputes. The important facts upon which a case was decided are represented by cases' *Ratio Decidendi* that is the reasons that a case was found in a particular way (Ashley 1990). Thus, *Ratio Decidendi* and the case differences allow the possibility of argumentation. Any argument between parties will therefore revolve around the similarities and differences between a current dispute and a cited precedent. These differences may be factual, some event in one case may not be true for another, or based on legislation – one case may have been decided by a higher court (Alevan 1992). Accordingly, for effective management of any construction dispute, a vast amount of information needs to be gathered, analyzed, and presented by lawyers. The application of AI techniques such as adversarial case-based reasoning (ACBR) could greatly reduce the time spent wading through such information.

3.1.3 Case Based Reasoning and Adversarial Case-Based Reasoning

Case based reasoning (CBR) is a method of examining and solving problems by analogy (Ashley 1990). It has been exploited in various construction-related applications that comprise prequalification systems (Ng and Smith 1998), bid decision-making (Chua *et al* 2001), cost prediction of structural systems (Doğan *et al* 2006), contract strategy formulation (Chua and Loh 2006), optimum international market selection (Ozorhon *et al* 2006), and for disputed change order (Chen and Hsu 2007). Because of its flexibility in handling and explaining data, CBR was also found to be an efficient tool for construction litigation (Arditi and Tokdemir 1999).

CBR systems provide an index for retrieving cases, methods for applying the retrieved solution to the current problem, and a way to feed the current dispute into the overall database (Ng and Smith 1998). The word 'case' suggests an obvious relationship to the legal domain where current disputes are judged, to a large extent, on the outcome of precedent disputes. Before a case can be used in an argument, the relevant factors that make up the case must be determined (Doğan *et al* 2001). The accurate determination of these factors is of great importance since CBR simply retrieves precedent disputes that are similar to the current dispute and makes a judgment based on the outcome of the precedent case (Chua and Loh 2006). However, the simple retrieval of relevant cases may not work well in legal conflict resolution because of the adversarial nature of these conflicts.

As a result, this has given rise to a special type of CBR called adversarial case based reasoning (ACBR). "*Adversarial case based reasoning involves justifying a conclusion about a problem by drawing an analogy to a similar past case and arguing that the problem should be decided in the same way*" (Brüninghaus and Ashley 2001). Simply retrieving cases and suggesting a result does not produce an argument similar to a real legal debate. ACBR uses the principal of *stare decisis* to ascribe the decision to the closest precedent case (Bench-Capon 1997). The cases will need to be reasoned with, or about. This is achieved by using the facts of a case to create an argument for one side or the other. Available in the literature are various examples of successful ACBR systems such as HYPO (Ashley and Rissland 1988), CABERET (Shalak and Rissland 1992), CATO (Aleven 1997), and others. Figure 3.1 shows the interaction of previous researches in the area of legal reasoning.
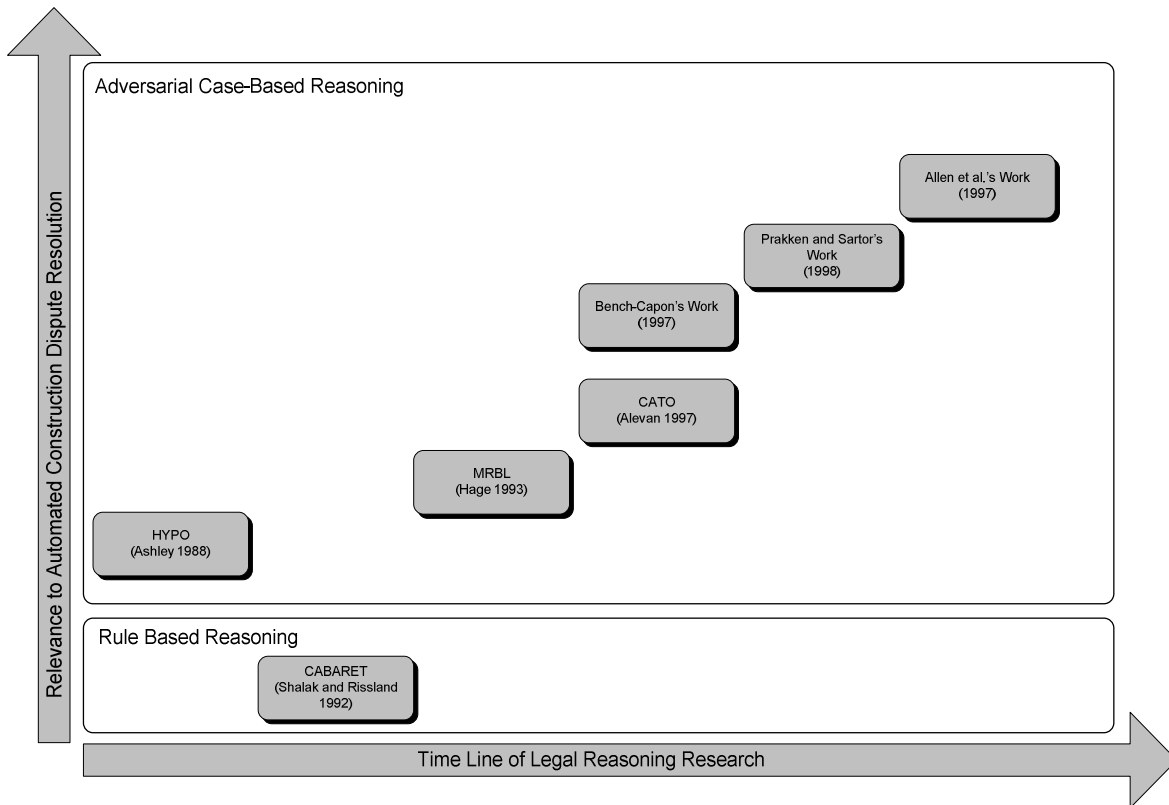
Figure 3.1: Developments in Legal Reasoning Research

HYPO performs adversarial reasoning with cases and hypotheticals drawn from the legal domain. It is a system for legal reasoning, which models disputes between a plaintiff and a defendant. Each move follows rules for analogizing and distinguishing factors within a precedent case. A case is cited if it shares at least one factor with the current dispute. If there are many cases, the cited case will be the one with the most inclusive number of common factors (i.e. as on point). The defendant may respond to the citation with a counter-case. The counter-case must be as 'on point' as the original case. The defendant can also distinguish between the case cited by the plaintiff and the new one by pointing out the differences between the two, especially those differences that tend to support the defendant (Bench-Capon 1997).

CATO is a legal tutoring system for teaching law students to make legal arguments using precedents. The program consists of a series of moves made up of individual arguments to build up a case. The available moves cover such things as analogizing a current

dispute to precedent disputes with a favorable outcome, emphasizing strengths, downplaying weaknesses and distinguishing cases (Aleven 1997).

Hage (1993) reconstructed HYPO's logic using Monological Reason Based Logic (MRBL). MRBL is concerned with rules and reasons. A reason is a set of facts that have some relevance to something else. There are reasons for behavior and one fact may be a reason to believe another fact. Rules are the underlying rationale that turns a set of facts into a reason. Rules have a condition part and a conclusion, or a conclusion and antecedent. The antecedent shows which facts count as a reason and the conclusion what they are a reason for. Although MRBL provides a method of analyzing legal reasoning, both case based and rule based, it does not provide the structure for conducting an argument.

Also, Prakken and Sartor (1998) reconstructed the reasoning processes of the HYPO/CATO system in a dialogue game logical framework where a citable case is found at the intersection of the set of factors between the current dispute and the precedent disputes. While HYPO and CATO marked factors as pros and cons, Prakken and Sartor (1998) proposed to represent precedents not as one argument but as a set of possibly conflicting arguments, including arguments that comparatively evaluate other arguments.

The fundamental rules derived from Prakken and Sartor (1998) cover the area of logical induction and include the following logical connectors: (1) negation as failure is essentially the absence of a factor from a particular case and is denoted by ($\sim$); (2) strong negation is the implication of a consequent resulting from the presence or absence of a fact and is denoted by ($\neg$); (3) strict rules are beyond debate and are denoted by ($\rightarrow$); (4) defeasible rules are the implications from facts that are arguable and are denoted by ($\Rightarrow$); (5) preferable rules are decisions expressed by a higher court and are denoted by ($\prec$). In other words, arguments are formed by chaining rules that ignore weakly negated antecedents. Moreover, conflicts between arguments are partly decided by utilizing priorities that are defined on the rules, and which induce a binary relation of defeat among arguments.

This chapter explores an innovative way pertaining to developing logical induction decision support for construction claims and disputes utilizing the principles highlighted by Ashley and Rissland (1988), Shalak and Rissland (1992), Bench-Capon (1997), Aleven

(1997), Prakken and Sartor (1998) and Allen *et al* (2000) in relation to logical induction in legal learning.

3.2 Methodology

3.2.1 Overview

In order to meet the goals and objectives of this chapter: (1) facts associated with construction change order cases were factorized into binary, dimensional, and abstract factors; (2) relevance was drawn between the developed factors and the disputing parties; (3) rules were generated as a result of the said factors; (4) factors were logically analyzed into distinct classifications; and (5) an 11 stage logical induction-based algorithm was mathematically modeled to put hands on similarities, differences, strengths, and weaknesses between current and precedent construction disputes.

3.2.2 Study of Change Order Construction Disputes

Resolution of change order construction disputes depend highly on gathering and putting hands on series of interrelated legal arguments that are usually scattered in set of documents and correspondences (Pickavance 2001). Thus, under this chapter, facts in relation to any construction dispute will be studied based on a number of factors representing the major facts of the dispute. Factors can be categorized as binary, dimensional, or abstract factors.

Binary factors are either existing (i.e. represented by 'ones') or non-existing (i.e. represented by 'zeros'). Dimensional factors are enumerated vector-like factors that have a distinctive magnitude (i.e. duration of delay as result of a change order). Finally, abstract factors proposed by (Aleven 1997), are ones that can be implicitly understood from the case, and have a hierarchy of sub-factors. Abstract factors can either be supported or militated against by sub-factors.

The proposed factorization is set in light of the cause and nature of disputes. Factorization was applied for change order related disputes as they represent the most significant source of disputes in the construction industry (Levin 1998). In all cases, those different types of factors can be either plaintiff *p-factors* (i.e. relevant and important to contractor) or defendant *d-factors* (i.e. relevant and important to owner). Thus, relevance was drawn between the developed factors and disputing parties. Moreover, a set of rules were

generated as a result of the aforementioned developed factors. These rules constitute the bases of the generation of all arguments pertaining to any construction dispute.

In order to insure the reliability of the proposed factorization as well as its associated rules and relevance for disputing parties, the same was submitted for review and evaluation by 5 top-notched industry arbitrators and mediators who have been in contact with construction disputes for more than 25 years in the U.S. and U.K. through an industry questionnaire that is included in Appendix A.

Accordingly, the proposed factorization as well as its associated rules and relevance for disputing parties were amended to the final setting shown in Tables 3.1 and 3.2 respectively.

Table 3.1: Factors in Change Order Disputes

Factor	Nature and Classification	Type
F1	Needs material available on site	Binary
F2	Needs material not available on site	Binary
F3	Allows change of material price	Abstract
F4	Delay effect in days	Dimensional
F5	Needs material in bill of quantities	Binary
F6	Needs material not in bill of quantities	Binary
F7	Needs local material	Binary
F8	Needs imported material	Binary
F9	Affected by new legislations and ordinances	Binary
F10	Carried out using available work force	Binary
F11	Carried out using additional work force	Binary
F12	Needs additional equipment	Binary

Table 3.1 illustrates that the existence of a binary factor implies a particular outcome either for or against plaintiff (i.e. contractor). Dimensions are enumerated such that higher or lower values are more favorable to one side or the other. In the investigated cases, F4's (i.e. delay effect of change order in days) higher values are more favorable to the contractor. It is worth noting that in this example we are arguing from the point of view of the plaintiff and therefore pro-contractor. This perspective does not indicate any kind of bias, but it is a mere standpoint in building the arguments under this work.

Table 3.2: Rules and Relevance of Factors

Rule	Factor	Relevance	Explanation
R1	F1	¬Contractor (d-factor)	Existence of F1 supports owner
R2	F2	Contractor (p-factor)	Existence of F2 supports contractor
R3	F3	Contractor (p-factor)	Existence of F3 supports contractor
R4	F4	Contractor (p-factor)	Existence of F4 supports contractor
R5	F5	¬Contractor (d-factor)	Existence of F5 supports owner
R6	F6	Contractor (p-factor)	Existence of F6 supports contractor
R7	F7	¬ Contractor (d-factor)	Existence of F7 supports owner
R8	F8	Contractor (d-factor)	Existence of F8 supports contractor
R9	F9	Contractor (p-factor)	Existence of F9 supports contractor
R10	F10	Contractor (p-factor)	Existence of F10 supports contractor
R11	F11	¬Contractor (d-factor)	Existence of F11 supports owner
R12	F12	Contractor (p-factor)	Existence of F12 supports contractor
R13	F5 ∨F7	F3	Existence of F5 and F7 supports F3
R14	F6 ∨F8 ∨ F9	¬F3	Existence of F6, F8, and F9, militate against F3

On the other hand, Table 3.2 highlights the rules associated with each factor and/or set of factors. For example, rules 13 and 14 state that the existence of F3 (i.e. no change of material price) is: (1) supported by the existence of F5 (i.e. needs material in bill of quantities) or F7 (i.e. additional quantity within contract limit) and (2) is militated against by the existence of F6 (i.e. needs material not in bill of quantities) or F8 (i.e. additional quantity not within contract limit) or F9 (i.e. affected by new regulations and ordinances). For the purpose of this research, the extraction of factors is carried out manually from case

documents. However, this process can be automated using Natural Language Processing techniques.

3.2.3 Factors Analysis

The starting point for the developed model depends on the analysis of the aforementioned factors in light of the logical relationship between precedent disputes and the current disputes. In this regard, logical connectives such as 'AND', 'OR', and 'XOR' are known as functions that take one or more truth values as input and returns a single truth value as output. An 'AND' operation yields true if all conditions are true and false if any condition is false, 'OR' yields true if any one of a sequence conditions is true, and false if all conditions are false, and 'XOR' or ($\oplus$) yields true if exactly one of two conditions is true. Previously, in construction research, logical connectives were employed by Rasdorf and Lakmazaheri (1990) to model design standards in civil works and in ontology development for knowledge management in highway construction (El-Diraby and Kashif 2005).

That said, the classification of the factors interrelated with construction disputes is set to be either 'A', 'B', 'C', and 'D'. Any precedent case can contain any combination between the 4 classifications. The said classification would depend on a rigorous scientific utilization of logical connectives that was proposed by Allen *et al* (2000). In light of the above, classification of factors depend on the similarities and differences between precedent cases and current dispute as follows:

1) 'A' factors are p-factors that the precedent dispute and the current dispute have in common;

2) 'B' factors are d-factors that the precedent dispute and the current dispute have in common;

3) 'C' factors are p-factors that are present in the current dispute but not in the precedent dispute and/or d-factors that are present in the precedent dispute but not in the current dispute;

4) 'D' factors are d-factors that are present in the current dispute but not in the precedent dispute and/or p-factors that are present in the precedent dispute but not in the current dispute.

Figure 3.2 shows the logic for the 'A', 'B', 'C', and 'D' classifications.
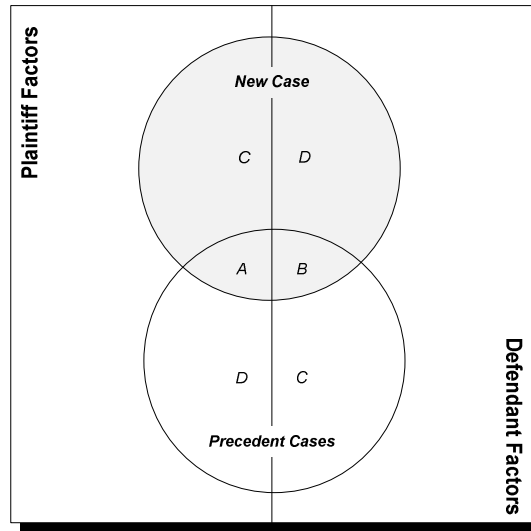
Figure 3.2: Factor Analysis (Adapted from Allen et al 2000)

3.2.4 Logical Induction Standpoint

The developed logical induction approach is modeled to argue from the plaintiff's perspective. That said, under the developed model and from the plaintiff's view, the following logical arguments are presented:

1) The highest numbers of 'A' and 'B' factors are desired because the number of 'A' and 'B' factors determine the degree of similarity between the current dispute and a precedent dispute.

2) The highest number of 'C' factors is needed because the more 'C' factors there are, the weaker the precedent dispute is with respect to the current dispute,

3) The fewest number of 'D' factors is desired because the less 'D' factors there is, the weaker the precedent dispute is with respect to the current dispute.

4) In the instance of disputes that will have a combination of 'C' and 'D' factors, the plaintiff would cite the dispute that would have the greatest positive difference between 'C' and 'D' factors.

That said, the contractor would cite relevant disputes based on combinations of analyzed factors pursuant to the following order of precedence: AC, ABC, AB, A, ACD, ABCD, ABD, and AD. In the event of two disputes having the same factor classification, the cited dispute will be the one with the greatest number of 'C' factors, fewest 'D' factors, greatest 'A' factors, or greatest 'B' factors in the same order of precedence.

3.2.5 Algorithm Design

Figure 3.3 highlights the flow of the various steps utilized for developed logical induction approach.
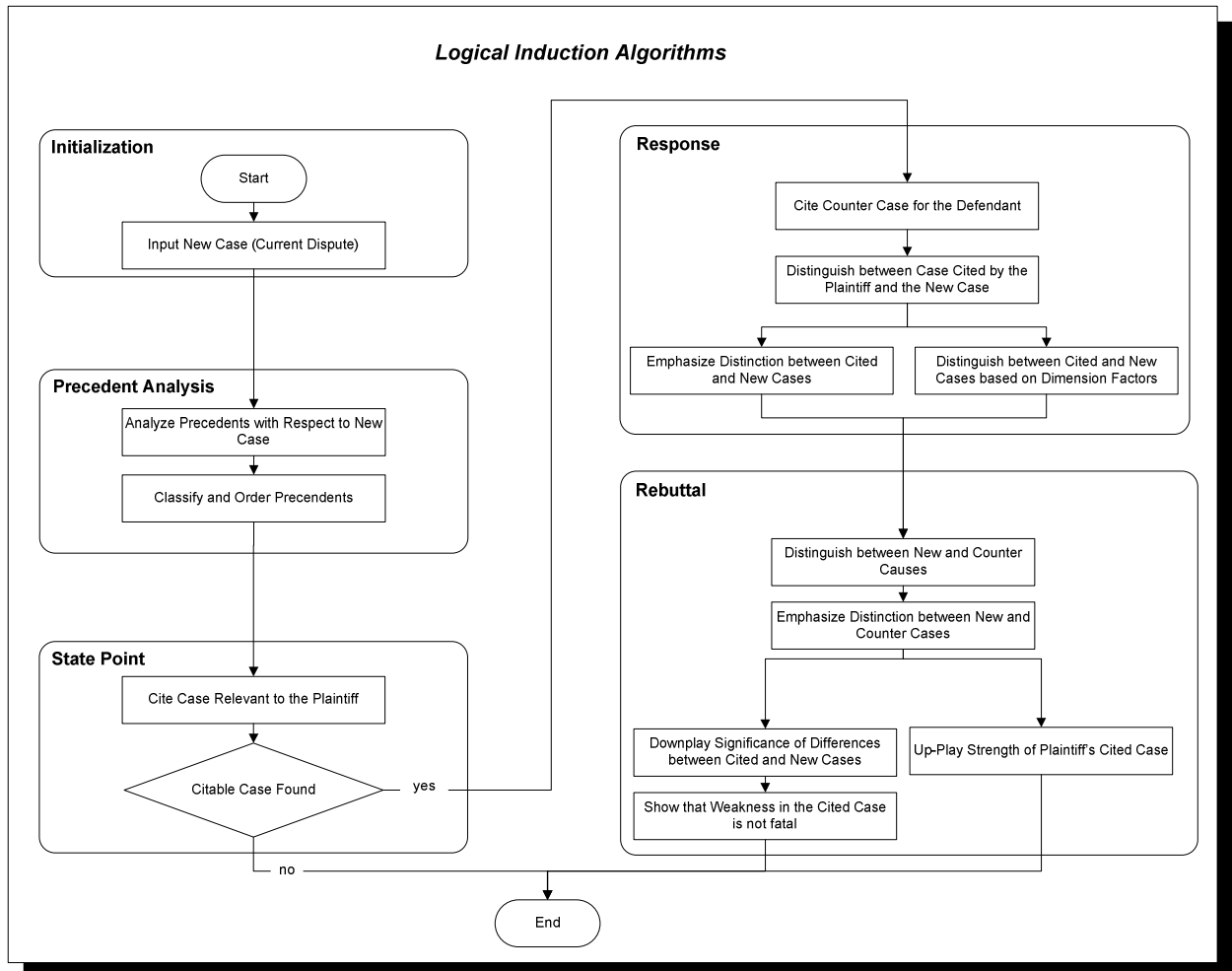
Figure 3.3: Logical Induction Arguments

As shown, the 3 main steps utilized for utilizing logical induction support system in construction claims and disputes, are:

1) State Point where plaintiff cites a supporting case;
2) Response where defendant counters plaintiffs' opening statement; and
3) Rebuttal where plaintiff rebuts defendant's response.

On the one hand, the response consists of:

1) Citing a counter case;

2) Distinguishing the cited case and the current dispute; and

3) Emphasizing any distinctions.

On the other hand, the rebuttal confines:

1) Distinguishing the counter case;

2) Downplaying distinctions;

3) Emphasizing strengths; and

4) Showing weaknesses not fatal.

Hereunder, a summarized description of the aforementioned algorithm, is presented:

1) New case is entered into the system and depending on the analysis of the factors in the new case and the precedent cases, the precedent cases will be classified and ordered.

2) *State Point (SP)* is the stage where the plaintiff states his supporting case. In order to decide on a citable case, we need to find the intersection of factors between the precedent disputes with plaintiff outcome and the current dispute.

3) *Counter Case (CC)* is the stage where a case is determined to be citable by the opposition (i.e. the defendant). A counter case must be a case with a defendant outcome. A case is *as on point* 'AOP' if it has the same 'A' and 'B' factors as the cited case and it is *more on point* 'MOP' if it has extra 'A' or 'B' factors. It is less on point, and therefore not citable, if it has fewer 'A' or 'B' factors. Thus, we get the matching factors between the precedent case with defendant outcome and the current dispute to decide the 'AOP' element. Then, we get the difference between the factors matching the counter case, current dispute, and the factors in the cited case after removing common factors between the counter case and the cited case to decide upon the 'MOP' element.

4) *Factor Distinction (FD)* is the stage where the case cited by the plaintiff is distinguished from the current dispute; we point out the unfavorable differences between the cited case and current dispute. We do not want to point out strengths but only weaknesses in cited case. Thus, an 'XOR' operation ($\oplus$) between the cited case and the current dispute gives a set of differences which is intersected with the factors within the cited case and the p-factors to get the absent p-factors in cited case. The

differences are also intersected with the factors of the current dispute and the d-factors to get the absent d-factors in current dispute.

5) *Emphasize Distinction (ED)* is the stage where factors that have been distinguished in the previous stage have this distinction emphasized. This emphasis comes from the fact that the combination of abstract factors implies the existence of other factors and, conversely, some other abstract facts imply the non-existence of other factors. Thus, to emphasize a distinction, we need to see if the differences between the current dispute and cited case support an abstract factor and whether that factor is not supported by another factor within the current dispute. Accordingly, we intersect the supporting factors with absent p-factors to emphasize p-factors. Likewise, we intersect the supporting factors with the absent d-factors to emphasize d-factors.

6) *Dimension Distinction (DD)* is the stage where the difference in dimensional factors (i.e. significance of effect) between the current dispute and cited case can also be illustrated. Therefore, this stage determines whether the value of a dimensional factor is more favorable to the defendant in the cited case than in the current dispute.

7) *Factor Rebuttal (FR)* is the stage where the differences between the current dispute and counter case that was cited by the defendant are illustrated. These differences should be 'C' factors that are weaknesses in the counter case with respect to the current dispute. Thus, an 'XOR' operation ($\oplus$) between the counter case and current dispute will give us a set of differences which will be intersected with p-factors and factors of the current dispute to determine plaintiff strengths in current dispute or intersected with d-factors and factors of the counter case to determine defendant strengths in current dispute.

8) *Dimension Rebuttal (DR)* is a similar stage to dimension distinction above in the sense that if the value of the dimension factors of current dispute is more favorable to the plaintiff than that of the counter case, they could be considered differences between the current dispute and the counter case.

9) *Downplay significance (DS)* is a similar stage to the aforementioned emphasize distinction stage. At this stage, we look for an abstract factor supported by a distinguished factor (i.e. a factor classified as a strength or weakness for the plaintiff)

in the counter case and which is also supported by a factor in the current dispute. Thus, we intersect the set of absent p-factors with the factors of the counter case and intersect the set of absent d factors with the militating weakness factors of the counter case.

10) *Up-play strengths (UPS)* is the stage where we need to capitalize and emphasize the factors that were classified as strengths for the plaintiff. Thus, we intersect the current dispute with the set of p-factors and an 'XOR' operation is carried between the result and the cited case to have the p-factors that are not in both cases which are either 'C' or 'D' factors. Likewise, we intersect the cited case with the set of d-factors and an 'XOR' operation is carried between the result and the cited case to have the p-factors that are not in both cases which are again either 'C' or 'D' factors.

11) *Weakness Not Fatal (WNF)* is the stage we need to find a case with a plaintiff outcome where the weakness were also present and applied. Thus, we need a case where the missing p-factors did not also apply or a case where the d-factors were present. This would show that either factor is not fatal.

The aforementioned carefully designed stages were integrated together to test the feasibility of using logical induction for decision support through highlighting similarities, differences, weaknesses, and strengths between current construction disputes and precedent ones.

3.3 Proof of Concept: A Detailed Worked Example

3.3.1 Overview

As a proof of concept, this chapter presents a detailed worked example where the developed approach was utilized to generate argument pertaining to five previously arbitrated construction cases where one of them was treated as a current dispute and the other four were regarded as precedent disputes. This arrangement was carried out based on the opinion of the previously mentioned top notched legal construction professionals in the US and UK who stated that usually they read the current dispute and relate its facts to a set of three to seven relevant precedent cases. The said cases were provided by a quantity surveyor and arbitrator at the Royal Institute of Chartered Surveyors and the Chartered Institute of Arbitrators. Under this example, the contractor is the plaintiff and the owner is the defendant.

3.3.2 Study of Construction Disputes

The following equations illustrate the factors present in the assumed 4 precedent disputes (P1 through P4 with their factors and their outcomes) as well as the presumed current dispute (CD):

P1= {F1, F2, F3, F4, F6, F7, F8, F9, F11} P   *(6)*

P2= {F3, F4, F5, F7, F9, F11} D   *(7)*

P3= {F1, F2, F3, F4, F5, F7} D   *(8)*

P4= {F2, F3, F4, F8, F10, F12} P   *(9)*

CD = {F1 to F12}   *(10)*

Table 3.3 provides a mapping between the peer reviewed factors and their availability in the cases under investigation.

Table 3.3: Mapping of Factors

Factor	Nature and Classification	P1	P2	P3	P4	CD
F1	Needs material available on site	1	0	1	0	1
F2	Needs material not available on site	1	0	1	1	1
F3	Allows change of material price	A	A	A	A	A
F4	Delay effect as result of change order (in days)	15	12	56	71	48
F5	Needs material in bill of quantities	0	1	1	0	1
F6	Needs material not in bill of quantities	1	0	0	0	1
F7	Needs local material	1	1	1	0	1
F8	Needs imported material	1	0	0	1	1
F9	Affected by new legislations and ordinances	1	1	0	0	1
F10	Carried out using available work force	0	0	0	1	1
F11	Carried out using additional work force	1	1	0	0	1
F12	Needs additional equipment	0	0	0	1	1
	Case Output	P	D	D	P	-

1 = Applicable
0 = Not Applicable
A = Abstract Factor

2.3.3 Factor Analysis

In order to determine the 'A' and 'B' classification, a series of 'AND' operations will be applied. In this regard, an 'AND' operation is performed on each respective factor in the precedent and current disputes. Then, to decide whether precedent dispute has an 'A' factor or not, an 'AND' operation is applied on the factor outcome of the precedent case where p-factors denoted as '1' and d-factors denoted as '0'. If the result has at least '1', then this is an

'A' factor. The very same methods applies to determine 'B' factors but with one difference that is d-factors are denoted as '1' and p-factors are denoted as '0'. Table 3.4 demonstrates that the precedent case P1 is determined to have 6 'A' factors. Also, Table 3.4 highlights that the precedent case P1 is determined to have 2 'B' factors.

Table 3.4: A and B Classification for P1 with Respect to Current Dispute

A-Test	1	1	-	1	0	0	1	0	1	1	0	0
B-Test	0	0	-	0	1	0	0	1	0	0	0	0

On the other hand and to assess the 'C' and 'D' classification, a series of 'AND' and 'XOR' operations will be applied. In this connection, an 'XOR' operation is performed on each respective factor of the precedent and current disputes. The 'XOR' on the two cases does not provide enough information as it tells the difference between the two cases but not the nature of this difference. To determine if these are plaintiff differences, another 'AND' operation will be applied on the 'XOR' result on the current dispute with the factor outcomes denoted as '1' for p-factors and denoted as '0' for d-factors and the result is a p-table. To determine if the p-differences are extra p-factors for the current dispute, an 'AND' operation is applied on the p-table with the factors in the current dispute and the result is a c-table. If the c-table has at least '1', then the precedent case has 'C' factors. Also, to determine if the p-differences are extra p-factors for the precedent case, 'XOR' operation is applied on the p-table and c-table and the result is a d-table. If the d-table has at least '1', then the precedent case has also 'D' factors. Table 3.5 illustrates that the precedent case P1 has one 'C' factor. Also, Table 3.5 shows that P1 has also 4 'D' factors.

Table 3.5: C and D Classification for P1 with Respect to Current Dispute

C-Test	0	0	-	0	0	0	0	0	0	1	0	0
D-Test	1	0	-	0	1	0	1	0	0	0	1	0

Thus, the precedent case P1 has an 'ABCD' factor classification. Likewise, the factors in the other precedent disputes are classified as follows:

- Case P2: ABCD
- Case P3: ABCD

• Case P4: ABCD

Moreover, Table 3.6 shows the similarities, differences, strengths, and weaknesses in precedent disputes in light of the current dispute. In fact, the aforementioned analysis would only show the high complexity of such cases and would substantiate the efforts towards developing a tool that would help industry professionals in generating arguments pertaining to the same.

Table 3.6: Similarities, Differences, Strengths, and Weaknesses in Precedent Disputes

Case	Related Factors			
	A	**B**	**C**	**D**
P1	6	2	1	2
P2	3	2	4	2
P3	2	3	5	1
P4	4	1	3	3

3.3.4 Mathematical Formulation of Algorithm

Following the aforementioned logical induction algorithm, the various stages of the model were applied to simulate legal discourse in relation to the cases under investigation. Details of such implementation are included in Appendix B.

3.3.5 Legal Discourse Simulation

Based on the developed algorithms, the following arguments were generated in relation to the current dispute under investigation as well as the relevant precedent disputes:

1) The state point algorithm will let the plaintiff (i.e. contractor) cite case P4 that has four common p-factors with the current dispute namely needs material not available on site, delay effect as result of change order, needs imported material, and carried out using additional work force.

2) The counter case algorithm will let the defendant (i.e. owner) cite case P3 that is 'more on point' to current dispute because it has five common d-factors namely needs material on site, needs material not available on site, delay effect as result of change order, needs material in bill of quantities, and needs local material Also, P3 is 'more on point' than cited case P4 because it has three additional d-factors namely needs material on site, needs material in bill of quantities, and needs local material.

3) The consequent algorithm is the defendant's response (i.e. owner) which consists of distinguishing the cited case through factor distinction, emphasize distinction, and dimension distinction. In this regard, three factors namely needs material on site, needs material in bill of quantities, needs local material, can be distinguished between cited case P4 and current dispute. Consequently, and as a result of the effect of militating abstract factors, five factors namely needs material on site, needs material not available on site, allows change of material price, delay effect as result of change order, and needs local material can be emphasized between cited case P4 and current dispute. Furthermore, the cited case P4 has larger delay than current dispute.

4) The following algorithm is the plaintiffs' rebuttal (i.e. contractor) which consists of distinguishing the counter case (i.e. P2) as well as downplaying these distinctions, emphasizing additional strengths, and showing any weaknesses not fatal in the cited case (i.e. P1). In this connection, five factors namely needs material not in bill of quantities, needs imported material, carried out using available work force, and carried out using additional work force can be distinguished between counter case P3 and current dispute. On the other hand, four factors namely needs material not in bill of quantities, carried out using available work force, carried out using additional work force, and needs additional equipment can be up-played between cited case and current dispute. Last, though the weaknesses in cited case namely needs material on site need local material were available in case P1, the same case had award in favor of the plaintiff and thus, these weaknesses can be demonstrated as not fatal.

The aforementioned arguments constitute all possible similarities, differences, weaknesses, and strengths between the current dispute and precedent ones. Realizing the complexity of such arguments, it is just normal that contract administrators, claims advisors, and lawyers spend lots of time, effort, and cost wading back and forth between construction cases to acquire such valuable and complex arguments. Thus, such generated arguments are highly needed by the all these professionals when evaluating current disputes in light of precedent cases and consequently, issuing relevant decisions and awards. The great value of such arguments as well as their complexity should foster any effort towards the automation

of utilizing logical induction and set theory approaches for generation of arguments change order construction disputes.

Moreover, and as might be also revealed from the complexity of those arguments, the known practices in the legal domain pertaining to the study of change order disputes that entails deep discussion on case by case basis even after acquisition of associated legal arguments made us decide to limit our model to generation of arguments without issuing actual awards. This inclination would preserve the soundness and realistic applicability of the model in the minds of industry practitioners and thus, encourage its gradual utilization in the market. However, this work is still conjectured as an important step towards the development of a logical induction-based algorithm that can produce a realistic award in construction disputes.

Meanwhile, the nature of change order disputes put limitations on our algorithm; this would not be the case in delay analysis disputes that entail a set of known rules in awarding time extensions. Thus, and as a future development for this work, logical induction will be utilized for generating arguments and issuing awards in delay analysis cases.

3.4 Summary and Conclusions

Construction disputes consume yearly around US $5 billion, which negatively affects the health of the construction industry as well as the nation's economy. This chapter developed an innovative method for studying construction claims and disputes using a logical induction approach. In this regard, (1) facts associated with construction change order cases were factorized into binary, dimensional, and abstract factors; (2) relevance was drawn between the developed factors and the disputing parties; (3) rules were generated as a result of the said factors; (4) factors were logically analyzed into distinct classifications, and (5) an 11 stage logical induction algorithm was mathematically modeled to put hands on similarities, differences, strengths, and weaknesses between current and precedent construction disputes. Also, and as a proof of concept, this chapter presented a detailed worked example where the developed approach was utilized to simulate the process of legal discourse based on adversarial precedent law where the developed algorithm will show, based on the relationship between the characteristics of dispute at hand and precedent ones, the best case to be cited by the plaintiff or the defendant and possible responses or rebuttals for each of the

two cases. The said example has presented a set of legal arguments that are highly needed by contract administrators, claims advisors, and lawyers when evaluating current disputes in light of precedent cases. The great value of such arguments as well as their complexity should foster any effort towards the automation of utilizing logical induction approach for decision support in change order construction disputes. Thus, and as a further development of this work, logical induction can be used to create a multi agent system for construction dispute resolution (MAS-COR). This anticipated development would insure reusability, scalability, and flexibility of logical induction and set theory as supportive tools in the area of construction dispute mitigation and resolution.

4.0 MULTI AGENT SYSTEM FOR CONSTRUCTION DISPUTE RESOLUTION

4.1 Background Information

4.1.1 Discrete Simulations in the Construction Industry

Construction researchers and practitioners have used various techniques for studying complex construction systems. These techniques covers a wide spectrum of methods that include networking techniques, queuing models, productivity models, probabilistic models, linear programming, game theory, fuzzy logic system dynamics, and artificial intelligent simulations. As quoted in Sawhney *et al* (2003) the most significant discrete simulation models in the construction industry would include:

- The "link node" developed by Teicholz (1963) for modeling construction operations;

- CYCLONE model developed by Halpin (1973) as an application of the construction process simulation that assists in productivity measurement, risk analysis, resource allocation, and site planning;

- Micro CYCLONE model developed by Luch and Halpin (1981) at Georgia Tech, which was the first micro computer version of CYCLONE;

- INSIGHT system was developed by Paulson (1987) and is based on the CYCLONE /methodology and has a more interactive interface;

- RESQUE model was developed by Chang and Carr (1987) and is also CYCLONE based with advanced resource handling capabilities;

- UM-CYCLONE was developed by Ioannou (1989) for advanced construction process modeling;

- Simphony was developed by AbouRizk *et al*. (1999) as well as Hajjar and AbouRizk (1999) at the University of Alberta to provide an advanced simulation environment specially tailored for construction researchers and practitioners;

- STROBOSCOPE was developed by Martinez (1996) as well as Martinez and Ioannou (1999) at the University of Michigan and now housed at Purdue University.

All such contributions have positively affected the process of construction management and triggered research for more employment of more new tools for more effective and efficient management of the construction process. Accordingly, and with the advancements

in the area of multi-agent systems, construction researchers have employed multi-agent simulations in order to study and solve various management related applications.

4.1.2 Multi-Agent Systems in the Construction Industry

As previously discussed, complexity is one of the major characteristics in the construction industry. In fact, the construction sector witnesses various and different project participants who need to get together in order to perform various construction activities. Moreover, these construction activities usually involve open, dynamic, and complex problems where the structure of the system itself is capable of changing dynamically. Anumba *et al* (2005) stated that the properties of such a system are that its components are not known in advance, changeable over time, and can consist of highly heterogeneous inputs that are implemented by different people, at different times, and with different techniques. Such a complex interactive environment requires agents to be able to inter-operate and collaborate with each other and to learn from one another. Being the case, MAS is perceived to have considerable potential to address various applications in the construction industry. Anumba *et al* (2005) states that using MAS in the construction industry could provide: (1) effective decomposition of large-scale construction problems; (2) improved collaborative and concurrent working; (3) easier and cheaper way to specialist information.

Based on a comprehensive literature review of previous research that utilized MAS in construction related applications, it was discovered that MAS was mainly used in material management, engineering design, supply chain management, scheduling and control, and negotiations in dispute resolutions. This research would cover agent-based integrated building design environment (Fenves *et al* 1994), agent-based collaborative design for construction (Heckel 1996), agent-based design changes system (Chiou and Logcher 1996), agent-based resource management system (Oliviera 1997), CONVINCER mediating agent-based model (Pena-Mora and Wang 1998), agent-based project schedule coordination agent based tendering system (Denk and Schnellenbach-Held 2002), MASCOT negotiating agent-based model (Ren *et al* 2003), and models for situational simulations in construction industry (Rojas and Mukherjee 2006). It is deemed that all of these research efforts would have a positive effect in increasing the efficiency and effectiveness of the construction process, contracting parties, and benefit the overall industry in return.

In the following two sections, a much more thorough and detailed review and critique will be presented for the two researches pertaining to the previously mentioned MAS models namely, CONVINCER and MASCOT. This special emphasis for those two specific models is given because they deal with the issue of construction dispute resolution.

CONVINCER: Agent-Based Mediating Model

CONVINCER is a collaborative negotiation methodology and computer agent that is used to mitigate the negotiation of conflicts in the construction industry. The said simulation model is primarily based on game and negotiation theories.

Negotiation theory is the qualitative method concerned with the interaction between parties, reconciliation of differences, and attainment of mutual agreements (Pena-Mora and Wang 1998). As previously mentioned, the construction industry is based on cooperation of competitive parties. Moreover, the involved parties are usually domain specific in the sense that each party has only a partial view of the whole problem of successfully completing the project. Therefore, each of those parties will usually have a strategy to maximize their gains during the course of the projects. Being the case, and as indicated by Pena-Mora and Wang (1998), it should be understood that the construction industry is naturally a none-zero sum game where all participants can share profits. Accordingly, an efficient utilization of game theories in mediation of construction disputes should be based on the above mentioned characteristics.

Game theory is known to be the quantitative study of a player's action that can affect the payoff of other players (Pena-Mora and Wang 1998). In this connection, it should be highlighted that the resolution of construction disputes can take place using direct negotiation, mediated negotiations, or binding adjudication. It is conjectured that the developed collaborative negotiation model is most suitable for mediated negotiations. Accordingly, Pena-Mora and Wang (1998) stated that the said model should incorporate basic elements such as the project, the participants, the negotiation process, the collaborative negotiation methodology, and the outcome.

As highlighted by Pena-Mora and Wang (1998), the development of CONVINCER is based on a five step methodology. It starts with building a decision tree that highlights the available choices, decision, or alternatives for each of the project participants. The

methodology continues by selecting the payoff function for each of the involved parties. Pena-Mora and Wang (1998) conjectured that the payoff function should be based on the premise that positive concessions made by any player usually decrease with time. Accordingly, and in order to be in line with this premise, the payoff function should have a first derivative that is less than zero and a second derivative that is less than or equal to zero. However, a matter of concern in formulating such a payoff function is the possible exaggeration by some parties as well as lack of knowledge of others.

Knowing the players' payoff function for a certain matter, the next step is to find the settlement point that will generate the maximum payoff for the same. Pena-Mora and Wang (1998) asserted that the maximum payoff is the point of agreement attained when plotting and adding the payoff functions on one coordinate system. Being the case, the settlement point can be found on the horizontal axis.

Knowing all the payoffs for the various alternatives, the suggested decision set can be explored using backward induction. According to Wikipedia (2007), "*backward induction is one of dynamic programming algorithms used to compute equilibrium in sequential games where the process proceeds by first looking at the last possible action, determine what the last player will do in each situation, and determine what the second to last player will do*". This method appears to be a reasonable technique for putting hands on the available actions' domain.

Later on, and in order to account for the matter of exaggeration and incomplete information, the model is designed to give some kind of warning to participants that are disadvantaged as a result of domain dependency or opportunistic strategies that were developed by other parties.

It is worth noting that the said model was implemented using JAVA and is operated by parties using the internet through Java Database Connectivity and Open Database Connectivity. The authors have presented a construction dispute that was attempted to be resolved once with direct negotiations and the other through mediated negotiations of CONVINCER. While in unassisted negotiations, agreement was not resolved up until the fourth negotiation round, CONVINCER helped the disputing parties to settle after two

rounds. This is mainly attributed that using CONVINCER, parties reveal their preferences to the mediating agent who in consequence suggests solutions based on those preferences.

It can be conjectured that the proposed model represents a reasonable step towards helping contracting parties to reach mutual settlement pertaining to matters in dispute. Nevertheless, it is also a fact that the said model has several weak points that would refute its dependability in critical application like mediating in construction disputes. In CONVINCER, each party's decision-making is determined by a simple game theory rule, which can hardly represent the real negotiation situation. Moreover, it is believed that the proposed model would be much more efficient if it does not only depend on statistical and inductive methods for calculating the payoff functions, but is also dependant on theories of social and psychological behavior as those are an integral part of the rational governing the negotiation process.

MASCOT: Agent-Based Negotiation Model

MASCOT is a multi-agent system for construction claims negotiation. In the said model, autonomous agents act on behalf of project participant and can directly negotiate with each other to resolve construction claims. Ren *et al* (2003) assert that while all previous researched for development of negotiating support systems have only concentrated on the analytical aspects, MASCOT would concentrate on the comprehensive view of the interactive negotiating process where attained settlement is based on Bayesian-Nash equilibrium. It is worth noting that the said model was developed using the ZEUS platform that incorporates the Java programming language.

In their quest to build MASCOT, Ren *et al* (2003) have deeply surveyed negotiation theories, game theories, behavior-oriented theory, and the characteristics claims negotiations. Pursuant to their study, they highlighted that:

- In negotiation, the four most important elements are parties, values, movements and outcomes. Moreover, Ren *et al* (2003) stated that while negotiation parties may be fixed, value and behaviors are modified to converge the parties' divergent opinions.

- Game theories are based on axiomatic and strategic approaches. Pursuant to Ren *et al* (2003), strategic approaches are much more relevant to construction negotiations as they allow for convergence of a solution. Accordingly, the paper made use of

Zeuthen's model where players first calculate the maximum probability of conflict they would be willing to accept in preference to acquiescing on the opponent's current offer, and then the player who has a lower maximum risk acceptability will make the next concession.

- The behavior-oriented theory covers issues in relation to negotiator's personality, perceptions and expectations, persuasive techniques, psychological and contextual factors, and the different learning approaches. Ren *et al* (2003) highlighted the importance of these concepts when modeling an agent based negotiation model suh as MASCOT.

- The construction claim negotiation is dependent on series of issues that are competitive-collaborative nature of the industry, domain specific knowledge, strategic decisions, and time. It should be noted that the first three characteristics were previously outlined by Pena-Mora and Wang (1998).

Pursuant to Ren *et al* (2003), MASCOT is based on five (5) basic premises that are: 1) negotiation will be a quantitative process in relation to settled amounts; 2) agents are rational in their decision pertaining to utility and risk; 3) utility function is fixed; 4) agents do not have complete information but can learn; and 5) information are not disseminated to agents during the negotiation process.

As previously mentioned, MASCOT's negotiation model utilizes the Zeuthen's model with the support of learning algorithms in order to account for the problem of incomplete information that is inherent in the Zeuthen's model. Furthermore, MASCOT's protocol specifies the kinds of deals that the agents can make, as well as the sequence of offers and counter-offers that are allowed. MASCOT's protocol is a Monotonic Concession Protocol where agents simultaneously present their offers, agreement is reached when offers match or one exceeds the other, continues to second round if no matching occurs, and no agent can reduce his offer in iterative rounds.

However, and in order to account for the characteristics of the construction industry, Ren *et al* (2003) have modified the Monotonic Concession Protocol in the sense that it can accommodate:

- Relaxed conflict deals using mediators/client agents and the expanded search process for solutions;
- Learning using the Bayesian mechanism where an agent receives an offer (or counteroffer) from its opponent, the agent will analyze the offer, modify its beliefs about the opponent; and
- Time penalty in utility functions to accelerate settlement process.

Ren *et al* (2003) have designed MASCOT in the sense that maximization of the utility function is done in two steps where first the separate items are maximized and then the total utility is maximized using tradeoffs between the separate elements. It should be noted that MASCOT assumes a linear relation between the two extremes of the utility function, which are the reservation and optimum values. Moreover, MASCOT allows concession between agents on the basis of amount of risk. In fact, Ren *et al* (2003) asserted that if each agent's willingness to risk conflict can be measured, the agent with less willingness to risk will make a concession. In other words, settlement will be reached when the maximum risk for both parties is zero.

Ren *et al* (2003) have also validated MASCOT for the efficiency, simplicity, and stability aspects of the negotiation process. This seems like a comprehensive validation for the model. In this regard, and as far as efficiency is concerned, the authors asserted that though the model, by its construction, may not generate a pareto-optimal solution a fair solution will not specify an agreement that could be improved for one participant without harming the other, still the agreement will be efficient at least in the final round. Also, it was conjectured by the authors that despite the model's adaptation of learning, it is still simple because calculations are done locally within each agent. Ren *et al* (2003) have conjectured that Nash equilibrium is not efficient for construction negotiations because information is not complete to all parties and thus, more stability was attained using the Bayesian-Nash equilibrium.

For testing and validation, MASCOT was employed in three negotiation examples between the contractor and the engineer in connection with a water supply project. Based on the presented results, it is obvious that MASCOT has played a significant role in providing timely and cost effective negotiation of construction disputes. Also, it is evident that MASCOT is considered to be a much more sophisticated model than CONVINCER as it is

based on Bayesian-Nash equilibrium, Zeuthen's model, and behavior oriented theories. Nevertheless, and despite that such a model is much more comprehensive and sophisticated than other developed models, it still did not provide a method that would enable the agents to reason the correctness of the information provided by the parties. In other words, MASCOT in its current form can only negotiate deals on claimed amounts without having the ability to reason the correctness/incorrectness of such values. Moreover, it is perceived that employment of Q learning should be investigated as it is hypothesized that the same could be very efficient in updating the knowledge of negotiating parties.

Meanwhile, the above-mentioned models have concentrated on conducting effective negotiations between the conflicting parties. However, none of these models used facts and outcomes of precedent disputes to provide decision for legal arguments in construction claims and disputes.

4.2 Methodology

4.2.1 Overview

In order to implement the algorithm formulated in the previous chapter towards a multi agent system for construction dispute resolution (MAS-COR): (1) an agent-based role model was developed to represent the said interrelated algorithms; (2) a communication protocol between the agents was envisaged to ensure proper and easy flow of information through the system; (3) system implementation to put the aforementioned model and protocol into an object-oriented program; and (4) testing and validating the developed system through syntactical debugging and number of test cases.

4.2.2 Development of Agent-Based Role Model

An agent-based model was chosen as the basis for the implementation in order to use a highly modular design that allows experimentation with more, fewer or different components possible and allows one to take advantage of any parallelism that may present itself in the algorithms (Anumba *et al* 2005). In this regard, the said model and its agents were developed in light of the 11 stage logical induction algorithm that was presented in the previous chapter. As previously noted there are various ways for agents learning. However, under MAS-COR development, logical induction is utilized to provide agents with a cournot (i.e. naive) belief-

based learning in their actions. In this regard, Figure 4.1 shows the developed agents as well as their anticipated roles.
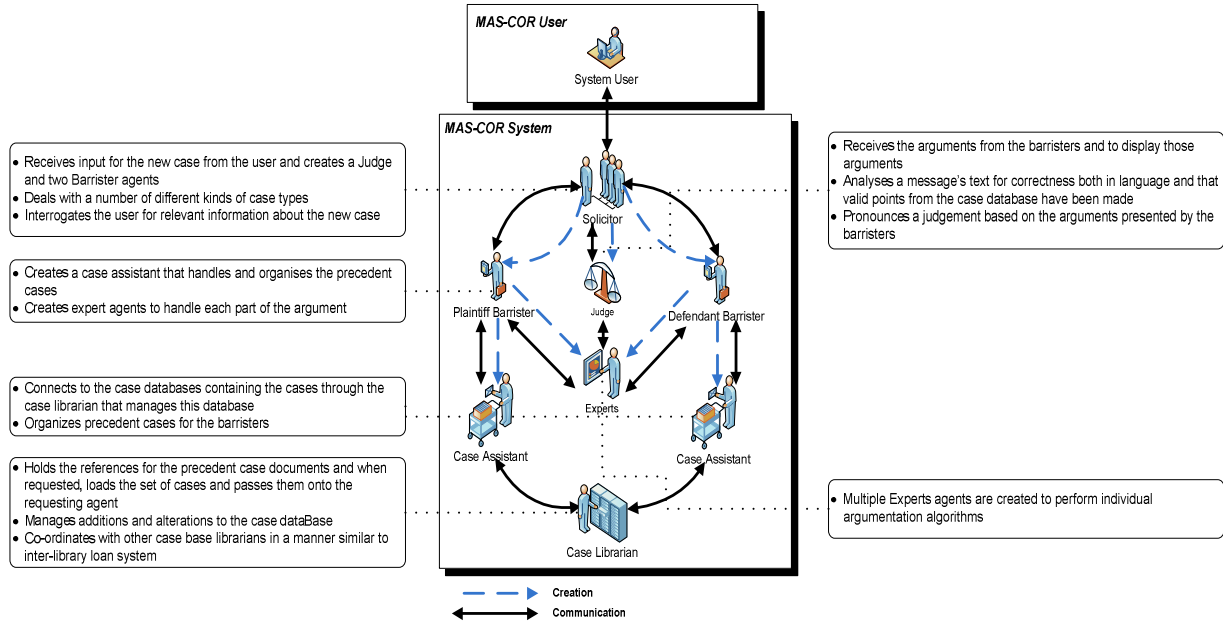
Figure 4.1: Agent-Based Role Model

The respective roles of each of the modeled agents are presented hereunder:

1) Solicitor: The solicitor receives input for the new case from the user and creates a Judge and two Barrister agents; one barrister represents the plaintiff and the other represents the defendant. Also, it will be able to interrogate for relevant information about the new case.

2) Case Librarian: The case librarian will provide agents with the library services pertaining to precedent cases in relation to available/non-available factors as well as the case outcomes. In this connection, the case librarian holds the references for the precedent case documents and when requested, loads the set of cases and passes them onto the requesting agent.

3) Case Assistants: There are two types of case assistants, one for the plaintiff and one for the defendant, both sub classed from the base case assistant. They connect to the case databases containing the cases through the case librarian that manages this database. They organize precedent cases for the barristers. The case assistant classifies and sorts the precedent cases according to the logical classification rules.

Defendant and plaintiff case assistants will sort cases in opposite order and then supply the best case to the barrister as requested.

4) Barristers: Barrister agent is cloned into two agents; a barrister for the plaintiff and a barrister for the defendant. The barristers' first task is to create a case assistant that handles and organizes the precedent cases. The barrister will create expert agents to handle each part of the argument.

5) Experts: Expert agents are created to perform individual argumentation algorithms. The envisioned expert agents include:

   i) State Point Expert, created by the solicitor to undertake the state point part of the algorithm where classifying and sorting of the case base is carried out to determine the citable case.

   ii) Counter Example Expert, looks for a counter example. This must be a case that is pro-defendant and at least "as on point" as the cited.

   iii) Distinguish Factors Expert, distinguishes on the new case and the cited precedent case. It is created and used by both the plaintiff and defendant barristers.

   iv) Distinguish Dimensions Expert, distinguishes between cases based on dimension factors and is utilized by both the plaintiff and defendant barristers.

   v) Emphasize Significance Expert, makes use of the factor hierarchy to emphasize the significance of a distinction.

   vi) Up-play Strengths Expert, searches the case base for factors to reinforce the plaintiffs' case.

   vii) Downplay Distinctions Expert, tries to do the opposite of the "emphasize significance" expert mentioned above.

   viii) Weaknesses Not Fatal Expert, searches for cases that also had the weakness but were still found for the plaintiff.

6) Judge: the judge will be receiving arguments from the barristers, rejecting invalid arguments, and displaying in return the valid arguments. Thus, it will be able provide decision support for legal arguments in the new construction dispute in light of a set of precedent disputes.

4.2.3 Creation of Communication Protocol

A communication protocol between the agents was designed to ensure proper and easy flow of information through the system utilizing a direct multi-agent planning approach. In this regard, as soon as the solicitor is fed with the details of the new case in dispute, it loads the case data and then creates a judge, case librarian, plaintiff barrister, and defendant barrister. Case librarian acquires the data for the precedent cases through the case assistants. At this point in time, plaintiff barrister receives a ready message from case librarian and will call its own argumentation methods creating a state point expert. Once the state point argument and its associated tasks of classification and ordering of cases argument are completed, the plaintiff barrister will cite a case. Now, the response part of the argument is ready to take place. Under the response algorithm, the counterexample expert, factor distinction expert, emphasize distinction expert, and dimension expert are created and run simultaneously using the same information that were initially passed to the state point expert. Afterwards, the rebuttal stage of the argument starts involves creating four experts namely emphasize significance expert, up-play strengths expert, downplay distinctions expert, and weaknesses not fatal expert. It should be noted that the *Judge* receives the argument strings from all experts during state point, response, and rebuttal arguments. Consequently, the judge will be able to provide decision support for legal argument in the new dispute in light of precedent disputes.

4.2.4 System Implementation

The multi agent system for construction dispute resolution (MAS-COR) was modeled as an object oriented program using Java programming language on NetBean's integrated development environment where the logical induction and set theory were utilized the bases for learning and utility functions of modeled agents. During the process of system implementation, main classes of MAS-COR were created in a manner that exactly represented all needed agents as previously detailed under agent-based role. Moreover, the attributes of those classes were added to be in complete accordance with the previously mentioned communication protocols between agents. In this regard, Figure 4.2 shows the UML diagram for the implemented system.
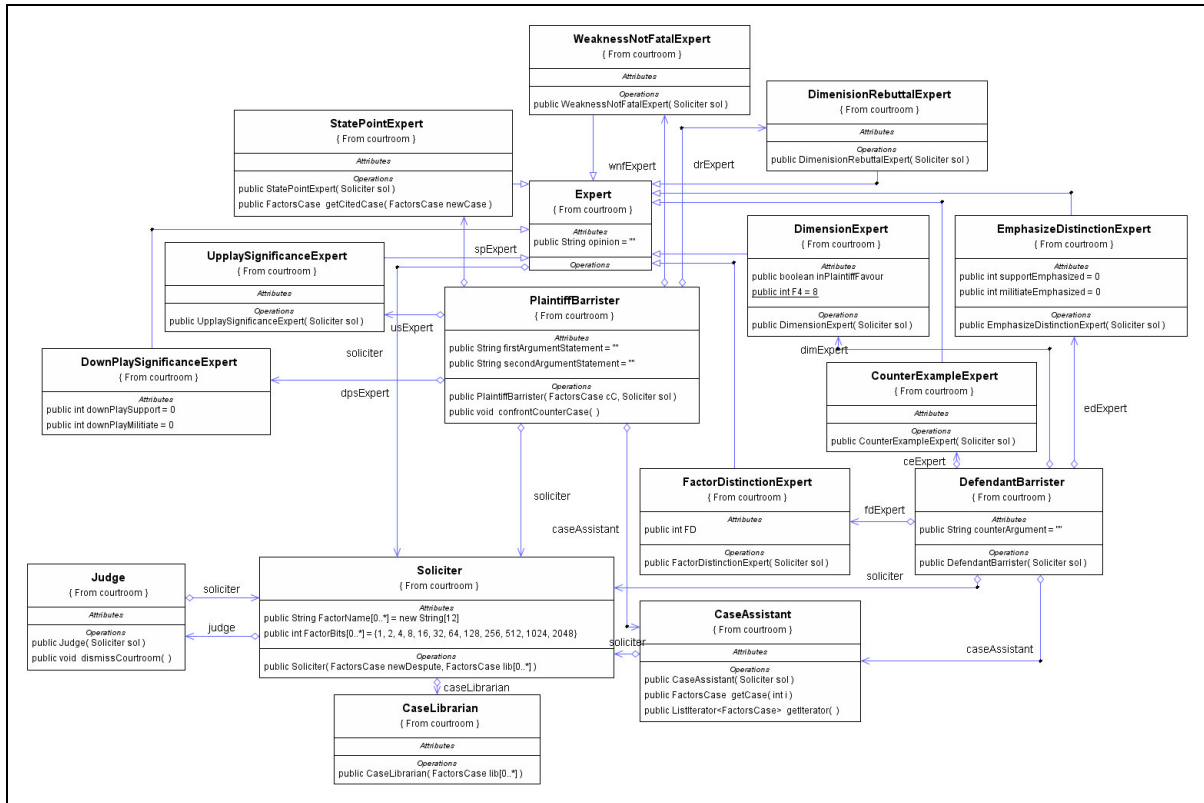
Figure 4.2: UML Class Diagram for MAS-COR

In order to graphically show the development and interaction of agents throughout the simulation period, the intention was to represent the java-build model on the recursive porous agent simulation toolkit (Repast). Repast is an agent-based simulation toolkit specifically designed for social science applications that permits the systematic study of complex system behaviors through controlled and replicable computational experiments and provides a core collection of classes for building and running agent-based simulations, and for collecting and displaying data in tables, charts, and graphs (North *et al* 2005). For the purpose of this study, RepastS (Repast Symphony, Java based) was to be utilized as it extends the Repast portfolio by offering a new approach to simulation development and execution, including a set of advanced computing technologies for application (North *et al* 2005). However, and after in-depth preliminary coding exercises, it was concluded that RepastS, as well as any other similar platform like Swarm or NetLogo, are not optimum platforms for such graphical process. This is because none of the developed agents under MAS-COR had, or needed; any internal state variables that lasted beyond any single operation they tried to perform during

the simulation except the librarian and case assistants, which are basically instruments for reading files. That said, the idea of graphical representation of agents, was abandoned for the purpose of MAS-COR development.

4.2.5 Testing and Validation

The last step of this methodology focused on testing and validating the implementation MAS-COR. The testing and validation steps will rigorously assess the developed agents and their integration into the MAS-COR system through syntactical debugging and number of test cases. Pursuant to Gatti and Staa (2006), there are at least three approaches for evaluating multi-agent systems: (1) theorem proving, which corresponds to checking that a set of formulas satisfies a goal; (2) model checking, which checks if agents are proved and modeled correctly pertaining to their on both the single and social aspects; and finally, (3) system testing, which consists in applying test cases to the system and checks if it behaves properly.

As far as MAS-COR is concerned, theorem testing was carried through the worked example presented in the previous chapter and which substantiated the effectiveness of logical induction for decision support in construction claims and disputes. In this regard, and for complete evaluation of MAS-COR's effectiveness, model checking and system testing are presented hereunder.

Model checking

According to Gatti and Staa (2006), developers also usually want to incrementally test their agents as they progress, testing functionality as it is added. Thus, model checking was done incrementally during software development in relation to agents' internal and external behaviors through regression and progression testing using simultaneous numerous fictitious cases that comprised all possible permutations and combinations between different factors and outcomes. Meanwhile, regression testing is used to insure that agents perform to their stated specifications and that modifications to agents do not affect existing message handling capabilities, progression testing supports agent developers as they add new messaging capabilities to their agents (Coelho *et al* 2006). Moreover, and in addition to individual testing of agents, society communication tests were applied to the agents of MAS-COR. This involved checking that each agent in the community/society receives the correct messages from the correct agent, provides the correct responses, and interacts with environment

correctly as a whole. Some of the main errors types that were encountered during development included:

1) Incorrectly addressing messages to other agent: For example, the message initiated by case assistants and which is loaded with details of cases, was not correctly passed to case librarian;

2) Improperly putting a request in a message so that receiving agent does not recognize the message: In this regard, the distinguish dimension expert was requesting the distinguished factors from the down-play expert rather than from the distinguish factor expert;

3) Created deadlocks into some messages exchanges between agents: In this instance, the state point expert was unable to access the sorted and classified precedent cases to cite a case supporting the plaintiff; and

4) Not developing sufficient code to accept all the messages that an agent is supposed to accept: In this connection, the messages directed to the judge from the solicitor representing the opinions of various agents were lost in the system;

Pursuant to this ongoing syntactical debugging, MAS-COR agents were remodeled and recoded to rectify the aforementioned errors. The final source code for MAS-COR is included in Appendix C.

System Testing

As far as system testing was concerned, MAS-COR was exposed to 30 actual previously arbitrated cases that were provided by a quantity surveyor and arbitrator at the Royal Institute of Chartered Surveyors and the Chartered Institute of Arbitrators in London. Those test cases were manually factorized and then sorted in 6 different simulations where each simulation is composed of one assumed new case and 4 assumed precedent cases. The first simulation represented the 5 test cases that were used in the worked example in the previous chapter. In this regard, Figure 4.3 shows the actual output of MAS-COR for the aforementioned worked example

Figure 4.3: Graphical User Interface of MAS-COR

Figure 4.4: Extract of MAS-COR Output for Worked Example in Chapter 4

The said output which fully complies with the results presented in the manual solution of the same worked in the previous chapter. This would provide preliminary confidence in MAS-COR, nevertheless; more detailed testing is required for more assurance with different case scenarios

4.3 Results and Analysis

The remaining factorized 25 cases were tested in 5 different simulations in order to ensure the reusability and scalability of MAS-COR as decision support for legal arguments in

change order construction disputes. Meanwhile Tables 4.1 through 4.5 show a mapping of factors in the other five tested cases, Table 4.6A and 4.6B show MAS-COR's results.

Table 4.1: Mapping of Factors in Test Case 1

Factor	Nature and Classification	P1	P2	P3	P4	CD
F1	Needs material available on site	1	1	1	0	0
F2	Needs material not available on site	0	0	1	0	0
F3	Allows change of material price	A	A	A	A	A
F4	Delay effect as result of change order (in days)	127	43	11	21	48
F5	Needs material in bill of quantities	1	1	0	0	1
F6	Needs material not in bill of quantities	0	0	0	1	0
F7	Needs local material	1	0	0	0	0
F8	Needs imported material	0	1	1	1	0
F9	Affected by new legislations and ordinances	0	1	1	0	0
F10	Carried out using available work force	0	0	1	0	1
F11	Carried out using additional work force	0	1	1	0	1
F12	Needs additional equipment	0	1	1	1	1
	Case Output	D	P	P	D	-

Table 4.2: Mapping of Factors in Test Case 2

Factor	Nature and Classification	P1	P2	P3	P4	CD
F1	Needs material available on site	0	1	0	1	0
F2	Needs material not available on site	1	0	0	1	0
F3	Allows change of material price	A	A	A	A	A
F4	Delay effect as result of change order (in days)	26	152	11	0	21
F5	Needs material in bill of quantities	1	1	0	0	1
F6	Needs material not in bill of quantities	0	1	0	1	1
F7	Needs local material	1	1	0	0	1
F8	Needs imported material	0	1	1	1	0
F9	Affected by new legislations and ordinances	1	0	0	0	0
F10	Carried out using available work force	0	0	1	1	0
F11	Carried out using additional work force	1	0	1	0	0
F12	Needs additional equipment	1	0	0	0	1
	Case Output	P	P	P	D	-

Table 4.3: Mapping of Factors in Test Case 3

Factor	Nature and Classification	P1	P2	P3	P4	CD
F1	Needs material available on site	0	1	0	0	0
F2	Needs material not available on site	1	0	1	1	1
F3	Allows change of material price	A	A	A	A	A
F4	Delay effect as result of change order (in days)	68	12	0	21	52

Table 4.3 (cont'd)

Factor	Nature and Classification	P1	P2	P3	P4	CD
F5	Needs material in bill of quantities	0	1	1	0	1
F6	Needs material not in bill of quantities	1	0	0	0	0
F7	Needs local material	0	0	1	0	1
F8	Needs imported material	1	1	0	0	0
F9	Affected by new legislations and ordinances	0	1	0	0	0
F10	Carried out using available work force	0	0	0	1	1
F11	Carried out using additional work force	0	0	0	0	0
F12	Needs additional equipment	0	0	0	1	0
Case Output		P	D	P	D	-

Table 4.4: Mapping of Factors in Test Case 4

Factor	Nature and Classification	P1	P2	P3	P4	CD
F1	Needs material available on site	1	1	0	1	0
F2	Needs material not available on site	1	0	1	1	1
F3	Allows change of material price	A	A	A	A	A
F4	Delay effect as result of change order (in days)	16	26	88	0	41
F5	Needs material in bill of quantities	0	1	1	0	1
F6	Needs material not in bill of quantities	1	0	0	1	0
F7	Needs local material	0	0	1	1	1
F8	Needs imported material	1	1	0	0	1
F9	Affected by new legislations and ordinances	1	0	0	0	0
F10	Carried out using available work force	0	0	0	1	1
F11	Carried out using additional work force	1	0	1	0	0
F12	Needs additional equipment	0	0	1	1	1
Case Output		P	P	P	D	-

Table 4.5: Mapping of Factors in Test Case 5

Factor	Nature and Classification	P1	P2	P3	P4	CD
F1	Needs material available on site	0	1	0	0	0
F2	Needs material not available on site	1	0	1	1	0
F3	Allows change of material price	A	A	A	A	A
F4	Delay effect as result of change order (in days)	0	32	28	67	71
F5	Needs material in bill of quantities	0	1	1	0	1
F6	Needs material not in bill of quantities	1	0	0	1	0
F7	Needs local material	0	0	0	0	0
F8	Needs imported material	1	1	1	0	1
F9	Affected by new legislations and ordinances	0	1	1	1	0
F10	Carried out using available work force	0	1	0	1	1
F11	Carried out using additional work force	1	0	1	1	1
F12	Needs additional equipment	0	1	0	1	1
Case Output		D	P	P	D	-

Test Case	Precedent	Factor Analysis				Case Classification	Plaintiff's Statement of Claim	Defendant's Statement of Defense	Plaintiff's Statement of Rebuttal
		A	B	C	D				
Case 1	P1	1	1	4	1	ABCD	State Point Expert, case P2 is in favor of plaintiff because of the following factors: 1- Delay effect as result of change order. 2- Carried out using additional work force.	There is no citable case for defense.	N/A
	P2	2	2	2	2	ABCD			
	P3	3	1	1	4	ABCD			
	P4	1	1	2	3	ABCD			
Case 2	P1	1	2	1	4	ABCD	State Point Expert, case P3 is in favor of plaintiff because of the following factors: 1- Delay effect as result of change order	Counter Case Expert, counter case P4 is more on point than P3 because it has common d-factor. 1- Delay effect as result of change order 2- Needs material not in bill of quantities Also, counter case P4 is more on point than CD because it has additional d-factors: 1- Needs material not in bill of quantities Factor Distinction Expert, the following factors can be distinguished between cited case and current case: 1- Needs material in bill of quantities 2- Needs local material 3- Needs imported material 4- Carried out using additional work force 5- Needs additional equipment Emphasize Distinction Expert, the following factors can be emphasized between cited case and current case: 1- Needs local material	Factor Rebuttal Expert, the following factors can be distinguished between counter case and current case : 1- Needs material on site Dimension Rebuttal Expert: 1- Counter case has delay less than current case Up-play Significance Expert, the following factors can be up-played 1- Needs material not in bill of quantities Weakness-Not Fatal Expert, absence of following factors is not fatal for case outcome. In precedent P2: 1- Carried out using additional work force 2- Needs material in bill of quantities 3- Needs local material In precedent P1: 1- Needs imported material 2- Carried out using additional work force 3- Needs local material 4- Needs additional equipment
	P2	2	2	1	2	ABCD	State Point Expert, case P3 is in favor of plaintiff because of the following factors : 1- Delay effect as result of change order		
	P3	1	0	1	6	ACD			
	P4	1	0	1	6	ACD			
Case 3	P1	2	0	1	4	ACD	State Point Expert, case P3 is in favor of plaintiff because of the following factors: 1- Needs material not available on site 2- Delay effect as result of change order	There is no citable case for defense.	N/A
	P2	1	1	2	3	ABCD			
	P3	2	2	1	0	ABC			
	P4	3	0	1	2	ACD			

Table 4.6A: MAS-COR Results

Test Case	Precedent	Factor Analysis A	B	C	D	Case Classification	Plaintiff's Statement of Claim	Defendant's Statement of Defense	Plaintiff's Statement of Rebuttal
Case 4	P1	3	0	2	6	ACD	Stand Point Expert, case P1 is in favor of plaintiff because of the following factors: 1- Needs material not available on site 2- Delay effect as result of change order 3- Needs imported material	Counter Case Expert, counter case P4 is more on point than P1 because it has common d-factors : 1- Needs material not available on site 2- Delay effect as result of change order 3- Needs local material 4- Carried out using additional work force 5- Needs additional equipment Also, counter case P4 is more on point than CD because it has additional d-factors : 1- Needs local material 2- Carried out using additional work force 3- Needs additional equipment Factor Distinction Expert, the following factors can be distinguished between cited case and current case : 1- Needs material in bill of quantities 2- Needs material not in bill of quantities 3- Needs local material 4- Carried out using available work force 5- Carried out using additional work force 6- Needs additional equipment Emphasize Distinction Expert, the following factors can be emphasized between cited case and current case : 1- Needs local material 2- Needs additional equipment	Factor Rebuttal Expert, the following factors can be distinguished between counter case and current case : 1- Needs material on site 2- Needs imported material Dimension Rebuttal Expert: 1- Counter case has delay less than current case Weakness-Not Fatal Expert, absence of following factors is not fatal for case outcome. In precedent P2: 1- Needs material not in bill of quantities 2- Carried out using available work force 3- Carried out using additional work force 4- Needs material in bill of quantities In precedent P3: 1- Needs material not in bill of quantities 2- Carried out using available work force 3- Needs material in bill of quantities 4- Needs local material 5- Needs additional equipment
	P2	2	1	3	2	ABCD			
	P3	2	3	2	1	ABCD			
	P4	3	2	2	2	ABCD			
Case 5	P1	3	0	1	4	ABCD	State Point Expert, case P2 is in favor of plaintiff because of the following factors : 1- Delay effect as result of change order 2- Needs imported material 3- Carried out using additional work force	There is no citable case for defense.	N/A
	P2	3	2	2	1	ABCD			
	P3	3	1	1	3	ABCD			
	P4	3	1	1	4	ABCD			

Table 4.6B: MAS-COR Results

In light of the above results, it is evident that MAS-COR has provided a full simulation to the legal proceedings based on adversarial precedent law through citing cases favorable for the plaintiff, citing counter cases favorable for the defendant, and highlighting the strengths and weaknesses for both cases. The outlined arguments present a crucial decision support system for contract administrators, attorneys, arbitrators, and mediators in putting hands on the most important facts in construction disputes and thus, issuing awards or decisions pertaining to the same. In fact, MAS-COR does not only successfully highlight legal arguments in construction disputes, but also it is evident that it can actually generate an award in relatively non complex cases as test cases 1, 3, and 5.

4.5 Summary and Conclusions

Multi-agent systems have impacted many construction-related applications including material management, engineering design, supply chain management, scheduling and control, and negotiations in dispute resolutions. In the previous chapter, the conceptual framework for 11 stage logical induction algorithm for decision support in construction claims and disputes was formulated. This chapter presented the actual implementation of the same towards the development of a multi agent system for construction dispute resolution (MAS-COR). In this regard: (1) an agent-based role model was developed to represent the said interrelated algorithms; (2) a communication protocol between the agents was envisioned to ensure proper and easy flow of information through the system; (3) system implementation to put the aforementioned model and protocol into an object-oriented program; and (4) testing and validating the developed system through syntactical debugging and number of test cases. The actual implementation of MAS-COR ensures reusability, scalability, and flexibility of logical induction as supportive tool in the area of construction dispute resolution. As a future development for the concepts of MAS-COR, it will be utilized for issuing awards in delay analysis dispute cases. It is conjectured that MAS-COR, with its current version and expected future development, will have a positive impact on the contracting parties, their projects, and the construction industry.

5.0 CONSTRUCTION RISK INSURANCE

5.1 Background Information

5.1.1 Bootstrapping

In finance, bootstrapping is a statistical technique that is applied in cases of scarcity of data for imposing correlation to generate large set data that is statistically identical to smaller one (Harte *et al* 2006). One of the important theories in this field is that of Iman and Conover (1982) which preserves the original distributional structure of each data series.

The chosen Iman and Conover (1982) procedure for inducing correlation has four attractive properties: (1) the procedure works well with any distribution function compared to other correlation techniques are aimed directly at standard distribution functions and cannot be used with other distribution functions; (2) the mathematics behind the procedure is not extremely complex and Cholesky factorization and inversion of matrices are the most exotic steps in the procedure; (3) the procedure can be used under any sampling scheme; and (4) the marginal distributions of interest are maintained throughout the procedure and the moments of the marginal distributions are not affected at all. The procedure is based on rank correlations, however; Iman and Conover (1982) point out that raw correlation numbers can be misleading when the underlying data is non-normal or contains outliers.

The theoretical basis for the method is that given a random matrix A whose columns have a correlation matrix I (the identity matrix) and a desired correlation matrix B, there exists a transformation matrix C such that the columns of AC' (where C' is the transpose of C) have a correlation matrix B. Since B is positive definite and symmetric, there exists a lower triangular matrix (the transformation matrix) C such that $B = CC'$. In this regard, Let X be a matrix of draws of marginal distributions of interest. Let R be a matrix of the same size that contains what Iman and Conover call "scores". In this regard, Iman and Conover suggest using ranks, random normal deviates, or Van Der Waerden scores (i.e. $\theta^{-1}(i / N+1)$ where θ^{-1} is the inverse of the standard normal distribution function, N is the number of draws, and i = 1, ..., N) as possible scores. Let T be the target rank correlation matrix for a transformation of the columns of X. Since T is positive definite and symmetric, there exists a lower triangular matrix P such that $T = PP'$. P can be found by Cholesky factorization. The transformed score matrix is $R* = RP'$. The columns of R* have a rank correlation matrix M, which is close to

the target rank correlation matrix T. When the elements of X are arranged in the same ranking as in R*, then the columns of the transformed X matrix will also have a rank correlation matrix equal to M, close to T. Iman and Conover method was successfully modeled in Microsoft Excel to induce a target correlation between as much as 70 different variables and generate as much as needed data that maintains the very same correlation. The developed model was revised by Dr. Dermot Hayes, Professor of Economics and Finance at Iowa State University, who has extensively conducted research using Iman and Conover (1982) and he praised the robustness and accuracy of the developed model.

In this regard, this chapter aims to develop a new method for addressing the issue of risks in the construction industry. Under this work, the possibility of insuring against construction risks that are not only beyond the control of contractors but are also not covered under surety policies are explored. In this regard, single and bundled insurance against construction such construction risks were created to evaluate the efficiency of both approaches and to decide upon the most feasible and economic choice of insurance.

5.2 Methodology

The methodology utilized for this study is function of the availability of the required data. Accordingly, a pilot study where contacts were initiated with various contractors and construction surety companies was set in order to collect detailed data about occurrence of risks in construction projects as well as the associated US$ amounts with such risks. Pursuant to this pilot study, it was concluded that scarcity of data will be the main challenge for this study. Consequently, this fact was an integral part of choosing the suitable methodology for this research project.

5.2.1 Overview

In order to meet the goals and objectives of this chapter, available bootstrapping technique in cases of scarcity of data were explored. Thus, the Iman and Conover method (1982) for inducing correlations was modeled in Microsoft Excel. Moreover, a method for pricing insurance premiums as an exotic option using the Monte Carlo method was determined. Consequently, five risks that are beyond the control of contractors, not covered by surety risk transfer policies, and negatively affect the financial and economic standing the contractors resulting in construction conflicts, claims, and disputes were identified. Lastly,

financial data were collected regarding the defined risks and applied the aforementioned bootstrapping and pricing techniques to calculate the indemnity rates for single and portfolio insurance of these risks.

5.2.2 Risks for Construction Projects

In order to define construction risks that are beyond the control of contractors, not covered by surety risk transfer policies, and negatively affect the financial and economic standing the contractors resulting in construction conflicts, claims, and disputes, the categorization of risks developed by Brown (2004) was utilized. Under the said categorization, five risks namely site, economic, political, design, and environmental risks were named. Table 5.1 highlights the sources of risks and their categorization under this study.

Table 5.1: Categorization and Sources of Risks

Source of Risk				
Site	**Economic**	**Political**	**Design**	**Environmental**
Floods, Earthquakes, Windstorms, etc	Economic Fluctuations	Social Legislation	Design Errors	Site Contamination
Differing Conditions	Embargoes	U.S. Congress Appropriations		Perceived Environmental Impact of Project
Expansive or Corrosive Soils		Federal Transit Administration		Long-term Degradation of Environment
Unexpected Utilities		War, Insurrection & Other Hostilities.		Toxic Spills
Archaeological Finds		Sabotage/Terrorism		Environmental Legislation
Sight-line Conflicts				

5.2.3 Design of Construction Risks Insurance Policy

In light of the aforementioned scientific principles, the new construction risks insurance policy is designed as follows:

1) In order to avoid the issue of adverse selection, continuous care should be given so that the policy is not only insuring contractors whose projects have witnessed risks but should extended to contractors who were parts to projects with no, minimal, or fewer risks;

2) The problem of moral hazards is not of concern under this policy because the insured risks are totally outside the control of the beneficiary (i.e. contractors);

3) The starting premium paid to insure against construction risks will be based upon past experiences with the predefined risks in a set of at least 3000 projects. Under all these projects, the insurer should be interested in the $ US associated with each risk that is certified by the architect, engineer, or project manager.

4) The fair indemnity rate (i.e. premium) paid to insure against each type of risk will be calculated as the fair rate for a put exotic option using the Monte Carlo method. The calculated fair rate can be adjusted by an additional 30% to 50% to account for cost of issuing and administering the policy, adjusting losses, and supplying the capital needed to reasonably assure that the group will be able to pay claims (Brocket *et al* 2006).

The required premium for each risk will be calculated and compared with the fair indemnity for insuring against all risks in one portfolio.

5.2.4 Date Collection

Contacts were initiated with the major construction surety companies in order to get detailed financial and accounting records about set of construction projects. In this regard, a California based surety company provided complete data set of five different construction projects that cover the spectrum of small, medium, large, and mega projects.

5.3 Results and Analysis

5.3.1 Quantification of Risks

Pursuant to the categorizations highlighted under Table 5.1, the $ US amount associated with the each of the predefined risks was calculated from the available dataset as shown in Table 5.2.

Table 5.2: Quantification of Risks in Investigated Projects

Project	Risks				
	Site ($ US)	Economic ($ US)	Political ($ US)	Design ($ US)	Environmental ($ US)
A	0	654,287	131,256	12,427,243	1,422,206
B	4,731,828	0	0	17,824,371	872,706
C	7,427,835	1,424,729	0	2,841,843	0
D	4,269,859	0	0	13,271,548	4,327,431
E	13,712,319	2,412,671	1,549,421	92,634	30,184,008

It is evident that five projects represent too little data to determine the fair premium of an insurance policy. However, in these instances, insurers utilize bootstrapping techniques (Hart *et al* 2006).

5.3.2 Bootstrapping

Correlation between site, economic, political, design, and environmental risk was calculated in Microsoft Excel and correlation matrix X was developed hereunder:

$$X = \begin{bmatrix} 1.000 & 0.797 & 0.808 & -0.770 & 0.826 \\ 0.797 & 1.000 & 0.825 & -0.950 & 0.768 \\ 0.808 & 0.825 & 1.000 & -0.678 & 0.988 \\ -0.77 & -0.95 & -0.67 & 1.000 & -0.64 \\ 0.826 & 0.768 & 0.988 & -0.645 & 1.000 \end{bmatrix}$$

The correlation matrix X was utilized by the modeled Microsoft Excel spreadsheet to produce the lower triangular matrix T shown here below:

$$T = \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.797 & 0.604 & 0.000 & 0.000 & 0.000 \\ 0.808 & 0.300 & 0.507 & 0.000 & 0.000 \\ -0.77 & -0.55 & 0.219 & 0.222 & 0.000 \\ 0.826 & 0.182 & 0.524 & -0.10 & 0.000 \end{bmatrix}$$

Afterwards, the Microsoft Excel spreadsheet continued to implement the Iman and Conover (1982) technique using the Cholesky decomposition. In this regard, a dataset composed of 5000 observations that carries the same statistical characteristics including correlation and probabilistic distribution was developed. Also, and in order to assure the

exactness of the method, Table 5.3 shows that the correlation over the new generated data is mostly the same as the correlation values over the original data.

Table 5.3: Comparison between Correlation Values in Original and Developed Dataset

Data	Risks				
	Site	Economic	Political	Design	Environmental
Original	1	0.79685	0.807982	-0.77025	0.825669
New	1	0.79254	0.806435	-0.76576	0.825681

5.3.3 Descriptive Statistics

The descriptive statistics for the newly generated set of site, economic, political, design, and environmental risks are presented in Table 5.4. The statistics included therein would demonstrate that the generated data is properly dispersed, which helps in putting hands on the value at risk (VAR) represented by the tail of the distribution. VAR is one of the important elements when pricing an insurance policy, as it provides a measure of the minimum loss that would be expected over a period of time for the given probability distribution (Chance and Brooks 2007). Also, the high value of the standard deviation over all types of risk is expected as a result of the high dispersion of the $ US amounts associated with those risks.

Table 5.4: Descriptive Statistics for the New Generated Dataset

Indicator	Risks				
	Site	Economic	Political	Design	Environmental
Mean	5,932,044.47	904,534.42	334,665.14	9,363,159.24	7,297,646.07
Median	5,926,554.91	910,578.18	325,336.82	9,302,272.95	7,253,089.44
Standard Deviation	5,010,144.71	1,036,660.02	674,538.54	7,427,294.81	12,766,211.80
Kurtosis	0.01	0.12	-0.02	-0.12	0.03
Skewness	0.03	-0.05	0.01	-0.03	0.02
Count	5,000	5,000	5,000	5,000	5,000

Moreover, and in order to decide upon the optimal way to price the indemnity rates according to the option theory principles using the Monte Carlo method, it was essential to fit the risks under investigation into probabilistic distributions. In this regard, Figures 5.1 highlights that site risks fit into normal probabilistic distribution.
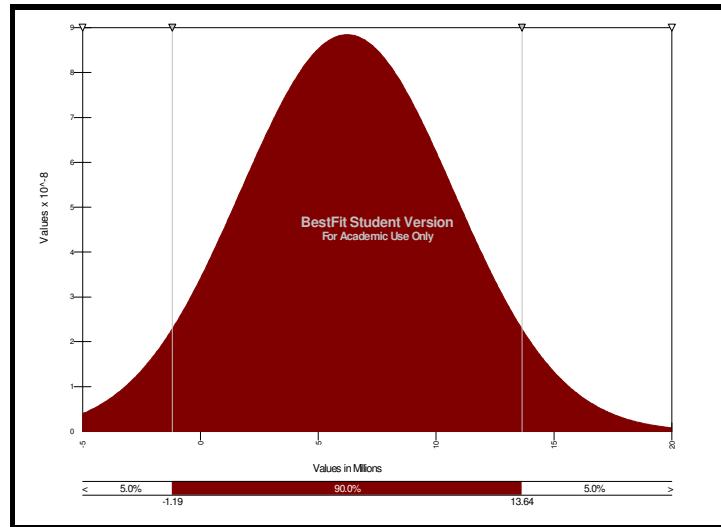
Figure 5.1: Normal Distribution of Site Risks

Likewise, a similar analysis would show the economic, political, design, and environmental risks follow as well the same normal distribution. This would accord with the conjecture that in most cases the occurrence of risks happens pursuant to a normal probability distribution (Paulson 2007). This would confirm that the premium for insuring construction claims risks in the said projects cannot be priced using the closed solution outlined by Black and Scholes (1973). Instead, the indemnity rate will be calculated using the unclosed solution outlined by Samuelson (1965) for exotic options. This unclosed solution will be modeled using the principles of Boyle (1977) for utilizing the Monte Carlo method for pricing exotic options.

5.3.4 Premium Calculations

Table 5.5 shows the calculated rates for different policy coverages. Fortunately, the calculated premiums are in line with the premiums available in market for different insurance policies. Thus, it is conjectured that insuring against construction risks would fit into the aforementioned 6 requirements for insurance. Now, an insurance coverage for each individual risk has been developed. However, it is important to investigate if there will be any benefit for contractors if they insured construction risk in a portfolio rather than on individual basis.

Table 5.5: Indemnity Rates

Coverage	Risk Indemnity in $ US/US $100 Based On Studied Pool				
	Site	**Economic**	**Political**	**Design**	**Environmental**
100%	5.648	0.910	0.410	8.836	8.366
95%	5.365	0.865	0.389	8.394	7.948
90%	5.083	0.819	0.369	7.953	7.529
85%	4.801	0.774	0.348	7.511	7.111

5.3.5 Single and Portfolio Insurance

Table 5.6 highlights the fair required premium for insuring against all risks in one portfolio compared with the premiums paid when insuring against each risk individually. The estimated premium for the proposed portfolio insurance product is well below: (1) the combined total of estimated premiums for insurance products covering each of the risk individually and (2) the estimated premium for even insuring against only one risk. This conclusion is sensible as it accords with the results outlined by Hennessy *et al* (1997) as well as Hart *et al* (2001 and 2006) when working with portfolio insurance. Portfolio risk insurance is more advantageous to contractors in both risk coverage and cost. In addition, it may provide higher coverage levels may be possible because of coverage diversification leading to lower risk and the limiting of potential moral hazard problems that occur with more specialized coverage.

Table 5.6: Single Vs Portfolio Insurance

Coverage	Risk Indemnity in $ US/US $100 Based On Studied Pool For All Risks	
	Single Insurance	**Portfolio Insurance**
100%	24.170	4.237
95%	22.962	4.026
90%	21.753	3.814
85%	20.545	3.602

5.4 Summary and Conclusions

This chapter developed a new method for addressing the issue of construction risks using principles of risk management. Under this study, the possibility of insuring against construction risks was explored. In this regard, single and portfolio insurance against such construction risks were created to evaluate the efficiency of both approaches and to decide upon the economically optimal choice of insurance. In order to meet the goals and objectives

of this study, bootstrapping technique in cases of scarcity of data were explored. Thus, the Iman and Conover method (1982) for inducing correlations was modeled in Microsoft Excel. Moreover, the method for pricing insurance premiums as an exotic option using the Monte Carlo method was utilized. Consequently, five risks that are beyond the control of contractors, not covered by surety risk transfer policies, and negatively affect the financial and economic standing the contractors resulting in construction conflicts, claims, and disputes were defined. Lastly, data regarding the defined risks was collected and applied the aforementioned bootstrapping and pricing techniques to calculate the indemnity rates for single and portfolio insurance of these risks.

Pursuant to this work, the generated 5000 point dataset has maintained all statistical parameters of the original data. This would confirm that the employed technique preserves the original distributional structure of the risks while imposing the desired correlation structure in transparent manner where the manipulation of the original data draws from the risk distributions is limited to a resorting of the draws. Also, the calculated premiums for insuring against each type of risk are in line with the premiums available in market for different insurance policies and thus, insuring against construction risks would fit into the requirements for insurance. Moreover, the estimated premium for the proposed portfolio insurance product is well below the combined total of estimated premiums for insurance products covering each of the risk individually and is more advantageous to contractors in both risk coverage and cost.

6.0 RISK RETENTION GROUP FOR CONTRACTORS CLAIMS

6.1 Background Information

6.1.1 Risk Retention Groups

Any risk management program is a combination of risk control and risk financing techniques. On one hand, risk control techniques, also commonly referred to as loss control techniques, are designed to reduce the frequency (i.e. loss prevention) and severity (i.e. loss reduction) of losses (Harrington and Niehaus 2004). On the other hand, risk financing involves arranging for funds to be available to pay losses that occur. The primary types of risk financing are risk retention and risk transfer. Losses can be retained through a number of methods including loss sensitive rating plans, qualified self-insurance, captive insurance subsidiaries, and risk retention groups (Harrington and Niehaus 2004).

The financial decision as to whether a business firm should self-insure a group of exposure units is a complicated process that depends on rigorous application of the utility theory (Brocket *et al* 1986). In this regard, the Liability Risk Retention Act was issued in October 1986 (Moore and Schmitt 1989).

A risk retention group (RRG) is a member-owned business association that is formed specifically for the purpose of pooling and sharing similar business risks (Moore and Schmit 1989). RRG must be licensed in at least one state or the District of Columbia, but once licensed, RRG are allowed under the federal Liability Risk Retention Act of 1986 to underwrite the insurance risks of its members nationwide, including giving preferential rates, terms and conditions to groups seeking liability insurance coverage (Steele 2007). The members of an RRG must be engaged in the same or similar businesses, at least so far as the liability exposures are concerned (Moore and Schmit 1989). RRG are usually established to cover risks that insurance companies are hesitant to insure due to lack of data or sverity of losses. A key benefit to the use of RRGs is that each policyholder has substantial incentive to engage in proactive risk management to try to avoid claims because each policyholder is also a member who participates in profits (Adkisson 2006). Also, the members can adopt better loss-control and more quickly identify other members whose risk-management is not efficient (Moore and Schmit 1989). Indeed, it is the fact that the members of an RRG know their business better than anybody else that often gives the RRG an underwriting edge over

insurance companies (Moore and Schmit 1989 and Adkisson 2006). As a result of such benefits, the number of risk retention groups has increased from 54 in 1988 to 253 in 2007 (Risk Retention Reporter 2007).

6.1.2 Construction Industry and Risk Retention Groups

Realizing the aforementioned characteristics of construction claims, it is evident that they fit into the first five principles of insurable products that were listed in Chapter 4. Nevertheless, none of the previous researchers has explored the idea of claims insurance. Along the same line, the concept of risk retention groups have been utilized in the construction industry through few groups that focus on issues like on some construction activities such as carpentry, door and window installation, masonry, plumbing, plastering, drywall and insulation, electrical works, concrete, heating and air conditioning, painting and paperhanging, floor covering, and others (Risk Retention Reporter 2007). However, none of the developed groups has focused on the alerting problem of construction claims and disputes that force many contractors to go out of business.

6.2 Methodology

6.2.1 Overview

In order to meet the goals and objectives of this study, the principles discussed in Chapter 5 for pricing insurance premiums using the options pricing theory and modeling the sophisticated exotic options pricing theory using the Monte Carlo method were utilized. Consequently, the principles required for optimal design of a risk retention group for construction claims were set up. Furthermore, the newly developed concept was applied on a set of 10,193 construction projects spanning over 12 different California districts.

6.2.2 Design of Contractors' Claims Risk Retention Group

In light of the aforementioned scientific principles of risk retention groups, the new contractors' claims risk retention group was designed as follows:

1) In order to avoid the issue of adverse selection, the risk retention group should not be composed only of contractors who engage in projects that witnessed constructions claims but should extend to contractors who were parts to projects with no disputes;

2) In order to avoid the problem of moral hazards, the risk retention group will have adjusters with rigorous experience in the matter of construction claims and disputes

who will evaluate each submitted claim on a 0 to 1 evaluation scale. Thus, the insured contractor will be paid in case of claim the percentage determined by the adjuster. Also, the collected premiums will be based on an adjusting factor equal to the mean of evaluation (i.e. 0.5);

3) The premium paid by the members of the group will be based on their previous experiences with claims in a set of at least 3000 completed projects. Under all these projects, the risk retention group would look for the project's contract amount, initial claimed amount by the contractor (if any), and settled claimed amount by negotiation, litigation, arbitration, or any ADR technique. It is worth noting that for simplicity of this newly created approach, the time for settlement of the dispute will be ignored.

4) The fair indemnity rate (i.e. premium) paid by the groups' members will be calculated as the fair rate for a put exotic option using the Monte Carlo method. The calculated fair rate will be adjusted by an additional 30% to 50% to account for cost of issuing and administering the policy, adjusting losses, and supplying the capital needed to reasonably assure that the group will be able to pay claims (Brocket *et al* 2006). However and again for the simplicity of this newly developed approach, the fair premium will not be adjusted to have the worst possible scenario.

5) The sufficiency of the collected premiums will be judged through comparison with the payouts that the risk retention group will incur during a specific period of time.

6.2.2 Data Collection

Contacts were initiated with various organizations that might have a database of at least 3000 projects project's contract amount, initial claimed amount by the contractor (if any), and settled claimed amount by negotiation, litigation, arbitration, or any ADR technique. The contacted institutions and companies included most of the nation's departments of transportation (DOTs), American arbitration association (AAA), and most of the top 400 companies listed in the engineering new record (ENR). In this regard, California department of transportation (CALTRANS) has willingly assisted in this matter. In this regard, CALTRANS provided a dataset that included 10,193 projects spanning over 12 California districts. The collected data was utilized in studying the newly developed model for the risk retention group of construction claims.

6.3 Results and Analysis

6.3.1 Overview

Despite the extreme lack of data faced under this study, the collected dataset was ideal for the intended purpose as a result of the following reasons: (1) the dataset included 10,193 projects which is 3.5 times more than the minimum required historic data outlined by Raviv (1979) and (2) the dataset confined 5588 project (i.e. 54.82 % of total projects) that did not witness any claims and disputes and thus, eliminating the problem of adverse selection. This would provide more confidence in the outcomes generated out of applying the concept of risk retention group on the CALTRANS dataset.

6.3.2 Descriptive Statistics

Tables 6.1 shows the descriptive statistics for the contract amounts, claimed amounts, and settled amounts under the investigated projects. The statistics included therein would demonstrate that the data is properly dispersed to cover wide spectrum of values in connection with contract amounts, claimed amounts, and settled amounts. This would insure that the collected dataset captures most if not all of the characteristics of construction projects.

Table 6.1: Descriptive Statistics for Contract, Claimed, and Settled Amounts in Dataset

Indicator	Contract Amount	Claimed Amounts	Settled Amounts
Mean	3,431,297.72	238,913.15	37,549.80
Median	642,000.00	0.00	0.00
Mode	500,000.00	0.00	0.00
Standard Deviation	23,067,128.02	2,522,487.71	503,714.51
Range	1,647,608,500.00	158,999,472.90	41,000,000.00
Minimum	21,500.00	0.00	0.00
Maximum	1,647,630,000.00	158,999,472.90	41,000,000.00
Sum	34,975,217,663.58	2,435,241,783.88	382,745,074.57
Count	10,193	10,193	10,193

Moreover, and in order to decide upon the optimal way to price the indemnity rates according to the option theory principles using the Monte Carlo method, it was essential to statistically fit the claimed amounts and settled amounts in the said projects into probabilistic distributions. In this regard, Figure 6.1 highlights that claimed amounts follow an exponential

probabilistic distribution. Likewise, a similar analysis shows that the settled amounts as well follow the same exponential distribution. This would confirm that the premium for insuring construction claims risks in the said projects cannot be priced using the closed solution outlined by Black and Scholes (1973). Instead, the indemnity rate will be calculated using the unclosed solution outlined by Samuelson (1965) for exotic options. This unclosed solution will be modeled using the principles of Boyle (1977) for utilizing the Monte Carlo method for pricing exotic options.
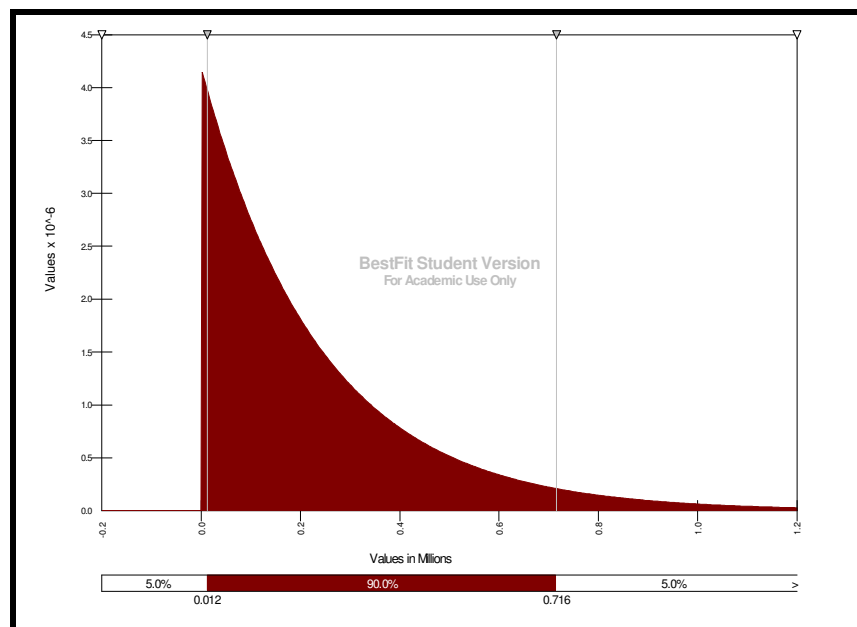
Figure 6.1: Exponential Distribution of Claimed Amounts in Dataset

6.3.3 Indemnity Rate Calculations

As the adjuster will evaluate any submitted claim on a 0 to 1 evaluation scale, the mean of his decision will be always 0.5. Thus, the risk retention group would anticipate that on average the payout of the insurance will be 0.5 of the submitted claims. In fact, this statistical contention is supported with the fact that over the dataset of 10,193 projects, on average the ratio between the settled claimed amounts and the original claimed amounts was 0.543. Thus, it is statistically and practically that the indemnity rate (i.e. premium) for the insurance policy would be based on a payout factor equals 0.5 of submitted claim. In this regard, Table 6.2 shows the calculated rates for different policy coverages. Fortunately, the calculated

premiums are in line with the premiums available in market for different insurance policies. Thus, it is conjectured that insuring against construction claims would fit into the 6 requirements for insurance outlined by Jaffee and Russell (1997) as well as Harrington and Niehaus (2004).

Table 6.2: Indemnity Rates

Coverage	Indemnity in $ US/US $100
100%	3.967
95%	3.769
90%	3.571
85%	3.372

6.3.4 Performance of Risk Retention Group

The performance of the risk retention group is judged by comparison between the collected premiums and the payouts incurred during a specific period of time in relation to sufficiency of funds. In this regard, it is worth noting this analysis assumes a worst case scenario for the group where the collected premiums are equal to the fair premiums without any adjusting amounts to account for cost of issuing and administering the policy, adjusting losses, and supplying the capital needed to reasonably assure that the group will be able to pay claims. This contention is acceptable especially that the developed group is composed of contractors whose primary aim is to protect themselves not benefit from each other. That said, Table 6.3 shows the collected premiums for the different coverages over the insured CALTRANS pool. It is clear that the collected amounts were sufficient to cover all claims incurred during the period. This would crystallize the efficient performance of the risk retention group. Also, the surplus monies could be invested by the group to provide the needed coverage during another period of time.

Table 6.3: Collected Premium Vs Payouts

Coverage	Collected Premiums ($ US)	Payouts ($ US)	Surplus ($ US)
100%	1,387,591,878.58	382,745,074.57	1,004,846,804.01
95%	1,318,212,284.65	382,745,074.57	935,467,210.08
90%	1,248,832,690.72	382,745,074.57	866,087,616.15
85%	1,179,453,096.79	382,745,074.57	796,708,022.22

On the other hand, and for explanation purposes, it is important to highlight that in case the collected premiums were not sufficient to cover incurred claims and this may be the case in other pools, the risk retention group had two possible alternatives to overcome such deficiency. On the one hand, the group could have multiplied the collected premium by a contingency factor to account for the deficient funding. On the other hand, the group could have utilized the concept of reinsurance where the group can protect itself against the risk of losses with other insurance companies providing the group with the benefits of risk transfer, income smoothing, surplus relief, and arbitrage Harrington and Niehaus (2004).

6.3.5 Benefits

Pursuant to the carefully designed concepts as well as the tested dataset, the newly developed risk retention group for construction claims has proved success from both, the insured and insurer sides. From the insured side, contractors are able to guarantee a method, that for only US $3.967 on every US $100 of contract amount, would provide a guarantee for an early financial and economic relief from claims and disputes that have already got many contractors out of business. On the other side, and from the insurer perspective, the risk retention group has ensured against the problem of moral hazards through the appointment of an experienced adjuster. Moreover, it has collected enough funds to pay the needed payouts in cases of losses (i.e. legitimate claims). Furthermore, the developed concept is wide enough to cover the problem of lack of sufficient funding, if any, using adjusting the fair premium with a contingency factor or using the reinsurance principle. That said, it is conjectured that insuring against construction claims through a contractors' risk retention group is a reality that could mitigate the negative effects of claims and disputes on contracting parties, executed projects, and the construction industry as a whole.

6.4 Summary and Conclusions

In this chapter, an innovative way for mitigating negative effects of contractor's construction claims and disputes using a risk retention approach was created. The developed risk retention group will function according to insurance principles and will confine set of contractors who in return for a certain paid premium would be indemnified against the risk of any legitimate construction claim. This would help contractors in getting an early relief from the financial and economic burden of construction claims. In this regard, the opportunity to

price insurance premiums using the warrant (i.e. option) pricing theory was innovatively investigated. Moreover, the sophisticated exotic options theory was modeled using the Monte Carlo method. Consequently, the principles required for optimal design of a risk retention group for construction claims were determined. Furthermore, the newly developed concept was applied on a set of 10,193 construction projects provided by CALTRANS. Pursuant to the carefully designed concepts as well as the tested dataset, it was verified that construction claims satisfy the required insurance principles. Moreover, the newly developed risk retention group for construction claims has proved success from both, the insured and insurer sides. It is conjectured that this study would lay basis for a leading risk management technique that could be extended over the nation for the benefit of relieving the negative consequences associated with lengthy claims and disputes resolution in the construction industry.

7.0 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This dissertation filled a research gap in the literature through developing an integrated and coherent methodology for mitigation of construction disputes using both, multi-agent based simulation concepts and risk management modeling principles. In this regard, the associated work carried out under this research has: (1) developed an innovative method that utilized logical induction as support tool for construction claims and disputes; (2) created a multi agent system for construction dispute resolution (MAS-COR) that simulates the legal discourse in construction disputes; (3) developed a new method for addressing the issue of risks in the construction industry using principle of portfolio insurance; and (4) created a leading way for mitigating negative effects of contractor's construction claims and disputes using a risk retention approach.

Under the first objective, an innovative method for studying construction claims and disputes using logical induction approach was developed. In this regard, (1) facts associated with construction change order cases were factorized into binary, dimensional, and abstract factors; (2) relevance was drawn between the developed factors and the disputing parties; (3) rules were generated as a result of the said factors; (4) factors were logically analyzed into distinct classifications; and (5) an 11 stage logical induction algorithm was mathematically modeled to put hands on similarities, differences, strengths, and weaknesses between current and precedent construction disputes. Also, and as a proof of concept, a detailed worked example where the developed approach was utilized to simulate the process of legal discourse based on adversarial precedent law where the developed algorithm will show, based on the relationship between the characteristics of dispute at hand and precedent ones, the best case to be cited by plaintiff, the best case to be cited by plaintiff, and possible attack for each of the two cases. The said example has presented set of legal arguments that are highly needed by contract administrators, claims advisors, and lawyers when evaluating current disputes in light of precedent cases. The great value of such arguments as well as their complexity should foster any effort towards the automation of utilizing logical induction approach for decision support in change order construction disputes.

Under the second objective, the previously formulated logical induction decision support algorithm was implemented towards the development of a multi agent system for construction dispute resolution (MAS-COR). In this regard: (1) an agent-based role model was developed to represent the said interrelated algorithm; (2) a communication protocol between the agents was envisaged to ensure proper and easy flow of information through the system; (3) system implementation to put the aforementioned model and protocol into an object-oriented program; and (4) testing and validating the developed system through syntactical debugging and number of test cases. The actual implementation of MAS-COR ensures reusability, scalability, and flexibility of logical induction as supportive tool in the area of construction dispute resolution.

Under the third objective, a new method for addressing the issue of construction risks using principles of risk management was developed. Thus, single and portfolio insurance against such construction risks were created to evaluate the efficiency of both approaches and to decide upon the economically optimal choice of insurance. In this regard, bootstrapping techniques in cases of scarcity of data were explored. Thus, the Iman and Conover method (1982) for inducing correlations was modeled in Microsoft Excel. Moreover, the method for pricing insurance premiums as an exotic option using the Monte Carlo method was utilized. Consequently, five risks that are beyond the control of contractors, not covered by surety risk transfer policies, and negatively affect the financial and economic standing the contractors resulting in construction conflicts, claims, and disputes were defined. Lastly, data regarding the defined risks was collected and applied the aforementioned bootstrapping and pricing techniques to calculate the indemnity rates for single and portfolio insurance of these risks. Pursuant to this work, the generated 5000 point dataset has maintained all statistical parameters of the original data. This would confirm that the employed technique preserves the original distributional structure of the risks while imposing the desired correlation structure in transparent manner where the manipulation of the original data draws from the risk distributions is limited to a resorting of the draws. Also, the calculated premiums for insuring against each type of risk are in line with the premiums available in market for different insurance policies and thus, insuring against construction risks would fit into the requirements for insurance. Moreover, the estimated premium for the proposed portfolio

insurance product is well below the combined total of estimated premiums for insurance products covering each of the risks individually and is more advantageous to contractors in both risk coverage and cost.

Under the fourth objective, an innovative way for mitigating negative effects of contractor's construction claims and disputes using a risk retention approach was created. The developed risk retention group will function according to insurance principles and will confine set of contractors who in return for a certain paid premium would be indemnified against the risk of any legitimate construction claim. This would help contractors in getting an early relief from the financial and economic burden of construction claims. In this regard, the opportunity to price insurance premiums using the warrant (i.e. option) pricing theory was innovatively investigated. Moreover, the sophisticated exotic options theory was modeled using the Monte Carlo method. Consequently, the principles required for optimal design of a risk retention group for construction claims were determined. Furthermore, the newly developed concept was applied on a set of 10,193 construction projects provided by CALTRANS. Pursuant to the carefully designed concepts as well as the tested dataset, it was verified that construction claims satisfy the required insurance principles. Moreover, the newly developed risk retention group for construction claims has proved success from both, the insured and insurer sides. It is conjectured that this study would lay basis for a leading risk management technique that could be extended over the nation for the benefit of relieving the negative consequences associated with lengthy claims and disputes resolution in the construction industry.

It is conjectured that accomplishing the aforementioned objectives would mitigate the negative effects of claims and disputes in the construction industry and thus, have a positive impact on the contracting parties, their projects, the construction industry as a whole, and consequently, the nation's economy.

7.2 Future Work

There are many opportunities for further developments for the work carried out under this research. In this regard, it is believed that the automation of factoring cases using natural language processing techniques would save lots of time that is consumed during the manual factorization. Furthermore, utilization of logical induction decision support for construction

claims should not be limited to change order disputes but should be extended to delay analysis disputes. In this regard, logical induction would provide actual awards rather than just providing support for contract administrators, attorneys, arbitrators, or mediators in issuing awards in disputes. Moreover, game theory, Bayesian-Nash equilibrium, and behavioral theories might be combined with logical induction to provide an integrated methodology for automated resolution of construction disputes. In addition, construction disputes can be mitigated through further risk management techniques including stochastic modeling of external risks as well as developing a vulnerability analysis for complex large-scale construction projects.

REFERENCES

Anumba, C. J., O. O. Ugwa, and Z. Ren (2005), *Agents and Multi-agent Systems in Construction*, Taylor & Francis, New York.

A.M, S. Ping Ho and Liang Y. Liu (2003), "How to Evaluate and Invest in Emerging A/E/C Technologies under Uncertainty", *Journal of Construction Engineering and Management*, Vol. 129 No. 1, pp. 16-24.

Allen, Mark, T. Bench-Capon, and G. Staniford (2000), 'A Multi agent Legal Argument Generator', *Proceedings of the 11th International Workshop on Database and Expert Systems Application,* pp. 10-1084

Aleven, V (1997), 'Teaching Case-Based Argumentation: Through a Model and Examples', *PhD Dissertation*, University of Pittsburgh, Pittsburg, PA, USA.

Arditi, David and Onur Tokdemir (1999), 'Comparison of Case-Based Reasoning and Artificial Neural Networks', *Journal of Computing in Civil Engineering,* Volume 13 No 3, pp. 162-169

Ashley, K. D. (1990). *Modeling Legal Argument: Reasoning with Cases and Hypotheticals*. Cambridge, *MA: MIT Press*, UK.

Ashley, K. D. and E. L. Rissland, (1988), 'A Case-Based Approach to Modeling Legal Expertise'. *IEEE Expert*, Volume 3 No 3, pp. 70-77.

Bench-Capon, T.J. M (1997), 'Arguing with Cases in Legal Knowledge Based Systems', *Proceedings of the 10th JUIRIX Conference*, pp. 85-100.

Bier, Vicki M. (2007), "Choosing What to Protect", *Journal of Risk Analysis*, Vol. 27, No. 3, pp. 607-620.

Boyle, Phelim (1977), "Options: A Monte Carlo Approach", *Journal of Financial Economics*, Volume 4, pp. 323-339.

Breuer, Michael (2005), "Multiple Losses, Ex Ante Moral Hazard, and the Implications for Umbrella", *Journal of Risk and Insurance*, Vol. 72, No. 4, pp. 525-538.

Brockett, Patrick L., Samuel H. Cox, Jr. and Robert C. Witt (1986), "Insurance Versus Self-Insurance: A Risk Management Perspective", *Journal of Risk and Insurance*, Vol. 53, 242-257.

Brown, Ernest C. (2004), *Bidding, Contracts, and Claims*, Ernest Brown & Company, California, USA.

Brüninghaus, S., and Ashley, K. D. (2001). 'The Role of Information Extraction for Textual CBR', *Proc. 4th International Conference on Case-Based Reasoning*, Springer-Verlag, Berlin, Germany, pp. 74-89.

Chance, Don and Robert Brooks (2007), *An Introduction to Derivatives and Risk Management*, Thomson Higher Education, Ohio, USA.

Cheeks, Richard J. (2003), 'Multistep Dispute Resolution in Design and Construction Industry'. *Journal of Professional Issues in Engineering Education and Practice*, volume 129, pp 84-90.

Chen, Jieh-Haur and S.C. Hsu, 'Hybrid ANN-CBR Model for Disputed Change Orders in Construction Projects', *Automation in Construction – Elsevier,* Volume 17, pp. 56-64.

Chiou, D and R.D. Logcher (1996), "Testing a Federation Architecture in Collaborative Design Process", *Report no. R96-01*.

Chiu, W. Henry (2000), "On the Propensity To Self-Protect", *Journal of Risk and Insurance,* 2000, Vol. 67, No. 4, pp. 555-578.

Chua, D. K. H D. Z. Li, and W. T. Chan (2001), 'Case-Based Reasoning Approach in Bid Decision Making', *Journal of Construction Engineering and Management*, Volume 127 No 1, pp. 35-45.

Chua, D. K. H and P. K. Loh (2006), 'CB-Contract: Case-Based Reasoning Approach to Construction Contract Strategy Formulation', *Journal of Computing in Civil Engineering*, Volume 20 No 5, pp. 339-350.

Coelho. Roberta de Souza, Uirá Kulesza, Arndt von Staa, Carlos José Pereira de Lucena (2006), Unit Testing in Multi-agent Systems using Mock Agents and Aspects, *Departamento de Informática*, Pontifícia Universidade Católica Do Rio De Janeiro, Rio De Janeiro, Brazil.

Del Cano, Alfredo and M. Pilar de la Cruz (2002), "Integrated Methodology for Project Risk Management", *Journal of Construction Engineering and Management*, Vol. 128, No. 6, pp. 473-485.

Denk H. and M. Schnellenbach-Held (2002), "Models for An Agent-based Public Tendering Procedure", *Proceedings of the 9th International EG-ICE Workshop Advances in Intelligent Computing in Engineering*, Darmstadt, Germany, pp. 40–48.

Doğan, Sevgi Zeynep, David Arditi, and H. Murat Günaydın (2006), 'Determining Attribute Weights in a CBR Model for Early Cost Prediction of Structural Systems', *Journal of Construction Engineering and Management*, Volume 132 No 10, pp. 1092-1098.

Doherty, Neil and Kent Smetters (2005), "Moral Hazard in Reinsurance Markets", *Journal of Risk and Insurance*, Vol. 72, No. 3, pp. 375-391.

Edmonds, B. (1998). *Social Embeddedness and Agent Development*. University of Manchester, UK.

El-adaway, Islam H. and A. Samer Ezeldin (2007), 'Dispute Review Boards: Expected Application on Egyptian Large Scale Construction Projects', *Journal of Professional Issues in engineering Education and Practices,* Volume 133 No 4, pp. 365-372.

El-Diraby, T. E. and K.F. Kashif (2005), 'Distributed Ontology Architecture for Knowledge Management in Highway Construction", *Journal of Construction Engineering and Management,* Volume 131 No 5, pp. 591-603.

Fenn, Peter (2002), "An Aetiological Approach to Construction Disputes", *Society of Construction Law*, UK.

Fenves, S., U. Flemming, C. Hendrickson, M.L. Maher, R. Quadrel, M. Terk and R. Woodbury (1996), *Concurrent Computer-Integrated Building Design*, Prentice-Hall, New Jersey, USA.

Finnerty, J. (1996), *Project Financing: Asset-Based Financial Engineering*, John Wiley & Sons, New York, USA.

Fischer, Black and Myron Scholes (1973), "The Pricing of Options and Corporate Liabilities"**,** *The Journal of Political Economy*, Vol. 81, No. 3, pp. 637-654.

Gatti , Maíra Athanázio de Cerqueira and Arndt von Staa (2206), "Testing & Debugging Multi-Agent Systems: A State of the Art Report", *Departamento de Informática* Pontifícia Universidade Católica Do Rio De Janeiro, Rio De Janeiro, Brazil.

Gilbert, N. and K. G. Troitzsch (1999). *Simulation for the Social Scientist*. University of Buckingham, Open University Press, UK.

Glasserman, Paul (2003), *Monte Carlo Method in Financial Engineering*, Springer Verlag, USA.

Gogol, Daniel (1993). "The Value of Information in Insurance Pricing", *Journal of Risk and Insurance*, Vol. 60 No. 1, pp. 119-128.

Groton, James P., Robert A. Rubin, and Bettina Quintas (2001), "A comparison of Dispute Review Boards and Adjudication", *The International Construction Law Review*, volume 18, pp 1-16.

Hage, J., (1993), 'Monological Reason-Based Logic', *Association of Computing Machinery*, pp 30-39.

Harmon, Kathleen M. J. (2003), 'Dispute Review Boards and Construction Conflicts: Attitudes and Opinions of Construction Industry Members', *PhD Dissertation*, Nova Southeastern University, Florida, USA.

Harrington, Scott and Gregory Niehaus (2004), *Risk Management and Insurance*, Mc-Graw Hill and Companies, New York, USA.

Hart, C. E., B. A. Babcock, and D. J. Hayes. 2001. "Livestock Revenue Insurance." *The Journal of Futures Markets,* Volume 21, pp. 553-580.

Hart, Chad E., Dermot J. Hayes, and Bruce A. Babcock (2006). "Insuring Eggs in Baskets: Should the Government Insure against Individual Risks" *Canadian Journal of Agricultural Economics,* Vol. 54 No. 1, pp. 121-137.

Hennessy, D. A., B. A. Babcock, and D. J. Hayes (1997), "The Budgetary and Producer Welfare Effects of Revenue Assurance." *American Journal of Agricultural Economic,* Volume 79, pp. 1024-34.

Iman, R. L., and W. J. Conover (1982),"A Distribution-Free Approach to Inducing Rank Correlation among Input Variables." *Communication in Statistics: Simulation and Computation* 11: 311-34.

Jaffee, Dwight and Thomas Russell (1997), "Catastrophe Insurance, Capital Markets, and Uninsurable Risks", *Journal of Risk and Insurance*, Vol. 62, No. 2, pp. 205-230.

Jannadi, Osama Ahmed and Salman Almishari (2003), "Risk Assessment in Construction", *Journal of Construction Engineering and Management*, Vol. 129, No. 5, pp. 492-500.

Janssen, Maarten C. W. and Vladimir A. Karamychev (2005), "Dynamic Insurance Contracts and Adverse Selection", *Journal of Risk and Insurance*, Vol. 72, No. 1, pp. 45-59.

Jergeas, George F. and Francis T. Hartman (1994), "Contractors Construction-Claims Avoidance", *Journal of Construction Engineering and Management*, Vol. 120, No. 3, pp. 553-560.

Klemetti, Ann (2006), "Risk Management in Construction Project Networks", *Report 2006/2, Laboratory of Industrial Management, Helsinki University of Technology*, Finland.

Kululanga, G. K., W. Koutcha, R. McCaffer, and F. Edum-Fotwe (2001), 'Construction Contractors Claim Process Framework', *Journal of Construction Engineering and Management*, Volume 127, pp 309-314.

Levin, Paul (1998), *Construction Contracts, Claims, and Disputes*, American society of Civil Engineers (ASCE), Reston, VA, USA.

Lee, Wondon and James A Ligon (2001), "Moral Hazard In Risk Pooling Arrangements", *Journal of Risk and Insurance*, Vol. 68 No 1, pp. 375-390.

Ling, Li, Robert Lee-Kong Tiong, Wong Wai Fan, and David Ah-Seng Chew (1999), "Risk Management In International Construction Joint Ventures", *Journal of Construction Engineering and Management*, Vol. 125 No. 4, 99-277-284.

Matyas, R. M., Mathews, A.A., Smith, R. J., and Sperry, P.E. (1995). *Construction Dispute Review Board Manual*. McGraw Hill Publishers, New York, New York.

Miller, R. and D. Lessard (2001), "Understanding And Managing Risks In Large Engineering Projects", *International Journal of Project Management,* Vol. 19, pp. 437-443.

Minato, Takayuki and David B. Ashley (1998), "Data-Driven of Corporate Risk Using Historical Cost Control Data", *Journal of Construction Engineering and Management*, Volume 124 No 1, pp. 42-47.

Mitropoulus, Panagiotis and Gregory Howell (2001), 'Model for Understanding, Preventing, and Resolving Project Conflicts', *Journal of Construction Engineering and Management*, volume 127, pp 223-231.

Moore, William T. and Joan T. Schmit (1989), "The Risk Retention Act of 1986: Effects on Insurance Firm Shareholders' Wealth", *Journal of Risk and Insurance*, Vol. 56, pp. 137-145

Moss, S. (2000), "Applications Centered Multi Agent Systems Design", *International Conference on Multi Agent Systems*, Boston MA, IEEE Computer Society

Ng, S. Thomas and Nigel J. Smith (1998), 'Validation and Verification of Case Based Prequalification System', *Journal of Computing in Civil Engineering*, Volume 12 No 4, pp. 215-226.

Oliveira, E. D., J.M. Fonseca, and A. Steiger-Garcao (1997), "MACIV: A DAI Based Resource Management System", *Journal of Applied Artificial Intelligence*, Volume 11. No 6, pp. 525–550.

Ozorhon, Beliz, Irem Dikmen, and M. Talat Birgonul (2006), 'Case-Based Reasoning Model for International Market Selection', *Journal of Construction Engineering and Management*, Volume 132 No 9, pp. 94-948.

Paulson, Nicholas (2007), *Risks and Uncertainty in Agriculture,* PhD Dissertation, Economics Department, Iowa State University.

Paek, James, Yong W. Lee, and Jong H. Ock (1993), "Pricing Construction Risk: Fuzzy Set Application", *Journal of Construction Engineering and Management*, Volume 119 No 4, pp. 743-756.

Peña-Mora, Feniosky and Wang Chun-Yi (1998), "Computer Supported Collaborative Negotiation Methodology", *Journal of Computing in Civil Engineering*, Volume 13 No 2, pp. 64-81.

Peña-Mora, F., C, Sosa,, and S. McCone, (2002), *Dispute Avoidance and Resolution in Large Scale Civil Engineering Systems*, MIT/Prentice Hall Civil and Environmental Engineering Textbook Series, USA.

Prakken. Henry and G. Sartor (1998), 'Modeling Reasoning with Precedents in a Formal Dialogue Game, *Journal of Artificial Intelligence and Law,* Volume 6 No 2, pp. 231-287.

Rahman, M. Motiar and Mohan M. Kumaraswamy (2004), "Potential for Implementing Relational Contracting and Joint Risk Management", *Journal of Management in Engineering,* Vol. 20, No. 4, pp. 178-189.

Raviv, Artur (1979), "The Design of an Optimal Insurance Policy", *American Economic Review*, Volume 69 No 1, pp. 84-96.

Ren, Z, C. J. Anumba, and O. O. Ugwa (2003), "Negotiation In multi-agent Systems for Construction Claim Negotiation", *International Journal of Applied Artificial Intelligence*, volume 16, No 5, pp 359-394.

Ren, Z, C. J. Anumba, and O. O. Ugwa (2001), 'Construction Claims Management: Towards An Agent Based Approach', *Journal of Engineering, Construction, and Architectural Management,* Volume 8, pp 185-197.

Risk Retention Reporter (2007), http://www.rrr.com, <Accessed December 1, 2007>.

Rojas, Eddy M. and Amlan Mukherjee (2006), "Multi-Agent Framework for General-Purpose Situational Simulations in the Construction Management Domain", *Journal of Computing in Civil Engineering,* Volume 20 No 3, pp. 165-176.

Rosdorf, William and Sivand Lakmazaheri (1990), 'Logic-Based Approach for Modeling Organization of Design Standards', *Journal of Computing in Civil Engineering,* Volume 4, No 2, pp. 102-123.

Russell, Jeffrey and Zhai Hauming (1996), "Predicting Contractors Failure Using Stochastic Dynamics of Economics and Financial Variables", *Journal of Construction Engineering and Management*, Vol. 122, No. 2, pp. 183-191.

Samuelson, Paul (1965), "Rational Theory of Warrant Pricing", *Industrial Management Review,* Volume 6 Number 2, pp.13-39.

Skalak, D. B. and E.L. Rissland (1992), 'Arguments and Cases: An Inevitable Intertwining', *Journal of Artificial Intelligence and Law*, Volume 11 No 1, pp. 3-42.

Schexnayder, Cliff, Sandra L. Weber, and Scott A. David (2004), "Transportation Agency Use of Owner-Controlled Insurance Programs". *Journal of Construction Engineering and Management*, Volume 130 No 4, pp. 517-524.

Smith, Gary R. and Caryn M. Bohn (1999), "Small To Medium Contractor Contingency and Assumption Of Risk", *Journal of Construction Engineering and Management*, Volume 125 No 2, pp. 101-106.

Tang, Wenzhe, Colin F. Duffield, and David M. Young (2006), "Partnering Mechanism in Construction: An Empirical Study on the Chinese Construction Industry", *Journal of Construction Engineering and Management*, Vol. 132, No. 3, pp. 217-229.

Taylor, R.I. (2003), "Agent-Based Modelling Incorporating Qualitative and Quantitative Methods: A Case Study Investigating the Impact of E-commerce upon the Value Chain" *Doctoral Thesis*, Manchester Metropolitan University, Manchester, UK.

Tesfatsion, Leigh, "Learning Algorithms: Illustrative Examples", *Lecture Notes: Multi-Agent Economics (Econ 308)*, Iowa State University.

Tsanakas, Andreas and Evangelia Desli (2005), "Measurement and Pricing of Risk in Insurance Markets", *Journal of Risk Analysis*, Vol. 25, No. 6, pp. 1653-1668.

Turner, J.R. (1999), *The Handbook Of Project-Based Management: Improving The Processes For Achieving Strategic Objectives*, McGraw-Hill, London, UK.

Vidal, Jose M. (2007), *Fundamentals of Multi-Agent Systems*, Department of Computer Science and Engineering, University of South Carolina, Columbia, USA.

Vlassis, Nikos (2003), *A Concise Introduction to Multi-Agent Systems and Distributed AI, Intelligent Autonomous Systems*, Informatics Institute, University of Amsterdam, Holland.

Vorster, M. C., (1993). *Dispute Prevention and Resolution: Alternative Dispute Resolution in Construction with Emphasis on Dispute Review Boards.* Austin, Texas, Construction Industry Institute.

APPENDIX A
Construction Industry Questionnaire

Dear Sir/Madam,

This is Islam El-adaway, PhD student in the department of civil, construction, and environmental engineering. I am working on my dissertation titled: "*Construction Dispute Mitigation through Muli-Agent Based Simulation and Risk Management Modeling*".

As part of my dissertation work, I need to set up a set of *factors* that the construction industry experts would regard as being the most important when determining a decision, award, or recommendation in two (2) types of claims and disputes is pertaining to construction projects. It is worth noting that this set of factors should be classified as being favorable for the contractor or the owner and should be categorized as being:

1) Binary, that can be present or not present in the case;
2) Dimensional, that can have a quantitative effect in the case;
3) Abstract, that can be implied for the case or the project proceedings.

In light of the above, I would kindly ask you to take 20 minutes of your precious time to complete the following industry survey. Your participation is really treasured as it is very important to compliment our efforts to mitigate the negative effects of construction disputes and positively affecting contracting parties, construction projects, and the construction industry as a whole.

Sincerely,
Islam El-adaway

P.S
1) All names and personal info are kept anonymous.
2) Should you need a copy of the final survey results, please indicate so at the end of the survey.

Section One:

Please complete in the blanks or underline the most appropriate under of the following:

1) Name: ………………………………………………………………………..

2) E-mail Address: ………………………………………………………………..

3) What is your profession in connection with construction industry?

    a) Engineer/Consultant/Expert Determinator/Quantity Surveyor/Project Manager;

    b) Contracts Administrator/Claims Advisor;

    c) Arbitrator/Adjudicator/Mediator;

    d) Lawyer;

4) Number of years in that profession:

    a) 0-5 years;

    b) 5-10 years;

    c) 10-15 years;

    d) 15-20 years;

    e) 20-25 years;

    f) 25+.

5) Were you involved or are you aware of construction disputes in the US and/or the UK?

    a) Yes (State number of times: --)

    b) No

6) Have you been a member of any arbitral tribunal for large-scale construction project?

    a) Yes (State number of times: --)

    b) No

7) Have you been a member of any ADR proceeding in large-scale construction project?

    a) Yes (State number of times: --)

    b) No

Section Two:

Please fill in the following table in relation to change order disputes.

Factor Name	Relevance		Categorization			
	For Contractor	For Owner	Binary		Dimensional	Abstract
			Present	Not Present		

Are you interested to have in the survey results? …………………………………………..

Thanks for your time and input that would hopefully have a positive effect on the construction industry.

Sincerely,

Islam El-adaway

APPENDIX B
Logical Induction and Set Theory for Legal Argument Generation

The general formulation for logical induction and set theory utilization for generation of arguments in change order construction disputes is given hereunder, under Appendix A.

1) State Point

PF is the set of pro-plaintiff factors; DF is the set of pro-defendant factors; and CD is the current dispute under investigation. The following equations show the same:

$$PF = \{ \ldots \} \tag{1}$$

$$DF = \{ \ldots \} \tag{2}$$

In the aforementioned cases, there are only 3 possible cases to cite because P3 was found for the defendant and thus, is not citable. Accordingly, finding the intersection of factors between the 3 cases and CD is as follows:

$$F1 = P1 \cap CD \tag{3}$$

$$F2 = P2 \cap CD \tag{4}$$

$$F4 = P4 \cap CD \tag{5}$$

We now check if any of the matching factors are p-factors by intersection between the F sets with PF as follows:

$$F1 \cap PF \tag{6}$$

$$F2 \cap PF \tag{7}$$

$$F4 \cap PF \tag{8}$$

However we want to know, in advance, the weaknesses and strengths (i.e. the 'C' and 'D' factors) which we can get by an 'XOR' operation ($\oplus$) as follows:

$$D1 = P1 \oplus CD \tag{9}$$

$$D4 = P4 \oplus CD \tag{10}$$

We now need to determine which of the factors in the D sets are strengths (i.e. 'C' factors) and which are weaknesses (i.e. 'D' factors). To be strengths the factors will be either p-factors in CD but not in the precedent or d-factors in the precedent but not in CD. If the set PF represents the p-factors and the set DF represents the d-factors, we can determine which

of the factors are p-factors and which are d-factors by intersection. This is carried out as follows:

$$DP1 = D1 \cap PF \tag{11}$$

$$DP4 = D4 \cap PF \tag{12}$$

The DP sets (i.e. DP1 and DP4) tell us the different p-factors between the precedent disputes and CD. They will be strengths if they intersect with CD and weaknesses if they intersect with the precedent case. This is clear as follows:

$$SP1 = DP1 \cap CD \tag{13}$$

$$SP4 = DP4 \cap CD \tag{14}$$

The SP sets (i.e. SP1 and SP2) show the extra p-factors that CD has over each case. To find the p-factor weaknesses, we intersect the DP sets with the precedent disputes as follows:

$$WP1 = DP1 \cap P1 \tag{15}$$

$$WP4 = DP4 \cap P4 \tag{16}$$

We now find the strengths and weaknesses of the d-factors by intersecting the D sets with DF as follows:

$$DD1 = D1 \cap DF \tag{17}$$

$$DD4 = D4 \cap DF \tag{18}$$

The intersection with the precedent case will give us the strengths and the intersection with CD the weaknesses as follows:

$$DS1 = DD1 \cap P1 \tag{19}$$

$$DS4 = DD4 \cap P4 \tag{20}$$

$$DW1 = DD1 \cap CD \tag{21}$$

$$DW4 = DD4 \cap CD \tag{22}$$

In order to combine the weaknesses and strengths, we can union the two sets as follows:

$$W1 = DW1 \cup WP1 \tag{23}$$

$$W4 = DW4 \cup WP4 \tag{24}$$

$$S1 = SP1 \cup DS1 \tag{25}$$

$$S4 = SP4 \cup DS4 \tag{26}$$

The aforementioned sets give the number of strengths and weaknesses between precedent cases and current disputes.

2) Counter Case

The counter case (CC) is the set of intersecting factors between P3 and CD as follows:

$$CC = P3 \cap CD \tag{27}$$

Now, we get the factors from CC that also matches the cited case P1 as follows:

$$U = CC \cap P \tag{28}$$

Then, we get the difference between the factors matching the counter case, CC, and the factors in the cited case but after removing common factors between the counter case and the cited case as follows:

$$D = CC - P \tag{29}$$

The D set gives us any extra matching factors between the counter case and CC. If this set is not empty then the counter case is 'more on point'.

3) Factor Distinction

An 'XOR' operation between the cited case and CD gives a set F that is the set of differences as follows:

$$F = P \oplus CD \tag{30}$$

To remove from F the p-factors in CD and d-factors in the cited case, we intersect F with P1 for p-factors and intersect F with CD and DF for d-factors. This is carried out as follows:

$$WP = F \cap P \cap PF \tag{31}$$

$$WD = F \cap CD \cap DF \tag{32}$$

WP and WD are sets of p-factors and d-factors absences respectively.

4) Emphasize Distinction

In our work, F3 is an abstract factor that is supported by factors F5 and F7, and militated against by factors F6, F8 and F9. This gives us the additional rules, R13 and R14. The factors involved are confined the sets SU (i.e. support) and MI (i.e. militate) as follows.

$$SU = \{... \} \tag{33}$$

$$MI = \{... \} \tag{34}$$

To show if a factor distinction can be emphasized, it must belong to the set SU and WD and there must not be a factor in CC that also belongs to SU. This is carried out for p–factors as follows:

$$SU \cap WP \tag{35}$$

If the set is empty, we cannot emphasize any factor. As for non empty sets, it is carried out as follows:

$$SD = SU \cap WD \tag{36}$$

$$SU \cap CC \tag{37}$$

5) Dimension Distinction

It is just a comparison between effect of delay as result of change order between cited case and current dispute.

6) Factor Rebuttal

An 'XOR' operation ($\oplus$) between the counter case and CD will give us the set of differences as follows:

$$R = P \oplus CD \tag{38}$$

An intersection of PF, CD and R will tell us if any of the factors in R are p-factors in CD and thus, strengths. Also, an intersection of DF, P3 and R will show if any of the factors in R are d-factors in P and therefore also strengths for CC. This is shown as follows:

$$DP = PF \cap CD \cap R \tag{39}$$

$$DD = DF \cap P3 \cap R \tag{40}$$

7) Dimension Rebuttal

This is similar to dimension distinction bit in an opposite sense. Thus, it is just a comparison between effect of delay as result of change order between counter case and current dispute.

8) Downplay Significance

Set of absent factors were shown in equations 31 and 32 and the sets of supporting and militating factors were shown in equations 33 and 34. Equation 41 shows the intersection of WP and SU that gives the set of p-factors in the cited case that support an abstract factor and equation 42 shows intersection of WD and M.

$$WP \cap SU \tag{41}$$

$$WD \cap MI \qquad\qquad (42)$$

Empty sets cannot be down-played.

9) Up-play strengths

The PF and DF factors are previously shown in equations 1 and 2. The intersection of CD and PF gives the set of p-factors in CD (i.e. C) and the intersection of P1 and DF gives the set of d-factors in P1 (i.e. D). These are shown as follows:

$$C = CD \cap PF \qquad\qquad (43)$$

$$D = P1 \cap DF \qquad\qquad (44)$$

An 'XOR' operation ($\oplus$) between C and P gives the p-factors that are not in both cases. This means they are either 'C' or 'D' factors. Equation 45 shows the same:

$$C1 = C \oplus P \qquad\qquad (45)$$

To create a set of 'C' factors, we intersect C1 with CD as follows:

$$C2 = C1 \cap CD \qquad\qquad (46)$$

To emphasize the result of equation 46, we need to look for a case where this result is an 'A' factor as follows:

$$P \cap C2 \qquad\qquad (47)$$

$$P4 \cap C2 \qquad\qquad (48)$$

10) Weakness Not Fatal

The set of absent factors WP was generated in equations 31. For the p-factors (WP), we need a case where they also did not apply. This is carried out as follows:

$$M1 = WP \cap P4 \qquad\qquad (49)$$

The resulting set would indicate the set of weakness-not fatal factors.

APPENDIX C
Source Code for MAS-COR

```java
/*
 * GUIAboutBox.java
 */

package gui;

import org.jdesktop.application.Action;

public class GUIAboutBox extends javax.swing.JDialog {

   public GUIAboutBox(java.awt.Frame parent) {
      super(parent);
      initComponents();
      getRootPane().setDefaultButton(closeButton);
   }

   @Action public void closeAboutBox() {
      setVisible(false);
   }

   /** This method is called from within the constructor to
    * initialize the form.
    * WARNING: Do NOT modify this code. The content of this method is
    * always regenerated by the Form Editor.
    */
   // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
   private void initComponents() {

      closeButton = new javax.swing.JButton();
      javax.swing.JLabel appTitleLabel = new javax.swing.JLabel();
      javax.swing.JLabel appDescLabel = new javax.swing.JLabel();
      javax.swing.JLabel imageLabel = new javax.swing.JLabel();
      jLabel1 = new javax.swing.JLabel();

      setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
      org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(gui.GUIApp.class).getContext().getResourc
eMap(GUIAboutBox.class);
      setTitle(resourceMap.getString("title")); // NOI18N
      setModal(true);
      setName("aboutBox"); // NOI18N
      setResizable(false);
```

```
        javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(gui.GUIApp.class).getContext().getAction
Map(GUIAboutBox.class, this);
        closeButton.setAction(actionMap.get("closeAboutBox")); // NOI18N
        closeButton.setName("closeButton"); // NOI18N

appTitleLabel.setFont(appTitleLabel.getFont().deriveFont(appTitleLabel.getFont().getStyle()
| java.awt.Font.BOLD, appTitleLabel.getFont().getSize()+4));
        appTitleLabel.setText(resourceMap.getString("Application.title")); // NOI18N
        appTitleLabel.setName("appTitleLabel"); // NOI18N

        appDescLabel.setFont(resourceMap.getFont("appDescLabel.font")); // NOI18N
        appDescLabel.setText(resourceMap.getString("appDescLabel.text")); // NOI18N
        appDescLabel.setName("appDescLabel"); // NOI18N

        imageLabel.setIcon(resourceMap.getIcon("imageLabel.icon")); // NOI18N
        imageLabel.setName("imageLabel"); // NOI18N

        jLabel1.setText(resourceMap.getString("jLabel1.text")); // NOI18N
        jLabel1.setName("jLabel1"); // NOI18N

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(imageLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 142,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addComponent(appTitleLabel)
                        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
284, javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addComponent(appDescLabel,
javax.swing.GroupLayout.PREFERRED_SIZE, 329,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
```

```
                .addGroup(layout.createSequentialGroup()

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .addComponent(closeButton)
                    .addContainerGap())))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addComponent(imageLabel,
javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
                    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
layout.createSequentialGroup()
                        .addComponent(appTitleLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(appDescLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE,
61, javax.swing.GroupLayout.PREFERRED_SIZE)))
                    .addContainerGap(19, Short.MAX_VALUE))
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
                    .addComponent(closeButton)
                    .addContainerGap())))
        );

        pack();
    }// </editor-fold>//GEN-END:initComponents

    // Variables declaration - do not modify//GEN-BEGIN:variables
    private javax.swing.JButton closeButton;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration//GEN-END:variables

}
```

```java
/*
 * GUIApp.java
 */

package gui;

import org.jdesktop.application.Application;
import org.jdesktop.application.SingleFrameApplication;

/**
 * The main class of the application.
 */
public class GUIApp extends SingleFrameApplication {

    /**
     * At startup create and show the main frame of the application.
     */
    @Override protected void startup() {
        show(new GUIView(this));
    }

    /**
     * This method is to initialize the specified window by injecting resources.
     * Windows shown in our application come fully initialized from the GUI
     * builder, so this additional configuration is not needed.
     */
    @Override protected void configureWindow(java.awt.Window root) {
    }

    /**
     * A convenient static getter for the application instance.
     * @return the instance of GUIApp
     */
    public static GUIApp getApplication() {
        return Application.getInstance(GUIApp.class);
    }

    /**
     * Main method launching the application.
     */
    public static void main(String[] args) {
        launch(GUIApp.class, args);
    }
}
```

```java
/*
 * GUIView.java
 */

package gui;

import courtroom.FactorsCase;
import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.BufferedReader;
import java.util.ArrayList;
import java.util.List;
import javax.swing.Timer;
import javax.swing.Icon;
import javax.swing.JDialog;
import javax.swing.JFrame;
import courtroom.*;
import javax.swing.JOptionPane;
/**
 * The application's main frame.
 */
public class GUIView extends FrameView {


    private  int caseCount=0;
    /* enterCurrentCase tell us that window is first time started
     * and we should enter current case factors
     * when we click next enterCurremtCase goes false
     * and that means we are entering precedans
     * radio buttons shiows.
     */

    private  boolean enterCurrentCase = true;
    private FactorsCase currentCase;
    private  List<FactorsCase> caseList = new ArrayList<FactorsCase>();
    private OutputDialog dlg;
    public GUIView(SingleFrameApplication app) {
```

```
    super(app);

    initComponents();
    jSpinner1.setEnabled(false);
    plaintiffLose.setVisible(false);
    JFrame mainFrame = GUIApp.getApplication().getMainFrame();
    dlg = new OutputDialog(mainFrame,true);

    plaintiffWon.setVisible(false);

    // status bar initialization - message timeout, idle icon and busy animation, etc
    ResourceMap resourceMap = getResourceMap();
    int messageTimeout = resourceMap.getInteger("StatusBar.messageTimeout");
    messageTimer = new Timer(messageTimeout, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            statusMessageLabel.setText("");
        }
    });
    messageTimer.setRepeats(false);
    int busyAnimationRate = resourceMap.getInteger("StatusBar.busyAnimationRate");
    for (int i = 0; i < busyIcons.length; i++) {
        busyIcons[i] = resourceMap.getIcon("StatusBar.busyIcons[" + i + "]");
    }
    busyIconTimer = new Timer(busyAnimationRate, new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            busyIconIndex = (busyIconIndex + 1) % busyIcons.length;
            statusAnimationLabel.setIcon(busyIcons[busyIconIndex]);
        }
    });
    idleIcon = resourceMap.getIcon("StatusBar.idleIcon");
    statusAnimationLabel.setIcon(idleIcon);
    progressBar.setVisible(false);

    // connecting action tasks to status bar via TaskMonitor
    TaskMonitor taskMonitor = new TaskMonitor(getApplication().getContext());
    taskMonitor.addPropertyChangeListener(new java.beans.PropertyChangeListener() {
        public void propertyChange(java.beans.PropertyChangeEvent evt) {
            String propertyName = evt.getPropertyName();
            if ("started".equals(propertyName)) {
                if (!busyIconTimer.isRunning()) {
                    statusAnimationLabel.setIcon(busyIcons[0]);
                    busyIconIndex = 0;
                    busyIconTimer.start();
                }
                progressBar.setVisible(true);
```

```
                progressBar.setIndeterminate(true);
            } else if ("done".equals(propertyName)) {
                busyIconTimer.stop();
                statusAnimationLabel.setIcon(idleIcon);
                progressBar.setVisible(false);
                progressBar.setValue(0);
            } else if ("message".equals(propertyName)) {
                String text = (String)(evt.getNewValue());
                statusMessageLabel.setText((text == null) ? "" : text);
                messageTimer.restart();
            } else if ("progress".equals(propertyName)) {
                int value = (Integer)(evt.getNewValue());
                progressBar.setVisible(true);
                progressBar.setIndeterminate(false);
                progressBar.setValue(value);
            }
        }
    });
}


    @Action
    public void showAboutBox() {
        if (aboutBox == null) {
            JFrame mainFrame = GUIApp.getApplication().getMainFrame();
            aboutBox = new GUIAboutBox(mainFrame);
            aboutBox.setLocationRelativeTo(mainFrame);
        }
        GUIApp.getApplication().show(aboutBox);
    }


    /** This method is called from within the constructor to
     * initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is
     * always regenerated by the Form Editor.
     */
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-
BEGIN:initComponents
    private void initComponents() {

        mainPanel = new javax.swing.JPanel();
        textName = new javax.swing.JTextField();
        jLabel1 = new javax.swing.JLabel();
        jCheckBox1 = new javax.swing.JCheckBox();
        jCheckBox2 = new javax.swing.JCheckBox();
        jCheckBox3 = new javax.swing.JCheckBox();
```

```
jCheckBox4 = new javax.swing.JCheckBox();
jSpinner1 = new javax.swing.JSpinner();
jCheckBox5 = new javax.swing.JCheckBox();
jCheckBox6 = new javax.swing.JCheckBox();
jCheckBox7 = new javax.swing.JCheckBox();
jCheckBox9 = new javax.swing.JCheckBox();
jCheckBox10 = new javax.swing.JCheckBox();
jCheckBox11 = new javax.swing.JCheckBox();
jCheckBox12 = new javax.swing.JCheckBox();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();
jCheckBox8 = new javax.swing.JCheckBox();
plaintiffWon = new javax.swing.JRadioButton();
plaintiffLose = new javax.swing.JRadioButton();
jLabel2 = new javax.swing.JLabel();
menuBar = new javax.swing.JMenuBar();
javax.swing.JMenu fileMenu = new javax.swing.JMenu();
javax.swing.JMenuItem exitMenuItem = new javax.swing.JMenuItem();
javax.swing.JMenu helpMenu = new javax.swing.JMenu();
javax.swing.JMenuItem aboutMenuItem = new javax.swing.JMenuItem();
statusPanel = new javax.swing.JPanel();
javax.swing.JSeparator statusPanelSeparator = new javax.swing.JSeparator();
statusMessageLabel = new javax.swing.JLabel();
statusAnimationLabel = new javax.swing.JLabel();
progressBar = new javax.swing.JProgressBar();
buttonGroup1 = new javax.swing.ButtonGroup();

mainPanel.setEnabled(false);
mainPanel.setName("mainPanel"); // NOI18N

textName.setHorizontalAlignment(javax.swing.JTextField.RIGHT);
org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(gui.GUIApp.class).getContext().getResourc
eMap(GUIView.class);
textName.setText(resourceMap.getString("textName.text")); // NOI18N
textName.setToolTipText(resourceMap.getString("textName.toolTipText")); // NOI18N
textName.setName("textName"); // NOI18N

jLabel1.setText(resourceMap.getString("jLabel1.text")); // NOI18N
jLabel1.setName("jLabel1"); // NOI18N

jCheckBox1.setText(resourceMap.getString("checkF1.text")); // NOI18N
jCheckBox1.setName("checkF1"); // NOI18N

jCheckBox2.setText(resourceMap.getString("checkF2.text")); // NOI18N
```

```java
    jCheckBox2.setName("checkF2"); // NOI18N

    jCheckBox3.setText(resourceMap.getString("checkF3.text")); // NOI18N
    jCheckBox3.setActionCommand(resourceMap.getString("checkF3.actionCommand"));
// NOI18N
    jCheckBox3.setName("checkF3"); // NOI18N

    jCheckBox4.setText(resourceMap.getString("checkF4.text")); // NOI18N
    jCheckBox4.setName("checkF4"); // NOI18N
    jCheckBox4.addChangeListener(new javax.swing.event.ChangeListener() {
       public void stateChanged(javax.swing.event.ChangeEvent evt) {
          jCheckBox4StateChanged(evt);
       }
    });

    jSpinner1.setName("spinDays"); // NOI18N

    jCheckBox5.setText(resourceMap.getString("checkF5.text")); // NOI18N
    jCheckBox5.setName("checkF5"); // NOI18N

    jCheckBox6.setText(resourceMap.getString("checkF6.text")); // NOI18N
    jCheckBox6.setName("checkF6"); // NOI18N

    jCheckBox7.setText(resourceMap.getString("checkF7.text")); // NOI18N
    jCheckBox7.setName("checkF7"); // NOI18N

    jCheckBox9.setText(resourceMap.getString("checkF9.text")); // NOI18N
    jCheckBox9.setName("checkF9"); // NOI18N

    jCheckBox10.setText(resourceMap.getString("checkF10.text")); // NOI18N
    jCheckBox10.setName("checkF10"); // NOI18N

    jCheckBox11.setText(resourceMap.getString("checkF11.text")); // NOI18N
    jCheckBox11.setName("checkF11"); // NOI18N

    jCheckBox12.setText(resourceMap.getString("check12.text")); // NOI18N
    jCheckBox12.setName("check12"); // NOI18N

    jButton1.setText(resourceMap.getString("jButton1.text")); // NOI18N
    jButton1.setName("jButton1"); // NOI18N
    jButton1.addActionListener(new java.awt.event.ActionListener() {
       public void actionPerformed(java.awt.event.ActionEvent evt) {
          jButton1ActionPerformed(evt);
       }
    });
```

```java
    jButton2.setText(resourceMap.getString("jButton2.text")); // NOI18N
    jButton2.setName("jButton2"); // NOI18N
    jButton2.addActionListener(new java.awt.event.ActionListener() {
       public void actionPerformed(java.awt.event.ActionEvent evt) {
          jButton2ActionPerformed(evt);
       }
    });

    jCheckBox8.setText(resourceMap.getString("checkF8.text")); // NOI18N
    jCheckBox8.setName("checkF8"); // NOI18N

    buttonGroup1.add(plaintiffWon);
    plaintiffWon.setSelected(true);
    plaintiffWon.setText(resourceMap.getString("plaintiffWon.text")); // NOI18N
    plaintiffWon.setName("plaintiffWon"); // NOI18N

    buttonGroup1.add(plaintiffLose);
    plaintiffLose.setText(resourceMap.getString("plaintiffLose.text")); // NOI18N
    plaintiffLose.setName("plaintiffLose"); // NOI18N

    jLabel2.setFont(resourceMap.getFont("jLabel2.font")); // NOI18N
    jLabel2.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel2.setText(resourceMap.getString("jLabel2.text")); // NOI18N
    jLabel2.setName("jLabel2"); // NOI18N

    javax.swing.GroupLayout mainPanelLayout = new
javax.swing.GroupLayout(mainPanel);
    mainPanel.setLayout(mainPanelLayout);
    mainPanelLayout.setHorizontalGroup(

mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
       .addGroup(mainPanelLayout.createSequentialGroup()
          .addContainerGap()

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING)
             .addGroup(mainPanelLayout.createSequentialGroup()
                .addComponent(plaintiffWon)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
91, Short.MAX_VALUE)
                .addComponent(plaintiffLose))
             .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addComponent(jCheckBox12)
```

```
                    .addComponent(jCheckBox10)
                    .addComponent(jCheckBox11)
                    .addComponent(jCheckBox9)
                    .addComponent(jCheckBox7)
                    .addComponent(jCheckBox6)


.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEA
DING, false)
                    .addComponent(jCheckBox5)
                    .addGroup(mainPanelLayout.createSequentialGroup()
                       .addComponent(jLabel1)


.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                       .addComponent(textName))
                    .addComponent(jCheckBox1)
                    .addGroup(mainPanelLayout.createSequentialGroup()
                       .addComponent(jCheckBox4)
                       .addGap(74, 74, 74)
                       .addComponent(jSpinner1,
javax.swing.GroupLayout.PREFERRED_SIZE, 78,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addComponent(jCheckBox2)
                    .addComponent(jCheckBox3))
                 .addComponent(jCheckBox8, javax.swing.GroupLayout.PREFERRED_SIZE,
147, javax.swing.GroupLayout.PREFERRED_SIZE))
                 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
                    .addComponent(jButton1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
161, Short.MAX_VALUE)
                    .addComponent(jButton2)))
              .addGap(14, 14, 14))
           .addGroup(mainPanelLayout.createSequentialGroup()
              .addGap(50, 50, 50)
              .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 191,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addContainerGap(58, Short.MAX_VALUE))
        );
        mainPanelLayout.setVerticalGroup(

mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
           .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
mainPanelLayout.createSequentialGroup()
              .addContainerGap()
```

```
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(jLabel1)
                    .addComponent(textName, javax.swing.GroupLayout.PREFERRED_SIZE, 24,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox3)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(jCheckBox4)
                    .addComponent(jSpinner1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox5)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox6)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox7)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox8)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox9)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox10, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox11)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jCheckBox12)
                .addGap(18, 18, 18)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
                    .addComponent(plaintiffWon)
```

```
        .addComponent(plaintiffLose))
      .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 9,
Short.MAX_VALUE)

.addGroup(mainPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BAS
ELINE)
        .addComponent(jButton2)
        .addComponent(jButton1))
      .addContainerGap())
  );

  menuBar.setName("menuBar"); // NOI18N

  fileMenu.setText(resourceMap.getString("fileMenu.text")); // NOI18N
  fileMenu.setName("fileMenu"); // NOI18N

  javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(gui.GUIApp.class).getContext().getAction
Map(GUIView.class, this);
  exitMenuItem.setAction(actionMap.get("quit")); // NOI18N
  exitMenuItem.setName("exitMenuItem"); // NOI18N
  fileMenu.add(exitMenuItem);

  menuBar.add(fileMenu);

  helpMenu.setText(resourceMap.getString("helpMenu.text")); // NOI18N
  helpMenu.setName("helpMenu"); // NOI18N

  aboutMenuItem.setAction(actionMap.get("showAboutBox")); // NOI18N
  aboutMenuItem.setName("aboutMenuItem"); // NOI18N
  helpMenu.add(aboutMenuItem);

  menuBar.add(helpMenu);

  statusPanel.setName("statusPanel"); // NOI18N

  statusPanelSeparator.setName("statusPanelSeparator"); // NOI18N

  statusMessageLabel.setName("statusMessageLabel"); // NOI18N

  statusAnimationLabel.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
  statusAnimationLabel.setName("statusAnimationLabel"); // NOI18N

  progressBar.setName("progressBar"); // NOI18N
```

```
    javax.swing.GroupLayout statusPanelLayout = new
javax.swing.GroupLayout(statusPanel);
    statusPanel.setLayout(statusPanelLayout);
    statusPanelLayout.setHorizontalGroup(

statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(statusPanelSeparator, javax.swing.GroupLayout.DEFAULT_SIZE,
299, Short.MAX_VALUE)
        .addGroup(statusPanelLayout.createSequentialGroup()
          .addContainerGap()
          .addComponent(statusMessageLabel)
          .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 181,
Short.MAX_VALUE)
          .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
          .addComponent(statusAnimationLabel)
          .addContainerGap())
    );
    statusPanelLayout.setVerticalGroup(

statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(statusPanelLayout.createSequentialGroup()
          .addComponent(statusPanelSeparator,
javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)
          .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BA
SELINE)
            .addComponent(statusMessageLabel)
            .addComponent(statusAnimationLabel)
            .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
          .addGap(3, 3, 3))
    );

    setComponent(mainPanel);
    setMenuBar(menuBar);
    setStatusBar(statusPanel);
  }// </editor-fold>//GEN-END:initComponents
```

```java
    private void jCheckBox4StateChanged(javax.swing.event.ChangeEvent evt) {//GEN-
FIRST:event_jCheckBox4StateChanged

    // enable spinner for F4

    jSpinner1.setEnabled(jCheckBox4.isSelected());
  }//GEN-LAST:event_jCheckBox4StateChanged

  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton1ActionPerformed

    /* this is beta version legacy
     * while we are making case library
     * make posible to output all cases to file
     * might need it again
     */
    if (textName.getText().equals(""))
    {
       JOptionPane.showMessageDialog(null , "Please enter case name ", "carefull",
JOptionPane.ERROR_MESSAGE);


       return;
    }


    /*
     * go trought check boxes and
     * make string that represent that case
     * int form 0101011110101
     */
    String abstractFact = new String("A");
    caseCount++;
    String strFacts = "";
    int daysLate = 0;
    if (jCheckBox1.isSelected())
    {
       strFacts+='1';
    }
    else strFacts+='0';

    if (jCheckBox2.isSelected())
    {
       strFacts+='1';
    }
```

```
else strFacts+='0';

// this is for abstract factor 3
if (jCheckBox5.isSelected() | jCheckBox7.isSelected())
{
    strFacts+='1';
}
else strFacts+='0';

if (jCheckBox6.isSelected()|jCheckBox8.isSelected()|jCheckBox9.isSelected())
{
    jCheckBox3.setSelected(!jCheckBox3.isSelected());
}


if (jCheckBox4.isSelected())
{
    strFacts+='1';
    daysLate = Integer.parseInt( jSpinner1.getValue().toString() );
}
else strFacts+='0';

if (jCheckBox5.isSelected())
{
    strFacts+='1';
}
else strFacts+='0';

if (jCheckBox6.isSelected())
{
    strFacts+='1';
}
else strFacts+='0';

if (jCheckBox7.isSelected())
{
    strFacts+='1';
}
else strFacts+='0';

if (jCheckBox8.isSelected())
{
    strFacts+='1';
}
else strFacts+='0';
```

```java
        if (jCheckBox9.isSelected())
        {
            strFacts+='1';
        }
        else strFacts+='0';

        if (jCheckBox10.isSelected())
        {
            strFacts+='1';
        }
        else strFacts+='0';

        if (jCheckBox11.isSelected())
        {
            strFacts+='1';
        }
        else strFacts+='0';

        if (jCheckBox12.isSelected())
        {
            strFacts+='1';
        }
        else strFacts+='0';

        // from string or file input we can make case and put it in librart
        //System.out.println( textName.getText() + "\n"+ strFacts + "\n" + daysLate + "\n" +
plaintiffWon.isSelected() );

        // first time subminting a case?
        // that is nor precedent it is a current case
        if (enterCurrentCase)
        {
            currentCase = new FactorsCase(textName.getText(), strFacts, abstractFact,
daysLate,true);
            plaintiffLose.setVisible(true);
            plaintiffWon.setVisible(true);
            jLabel2.setText("Enter precedent case factors");
            enterCurrentCase = false;
        }
        else
        {
        // adding new case to list
        caseList.add(new FactorsCase( textName.getText() , strFacts,abstractFact,
daysLate,plaintiffWon.isSelected()));
```

```
    }

    // uncheck all boxes prerearing us for next case input
    jCheckBox1.setSelected(false);
    jCheckBox2.setSelected(false);
    jCheckBox3.setSelected(false);
    jCheckBox4.setSelected(false);
    jCheckBox5.setSelected(false);
    jCheckBox6.setSelected(false);
    jCheckBox7.setSelected(false);
    jCheckBox8.setSelected(false);
    jCheckBox9.setSelected(false);
    jCheckBox10.setSelected(false);
    jCheckBox11.setSelected(false);
    jCheckBox12.setSelected(false);
    textName.setText("");
}//GEN-LAST:event_jButton1ActionPerformed

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-
FIRST:event_jButton2ActionPerformed

    /* when finished last case that was entered
     * must be put in library
     * apply jButtton1ActionPerformed that will
     * put him there
     */

    jButton1ActionPerformed(evt);
    if (currentCase==null)
        return;
    /* Creating a soliciter, starting a new trial
     * when soliciter is created he will create
     * other court personel and put them into
     * the service
     */

    Soliciter sol= new Soliciter(currentCase, caseList);

    /*
     * trial is over, judge displays arguments
     * in new dialog window
     */
    dlg.setText(sol.judge.Claim);

    /*
```

```
     * show dialog with arguments
     */
    dlg.setVisible(true);
    /*
     * work done exit application
     */
    System.exit(0);

}//GEN-LAST:event_jButton2ActionPerformed

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JCheckBox jCheckBox1;
private javax.swing.JCheckBox jCheckBox10;
private javax.swing.JCheckBox jCheckBox11;
private javax.swing.JCheckBox jCheckBox12;
private javax.swing.JCheckBox jCheckBox2;
private javax.swing.JCheckBox jCheckBox3;
private javax.swing.JCheckBox jCheckBox4;
private javax.swing.JCheckBox jCheckBox5;
private javax.swing.JCheckBox jCheckBox6;
private javax.swing.JCheckBox jCheckBox7;
private javax.swing.JCheckBox jCheckBox8;
private javax.swing.JCheckBox jCheckBox9;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JSpinner jSpinner1;
private javax.swing.JPanel mainPanel;
private javax.swing.JMenuBar menuBar;
private javax.swing.JRadioButton plaintiffLose;
private javax.swing.JRadioButton plaintiffWon;
private javax.swing.JProgressBar progressBar;
private javax.swing.JLabel statusAnimationLabel;
private javax.swing.JLabel statusMessageLabel;
private javax.swing.JPanel statusPanel;
private javax.swing.JTextField textName;
// End of variables declaration//GEN-END:variables

private final Timer messageTimer;
private final Timer busyIconTimer;
private final Icon idleIcon;
private final Icon[] busyIcons = new Icon[15];
private int busyIconIndex = 0;
```

```java
    private JDialog aboutBox;
    private JDialog output;
    private void LoadCases()
    {
       try {
       String name;
       String facts;
       int delay;
       boolean pWon;

       BufferedReader reader = new BufferedReader( new FileReader("input1.cf"));

       name = reader.readLine();
       facts = reader.readLine();
       delay = Integer.parseInt(reader.readLine());
       pWon = Boolean.parseBoolean(reader.readLine());

       currentCase = new FactorsCase(name, facts, "A", delay, pWon);

       while ( true )
         {
          name = reader.readLine();
          facts = reader.readLine();
          delay = Integer.parseInt(reader.readLine());
          pWon = Boolean.parseBoolean(reader.readLine());
          caseList.add(new FactorsCase(name, facts, "A", delay, pWon));
         }
       }
       catch (Exception e)
       {
          System.out.println("Finished loading cases");
       }
     }

}


/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;
```

```java
import java.util.Collections;
import java.util.List;
import java.util.ListIterator;

/**
 * Person who is responsoible to berrister and helps him pass trougt
 * library
 * @author Islam El-adaway
 */
public class CaseAssistant {
   Soliciter soliciter;
   public List<FactorsCase> library;
   public String  sorting = "";
/**
 *
 * @param sol link to parrent soliciter
 */
   public CaseAssistant(Soliciter sol) {
      soliciter = sol;
      library = soliciter.caseLibrarian.library ;

      // temp vars for factors analysis
      int A,B,C,D;
      FactorsCase pCase;
      // string in wich will go to judge aka output
      sorting = "A) Factor Analysis :\n";
      for (int i = 0; i < library.size(); i ++)
      {
        pCase = this.getCase(i);
        sol.NewCase.testAgainst( pCase);
        A = pCase.Count(pCase.Afactors);
        B = pCase.Count(pCase.Bfactors);
        C = pCase.Count(pCase.Cfactors);
        D = pCase.Count(pCase.Dfactors);
        sorting += "Case " + pCase.Name + " A=" + A + " :B="+ B + " :C="+ C + " :D=" + D
+ "\n";

      }

      // sorting library using Comparable interface
      Collections.sort(library);

      // preparing case classificarion for output
      sorting +="\nB) Case classification:\n";
      for (int i=0;i<library.size();i++)
```

```java
        {
            pCase = this.getCase(i);
            sorting += "Case " + pCase.Name + ":"  + pCase.factorClass + "\n";
        }
    }
    /**
     * accesing factor case with index
     * @param i case index
     * @return Factor case with index i
     */
    /**
     * Get case returns factors case with index i
     * @param i index int library
     * @return case with index i
     */
    public FactorsCase getCase(int i)
    {
        return (FactorsCase)library.get(i);
    }


    /**
     * If we want to go trougt whole library we need this irerator
     * @return ListItarator i
     */
    public ListIterator<FactorsCase> getIterator()
    {
        return soliciter.caseLibrarian.library.listIterator();
    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

import java.util.List;

/**
 * Holds library
 * @author Islam El-adaway
 */
public class CaseLibrarian {
```

```java
    public List<FactorsCase> library;

    public CaseLibrarian(List<FactorsCase> lib) {
        library = lib;

    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

import java.util.ListIterator;

/**
 * Defendents Expert responsible for finding a suitable counter case
 * @author Islam El-adaway
 */
public class CounterExampleExpert extends Expert{
    public FactorsCase counterCase;
  /**
   * Every preson in courtrum is directly or indirectly related to soliciter
   * @param sol link to parent soliciter
   */
    public CounterExampleExpert(Soliciter sol) {
        soliciter = sol;
        CaseAssistant cAssistant = new CaseAssistant(sol);

        int citedABcount =
sol.citedCase.Count(sol.citedCase.Afactors)+sol.citedCase.Count(sol.citedCase.Bfactors);
        FactorsCase precedent = null;
        int precABcount = 0;
        int max = 0;
        int CE, U, D;

        ListIterator<FactorsCase> i  = cAssistant.getIterator();
        while (i.hasNext())
        {
            precedent = i.next();
            if (!precedent.PlaintieffWon) // if plaintiefwon skip it
            {
```

```
        precABcount =
precedent.Count(precedent.Afactors)+precedent.Count(precedent.Bfactors);
          if ((precABcount >= citedABcount)&(precABcount>max))  // this is current max
          {
             counterCase = precedent;
             max = precABcount;
          }
       // if plaintiefwon skip it
        }
      }

      if (counterCase != null)
      {
      opinion = "Counter Case Expert :\n counter Case " + counterCase.Name +" is more on
point " + "than " + soliciter.citedCase.Name ;
      opinion +=  "because it has common d-factors : \n";  // TODO

      CE = counterCase.Facts & sol.NewCase.Facts & (counterCase.dTypeFactors |
counterCase.pTypeFactors);
      U  = CE & sol.citedCase.Facts;
      D  = CE & ~sol.citedCase.Facts;
      int count = 0;

      if (CE>0)
      {
      for (int ii =0;ii<12;ii++)
         {
           if ( (CE & soliciter.FactorBits[ii]) != 0 )
           {
              count ++;
              opinion  += count +") "+soliciter.FactorName[ii] + "\n";
           }

         }
      }

      if (D>0)
      {
      opinion += "Counter Case " + counterCase.Name +" is more on point " + "than " +
soliciter.NewCase.Name ;
      opinion +=  "because it has additinoal d-factors : \n";   // TODO
      count = 0;
      for (int ii =0;ii<12;ii++)
         {
           if ( (D & soliciter.FactorBits[ii]) != 0 )
```

```
                {
                    count++;
                    opinion  += count + ") "+soliciter.FactorName[ii] + "\n";
                }

            }

        }
        }
    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;


/**
 * Person responsoble for finding coutra argument and
 * respond to plaintiff satement of claim
 *
 * @author Islam El-adaway
 */
public class DefendantBarrister {
    /**
     * @param counterArgument response to plaintiff arguments
     */
    public String counterArgument = "";
    /**
     * @param currentCase case currenty argumenterd
     */
    public FactorsCase currentCase = null;
    public FactorsCase citedCase;
    public FactorsCase counterCase = null;
    public Soliciter soliciter;
    public CounterExampleExpert ceExpert;
    public FactorDistinctionExpert fdExpert;
    public DimensionExpert dimExpert;
    public EmphasizeDistinctionExpert edExpert;
    public CaseAssistant caseAssistant;
    public DefendantBarrister(Soliciter sol)
```

```java
    {
        soliciter = sol;
        caseAssistant = new CaseAssistant(sol);
        ceExpert = new CounterExampleExpert(sol);

        if (ceExpert.counterCase != null)
        {
        counterArgument = "\nD) Defendent Statement of Defense :\n";

        counterCase = ceExpert.counterCase;
        counterArgument += ceExpert.opinion;
        soliciter.counterCase = counterCase;

        fdExpert = new FactorDistinctionExpert(sol);
        counterArgument += fdExpert.opinion;

        edExpert = new EmphasizeDistinctionExpert(sol);
        counterArgument += edExpert.opinion;

        dimExpert = new DimensionExpert(sol);
        counterArgument += dimExpert.opinion;
        }
        else
        {
            counterArgument = "\n\nThere is no citable case for Defense! ";
        }


    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

/**
 * Similar to Dimension expert
 * @author Islam El-adaway
 * @see    DimensionExpert
 */
public class DimenisionRebuttalExpert extends Expert{
```

```java
    public boolean inPlaintiffFavour;
    /**
     * Every preson in courtrum is directly or indirectly related to soliciter
     * @param sol link to parent soliciter
     */
    public DimenisionRebuttalExpert( Soliciter sol) {
       soliciter = sol;
       if (sol.counterCase.Days < sol.NewCase.Days)
       {
          inPlaintiffFavour = true;
          opinion = "Dimesion Rebuttal Expert : Counter case has delay less than current
case\n";

       }
       else
          inPlaintiffFavour = false;
    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

/**
 * Dimension expert
 * @author Islam El-adaway
 */
public class DimensionExpert extends Expert{
    /**
     * @param inPlaintiffFavour holds expert opinion
     */
    public boolean inPlaintiffFavour;
    public static final int F4 = 8;
    /**
     * Every preson in courtrum is directly or indirectly related to soliciter
     * @param sol link to parent soliciter
     */
    public DimensionExpert(Soliciter sol)
    {
       soliciter = sol;
```

```java
      // test if newCase has latency
      if ( (sol.NewCase.Facts & F4) != 0 )   // is this a
      {
      if (sol.citedCase.Days > sol.NewCase.Days)
      {

        inPlaintiffFavour = false;
        opinion = "Dimension Expert : This cited case has larger delay than current case";
      }
      else
        inPlaintiffFavour = true;
        // we dont need to opinion this becouse it is in plaintiff favour


      }


   }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

import java.util.ListIterator;

/**
 * Downplay importance of factors in counter case
 * @author Islam El-adaway
 */
public class DownPlaySignificanceExpert extends Expert{
   public int downPlaySupport = 0;
   public int downPlayMilitiate = 0;
   /**
    * Every preson in courtrum is directly or indirectly related to soliciter
    * @param sol link to parent soliciter
    */
   public DownPlaySignificanceExpert (Soliciter sol )
   {
     soliciter = sol;
     int diff = sol.citedCase.Facts ^ sol.NewCase.Facts;   //diff iz F in worked example
```

```java
        int WP = diff & sol.citedCase.Facts & sol.citedCase.pTypeFactors;
        int WD = diff & sol.NewCase.Facts & sol.citedCase.dTypeFactors;

        int SU = 16 + 63;       // setting F5 and F7 mask
        int MI = 32 + 128 + 256; // setting F6 , F8 and F9 mask


        if ( (WP&SU) != 0)
        {
           downPlaySupport = WP&SU;
        }

        if ( (WD& SU) != 0)
        {
           downPlayMilitiate = WD & MI;
        }
        int both = downPlayMilitiate | downPlaySupport;
        if (both!=0)
        {
        opinion = "Downplay Significance Expert : \nsignificance of the following factors can
be downplayed :\n";

        for (int ii =0;ii<12;ii++)
            {
               if ( (both & soliciter.FactorBits[ii]) != 0 )
               {
                  opinion  += soliciter.FactorName[ii] + "\n";
               }
            }
        }

    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

/**
 *
 * @author Islam El-adaway
```

```java
 */
public class EmphasizeDistinctionExpert extends Expert{

   public int supportEmphasized=0;
   public int militiateEmphasized=0;
   /**
    * Every preson in courtrum is directly or indirectly related to soliciter
    * @param sol link to parent soliciter
    */

   public EmphasizeDistinctionExpert(Soliciter sol) {
      soliciter = sol;
      if (sol.counterCase==null) return;

      int SU = 16 + 63; // setting F5 and F7
      int MI = 32 + 128 + 256; // setting F6 , F8 and F9

      int diff = sol.citedCase.Facts ^ sol.NewCase.Facts;
      int WP = diff & sol.citedCase.Facts & sol.citedCase.pTypeFactors;
      int WD = diff & sol.NewCase.Facts & sol.citedCase.dTypeFactors;

      if ((SU&WP)== 0)  // set intersection is empty
      {

         // this means that we cannot emphasize F6 and F8
      }
      else
      {
         militiateEmphasized = SU & WP & sol.citedCase.Facts;

         opinion = "Emphasize Distinction Expert:\nThe following factors can be emphasized
\n";
         int count = 0;
         for (int ii =0;ii<12;ii++)
            {
               if ( (militiateEmphasized & soliciter.FactorBits[ii]) != 0 )
               {
                  count ++;
                  opinion  += count +") "+soliciter.FactorName[ii] + "\n";
               }
            }

         //
      }
```

```
        if ( (SU & WD)== 0 ) // set intersection is empty
        {
           // we canot emphasize F5 and F7 factors
        }
        else
        {
           // we can emphasize F5 and F7

           supportEmphasized = SU & sol.counterCase.Facts;
           opinion = "The following factors can be emphasized \n";
           int count = 0;
           for (int ii =0;ii<12;ii++)
             {

                if ( (supportEmphasized & soliciter.FactorBits[ii]) != 0 )
                {
                   count ++;
                   opinion  += count +") "+soliciter.FactorName[ii] + "\n";
                }
             }


        }


   }


}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

/**
 * Superclass for various experts
 *
 * @author Islam El-adaway
 * @param  soliciter Every expert is related indirectly to soliciter
 * @param  opinion   Every expert has some kind of opinion
 */
public class Expert {
```

```java
    public Soliciter soliciter;
    public String opinion = "";
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

/**
 * Factor distinction mesures distinction between current and cited case
 * @author Islam El-adaway
 */
public class FactorDistinctionExpert extends Expert{

    public int FD;

    /**
     * Every preson in courtroom is directly or indirectly related to soliciter
     * @param sol
     */
    public FactorDistinctionExpert(Soliciter sol) {
        soliciter = sol;
        int diff = sol.citedCase.Facts ^ sol.NewCase.Facts;
        int WP = diff & sol.citedCase.Facts & sol.citedCase.pTypeFactors;
        int WD = diff & sol.NewCase.Facts & sol.citedCase.dTypeFactors;
        FD = WP | WD;

        if (FD!=0)
        {
        opinion = "Factor Distinction Expert :\nThe following factors can be distinguished
between cited case and current case \n";
        int count = 0;
        // we go trought factors of FD and print them in experts opinion
        for (int ii =0;ii<12;ii++)
            {
                if ( (FD & soliciter.FactorBits[ii]) != 0 )
                {
                    count ++;
                    opinion  += count + ") "+ soliciter.FactorName[ii] + "\n";
                }
            }
```

```
        }


    }


}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

/**
 *
 * @author Islam El-adaway
 */
public class FactorRebuttalExpert extends Expert{
    public int strengthFact;
  /**
     * Every preson in courtrum is directly or indirectly related to soliciter
     * @param sol link to parent soliciter
     */

    public FactorRebuttalExpert(Soliciter sol)
    {
        soliciter = sol;
        int diff = sol.NewCase.Facts ^ sol.counterCase.Facts;
        int df = sol.NewCase.Facts & diff & sol.NewCase.pTypeFactors;
        int dd = sol.counterCase.Facts & diff & sol.NewCase.dTypeFactors;
        strengthFact = df | dd;
        opinion = "Factor Rebuttal Expert : \nThe following factors can be distinguished
between counter case and current case \n";
        int count = 0;
        for (int ii =0;ii<12;ii++)
        {
            if ( (strengthFact & soliciter.FactorBits[ii]) != 0 )
            {
                count ++;
                opinion  +=count+") "+ soliciter.FactorName[ii] + "\n";
            }
        }
```

```java
    }
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

/**
 *
 * @author Islam El-adaway
 */
public class FactorsCase implements Comparable  {
    public static final int pTypeFactors =  1962; //1834;
    public static final int dTypeFactors = 2129;
    public static final int[] precedanceTable = {0,0,0,0,0,0,0,0,0,5,1,8,4,6,2,7,3};
    public String Name;
    public int Facts;
    public String AbstractFact;
    public int Days;
    public boolean PlaintieffWon;
    public int precedence;
    public int Afactors;
    public int Bfactors;
    public int Cfactors;
    public int Dfactors;
    public String factorClass;

    public void writeItDown(int numb)
    {
        String s = "";
        int power = 1;
        for (int i=0;i<12;i++)
        {
            if ( (power & numb) == power)
            {
                s =  s + '1';
            }
            else
                s = s + '0';
            power *= 2;
        }
        System.out.println(s);
```

```
    }

    public void testAgainst(FactorsCase pCase)
    {
        pCase.Afactors = pCase.Facts & pTypeFactors & Facts;
        pCase.Bfactors = pCase.Facts & dTypeFactors & Facts;
        pCase.Cfactors = ((Facts & ~pCase.Facts) & pTypeFactors ) | ((~Facts &
pCase.Facts) & dTypeFactors);

        pCase.Dfactors = (dTypeFactors & Facts & ~pCase.Facts)  | (pTypeFactors & ~Facts &
pCase.Facts);

        /*
        System.out.println("P-type");
        writeItDown(pTypeFactors);
        System.out.println("CD");
        writeItDown(Facts);
        System.out.println(pCase.Name);
        writeItDown(pCase.Facts);
        System.out.println("CD agains " + pCase.Name);
        System.out.print("A - ");
        writeItDown(pCase.Afactors);
        System.out.print("B - ");
        writeItDown(pCase.Bfactors);
        System.out.print("C - ");
        writeItDown(pCase.Cfactors);
        System.out.print("D - ");
        writeItDown(pCase.Dfactors);
        System.out.println("------");
        */
        int A,B,C,D;
        pCase.precedence = 0;
        pCase.factorClass = "";
        if ((A = Count(pCase.Afactors))>0)
        {
            pCase.factorClass +="A";
            pCase.precedence +=8;
        }
        if ((B = Count(pCase.Bfactors))>0)
        {
            pCase.factorClass +="B";
            pCase.precedence +=4;
        }
        if ((C = Count(pCase.Cfactors))>0)
        {
```

```
        pCase.factorClass +="C";
        pCase.precedence +=2;
    }
    if ((D = Count(pCase.Dfactors))>0)
    {
        pCase.factorClass +="D";
        pCase.precedence +=1;
    }

}

/*
 * Count how many Afactors, Bfactors,Cfators or Dfactors is
 * int var factors, we will use this function in following form
 * caseList.Count(AFactors)
 * @param factors counter
 */
public int Count (int factors)
{   // ovde sam napravio neko sranje
    int count = 0;
    int power = 1;
    for (int i=0; i < 12; i++)
    {
        if ( (factors & power)!= 0 )
        {
            count++;
        }

        power *= 2;
    }
    return count;
}

public int compareTo(Object o) {
    FactorsCase fc = (FactorsCase)o;

    if (precedanceTable[fc.precedence] > precedanceTable[this.precedence] ) return 1;
    if (precedanceTable[fc.precedence] < precedanceTable[this.precedence] ) return -1;

    if (Count(fc.Cfactors) > Count(Cfactors)) return 1;
    if (Count(fc.Cfactors) < Count(Cfactors)) return -1;

    if (Count(fc.Dfactors) < Count(Dfactors)) return 1;
    if (Count(fc.Dfactors) > Count(Dfactors)) return 1;
```

```
      if (Count(fc.Afactors) > Count(Afactors)) return 1;
      if (Count(fc.Afactors) < Count(Afactors)) return -1;

      if (Count(fc.Bfactors) > Count(Bfactors)) return 1;
      if (Count(fc.Bfactors) < Count(Bfactors)) return -1;

      return 0;
   }

   /**
    * Constructor for FactorsCase class
    *
    * @param name name of case
    * @param facts array of s and n exactly 12 of them
    * @param abstractFacts usualy A
    * @param days Factor3's attribute days late
    * @param whoWon who won the case plantieff or defender
    * @see Image
    */
   public FactorsCase(String name,String facts, String abstractFacts, int days, boolean
whoWon)
   {
      Name = name;
      Facts = 0;
      // Facts = facts;
      int power = 1;
      for (int i = 0 ;i < 12 ; i++)
      {
         if (facts.charAt(i)=='1')
         {
            this.Facts = this.Facts + power;

         }          power *= 2;
      }
      AbstractFact = abstractFacts;
      Days = days;
      PlaintieffWon = whoWon;
   }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
```

```java
package courtroom;

/**
 *
 * @author Islam El-adaway
 */
public class Judge {
    public FactorsCase currentCase;
    public FactorsCase citedCase;
    public FactorsCase counterCase;
    public Soliciter   soliciter;
    public String      Claim;
    public Judge(Soliciter sol) {
        soliciter = sol;
    }

    public void dismissCourtroom()
    {

    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

/**
 * Barrister that is submiting the claim
 * @author Islam El-adaway
 */
public class PlaintiffBarrister {
    /**
     * @param soliciter link to parent Soliciter
     */
    public Soliciter soliciter;
    public FactorsCase currentCase;
    public FactorsCase citedCase;

    public String firstArgumentStatement = "";
    public String secondArgumentStatement = "";
```

```
// first argument expert
/**
 *  @param spExpert Plaintiff's expert who is looking for best supporting precedent case
 *
 */
public StatePointExpert spExpert;

// second argument experts
/**
 * @param frExpert
 */
public FactorRebuttalExpert frExpert;
public DimenisionRebuttalExpert drExpert;
public DownPlaySignificanceExpert dpsExpert;
public UpplaySignificanceExpert usExpert;
public WeaknessNotFatalExpert wnfExpert;
public CaseAssistant caseAssistant;
public PlaintiffBarrister(FactorsCase cC,Soliciter sol)
{
   soliciter = sol;
   caseAssistant = new CaseAssistant(sol);
   currentCase = cC;
   spExpert = new StatePointExpert(sol);
   citedCase  = spExpert.getCitedCase(currentCase);
   if (citedCase!=null)
   {
   firstArgumentStatement = caseAssistant.sorting;
   firstArgumentStatement += "\nC) Plaintiff Statement of Claim :\nStand Point Expert
\nThe case "+citedCase.Name +" is in favour of plaintiff because of the following factors: \n"
+ spExpert.opinion;
   }
}

public void confrontCounterCase()
{
   secondArgumentStatement += "\nE) Plaintiff Statement of Rebuttal: \n";
   frExpert = new FactorRebuttalExpert(soliciter);
   secondArgumentStatement += frExpert.opinion;
   drExpert = new DimenisionRebuttalExpert(soliciter);
   secondArgumentStatement += drExpert.opinion;
   dpsExpert = new DownPlaySignificanceExpert(soliciter);
   secondArgumentStatement += dpsExpert.opinion;
   usExpert = new UpplaySignificanceExpert(soliciter);
   secondArgumentStatement += usExpert.opinion;
```

```java
        wnfExpert = new WeaknessNotFatalExpert(soliciter);
        secondArgumentStatement += wnfExpert.opinion;

    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

import java.util.List;

/**
 * Soliciter manages courtroom workflow
 * @author Islam El-adaway
 */
public class Soliciter {

/**
 * @param Newcase Current claim
 */
   public FactorsCase NewCase;

/**
 * @param caseLibrarian person that runs library
 */
   public CaseLibrarian caseLibrarian;
   /**
    * @param judge listen Barristers statements
    */
   public Judge judge = new Judge(this);
   /**
    * @param citedCase Cited Case by Plaintiff barrister
    */
   public FactorsCase citedCase;
   /**
    * @param counterCase case cited by Defendent barrister
    */
   public FactorsCase counterCase;
   public String[] FactorName = new String[12];
   public int[]    FactorBits = {1,2,4,8,16,32,64,128,256,512,1024,2048};
```

```
/***
 * Soliciter start process with a new claim
 * @param newDespute current dispute
 * @param lib library that holds all past cases
 */
public Soliciter(FactorsCase newDespute, List<FactorsCase> lib ) {

   // making factor name table
   FactorName[0] = "Needs material on site";
   FactorName[1] = "Needs material not available on site";
   FactorName[2] = "Allows change of material price";
   FactorName[3] = "Delay effect as result of change order";
   FactorName[4] = "Needs material in bill of quantities";
   FactorName[5] = "Needs material not in bill of quantities";
   FactorName[6] = "Needs local material";
   FactorName[7] = "Needs imported material";
   FactorName[8] = "Carried out using available work force";
   FactorName[9] =  "Carried out using additional work force";
   FactorName[10] = "Carried out using additional  work force ";
   FactorName[11] = "Needs additional equipment";

   // assigning newDespute to soliciter
   NewCase = newDespute;
   // Courtroom personel

   // assigning
   caseLibrarian = new CaseLibrarian(lib);
   // Creating plaintiff barrister and leting him do his work
   PlaintiffBarrister plaintiffBarrister = new PlaintiffBarrister(newDespute,this);
   // plaintifs cited case is officialy Cited Case
   citedCase = plaintiffBarrister.citedCase;

   // if there is no cited case judge dismiss Courtoom
   if (citedCase==null)
   {
      System.out.println("There is no citable case, claim droped \n");
      judge.dismissCourtroom();
   }

   // Creating defendent barrister
   // He will employ various experts

   DefendantBarrister defendentBarrister = new DefendantBarrister(this);
   // counter case precedan is given to soliciter
```

```java
      // so becomes official
      counterCase = defendentBarrister.counterCase;

      // if there is no counter case than finish discution
      // else confront counter argument
      if (counterCase != null )
      {
      plaintiffBarrister.confrontCounterCase();
      }
      // give all arguments to judge

      judge.Claim = plaintiffBarrister.firstArgumentStatement;
      judge.Claim += defendentBarrister.counterArgument;
      judge.Claim += plaintiffBarrister.secondArgumentStatement;


  }
}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;


import java.util.Collections;
import java.util.ListIterator;

/**
 * Looks for best suitable precedent for current dispute
 * @author Islam El-adaway
 */
public class StatePointExpert extends Expert{
   /**
    * Every preson in courtrum is directly or indirectly related to soliciter
    * @param sol link to parent soliciter
    */

   public StatePointExpert(Soliciter sol) {
      soliciter = sol;
   }

   /**
    * test new case against all precedents so he can find best
```

```java
 * @param newCase new case
 * @return
 */
public FactorsCase getCitedCase(FactorsCase newCase)
{
   CaseAssistant caseAssistant= new CaseAssistant(soliciter);

   // test new case agains whole library
   for (int i = 0; i < caseAssistant.library.size(); i ++)
   {
      newCase.testAgainst( caseAssistant.getCase(i));
   }
   // sort library in best way for plaintiff using comprable interface
   Collections.sort(caseAssistant.library);

   FactorsCase currentBest;

   // in ordered library go from begining and look first case
   // that has been won by  plaintiff
   ListIterator<FactorsCase> i = caseAssistant.getIterator();
   while (i.hasNext())
   {
      currentBest = i.next();
      if (currentBest.PlaintieffWon)
      {
         // we found best one
         int facts = currentBest.Afactors;
         int count = 0;
         // print common factors
         for (int ii =0;ii<12;ii++)
         {
            if ( (currentBest.Afactors & soliciter.FactorBits[ii]) != 0 )
            {
               count ++;
               opinion  += count + ") " +soliciter.FactorName[ii] + "\n";
            }

         }
         return currentBest;
      }
   }
   return null;
}
}
```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

import java.util.ListIterator;

/**
 *  Capitalize factors that where classified as strengths
 * @author Islam El-adaway
 */
public class UpplaySignificanceExpert extends Expert{
  /**
   * @param upPlayStrength Case wich can be used to emphasize strengths
   *
   */
  public FactorsCase upPlayStrength = null;

   /**
   * Every preson in courtrum is directly or indirectly related to soliciter
   * @param sol link to parent soliciter
   */

  public UpplaySignificanceExpert(Soliciter sol)
  {
    CaseAssistant caseAssistant = new CaseAssistant(sol);
    soliciter = sol;
    int C = sol.NewCase.Facts & sol.NewCase.pTypeFactors;
    int C1 = C ^ sol.citedCase.Facts;
    int strength = C1 & sol.NewCase.Facts;

    ListIterator<FactorsCase> i = caseAssistant.getIterator();
    FactorsCase precedent;

    while ( i.hasNext() )
    {
      precedent = i.next();
      if ( ( (precedent.Facts & strength)!= 0 ) & (precedent.PlaintieffWon ) )
      {
        // TODO find one or find more precedents
        upPlayStrength = precedent;
        break ;
      }
```

```
        }

      if (upPlayStrength!= null)
      {
        opinion = "Up-play Significance Expert : \nThe Following factors can be up-
played\n";
        int count = 0;
        for (int ii = 0; ii< 12 ; ii++)
        {
          if ( (strength & soliciter.FactorBits[ii]) != 0 )
          {
            count ++;
            opinion += count +") "+ soliciter.FactorName[ii] + "\n";
          }
        }
      }

    }

}

/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */

package courtroom;

import java.util.ArrayList;
import java.util.List;
import java.util.ListIterator;


/**
 *
 * This Expert is looking to find a case with a plaintiff outcome
 * where the weakness were also present and applied
 * @author Islam El-Hadavay
 */
public class WeaknessNotFatalExpert extends Expert{

  public List<FactorsCase> supportCases;
  public List<Integer>    supportFacts;

    /**
```

```
 * Every preson in courtrum is directly or indirectly related to soliciter
 * @param sol link to parent soliciter
 */

public WeaknessNotFatalExpert (Soliciter sol)
{
   soliciter = sol;
   CaseAssistant caseAssistant = new CaseAssistant(sol);

   int diff = sol.citedCase.Facts ^ sol.NewCase.Facts;   //diff is F in worked example
   int WP = diff & sol.citedCase.Facts & sol.citedCase.pTypeFactors;
   int WD = diff & sol.NewCase.Facts & sol.NewCase.dTypeFactors;
   boolean found = false;

   int M1,M2,T1,T2;
   supportCases = new ArrayList<FactorsCase>();
   supportFacts = new ArrayList<Integer>();


   ListIterator<FactorsCase> i = caseAssistant.getIterator();
   FactorsCase precedent;

   while (i.hasNext())
   {
     precedent = i.next();
     if ( precedent.PlaintieffWon)
     {
       M1 = WP & precedent.Facts;
       M2 = WP & ~M1 ; // this is minus M2 - M1
       if (M2!=0)
       {
         found = true;
         opinion += "in precedent : " + precedent.Name + "\n";
         supportCases.add(precedent);
         supportFacts.add( new Integer(M2));
         int count = 0;
         for (int ii =0;ii<12;ii++)
         {
           if ( (M2 & soliciter.FactorBits[ii]) != 0 )
           {
             count ++;
             opinion  += count + ") " +soliciter.FactorName[ii] + "\n";
           }
         }
       }
```

```
            //
            T1 = WD & precedent.Facts;
            if (T1 != 0 )
            {
               found = true;

               opinion += "in precedent : " + precedent.Name + "\n";
               supportCases.add(precedent);
               supportFacts.add( new Integer(M2));
               int count = 0 ;

               for (int ii =0;ii<12;ii++)
                 {
                    if ( (T1 & soliciter.FactorBits[ii]) != 0 )
                    {
                       count ++;
                       opinion  += count + ") " +soliciter.FactorName[ii] + "\n";
                    }
                 }

            }

         }

      if (found)
      {
         opinion = "Weakness not fatal Expert :\nAbsence of following factors is not fatal for
case outcome: \n" + opinion;
      }


   }
}
```