WAMAN[§]: Web-mining-Assisted Mobile-computing-enAbled on-line optioN pricing- a software Architecture towards autonomic computing

by

Kiran Kumar Reddy Kola

A thesis
submitted to the University of Manitoba
in partial fulfillment of the requirements for the
degree of Master of Science
in
Computer Science

Winnipeg, Manitoba, Canada

[§] WAMAN is an incarnation of Lord Vishnu (one of the Hindu Gods) as a very tiny boy. Waman tricks the demon king Bali to grant him as much of his empire as he could measure in three steps (or paces) of land in his kingdom. The boy grows into humongous form and measures the three strides to cover earth by one-step, cover universe another step, and when there was no space, he puts third step on king's head. This small study has large potential to grow in many directions while avoiding lack of knowledge and information. (Please refer to http://www.sanatansociety.org/hindu_gods_and_goddesses/vamana.htm for more information on this story)

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

WAMAN§: Web-mining-Assisted Mobile-computing-enAbled on-line option pricing- a software Architecture towards autonomic computing

BY

Kiran Kumar Reddy Kola

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

OF

MASTER OF SCIENCE

Abstract

The problem of pricing in finance industry is growing in computational complexity due to demands for self-managing (autonomic), high-speed and real-time solutions for complex mathematical problems such as option pricing. In current option trading scenario, determining a fair price for options "any time" and "any-where" has become vital yet difficult computational problem. In this study, we have designed, implemented, and developed the architecture for pricing options on-line using a hand-held device that is J2ME-based, mobile computing-enabled, and is assisted by web mining tools. In our architecture, the client is a MIDP user interface, and the back end servlet runs on a stand-alone server bound to a known port address. In addition, the server uses table-mining techniques to mine real-time data from reliable web sources upon the mobile investor's directive. The server performs all computations required for pricing options since mobile devices have limited battery power, low bandwidth, and low memory.

To the best of our knowledge, this is one of the first studies that facilitate the mobile-enabled-investor to compute the price of an option (using couple of different techniques such as binomial lattice, and finite differencing technique) in ubiquitous fashion (everywhere). In addition, we can incorporate any of the latest pricing (numerical) technique to our architecture, seamlessly. This architecture aims at providing the investor with various computational techniques to avail (to provide results from approximate to accurate results) while on-the-go. In addition, our architecture provides value-added services such as healthy bids, risk free zone, and favored stocks in the form of tables and graphs. Moreover, the user has an option of personalizing the computed results to his/her wireless device or retrieve from backend compute server and hence can make important and effective trading decisions that will ensure higher returns on investments in option trading.

We provide sample results from couple of case studies for options written on technology (blue chip) stocks. Without loss of generality, we claim that our architecture autonomically computes the option prices except for the user intervention at the start and at the end of the pricing process. However, by setting the user preferences apriori on variety of options, our architecture can self-manage in computing the price of options, inform the user continuously, and can even made to decide for the user.

Keywords: *Options, Option Pricing, Servlet, Table Mining, J2ME*

Categories & Subject Descriptors: *Option Trading: Option Pricing, Web Mining- Text Mining, Mobile Computing- Interface Design*

This thesis is dedicated to my parents and my family

Acknowledgement

I would like to express my gratitude to all those who gave me the possibility to complete this thesis. First, my thesis committee members, Dr. Renee B. Y. Kim, Dr. Pourang Irani and Dr. Neil Arnason, whom I thank for their valuable comments and suggestions that helped to improve the quality of the final document.

I feel very privileged to have worked with my supervisor Dr. Ruppa K. Thulasiram (Tulsi). I owe a great debt of gratitude for his guidance, inspiration, and patience. Dr. Thulasiram exposed me to the field of computational finance in the early stage of my degree and I found it very interesting and challenging. I want to thank for his countless hours of discussion, advice and guidance regarding research, writing, speaking and life in general. Under his guidance, I learnt the subtleties of writing, the peculiarities of publishing papers in computer science and the importance of public speaking. His tireless editorial effort vastly improved the quality of this thesis and their valuable comments gave me confidence for my defense.

Especially, I would like to give my special thanks to Dr. Neil Arnason and Dr. Parimala Thulasiraman for the help extended to me when I approached them and the valuable discussion that I had with them during the course of research.

My father has been my inspiration all my life and he always stood by me and my decisions. The values, care and love given by my family are responsible for making me who I am today.

This thesis would not have been possible without the few good people who somehow crossed my life at the right moment in time, I would like to thank my friends Varghese Devassy, Amit Chhabra, and Santan Challa for their support, ideas and when I needed a timeout.

TABLES OF CONTENTS

TABLES AND FIGURES

Glossary

- *CBOE: Chicago Board Options Exchange*
- *NASDAQ: National Association of Securities Dealers Automated Quotation*
- *CBOT: Chicago Board of Trade*
- *GCF: Generic Connection Framework*
- *J2ME: Java 2 Micro Edition*
- *SMS: Short Message Service*
- *MIDP: Mobile Information Device Profile*
- *ASF: Apache Software Foundation*
- *API: Application Programming Interface*
- *CLDC: Connected Limited Device Configuration*
- *CDC: Connected Device Configuration*
- *HTTP: HyperText Transfer Protocol*
- *HTTPS: Secure HyperText Transfer Protocol*
- *RPC: Remote Procedure Call*
- *OTA: Over-The-Air Provision*
- *JSP: Java servlet Page*
- *KVM: Kilo Virtual Machine*
- *CVM: C Virtual Machine*
- *ASF: Apache Software Foundation*
- *AMS: Application-Management System*

Chapter 1

Introduction

This thesis addresses the study of mobile technology and computational finance combined with Internet mining for successful trading that would pave way for effective investments for better profitability and higher returns on investments in option trading. In effect, this study combines e-commerce and e-finance with the modern mobile technology to create M-business environment. Option pricing forms a fundamental objective and backbone of financial risk management and decision-making solutions. Active trading takes place either at the trading floor or through computers with instructions from investors or investment managers. However, once an investor steps away from the workplace, the investor encounters problems of interrupted trading, as the required information is no longer available. In such cases, investors have to rely on the data provided by some source (such as electronic board display). If the investor is away from the building, he/she has to be in continuous touch with some other sources such as a broker to get some basic information about the market to analyze the market behaviour. However, the information provided by intermediary brokerage firms is generally inadequate especially in the case of computing the option values.

Over the last decade, sophisticated financial instruments called derivative securities have become increasingly important in financial markets. *Futures* and *Options* are the most common type of derivatives and both are actively traded on many exchanges. The current research focuses only on the option pricing and therefore, the discussion of futures and other derivatives is beyond the scope of this thesis.

1.1 Motivation of WAMAN

The purpose of this thesis is to design and develop a novel software architecture that enables an active investor to price options on-line using various computational techniques. The motivation of this study stems from the objective of enabling an investor to remain in continuous touch with the ever-changing market while on-the-move.

In pricing options, every investor must understand the possible future trends of an underlying asset and the potential for speculation and hedging. In the derivative market, accuracy and ability to respond quickly and ubiquitously to the fluctuating market is extremely important for every active investor. Therefore, to achieve ubiquity in option trading we need an infrastructure, which will enable the trader to avail various computational techniques for accurate and immediate results. This can accelerate decision-making process. The mobile technology is riding a new wave of business innovation. The use of mobile technology for e-business and decision-making strategy is slowly changing the dialogue between investors and traders on the floor of a stock exchange into M-business deals.

In this study, our focus is on three major issues to achieve ubiquity in derivative markets: (i) mobile commerce aspects in derivative markets (particularly financial options) (ii) computational techniques used to price options. (iii) mining a real-time finance data from web sources. We have incorporated all these issues to provide a value-added, any time/any-where service to the trader on the go. We use the terms ubiquitous and pervasive interchangeably in this thesis.

The architecture developed in this study works successfully for both the intended scenarios: (i) short-range wireless connectivity with PDA and (ii) enabling wide-range connectivity using J2ME-enabled mobile devices [48]. It is easy to make our architecture more effective by parallelizing the computational problem at hand.

1.2 Challenges of Thesis

Fundamental challenges to design this software architecture for ubiquitous option trading are from three different scientific domains: (i) option trading, (ii) web mining, and (iii) mobile computing. Every domain has its own challenges for its functionality. We list the challenges that we faced to build the architecture and our solution to each of the respective challenges and thereby my contribution to the emerging business scenario:

Computational issues: (i) option pricing by itself is a computationally intensive problem; (ii) parameters required for computing option price are continuously changing due to market fluctuations. In such a situation, accurate results will depend heavily on appropriate use of current market conditions; (iii) configuring and implementing the existing computational algorithms and handling multiple traders simultaneously and remotely is a challenging task.

Solution and Contributions: we have implemented couple of computational techniques (binomial tree method and finite-differencing technique) maintaining accuracy to a larger extent. Higher accuracy can be obtained by introducing appropriate new technique (when and if available or developed) in this module of our architecture. This is described in Chapter 2

Challenges in Web Mining: mining real-time finance data from reliable web sources and forwarding the observed results.

Solution and Contribution: Rules for single dimensional and two-dimensional tables are employed that are simple yet significant for this research. Related work is described in Chapter 4 and solution strategy is described in Chapter 5.

Challenges in mobile computing: (i) designing and building mobile trading terminal for computing option price any *time* and *anywhere* is one of the main challenging components of the current research; (ii) as the architecture depends heavily on real time data access, network connectivity and security between mobile client and back end server remain an important issue.

Solution and Contribution: To handle these issues we have employed iterative secured-flow-design-approach for building screen logic and layout design of the Mobile interfaces. In addition, to provide additional security to wireless devices, we employed lightweight API called "Bouncy Castle". This is described in

Chapter 5.

The main essence of this framework lies in its novelty, which stems from three different domains to enable the mobile trader to compute the price of an option in pervasive fashion, which is an important advancement in option trading application. Hence, this is one of the first studies that facilitate the service of option pricing on-the-go.

1.3 Advancing towards Autonomic Computing

Before addressing the automation of a particular problem, we might have to consider several issues, for example, true need for automation of that problem in the real world situations, possibility of risk factors, security issues, profitability, reliable infrastructure, etc. If so, does option trading need any automation or self-managing system? To answer this question we will elucidate the need for autonomic computing in option trading in the following paragraph.

Major stock exchanges trade options. Ever since the option market started, volume of trade has increased significantly. In most cases, investors participate or involve in trading activity via electronic trading board or through a brokerage firm. As in case when investor cannot reach the exchange while he/she is on the move, a brokerage firm can take the orders over the phone based on their personal account information. In option trading, accuracy and ability to respond quickly to the fluctuating market is vital for every active investor. There are many underlying factors that option investor has to remember before he/she takes any then describe our overall ubiquitous software architecture design for real-time option pricing in Chapter 5. We discuss computational results and trading scenarios in Chapter 6 and conclusions and future work in Chapter 7.

Chapter 2

Background Work in Finance Computation

In this Chapter, we introduce some vocabulary relevant to the current study with a motivation for Mobile-Pricing (M-pricing) in option trading. In addition, we explain the fundamental working principles of the computational techniques that we are using for this thesis.

2.1 Terminology

Some of the technical terms from the area of option pricing [1] are described in this section. An **option** is a security that gives its owner the right without creating any obligation to trade-in a fixed number of shares of a specified *asset* (for example, stocks) at a fixed price (*strike price*) at any time on or before a specified future date (*maturity date*).

There are two parties involved in the option trading namely *holder* and *writer*. The holder of the option gets the right to **buy** (**call**) or **sell** (**put**) assets at a predetermined time in the future for a predetermined value; the writer of the option on the other hand is obliged to deliver (call) or take delivery (put) of the underlying asset.

Example 1:

CALL option:

An investor buys a call option to purchase 100 INTEL (INT) shares. Strike price: $40; Current stock price: $38; Price of an option to buy one share = $5; Initial investment is 100 x $5 = $500

The outcome:

At the time of expiration of the option, let us assume INTEL stock price to be $55. At this time, the option is exercised for a gain of

($55 - $40) x 100 = $1,500 (that is when the stock price goes up the call option becomes beneficial). When the initial cost of the option is taken into account, the net gain is: $1,500 - $500 = $1,000. If the stock price is less than $40, the holder will not exercise the right to buy. In this circumstance, the investor loses the initial investment of $500.

PUT option:

An investor buys a put option to sell 100 IBM shares. Strike price: $70; Current stock price: $65; Price of an option to buy one share = $7; Initial investment is 100 x $7 = $700

The outcome:

At the time of expiration of the option let us assume, IBM's stock price is $55. At this time, the investor buys 100 IBM shares and, under the terms of the put option, sells them for $70 per share to realize a gain of $15 per share or $1500 in total. When the initial cost of the option is taken into account, the net gain is:

\$1500 - \$700 = \$800 (that is when the stock price goes down the put option becomes beneficial)

If the final stock price at the time of expiration is above \$70, there is no way that the investor will make a gain. In this case, the put option expires worthless and the investor loses \$700.

Example 2:

Let us assume the price of options on Intel, May 29, 2005; stock price =20.83

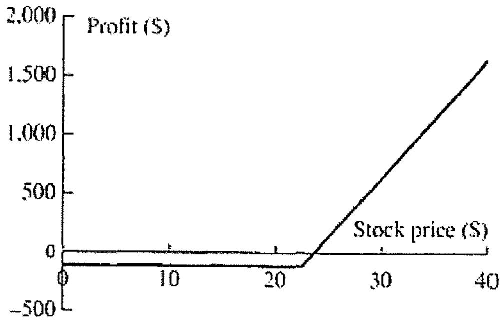
	Calls			*Puts*		
Strike price (\$)	*June*	*July*	*Oct.*	*June*	*July*	*Oct.*
20.00	1.25	1.60	2.40	0.45	0.85	1.50
22.50	0.20	0.45	1.15	1.85	2.20	2.85

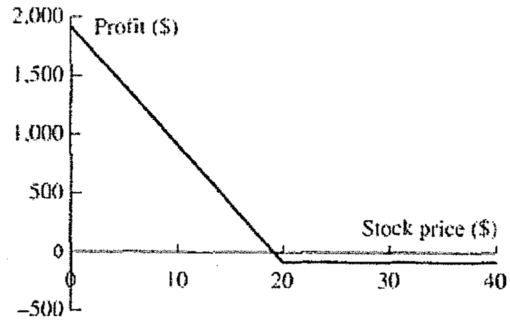
The holder may or may not exercise this right, unlike forwards and futures contracts where the holder is obligated to buy or sell the underlying asset. One difference however is, while it costs nothing to enter into a forward or futures contracts, there is a cost to acquiring an option.

The largest exchange in the world for trading stock options is the Chicago Board Option Exchange (CBOE; www.cboe.com). The above table gives the closing prices of some of the American options trading on Intel on May 29, 2005. The option strike prices are \$20 and \$22.50. The maturities are June 21, 2005, July 19, 2005 and October 18, 2005. Intel's stock price at the close of trading on May 29, 2005, was \$20.83.

Let us discuss a scenario. An investor instructs his/her broker to buy one October call option contract on Intel with a strike price of \$22.50. The broker will relay the instructions to a trader at the trading floor, who in turn will identify another trader who is willing to sell 1 October call contract on Intel with a strike price of $22.50, to agree on the price. As seen from the above table, the price for an option to buy one share is $1.15. Hence, the investor has to arrange for $115 to be remitted to the exchange through the broker. The exchange will then arrange for this amount to be passed on to the other party (writer – who is selling the call option). In this example, at a cost of $115 the investor has obtained the right to buy 100 Intel shares for $22.50 each. The writer has received $115 and has agreed to sell 100 Intel shares for $22.50 each in the event that the investor (holder) decides to exercise the option. The holder will not exercise the option if the price of Intel does not rise above $22.50 before October 18, 2005, the expiration date. In this situation the investor will loose the $115 he/she paid at the beginning. However, if the Intel share becomes worth $30 per share, the holder will exercise the option and buy the 100 shares at $2250. By selling it immediately, the investor can make a profit of $635 (3000-2250-115).

An alternative trade for the investor would be to purchase one July put option contract with a strike price of $20. From the preceding table we see that this would cost $85 to get the right to sell 100 Intel shares for $20 per share before July 19, 2005. If the Intel share price stays above $20, the investor will not exercise the option and hence will loose $85. If the price is $15 by exercising the option, the investor can make a profit of $415 (after including the initial cost) by buying 100 Intel shares at $15 and selling them at $20 per share. The following figure shows the net profit per share from (a) purchasing a contract consisting of 100 Intel October call options with a strike price of $22.50 and (b) purchasing a contract consisting of 100 Intel July put options with a strike price of $20.00





Two basic styles of options are *European and American.* A *European option* can be exercised only at maturity while an *American option* can be exercised at any time prior to maturity. The majority of exchange-traded options are American. Since investors have the freedom to exercise the freedom to exercise their American options at any point during the life of the contract, they are more valuable then European options, which can only be exercised at maturity. It is in the case of American options, allowing for the possibility of early exercise, that binomial approximations are useful. At each node, we calculate the value of the option as a function of the next period's prices, and then check for the value exercising of exercising the option now. However, for this research we have only considered European options.

Every option has a set of parameters that are required to compute the price of the option. These are strike price, stock price, risk free interest rate, period of contract, and volatility of the underlying asset. The *strike price* of a call (put) option is the contractual price at which the underlying asset will be purchased (sold) in the event that the option is exercised. The risk-free interest rate (r) is the rate at which an investment (such as simple deposit) would grow without incurring any risk to the capital. The time in years until the expiration of the option is called *maturity date.* A measure of the change (either up or down movement) in the value of underlying security over a given period is known as *volatility* (σ).

In option markets, accuracy and ability to respond quickly to the fluctuating market is vital for every active investor.

2.2 Motivation for Ubiquity Pricing

The use of mobile technology extends the nature and scope of the e-commerce providing additional advantages by enabling continuous communication and information access regardless of locale, thus resulting in continuous touch with the market. Ubiquity in the market place is concerned with the use, applications, and integration of diverse services. Many researchers attempted to address innovative research issues and possible solutions in mobile commerce. Research efforts by Varshney and Vetter [2] emphasize that mobile financial application is one of the important component of m-commerce, which can replace banks, ATMs, and manual methods by wireless aided services such as online brokerage, and micro payments, etc. Readers can find several interesting issues and research problems that are related to mobile commerce [3, 4, 5, 6]. The current research is motivated by the concept of trading options while-on-the-move and providing value-added services, which are driven by the following principles [7, 8]:

(i) Operational focus: Traders have access to handheld devices (mobile device or any wireless device) all the time. Therefore, as a value-added service, a trader should be able to compute the option price irrespective of his/her physical location. This type of trading is called Martini trading (trading any time and any-where). A recent report on Mobile Commerce (M-Commerce) by Durlacher Research states [9] “The ability to receive information and perform transactions from virtually any location is especially important to time-critical applications, such as stock and options trading as well as betting. Providing mobile traders with a similar level of access and information to that available in the fixed line environment is the key”.

(ii) Personalization: Wireless devices like mobile devices are typically operated by and configured for a single trader. Thus, the trader can receive personalized information on his/her investments as and when new information (for example, new price, new contract etc.) becomes available.

(iii) Channel Extension: Customer prefers a choice in the channel through which they do business. When a trader cannot associate himself with the main trading terminal as in cases such as (i) having power failure (ii) being away from the trading floor and (iii) experiencing a non functional terminal, a mobile device can act as an alternative trading terminal.

(iv) Handiness: Mobile devices are compact, low-cost, and have improved security thus making them popular for wireless trading.

2.3 Computational techniques for Option Pricing

Many techniques, such as the binomial tree method, the finite differencing method, and the Monte-Carlo method are being used for pricing options. In this section, we describe two of these computational techniques used for pricing options. For more information on other computational techniques, we suggest readers to refer [10-13]. We can use any of these techniques as stand-alone modules for option pricing in our architecture described in Chapter 5 and we can incorporate any of the latest pricing techniques in addition to these modules, if necessary.

2.3.1 Binomial tree method

Cox et al. [14] introduced a technique called the binomial tree method for pricing options. The binomial method captures the price movement of an underlying asset as different possible paths over the life of the option. This method assumes that the asset price changes upward by a proportional amount u (from current price S to new price Su) or moves down by a proportional amount d (from current price S to new price S_d) from a given current price S.

The method proceeds in two phases. First, asset prices are calculated (either as up or down movement of the price) at each time interval during the option's life in generating the tree towards the maturity data. Second, the pay-offs and option value will be calculated by working backward through the tree from the final time T (maturity date).

Figure 1 shows a general two-step binomial tree with stock prices. In this figure, S_0 is the initial asset (stock) price. The initial asset price S_0 can go up to S_{0u} with probability p or go down to S_{0d} with probability $(1-p)$ and so on during the small time interval dt where $dt=T/N$ and T is the total option contract period and N is total number of steps the option contract period T is divided into).

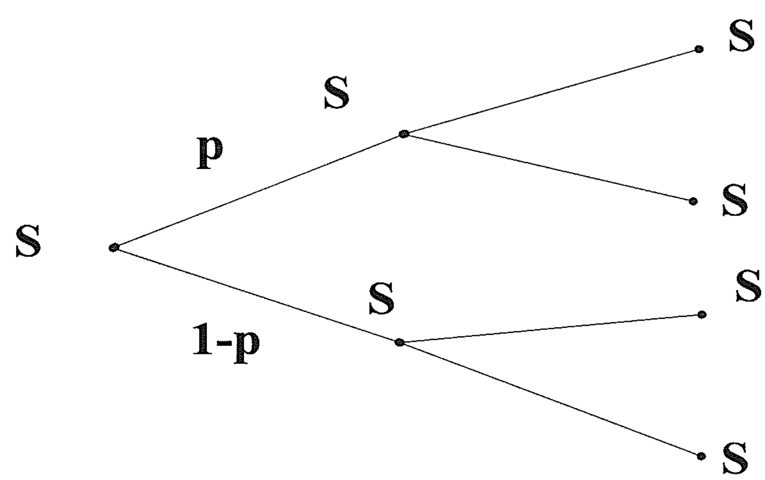


Figure 1: Stock and option prices in a general two-step tree

Cox et al. [14] defined the probability p as $p = (e^{r*dt}-d)/(u-d)$ where r is the risk-free interest rate, $u = e^{\sigma*\sqrt{dt}}$ and $d = e^{-\sigma*\sqrt{dt}}$ where σ is the volatility of asset prices.

The option value (local pay-off) of a call option at the leaf nodes (expiration time T) is given as:

$$F(S_i, T) = \mathrm{Max}[S_i - K, 0] \qquad (1)$$

where K is the strike price and S_i is the asset price at node i among the leaf nodes.

The value of the option at time $(T-1)$ and node j is given [10] as:

$$F(S_j, T-1) = e^{-r*dt}(p * F(S_{ju}, T) + (1-p) * F(S_{jd}, T)) \qquad (2)$$

Now, working backwards through the tree from the expiration time (i.e. leaf nodes), option value can be calculated. In the following, we explain the binomial tree method with a four-step binomial tree. We assume that company X had stock selling at \$31/share, and volatility of the stock price was 25% (annually) and risk free rate was 1%. Now we use binomial option pricing model to price a call option on this stock with the strike price of \$30, and contract expiring in one month. Therefore, parameters are:

S_o = 31 (stock price); K = 30 (strike price); T = 0.083333 (1/12)); σ = 0.25 (volatility); r = 0.1 (interest rate);

dt = 0.208333 (four time steps, i.e., (1/12)/4) (step size);

$u = 1.03675$; $d = 0.96455$; $p = 0.493866$

Asset price at a given (next) time step is determined by multiplying the (current) stock price by u or d, if the price is following up or down movement from the current node. In this example, initial price S_o = 31 and hence the price at 1st time step could be:

$S_u = 31 * 1.03675 = 32.13925$ or

$S_d = 31 * 0.964559 = 29.90133$

Figure 2 represents the general layout of four step binomial tree. Using the above procedure, we have constructed the whole tree that has four time steps.

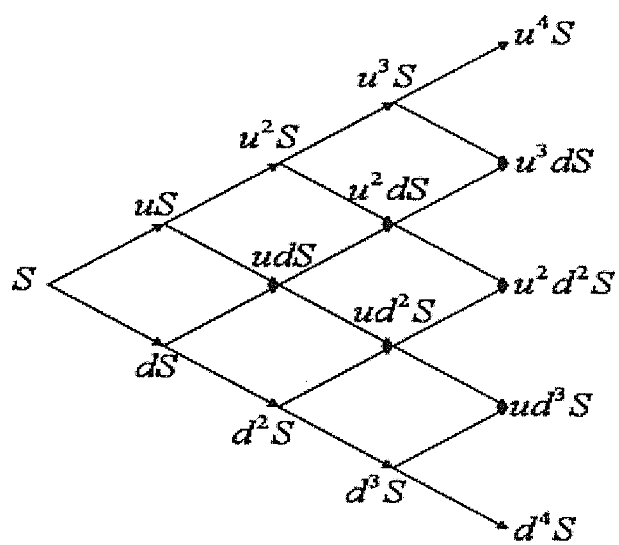


Figure 2: Lattice layout for four-step binomial lattice

Now, working backwards through the tree from the expiration time (i.e. leaf nodes), option values can be calculated at each node and fill out our lattice as shown in the following Figure 3.

1. First the local pay off at the leaf nodes (t=T) are computed using eq (1), which is (C (4, y) at the 4^{th} (leaf) time step.
2. The option value at the nodes t =T-1 are computed using eq (2). This process is repeated at earlier time steps until the initial node is reached.

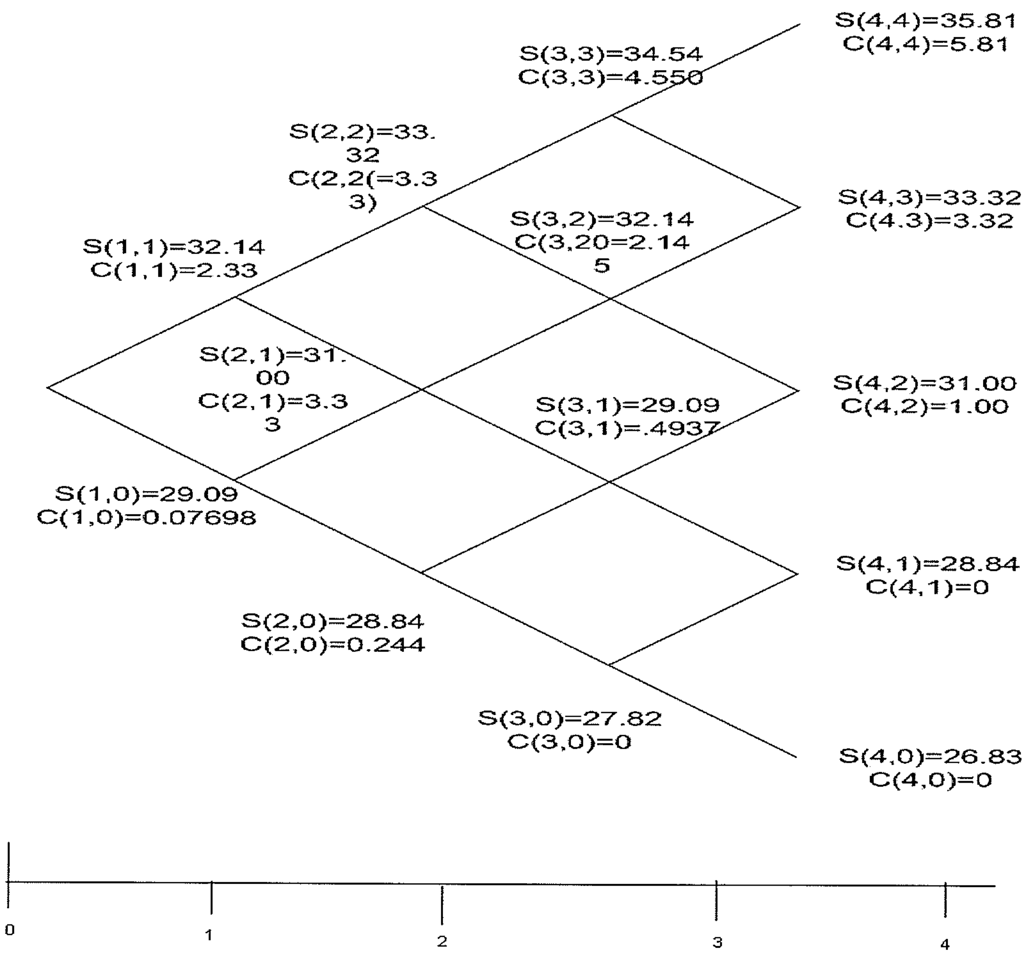


Figure 3: Stock price movements of four-step binomial lattice model

We have implemented this algorithm and have integrated with the proposed framework (Chapter 5) for this study.

2.3.2 Finite differencing technique

Finite differencing technique is one of the modern numerical approaches for pricing financial securities. There are several finite-difference schemes developed for engineering problems such as, McCormack scheme, Richardson scheme, Crank-Nicholson, forward-differencing, backward-differencing, central-differencing [15,16] that can be directly applied to finance problems. Consider the Black-Scholes model [1], a classical option-pricing model

$$\frac{\partial u}{\partial t}+\frac{\sigma^2 S^2}{2}\frac{\partial^2 u}{\partial S^2}+rS\frac{\partial u}{\partial S}-ru=0 \qquad (3)$$

where u is the option price (dependent variable- the unknown to be computed), t is time, S is the asset price (t and S are the independent variables in the problem), σ is volatility and r is the interest rate (σ and r are two of the many parameters of the problem) with initial and boundary conditions: $u(0;\ t) = 0$; and $\lim_{S\to\infty} u(S;\ t) = S$, $u(S;\ T) = max(S\text{-}E;\ 0)$ for a call option and $u(S;\ T) = max(E\text{-}S;\ 0)$ for a put option. We will only consider a call option here. The appropriate discretization for each term in equation (1) is dictated by the individual terms of the PDE together with the required precision and performance constraints. The accuracy of the results can be controlled by the use of a finer grid in the computational time direction as well as the space direction. That is, we iterate the solution process over many computational time steps until we reach a steady state solution. For our research, we have implemented FTCS finite-difference scheme to price options. This is a forward differencing in time direction and central differencing in space direction (for further details on finite-difference schemes, please refer to [15, 16].

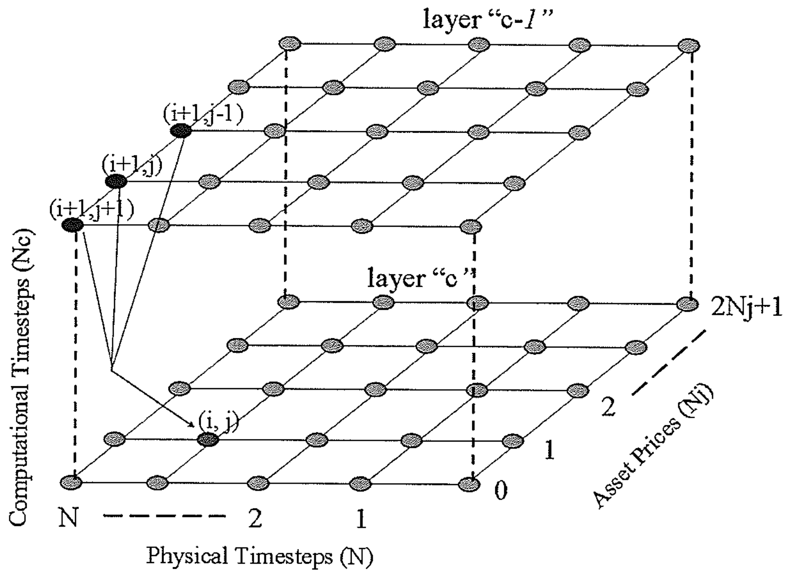


Figure 4: Option pricing computational domain

The implementation details of the algorithm are described below. The solution scheme is iterated over many *computational* time steps until it reaches a steady state. Steady state is defined as a scenario when the solution changes very little between two consecutive computational time steps. Figure 2 shows an instance during the computation of the option price. The computed values of the horizontal layer $(c - 1)$ are used to calculate the values of layer c. From Taylor series expansion [15] for the functions $u(x_o + \Delta x, y_o)$ and $u(x_o - \Delta x, y_o)$ at $x = x_o$ and $y = y_o$: $u(x_o + \Delta x, y_o) =$

$$u(x_o, y_o) + \frac{\partial u}{dx}\Delta x + \frac{\partial^2 u}{\partial x^2}\frac{(\Delta x)^2}{2!} \qquad (4)$$

$$y = y_o\text{: } u(x_o - \Delta x, y_o) = u(x_o, y_o) - \frac{\partial u}{dx}\Delta x + \frac{\partial^2 u}{\partial x^2}\frac{(\Delta x)^2}{2!} \qquad (5)$$

Adding equation 4 and equation 5 we get the central-differencing form of the second order term:

$$u(x_o+\Delta x, y_{o)} + u(x_0-\Delta x, y_{o)} = 2u(x_o, y_o) + \frac{\partial^2 u}{\partial x^2}(\Delta x)^2 + O(\Delta x)^4$$

$$\frac{\partial^2 u}{dx^2} = \frac{u(x_o+\Delta x, y_o) - 2u(x_o, y_o) + u(x_0-\Delta x, y_o)}{(\Delta x)^2}$$

Changing the notation to *i, j* gives

$$\frac{\partial^2 u}{\partial x^2} = \frac{u_{i+1j} - 2u_{ij} + u_{i-1j}}{(\Delta x)^2} + err \qquad (6)$$

By subtracting equation 5 from equation 4 and changing to notation *i, j* gives the central-differencing form of the first order term:

$$\frac{\partial u}{\partial x} = \frac{u_{i,j+1} - u_{i-1,j}}{2\Delta x} + err \qquad (7)$$

In the same way, forward and backward approximations of first order derivative can also be found by rearranging the Taylor series expansion of function $u(x, y+\Delta y)$. $\frac{\partial u}{\partial x}$ can also be approximated by forward and backward approximations as follows:

$$\frac{\partial u}{\partial x} = \frac{u_{i,j+1} - u_{i,j}}{\Delta x} \qquad (8)$$

$$\frac{\partial u}{\partial x} = \frac{u_{i,j} - u_{i,j-1}}{\Delta x} \qquad (9)$$

The number of computational layers depends on the relative error. To calculate the option values at the c^{th} layer at grid point (*i, j*) we use the values from the $(c - 1)^{th}$ layer, which can be expressed as

$$u_{i,j}^{c} = \rho u_{i+1,j-1}^{c-1} + (1-2\rho)u_{i+1,j}^{c-1} + \rho u_{i+1,j+1}^{c-1}$$

Here, $\rho = l / h^2$, where l denotes step size in time direction τ and h denotes step size in price direction x. If τ is from 0 to T and x is from x_{min} *to* x_{max}, then $N \times l = T$ and $2Nj \times h = x_{min} - x_{max}$. Therefore, note that, the terminal condition $t = T$ is transformed to the initial condition τ = 0 [17]. The relative error is calculated as given below and computation stops once the "*err*" falls below certain preset threshold value: $err = u_{i,j}^{c} - u_{i,j}^{c-1}$.

We describe the background material on M-Commerce and web mining in the following chapters through the related work discussion.

2.4 Summary

In this Chapter, we provided basic terminology of option trading and briefly described few computational techniques such as binomial lattice and finite differencing technique for pricing the option contract. In next Chapter, we describe the need for mobile technology in option trading together with some related work.

Chapter 3

Mobile Computing Developments in Trading

Mobile trading is a logical extension of e-business to address new customer channels and opportunities that deals with many services and applications in ubiquitous fashion. Even though little is reported on the confluence of derivative trading with wireless devices, researchers have recognized the need for such incorporation. In this chapter, after listing possible benefits of ubiquity in trading, we classify the use of the ubiquity into three broad aspects (i) commodity trading (ii) risk management and (iii) services and software.

3.1 Mobility in Derivative Markets

The frequent fluctuations in prices or the increasing volume of transactions are often overwhelming due to dependence on time-critical information of breaking news on a company's earnings, losses, or a change in their structure. In order to make conscious decisions in an uncertain market place, every investor needs time-critical information in a ubiquitous fashion [7]. Kargupta et al. [18] justify the needs and benefits of reporting time-critical information of stock data through wireless networks. These authors have introduced a mobile data mining system that facilitates intelligent monitoring of time critical financial data from a hand held device such as PDA (Personal Digital Assistant) and mobile. Ajenstat [19] proposed a new idea for automation of on-line derivative (stock options) trading in any place and at any time without the presence of a decision maker. More specifically, a *virtual decision maker* is a network of cooperating intelligent agents. These agents focus on some rule-based environment and expert-validated knowledge to analyze the behavior of the stock market. Furthermore, the author emphasizes that the integration of wireless commerce (option of accessibility) with web-based virtual decision-making systems will play a vital role in the future. For our research, we integrate time-critical information pervasively. That is, with predefined threshold and boundaries on the price movements of the asset in question, architecture will initiate new computation in a pervasive fashion whenever "real-time" price of the asset deviates from a predefined value or a predefined range. The "real time" prices on an asset are monitored continuously through collection of this information from on-line web sources with wireless devices. For instance, time-critical information for option pricing such as "volatility" and "prime rate" are mined from reliable web sources (presently Yahoo! Finance and Money Cafe). In addition, we have developed the architecture to choose one computational technique (at a time) among various techniques that we have implemented on back end server. These techniques exploit the time-critical information in order to compute the price of an option accurately for a particular underlying stock. In the following section, we describe some of the works related to the requirements of mobility in the derivative market.

3.2 Round-the-clock Trading

There are several factors behind the large dependence on wireless trading. Some of the most compelling factors are first, trading activity is becoming a 24-hour-a-day business - especially for those who invest in commodities have to be following the Asian and European markets, in sync with the north American market; there is higher danger of missing market movements without wireless connectivity. In other words, wireless devices make this possible, which enable the trader to track the various market movements anytime or anywhere. Second, Roche [20] states that, on black Monday (1987), one option trader lost £55,000 in 5 minutes just for leaving the market for such a short period. Research by Roche [20, 21, 22] focuses on the need for wireless services in equities and derivative markets. The author emphasizes that several market participants would prefer on-line trading while on the move, due to many advantages: (1) Traders and brokers who are not bound to desktop trading such as farmers or exporters; (2) General investors who want access to the market information while on the go; (3) Traders and brokers who need assurance of an alternative-trading terminal in case their main terminal gets disconnected.

Trading derivatives with hedging strategies has become one of the most important recent developments in the financial market. Currently, investors price options on-line from their desktop computers using various software tools [23, 24, 25]. However, once an investor steps away from the work place, he/she is disconnected from the market place. There have been some recent advancements to aid traders on the move, SMS (Short Message Service) being one of them. "Push" and "pull" models are specific M-commerce services in wireless trading. In the push model, text messages are sent to a wireless device via SMS without any previous trader request. Examples of push models include text messages sent to a mobile phone to alert clients regarding financial news. The expansion of SMS-related services reflects higher volumes of remote stock trading. Presently, a trader can subscribe to services such as CommSec [26] and Quo Trek [27] in order to be connected on the move. Technically, by subscribing to *equity-alert-services*, one can receive personalized equity information (real-time price information for personalized stock options) over a mobile phone. On the other hand, in pull messaging, mobile client invokes server-side applications and the resulting output is pulled from the web-server. However, information provided by brokerage firms, via SMS, is insufficient (for only limited information on the asset and period of contract is available through this message). Thus, the information from brokerage firms cannot be used to compute the option price for a particular option contract.

As a first step towards an organized use of mobility devices, Mobility Partner Advisory Council recently announced [28] that "The Chicago Board of Trade (CBOT) is deploying up to 10,000 wireless enabled pocket PC devices in the two years (2004-2005) to floor traders to automate the trading process". CBOT is integrating with Leapfrog technologies to introduce wireless technology for commodity and option trading.

The research effort by Web [29] focuses on software solutions for hedge fund-managers. According to Web, investors are becoming far more cautious and in fact want to see the risk management and reporting system before they invest on any asset. Moreover, they are not ready to accept basic risk analysis offered by hedge fund administrators. Roche [21] focuses on the risk management needs in the area of farm production and financing. The author emphasizes that incorporating commodity risk management into global on-line trading will significantly increase the use of commodity risk management tools worldwide. In addition, the integration of wireless communication and the commodity market, through appropriate software and hardware, will increase the agriculture derivative's business with consequent benefits.

Wireless traders do not have the time or power to browse on-line information from hand-held devices and calculate the risk level of a particular asset. However, a trader seeks personalized information to be delivered in a ubiquitous fashion. Chang and Cheng [30] emphasize a need of wireless solution for derivative trading and risk management. An International Task Force on commodity risk management explored new mobile assisted market-based approaches, in order to manage their vulnerability to commodity price fluctuations. Furthermore, Chang and Cheng [30] stress that such market-based risk management strategies increase the trading volumes of commodity derivative.

Patsystems [28] has developed software called H-trader, which assists a trader to do Martini Trading. H-trader operates on a mobile phone to trade derivatives across the world. According to Patsystems, the integration of an H-trader with some risk management tools will make more efficient trading environment. Thinkorswim [31], a brokerage company in Chicago enables the registered trader to perform trading (stock options, and other securities) via web-based, PDA, or Mobile devices. Using these services, trader can follow real-time audio commentary in stock Exchanges simultaneously has a feature of selling or buying commodities.

Windale technologies [15] and FIS-Group [16], present innovative pricing software with various option-pricing techniques that evaluate American and European style call and put options. In addition, support for both desktop environment and Pocket PC versions is available.

Ineffectiveness of these technologies is as follows: (a) The above-mentioned enterprise versions are high-priced products; (b) Frequent changes in these products require new updates to the software each time; (c) The computing technique(s) employed for option pricing and their working principles are not explained in their products to the end user; and (d) These products do not present the trader with real-time prices and other parameters (prime interest rate, volatility). Unfortunately, parameters (for example, volatility) used in the computational techniques are highly sensitive to the market fluctuations.

In the current study, we have used various computational techniques as mentioned before. For each technique, our goal is to use real-time information that is mined from reliable web-sources such as Yahoo! Finance. Proposed architecture uses the cost-effective implementation of a client and server scheme (for instance, Apache Server, and MIDP-interface development are open source environments) [29]. In addition, if there are any new updates in computation techniques, we just need to upload the techniques to the back end server rather than uploading them to client. This way, client saves the overhead cost each time when there is a new update. The design and implementation to make mobile trading a possibility using the current technology, and the proposed architecture to address this problem are presented in Chapter 6.

3.3 Summary

In this Chapter, we described about mobility issues in derivative markets and describe several factors behind the large dependence on wireless trading. In the next Chapter, we describe various web-mining techniques and their applications. In this thesis, our focus is on extracting some real-time financial data (parameters required for pricing option) to compute the option contract in ubiquitous fashion.

Chapter 4

Web Mining

Web mining finds its origins in the principles of data mining. Web mining is a method of extracting knowledge from hypertext data. Hypertext documents contain text, web links, and images. There are millions of text documents on the World Wide Web (WWW). Therefore, extracting information from the web should be done using appropriate mining tools.

According to Kosala and Blockeel [27], mining is categorized into web content mining, web usage mining and web structure mining. *Web content mining* involves extracting useful information from the data present in the web sources such as the HTML or the XML documents. *Web usage mining* determines the usage patterns of the websites by end users (persons browsing the web). Web usage mining extracts the data derived from the interactions of the users while interacting with the Web. Moreover, web usage mining is a technique, which is essentially employed to understand the requirements of the user and improve existing web services. *Web structure mining* determines the structure (the hyperlink structure) of the entire website or merely the web page. Web structure mining plays a vital role in determining the hierarchical structure of the web pages and helps in navigating through the website. Our research involves web content mining only. Therefore, further details on web usage and web structure mining are beyond the scope of this proposal. In our research, we intend to use real-time data. We intend to obtain the real-time data from portals that provide the latest market parameters on derivatives such as *strike price, volatility*, and *prime rate*. These parametric values do change periodically (prime rate) and continuously (volatility). Moreover, these values are present in the form of tables in a web page. Therefore, the information extraction strategy that we intend to use for my research falls into the category called "web table mining", and we describe this part of my strategy in chapter 6.

4.1 Web content mining

According to Liu and Kevin, some of the characteristics of the web content mining are as follows [32]:

- The amount of information available on the web is enormous and developments of upcoming technologies will eventually increase the web data.
- All kind of data are available on the Web, e.g., tables, figures, audio, video, text, hyperlinks, etc.
- In general, in a website, the representation (formats or syntax) of web data for each page is completely different. Hence, integrating data into single useful information is considered as an integral part of mining.

- WWW is usually packed and noisy with advertisements, links, navigation panels, notices, etc. For particular application, only particular data is used and remaining data is considered as noise.
- Mainly in financial related web sites, the content (web databases) of the website is dynamically updated. Therefore, monitoring those changes and extracting the updated data is an impotent issue for many services (for example, stock trading).

Some of the research efforts on web content mining are discussed briefly in the following paragraphs. Our focus is mainly on few approaches such as (i) structured data extracting (ii) unstructured data extracting (iii) information integration (iv) building concept hierarchies (v) detecting noises (vi) opinion sources

Structured data extracting is one of the important topics in web content mining. The recognition of structured data extracting is due to their representation of the essential information, for example, list of company services and their products for comparative study and Meta searching. Mining such information is useful for many applications and searching strategies. There are many approaches for structured data extracting such as wrapper generation, wrapper learning, and automatic approach [33]. In *wrapper generation*, based on the web site format patterns one can create rules to extract the data. This approach is complex and time-consuming process. In wrapper learning, user rules are applied on target items of the webpage. In *automatic fashion*, certain grammar rules are applied to extract the knowledge data. On other hand, extracting *unstructured data* is mainly based on machine learning and n*atural language processing*. In these approaches, the user can automatically give required patterns.

To make use of the extracted information of the web, one must semantically integrate multiple sources. Research efforts by Chang, He and Zang [35] proposed holistic mining in their study of dynamic on-the-go semantic discovery for large-scale integration on web databases. Their focus is on three factors such as interface extraction, schema matching and query translation.

Web size is huge. Organizing the search results of a given query from the web is an important issue. The standard method of information organization is concept hierarchy or categorization. *Text clustering* is one of the method in which search results are grouped in a hierarchical fashion. In few cases to extract the concepts or sub-concepts and their relationships, clustering is based on the organization structures, redundancy property, and semi-structure nature, for example, by using tags, language patterns, etc.

In general, Web page is a collection of many different blocks for example, navigation area, content, advisements etc. In practical applications, only some part of the web page is useful and remaining information are noises (as far as those applications are concerned). To remove those noises and identifying the main content automatically for several applications to produce much better result is a challenging part of the mining.

Mining web opinion sources is a new trend as a market explorer and analyzing agent for online sites. A recent study by Gruhl et.al [32] studies the structures and behaviours in the blogspace; by doing so, they discovered characterization of the information propagation, which allow applications to take advantage of various web facts.

4.2 Web structure mining

Web pages are usually linked to another Web page or web pages directly or indirectly as a neighbour. In such cases, determining the relationships among those Web pages is vital for discovering significant patterns to improve the navigation. Web structure mining determines link structure of the hyperlinks at the inter-document level to generate structural summary of the website or a webpage. Summary generation is mainly based on the hyperlinks, categorization of the website, and nature of the hierarchy of a particular domain. HITS (Hyperlink-Induced Topic Search) is one such algorithm that identifies relevant web pages based on the link structure [36]. In this algorithm, web is viewed as a directed graph whose nodes are documents and edges are hyperlinks between them. DIPRE (Dual Iterative Pattern Relation Expansion) framework is useful to extract relations of pair of books on the Web [37]. In addition, the same framework can be used to build a list of people, product database, bibliography works, and other useful resources.

4.3 Web usage mining

Web usage mining technique discovers usage patterns of web data in order understand user's behavioural interests and needs. Web usage mining technique has three main stages such as *pre-processing, pattern discovery* and *pattern analysis* [38].

In *pre-processing* stage, all usage data, content data, and structured data is converted into abstract data to discover patterns. The *pattern discovery* techniques are based on many algorithms that are stemming from various fields such as data mining, machine learning, and pattern recognition. Some of the valid techniques currently used for extracting knowledge data are as follows [39]: (i) statistical analysis (ii) association rules (iii) clustering (iv) classification (v) sequential patterns (iv) dependency modeling. In last stage, i.e., pattern analysis, all unrelated or uninterested patterns are filtered to extract the knowledge data. *Pattern analysis* of extracted data can be performed using knowledge query mechanisms such as SQL or loading usage data into data cubes in order to perform OLAP operations. Finally, by using visualizing techniques such as graphic patterns or an assortment of colors to different values that can highlight the significant patterns or trends in the data.

4.4 Summary

In this Chapter, we described three main techniques of web mining: (i) web content mining (ii) web structure mining (iii) web usage mining. In this thesis, to extract the knowledge information of option contract in real-time fashion we employed web content mining. Our focus is on table mining technique, which is described in chapter 5. In the next Chapter, we describe our overall architecture and its functionalities.

Chapter 5

WAMAN Software Architecture

To develop mobile infrastructure for option pricing our research focuses on three major issues: (i) mobility aspects in the derivative market, such as developing a mobile interactive interface, enabling interoperability together with Martini trading (ii) real-time mining of on-line finance data from the web sources using *Table Mining techniques* (iii) implementing various techniques (such as binomial tree method, finite differencing technique) to compute the option price, using the information in (i) & (ii). We incorporate all of the above components in order to provide client-server architecture that addresses a value-added service(s) to the trader on-the-go.

In the following sections, we explain the tasks performed by the client and server model together with their interface design and building, working approach and configurations.

5.1 Building Mobile Interface

There are few major platforms available for developing mobile applications such as BREW, Windows Mobile, Symbian, WAP (Wireless Application Protocol) and J2ME (Java 2 Micro Edition) [40]. For our research, we have developed the architecture on a J2ME platform. J2ME is supported by major carriers (for example Nokia, Motorola) and it is relatively easy to deploy the application for a trader to download and install on mobile devices. We describe in this section J2ME and its implementation details for the current study. J2ME is a stripped down version of Java aimed at machines with limited hardware resources such as a PDA or a mobile phone [41]. The J2ME platform has two configurations: CDC (Connected Device Configuration) and CLDC (Connected Limited Device Configuration). The choice of configuration will depend on the memory constraints of the particular device. CLDC targets for devices with a constrained CPU and memory, which is generally a 32-bit CPU with 160KB-512KB memory. CLDC uses the Kilo Virtual Machine (KVM), a specialized virtual machine that supports only a limited memory. CDC is targeted for devices with more resources, usually a 32-bit CPU with more than 2 MB of memory. A CDC configuration layer runs on top of the C Virtual Machine (CVM). On top of these configurations, other profiles such as MIDP (Mobile Information Device Profile) and optional APIs can be layered in order to support user interface and other network functionalities.

In the following paragraphs, we provide details about three important design issues for J2ME that are essential for our study: (i) designing and building a MIDP User Interface (UI) (ii) communication of the MIDP and back end server and (iii) security issues of the network. MIDlet is a MIDP application [42] similar to applet, a MIDlet is a managed application. A web browser manages applets, whereas, the Application-Management System (AMS) manages MIDlet. Every MIDlet class handles its own logic and life cycle, which reflects the methods of the MIDlet class. There are three possible methods in a MIDlet's life-cycle such as startApp(), pauseApp() and destroyApp(). MIDlet enters the active state after the application manager calls startApp(); MIDlet remains in the active state until the application manager calls pauseApp() or destroyApp(). In the pause-App() method, MIDlet is temporarily suspended whereas in destroyApp(), the MIDlet completely terminates the application itself and awaits garbage collection. In MIDP, UI classes are located in the javax.microedition.lcdui package of J2ME. In J2ME, commands are used to create UI objects that behave like buttons (action events in Java); commands such as OK, EXIT, and HELP are characterized by instances of command class.

Option pricing is computationally intensive. Since the required processing power and the memory are both in short supply on mobile devices, computation of option for particular asset is done on the server-end by utilizing the total functionality of Java 2 Standard Edition (J2SE). Moreover, in order to optimize the consumption of resources on mobile devices, it is desirable to keep the communication to a minimum. Therefore, the connection between the server and the device is kept open just long enough to exchange user data (for example, name of the computational technique to be used to compute the option value together with the required accuracy).

5.1.1 Network Connectivity and Security

The connection to dissimilar types of wireless devices will need different forms of connection interfaces. The Generic Connection Framework (GCF) [43] is available in J2ME/CLDC to reflect the need for small-footprint networking for a range of mobile devices. GCF is a hierarchy of interfaces defined in the "javax.microedition.io" package that allows mobile applications readily available to the trader on the network. The GCF interfaces reflect different capabilities and ensure the operations in a logical fashion. MIDP simplifies this GCF to a single connection type called HTTP (Hyper Text Transfer Protocol) and HTTPS (secure HTTP available in MIDP 2). HTTP is built around client requests and server responses, and it has two parts: header and content. The communication format (for example XML, text, and binary) between MIDlets and the back end server in the body of HTTP depends on the design of the application. We tried with GET, HEAD, and POST methods, which are simple to implement and then with XML-RPC and KXML-RPC** over HTTP/HTTPS. In our emulation via j2me wireless tool kit there is a feature of testing the speed of data transmission of particular

** XML-RPC is a standard way of implementing remote procedure calls (RPC) using XML and HTTP. To accomplish this, it uses XML to mark up all of the method and uses HTTP to transfer the methods communication format. From our observations of speed and bandwidth tests, XML tends to have heavy bandwidth between the mobile and the server rather than byte arrays (either it is a string or data of any sort).

In order to provide enough security for data transmission, we will use secure HTTP (HTTPS) provided by MIDP 2.0 (If device support MIDP 2.0, it has default HTTPS; for example, Motorola E390 supports MIDP 2.0). On top of that, to provide additional security, we use open source lightweight API called the Bouncy Castle library that supports a large number of cryptography algorithms [44]. Therefore, the mobile component of the architecture will be secured. Finally, the task of deployment of the above application (MIDlet Suites) to a specific mobile device can be done using OTA (Over-The-Air) installation of MIDlets or Infrared (IR) or Bluetooth technology.

5.1.2 Additional Services

There are several companies that provide on-line equity information to mobile devices such as CommSec [22] and Quo Trek [23]. In our research, in order to remit few parameters such as *stock price* and *maturity period*, period of the option contract to the client module, we emulate SMS service using J2ME wireless tool kit.

Before illustrating our server design, we describe the design layout of the architecture with aid of diagrammatic representation.

5.2 Server and its Functionality

In this architecture (Figure 5), a broker remits a strike price and a contract period of a particular option (with a specific asset) to a subscribed Mobile/PDA/wireless trader. The information provided by the broker is incomplete and can only partially aid in computing the fair value of the option. If the trader/investor (client) needs to decide if entering the option contract is beneficial, the client has to compute the option value by selecting the *underlying assets* and *computational technique* to be used for computation at the back end server. Client has given the choice of selecting few parameters that are used for computing options. Selection priority is given to client for the purpose of accuracy and high-speed results. Moreover, the client enters the *number of time steps* to be processed for computing the option price. The technique we employed for computing options is binomial tree method (Chapter 2). Binomial tree method is relatively speed when compared with other computational techniques. For our research, option value is computed to fourth decimal point, which is fine for academic purposes. However, in real world scenario, 10^{th} decimal point of option value is also considered so higher accuracy can be achieved by adopting simulation methods such as Monte Carlo simulation. All the above values entered by the client are sent to the web/compute server for computation.

Once the client submits time steps, underlying asset and the computational technique, back end servlet mines (using table mining technique) real-time values such as spot price, volatility, and prime interest rate from the web sources. The server then computes option price with the above real-time values, using binomial tree method and finite-differencing technique described in Chapter 2, or other computational techniques.

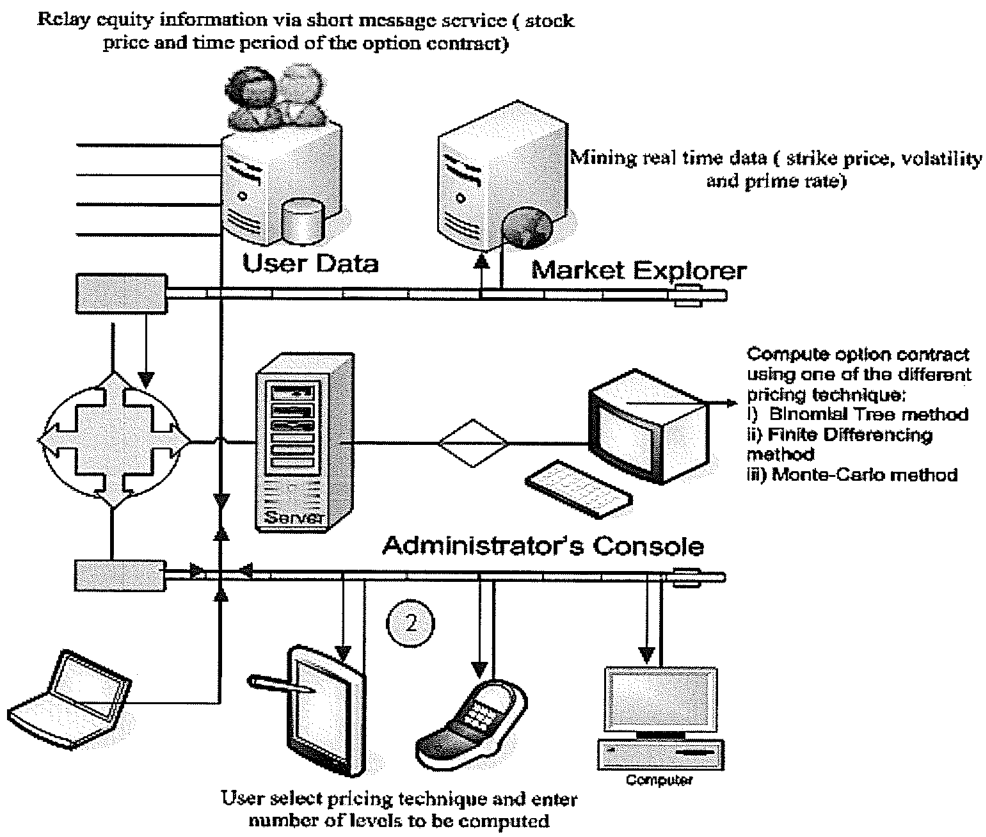


Figure 5: Mobile Infrastructure

The server program receives requests from the client. The inputs from the user comprises of number of time steps, an underlying asset and the computation technique to be used in addition to the strike price, as mentioned earlier. Web mining rules being developed for use in the web server is explained in the following subsection.

Theft and misappropriation are greatest vulnerable factors in wireless trading. Mobile devices can be easily stolen and misused which may result a financial debacle. Consequently, access control and identification of authentic client are undoubtedly vital in wireless trading. In our framework, transmission mode between client and server are secured in every aspect (as mentioned in section V). In addition, in order to access his/her portfolio, a client has to setup and will able to access their accounts that will facilitate customized tables and data analysis based on the underlying computation (for example, to access tables such as risk-free zone, healthy bids, and favored stocks which are discussed in Chapter 6). Client's devices are enabled by security token and each token generates 5-digit security code that is periodically changed, updated, and acknowledged by the trader to the server or vice-versa.

5.2.1 Mining real-time Finance Data from web sources

To extract the information from the web sources, there are number of straightforward methods. For instance, the entire source page can be passed into the search function and the required information (stock price, ASK price, volatility, prime interest rate, strike price) returned to the web server. However, when there is a change in web structure or HTML pattern, this method would fail. We have made a systematic study on the structures of both simple and complex tables, and proposed some generic web table mining rules that can be theoretically applied to specific web site domain.

```
/*sample pseudo code for data processing and data extraction*/
EXTARCT hypertext data (data extraction)
PROCESS data (pruning the raw data)
GET table (genuine table recognition)
COMPUTE TABLE (classification of table)
IF(single row with multiple cells)  or IF (multiple row with single cells)
  follow SDT rules
ENDIF
ELSE
  follow TDT rules
```

In this research, mining the real-time data (for option contract) from web sources undergoes three main stages: (i) *data extraction* (ii) *pruning the raw data* (iii) *classification of the data*. In *data extraction* stage, all raw data is extracted and temporarily stored for cleansing and processing. In next stage, the extracted data is pruned to remove unwanted data based on some assumption made by observations of web portals. Once the data is pruned, we will identify the exact table and classify the table as single dimensional table or two-dimensional table. If we know that we are dealing with single dimensional table then we use one set of rules or if it is a two-dimensional table employ a different set of rules to extract the required data from the table. All these rules and techniques are based on web table mining.

Web table mining plays a very important role in comprehending and analyzing a web document. In the current context, table mining is mining data from tables present in HTML documents. In all HTML texts, tables begin with a tag <table>. An important point is that a table tag (<table> </table>) is used for both web page formatting and representing one or two-dimensional tables with values (the ones we require for computing option price). Almost all of the web pages use tables to format their layout. Moreover, normal tables, for example tables that contain stock values, use the table tag. Wang and Hu [45] show that, although there are numerous tables, only a fraction of the tables are genuine two-dimensional tables (from authors' study, out of 11477 tables only 15% are genuine) showing consistent relationship within the cells. Apparently, identifying the genuine tables among all the tables in a web page is a not an easy task. For the current implementation, we have followed certain rules to detect genuine tables. These rules are based on the research efforts of Yang and Luk [46], and Chen et al. [47]. Our primary task is to obtain real-time information for a particular option contract. There is no standard representation for instantly recognizing these values (though they are present in tables). Some websites represent these values in a single dimensional table, while others have them in two-dimensional tables. Therefore, we have adopted a set of rules for retrieving these values from either a one or two-dimensional table specific to a particular website. In other words, we have created a parser with these rules. In particular, our focus at the current time is on three websites: Yahoo! Finance, Money Cafe, and NASDAQ. Please note, in these websites, there are no hyperactive links on stock symbols in table as there may be in some other websites.

Rules for identifying a valid single dimensional table:

Our first presumption is that a table under consideration does not have images or hyperlinks. The table does not contain heading rows or columns. While detecting any genuine table, it is very important to identify whether each cell in table has a certain limit on the number of characters. For instance, in a genuine table, each cell in general will not have more than 10 characters. A non-genuine table always exceeds 10 characters and may have numerous hyperlinks (and even images). Therefore, if any of the above rules apply to any cell in a table, we have disregarded that table and move on to the next table. A single dimensional table contains either a single row with multiple cells or multiple rows with a single cell. If this rule fails, then the table is a two-dimensional. However, a simpler rule can be framed by identifying labels (that represents parameters of the option pricing) in a table. We have identified from various stock websites that a colon always follows a label (for example, *Change:* (to represent change in the asset price between two updates or from inception or initial public offerings (IPO)). Using a label, we can identify if a table is a single dimensional table. A single dimensional table always contains a single row label or a single column label.

```
/*sample pseudo code for single dimensional table*/
 WHEN table has images or hyperlinks
   skip over to next table
 WHEN table has heading rows or columns
   skip over to next table
  IF table cell characters > 10 (table filtering)
   COMPUTE genuine table
    REPEAT for every table
    UNTIL the row containing the label followed by a colon
    IDENTIFY label (retrieving table value)
    SHOW value
 ENDIF
 ELSE
 Return false
```

Rules for identifying a valid two-dimensional table:

First, we have detected a genuine table using the above-mentioned rules. The rules for identifying single dimensional table can be extended to a 2-dimensional table as well. A two dimensional table contains more than one row label or column label. Once more than one row label or column label is identified, we concluded that the table under consideration is a two dimensional table.

Our observations reveal that a two-dimensional table containing the required values (*change, open, close,* etc) for option pricing does not exceed seven rows. Therefore, tables having more than seven rows are disregarded for my study.

Once we know the website we are dealing with (among the three websites mentioned earlier), we determined whether a single or a two dimensional table contains the information. For example, if we know that we are dealing with Yahoo! finance, then we applied the rules for two-dimensional tables. Likewise, if we use Money Cafe, we have adopted rules for single dimensional tables.

Once a genuine table is identified, the next task is to retrieve the values. This requires identifying the row containing the label. Once we identify the row that contains the required label (for example, *change:*), the immediate next cell in the row contains the required value.

Once the required parameters are mined from a table on-line, we have used them in the computational technique for pricing the option under consideration. Figure 6 is an example of mining real-time data from web sources to servlet for pricing the option contract. Required information for computing options is available on different web sources and representation is in the form of tables. Figure 6 represents the finance data in tables. In this research, we are focused only on few finance portals such as YAHOO!, IVOLATILITY, and MONEY CAFE.

```
/*sample pseudo code for two-dimensional table*/
 WHEN table has images or hyperlinks
 skip over to next table
 WHEN table has heading rows or columns
 skip over to next table
  IF table cell characters > 10 (table filtering)
   COMPUTE genuine table
 ENDIF
 WHEN table has more then one row label or column label
 SET TDT rules
 IF table rows < 7
  COMPUTE genuine table
 REPEAT for every table
 UNTIL the row containing the label followed by a colon
 IDENTIFY label (retrieving table value)
 ENDIF
 SHOW value
 ELSE
 disregard table
```

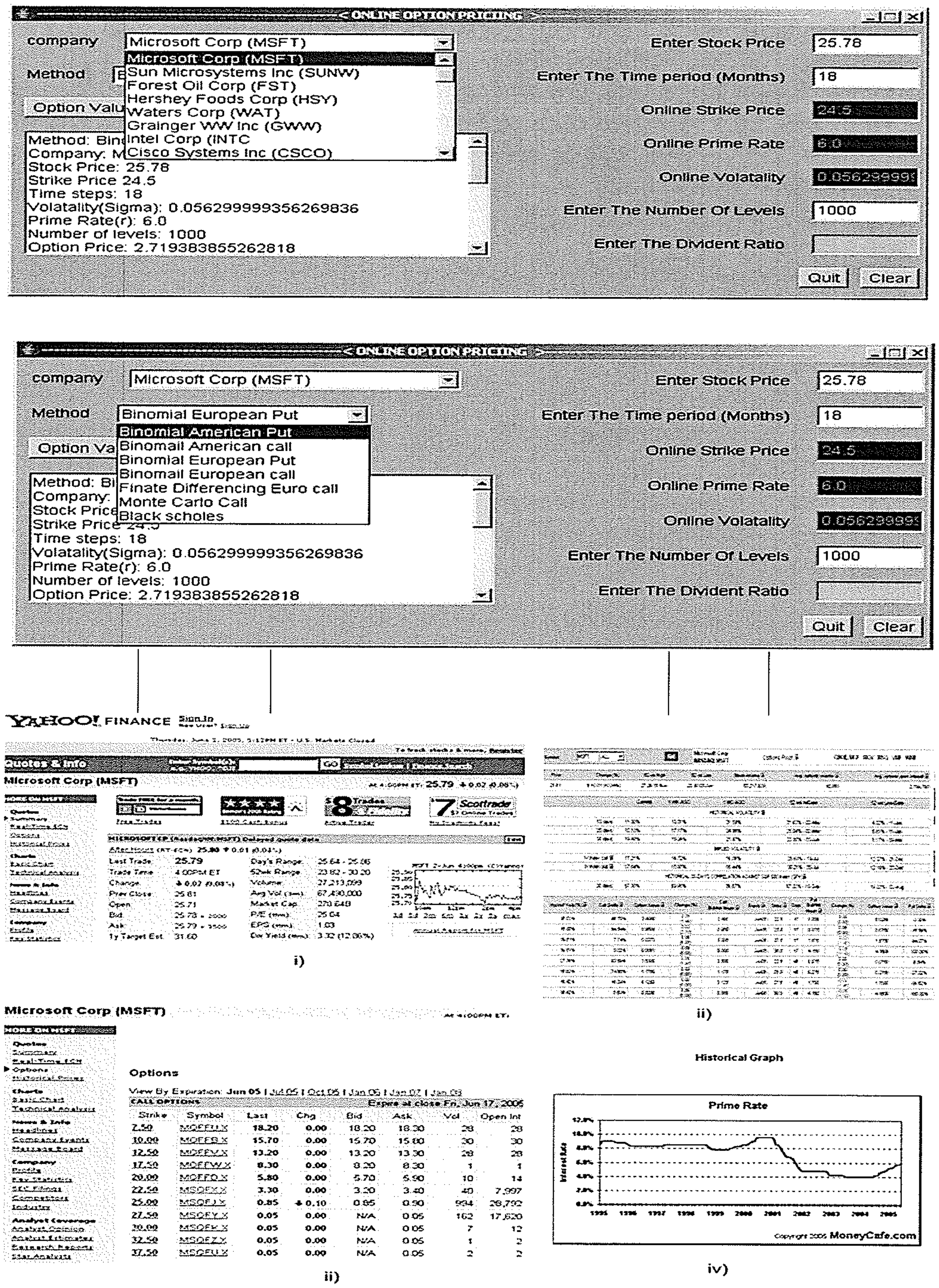


Figure 6: Mining finance data from multiple web sources

5.2.2 Computational Techniques used for pricing Options

Once the client submits time steps, underlying asset and the computational technique, back end servlet mines (using table mining technique in the previous Chapter 5.2.1) real-time values such as spot price, volatility, and prime interest rate from the web sources. The server then computes option price with the above real-time values, using any of the computational techniques described in Chapter 2 suggested by the client. Therefore, by following this procedure we can enable client with various computational techniques to avail (to provide results from approximate to accurate results) while on-the-go and to make important and effective trading decisions using the results that will ensure higher returns on investments in a ubiquitous fashion.

5.3 Beneficial factors and Enhancements

When compared with the existing pricing software, WAMAN architecture provides additional features that are beneficial for the traders for successful and profitable business. This architecture enables the trader for real-time pricing regard less of place and time. As in case when investor cannot reach the exchange while he/she is on the move, a brokerage firm can take the orders over the phone based on their personal account information. In addition, accuracy and ability to respond quickly to the arbitrary market is essential every investor. They are certain issues that option investor have to remember before he/she takes any decision on option contract. Real-time data availablity and enough background research from market explorers is the key in option trading. Moreover, investor should able to speculate the results and take effective decisions on contracts. WAMAN can autonomously get the real-time data, do necessary complex computation by itself, analyze the results, and instruct the investor with various possible scenarios in ubiquitous fashion.

Eanchements are possible in three modules of my architechture such as: (i) incorporating with efficient computational techniques (ii) improving the data extraction rules with effective data mining algorithams (iii) user navigation.

5.3 Summary

In this Chapter, we described our overall ubiquitous client-server architecture. We have employed J2ME technology to build this framework. We have proposed table-mining rules for single dimensional table and two-dimensional table to extract the data from web sources. We incorporated components from finance, web mining, and mobile computing to provide a value-added service(s) to the trader on-the-go.

Chapter 6

Results and Evaluation

In this section, we report our experimental results of the framework. We have divided our results into two sections i.e. (i) Trading scenario using PDA (ii) Trading scenario using J2ME-enabled mobile device. For this thesis, our focus is on option pricing and mobility aspects hence, these sections will only focus on computational results and trading Platforms. Discussions of network parameters such as bandwidth, performance, efficiency etc are beyond the scope of this research.

6.1 Trading scenario using PDA

In this section, we present computational results and real-time option pricing scenarios using PDA device.

6.1.1 Computational Results

In Figure 7, we present one set of results on the computed call option values. This scenario enables the client to view how the various input parameters (real-time values) vary against the output parameters with the help of 2D graphs. In addition, client has a facility to compare the outputs of the other models, which fall under the same underlying asset and option type

In our preliminary results, we notice that as the strike price increases the call option value decreases, as expected from the computation of local pay off (= *max(S-K, 0))* where *S* is at the asset price and *K* is the strike price. Validations of these results are done by simple manual calculation of a smaller tree with identical parametric conditions. Moreover, for the current study, accuracy on option value computed is to the fourth decimal place. This accuracy is sufficient for an academic exercise. However, we can obtain more accurate results, which will incur higher computational cost. Implementing the binomial lattice technique in a parallel environment [49] will circumvent the computational cost. This is not the objective of the current study, however. In addition, a client can opt for other numerical techniques for better accuracy, if required.

<u>Microsoft Corp (MSFT) CALL</u>:

We calculated option values for MSFT CALL on-line with varying strike prices (for both actual and speculated values) and stock prices.

	S = 26.20	S = 25.93	S = 25.79	S = 25.43	S = 25.25	S = 25.15
K=12	15.23294	14.96296	14.82299	14.46301	14.28303	14.183044
K=14.5	12.19873	11.92873	11.78873	11.42874	11.24874	11.148744
K=17	10.68565	10.4185	10.2819	9.247172	9.747298	9.6488172
K=19.5	8.4883	8.229661	8.101563	7.753729	7.583463	7.4891154
K=22	6.449714	6.209979	6.09804	5.772232	5.616749	5.5309261
k=27	3.89639	3.726372	3.545732	3.014906	3.270288	3.2548661
K=29	3.036333	2.890893	2.751696	2.631274	2.540916	2.4914981
k=30	2.668124	2.534558	2.413785	2.297212	2.214929	2.1697852

Table 1: MSFT (CALL) Option values for varying stock and strike prices

Table 1 presents the option values computed at various stock and strike prices. The real time data S for MSFT is 25.79 (at the time of computation of these results). To make some speculation we have computed the option values for stock, strike prices around the currently traded stock, and strike prices. These results are presented in tables 2 and 3. This is done so that the web/compute server can pick an appropriate "*healthy bid*" with which he/she can enter the option contract. In addition, calculation of a healthy bid can be done by the client (following the mechanism described below) if appropriate parameters are made available to the client from the web server.

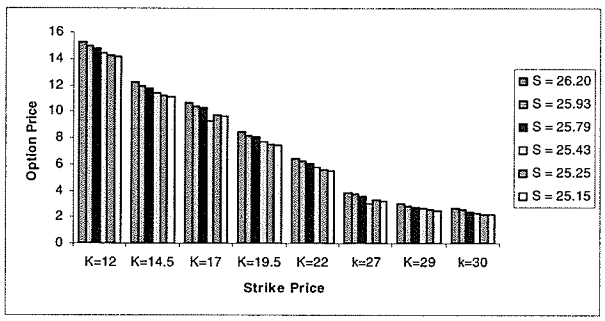


Figure 7: Call option values at various stocks and strike prices

Figure 7 shows solid lines (black and grey) that represent online values while other pattern-bars represent the speculation of the stock in increase and decrease of the stock price.

Safe Zone: To be in the safe zone, we have proposed a mechanism to compute the healthy bid based on the ASK price and Error rate of online contracts with stock prices 25.79 and 25.43. In tables 2-4, ASK prices are the real world asking prices on certain MSFT options.

Ask price can be defined as the price at which a writer is willing to sell (buy) an asset; also called the offer price. The proposed mechanism involves four steps for calculating healthy bids of the option contract.

1. Ask price is mined from online web sources (for example Yahoo!) and option price is computed (as mentioned in Chapter 5.2.1) based on other real time values. Once we have option price of the contract and online ASK value, we can calculate percentage error, mean error, and finally healthy bid.

2. Calculating percentage error: *percentage error (PE)* can be calculated as:

$$Ask\ price \quad = a$$

$$Option\ price = p$$

$$Healthy\ bid = b$$

$$\frac{a - p \times 100}{a}$$

For the current study, we consider errors within the range of (0-10%). If the error rate is more then the specified range, it is disregarded. For practical purposes, the real-time error could be improved with advanced computational techniques mentioned earlier. If error is within the specified range, we can continue with the calculation of the *healthy bid.*

3. *Mean error* is calculated as:

$$ME = \left| \frac{a + p}{2} - a \right|$$

4. If $(a - p) < 0$: $b = a - ME$ and

If $(a - p) > 0$: $b = p + ME$

Basically, an easy way to understand this is from the computed value of the option (p) the strategy is to go towards the asking price from either direction (i.e., either a<p or a>p.

Strike Price	Maturity-Period	ASK	Option price	Error-R %	Healthy BID
12	18	14.1	14.82299202	5.10%	13.73850399
14.5	7	11.5	11.78873399	2.51%	11.35563301
17	18	9.4	10.28189963	9.30%	8.9590185
19.5	18	7.3	8.101562999	10.90%	6.85905085
22	18	5.3	6.098039591	15.50%	4.90098021
27	18	2.05	3.545731939	72.96%	NA
29	18	1.2	2.751696068	129%	NA
30	18	1.05	2.413785081	129.90%	NA

Table 2: Healthy bids for option contract (MSFT CALL) when stock is 25.79

Strike Price	Maturity-Period	ASK	Option Price	Error-R%	Healthy BID
12	18	13.7	14.46300682	5.56%	13.31849659
14.5	7	11.2	11.42873735	2.04%	11.08563133
17	18	9.1	9.247171737	9.06%	9.026414315
19.5	18	7.1	7.753729176	9.20%	6.773135412
22	18	5.1	5.772231797	13.18%	NA
27	18	2.7	3.014905712	11.66%	NA
29	18	4.4	4.276165404	3%	4.3319173
30	18	4.8	4.999409394	4.10%	4.7005

Table 3: Healthy bids for option contract (MSFT CALL) when stock is 25.43

Tables 2 and 3 present various scenarios for option values and a healthy bid. These two tables are made available to the user (client) from the web/compute server. Depending on the current knowledge of the user on the behaviour of the underlying asset, the user will be able to select one of the healthy bids and instruct the broker to enter the option contract at that bid. If the writer finds this bid comfortable he/she will agree to this price and will agree to sell the underlying asset at the agreed upon strike price at the maturity date. In essence, the user (client), therefore, has used wireless computing ubiquitously to value an option and enter the contract with a level of comfort that as long as the underlying asset behaves as it has been in the past several months the user can expect a profit from the option contract. Similarly, we priced options on-line for MSFT PUT with varying strike and stock prices. The result from this study is presented in Figure 8 and Table 4.

Microsoft Corp (MSFT) PUT:

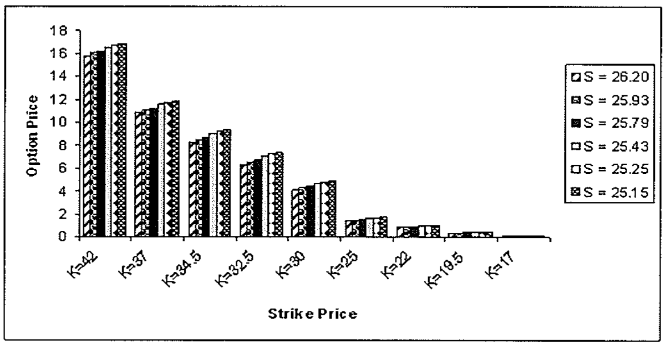


Figure 8: Put option values for varying stock and strike prices.

Strike Price	Maturity-Period	ASK	Optionprice	Error-R%	Healthy BID
42	7	16.3	16.209999	0.50%	16.3450005
37	18	11.3	11.209999	0.80%	11.2549995
34.5	18	8.8	8.709	1.04%	8.7545
32.5	18	6.8	6.711	1.32%	6.7555
30	30	4.9	4.469271582	9.60%	4.684635791
25	30	2.2	1.578788681	13.80%	NA
22	18	0.85	0.931877242	9.60%	0.809
19.5	18	0.45	0.407172418	9.50%	0.1598
17	18	0.25	0.138407292	44%	NA

Table 4: Healthy bids for option contract (MSFT PUT) when stock is 25.43

<u>**Favoured Stock:**</u> Our design also allows choosing a favourable asset (option) among many options. By favourable option we mean an option that has a well

performing underlying asset. This knowledge on favoured stock would be highly valuable to a frugal investor, who has less money at his/her disposal when he/she has to select one option among various possible options available – the options could be written on various underlying assets. Our architecture can provide this value-added service as well. This requires computation of option prices with individual underlying asset. To find out the preferred asset among various sectors, we introduced gain factor G_i, which is based on yield point. G_i is calculated using a formula: $G_i = \frac{C_i \times 100}{S_i - \bar{S}_i}$ where $\bar{S}_i$ is the *mean* (S_i); C_i is the option price; S_i stock price; $Max(G_i)$ is the favoured stock.

Symbol	Strike price	Stock Price	Historic Volatility	Option Value	Gi
MSFT	25.5	25.48	15.56%	3.137521933	4.50%
YHOO	30	37.41	24.19%	10.7066896	16.44%
MNT	45	41.53	27.78%	5.786570944	11.54%
SUN	105	108.68	25.94%	20.24165532	NA
GOOG	280	284.84	31.95%	1.121790707	0.60%

Table 5: Yield point of various companies' CALL contracts

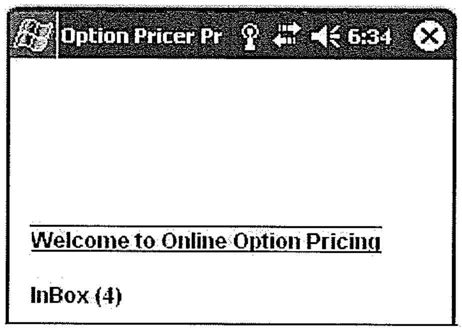
Symbol	Strike price	Stock Price	Historic Volatility	Option Value	Gi
MSFT	30	25.48	15.56%	4.52	6.55%
YHOO	35	37.41	24.19%	2.154014463	3.36%
MNT	45	41.53	27.78%	6.074323101	1.12%
SUN	105	108.68	25.94%	8.398909621	NA
GOOG	280	284.84	31.95%	31.6684338	17.49%

Table 6: Yield point of various companies' PUT contracts

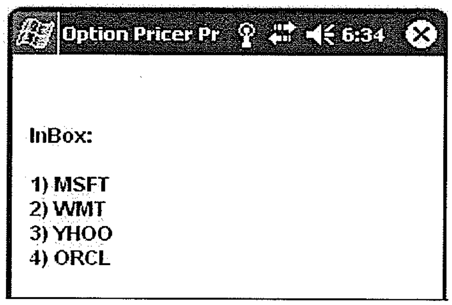
6.1.2 Trading scenarios using PDA device

The way in which a PDA communicates with a PC is referred to as *data synchronization*. The communication of PDA and the web/compute server can be done in a number of ways: (1) An USB/serial port; (2) an infrared radiation (IR); (3) Bluetooth functionality (short-range wireless); (4) WiFi functionality (medium-range wireless). For the current research, we are using WiFi as a communicator between the PDA and the backend server. We used Java programming to develop this trading scenario.

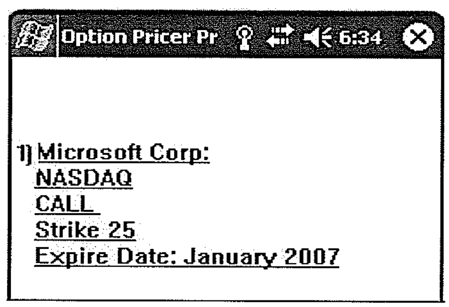
The following PDA-screens (1-5) describe the working procedure of the framework. This framework enables pricing multiple stock options with various strike prices in ubiquitous fashion. The contract/pricing information of particular stock is updated continuously to the floor-traders by aid of floor-services (exchanges) in time-to-time fashion. Once the clients get the option price from the web server, he/she will utilize built-in wireless device's small computing power to calculate healthy bids (if not provided by the web server) for various stocks in different sectors and simultaneously choose favoured stock among them.



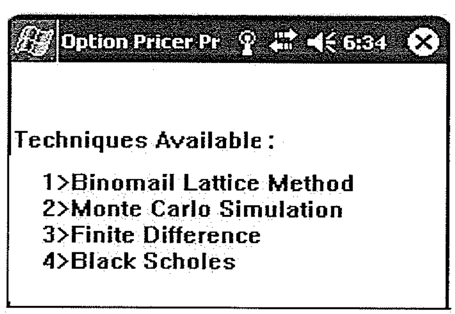
Screen1: Instant messages from the trading floor-services (exchanges)



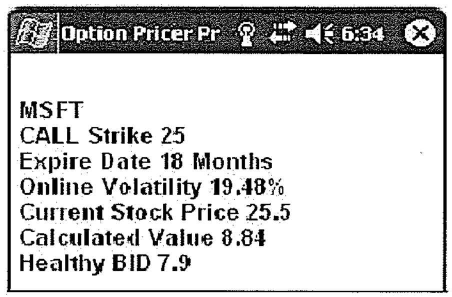
Screen 2: Active stocks on the trading floor



Screen 3: Contract details of option contract



Screen 4: Computational techniques available for pricing



Screen 5: Computed results from the web server (backend servlet)

6.2 Trading scenario using Mobile device

In this section, we present computational results and real-time option pricing scenarios using mobile device.

6.2 .1 Computational results

In Figure 9, we present one set of results on the computed call option values. The observations are same as the one in figure 7 obtained using a using PDA. In this scenario, we calculated option values for INTC CALL (as an underlying asset) on-line with varying strike prices (for both actual and speculated values) and stock prices.

Table 7 presents the option values computed at various stock and strike prices. The real time data for S is 25.52. To make some speculation we have computed the option values for stock, strike prices around the currently available stock, and strike prices. This is done to come up with a *healthy bid* to enter the option contract as presented in table 8.

	S= 26.24	S = 25.88	S = 25.52	S = 25.16	S = 25.8	S = 24.54
K=15	11.77553	11.44067	11.1058	10.77094	10.43608	10.19423
K=17.50	9.670142	9.345136	9.0259	8.706682	8.387466	8.156904
K=20	7.720975	7.401749	7.0826	6.763296	6.44406	6.213517
K=22.50	5.945113	5.67445	5.40381	5.133154	4.862501	4.667031
K=25	4.413729	4.143077	3.87243	3.601772	3.383281	3.255619
K=27.50	3.21096	3.034197	2.85743	2.680671	2.503908	2.376246
K=30	2.33158	2.154825	1.979806	1.801299	1.624536	1.502361
K=32.50	1.544512	1.457855	1.371199	1.284542	1.197885	1.134185

Table 7: INTC (CALL) option values for varying stock and strike prices

Figure 9 shows horizontal blocks patterned line (with S=25.52), which represents online values while other pattern-bars represent the speculation of the stock in increase and decrease of the stock price.

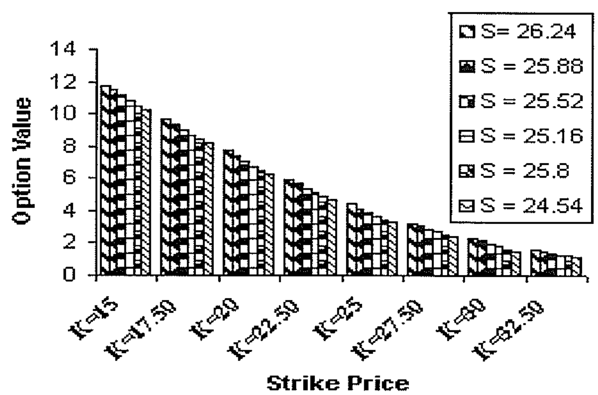


Figure 9: Call Option values at various stock and strike prices

Risk-free Zone: To be in the risk-free zone, I have set up a Healthy Bids based on the ASK price and Error rate of online contracts with stock price 25.52. As discussed earlier, we use four-step procedure for calculating "healthy bids" of the option contract. As seen in Table1, a client will be provided with various tables with speculated stock values to provide various scenarios of option values and a healthy bid. These tables are made available to the client from the web/compute server. Depending on the current knowledge of the client on the behaviour of the underlying asset, the client will be able to select one of the healthy bids (please see table 8 - an example for Intel call option) and instruct the broker to enter the option contract at that bid (these healthy bids, error rate, etc are computed using the same formula described earlier in Chapter 6.1). If the writer finds this bid comfortable he/she will agree to this price and will agree to sell the underlying asset at the agreed upon strike price at the maturity date. In essence, the client, therefore, has used his/her mobile device ubiquitously to value an option and enter the contract with a level of comfort that the investor can expect a profit from the option contract.

Strike Price	ASK-Price	Option Price	Error-R%	Healthy BID
15	11.4	11.1058	2.50%	11.25.29
17.5	9.7	9.0259	6.94%	9.3559
20	7.9	7.0826	10.34%	7.4913
22.5	6.2	5.40381	12.84%	5.8125
25	4.8	·3.87243	19.00%	4.33613
27.5	3.6	2.85743	20%	3.05099
30	2.6	1.979806	NA	NA
32.5	1.85	1.3711989	NA	NA

Table 8: Healthy bids for option contract (INTC) when stock is 25.52

6.2.2 Trading scenario using Mobile device

Mobile emulation is done on NetBeans, which is an open source platform with integration of J2ME Wireless Toolkit platform. Moreover, UEI (Unified Emulator Interface) compatible emulators allow us choosing different devices (for example, NOKIA and MOTOROLA) from various companies. Figure 10 correspond to the navigation of my interface design created in Net Beans.

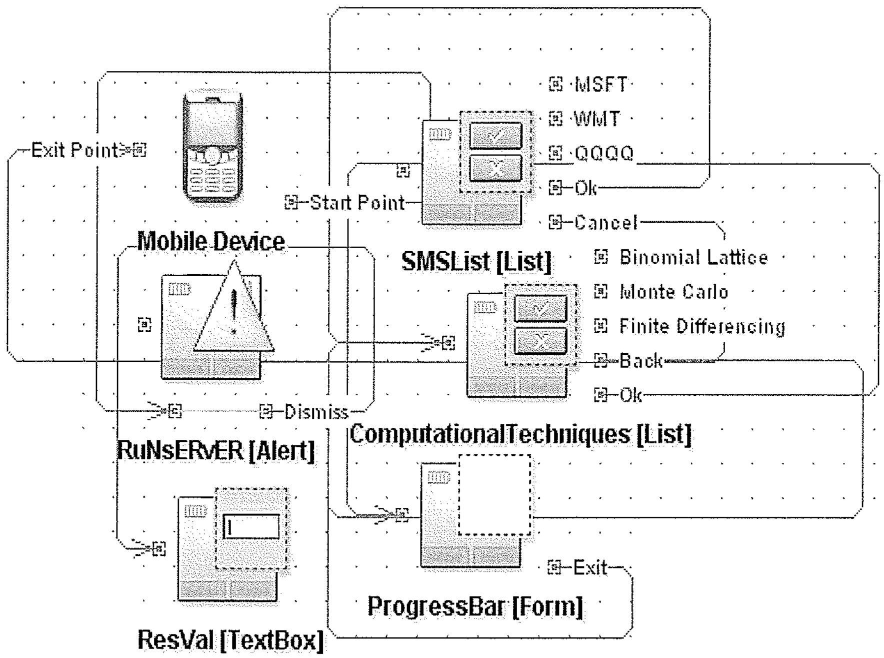
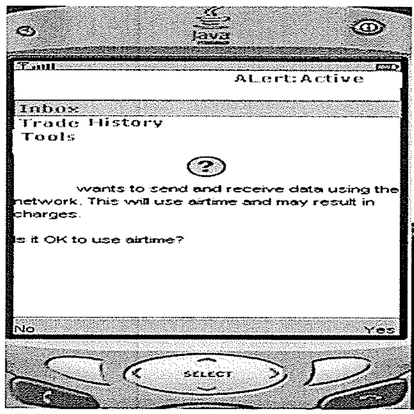


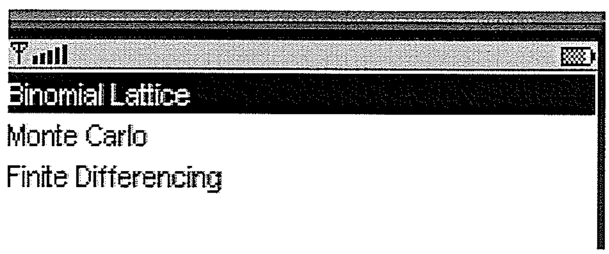
Figure 10: Flow design of mobile terminal and server response

The following Mobile screens (6-11) describe the flow design and the trading scenario. This architecture enables pricing multiple stock options with various strike prices in autonomic fashion. The contract/pricing information of particular stock is updated continuously to the trader on the move by aid of floor-services (exchanges) or brokers in time-to-time fashion. Once the clients get the option price from the web server, he/she will utilize built-in wireless device's small computing power to calculate healthy bids for various stocks in different sectors and simultaneously choose favoured stock among them.

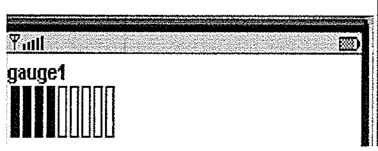


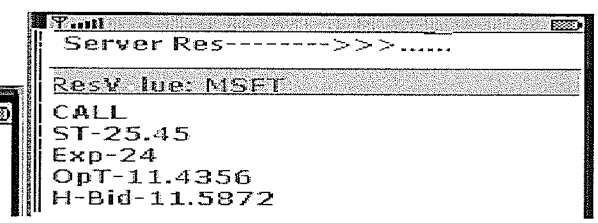
(6) (7)

Screen 6 and 7: Alerts from the trading floor-services or brokers; and Active stocks on the trading floor



Screen 8: Details of the contract and techniques available for pricing





(9) (10)

Screen 9 and 10: Response of the computed results from the web server

6.3 Summary

In this Chapter, we reported results in aspects of computational techniques and wireless trading scenarios such as PDA and J2ME-enabled mobile device. By using these computational technique's results (risk-free-zone based on healthy bids, speculated values, favoured stocks) with support of wireless framework a client can accelerate decision-making process, remain in continuous touch with the ever-changing market, and ensure higher returns on investments in option trading.

Chapter 7

Conclusions and Future work

The software architecture **WAMAN** is service based application software for individual investors, professional money managers, corporate accounting departments, and others involved in pricing of derivative securities, or risk management. WAMAN enables an active investor to price options in real-time using various computational techniques in ubiquitous fashion. The purpose of this study stems from the objective of enabling an investor to be in continuous touch with the ever-changing market while on-the-go. Fundamental challenges to design this autonomic software architecture for option trader are from three different scientific domains: option trading, web mining, and mobile computing.

Conclusively, this study is a combination of e-commerce and e-finance with modern mobile technology combined with Internet mining for successful trading in self-managing way that would pave way for effective investment trading towards higher profitability. Our objective in the current study is to enable wireless trading strategies in an autonomic fashion and ensure portability for the trader. We have experimented and validated the architecture on a PDA and on J2ME-based mobile emulators, which are applicable for limited and broad range of wireless range networks respectively.

As a value added service our architecture can also handle computation and selection of best possible bid or healthy bid, when the battery and compute power becomes larger in the hand-held devices. Also, for frugal investor our architecture can select one or more best options on various underlying assets.

This study becomes a precursor to many activities that could be continued both from the theoretical and practical aspects. In the technology side, there are now various ways of trading such as e-commerce, m-commerce, s-commerce (silent – with embedded sensors monitoring the investors practices and making decision for investment without the intervention of the investor), t-commerce (television – business through television), u-commerce (ubiquitous), v-commerce (voice). Most of these are done through w-commerce or wireless commerce. The current thesis is one of the early studies that falls under the broad categories of m-commerce and u-commerce.

To make this architecture more effective our plan is to introduce parallel computing of the various technique used to price options. This research is an initial step towards autonomic computing in the options markets. Since user intervention is required in three stages, the current study is not a fully self-managing system. In addition, we plan to introduce option sensitivities such as delta, gamma, Vega, theta, and rho, which are widely used by active traders to compute the exposures of portfolios that contain options. Each of these sensitivities measures how the real time portfolios should respond to a change in the underlying variable.

Bibliography

[1] J.C. Hull. *Options, Futures and Other Derivatives*. Prentice Hall, Upper Saddle River, NJ, 5 edition, 2002.

[2] U. Varshney and R. J. Vetter, *Mobile Commerce:Framework, Applications and Networking Support*, MONET, vol. 7, 3, 2002.

[3] U. Varshney, 2003. Location management for mobile commerce applications in wireless Internet environment. *ACM Trans. Inter. Tech.* 3, 3 (Aug. 2003), 236-255.

[4] K. Lyytinen, 2001. Mobile Commerce: A New Frontier for E-business. In *Proc. 34th Annual Hawaii international Conf. on System Sciences (HICSS-34)-Volume 9* - (January 03-06, 2001).

[5] M. Norman. Sadeh, *M-Commerce: Technologies, Services, and Business Models*, Wiley, 2002

[6] M. Sheshagiri, N. Sadeh and F. Gandon, *Using Semantic Web Services for Context-Aware Mobile Applications* MobiSys 2004 Workshop on Context Awareness, Boston, June 2004.

[7] Sun, S., Su, C., and Ju, T. 2005. *A study of consumer value-added services in mobile commerce: focusing on domestic cellular phone companies in Taiwan*, China. In *Proc. 7th Intl. Conf. on Electronic Commerce* (Xi'an, China, August 15 - 17, 2005). ICEC '05, vol. 113. 597-606; ACM Press, New York.

[8] Geoffrey. *Mobile Commerce and Wireless Computing Systems*. Pearson. Addison Wesley, first edition, 2004.

[9] Durlacher Corporation. *Mobile commerce report*. Durlacher Research Ltd, June 2000. http://www.durlacher.com/ investrel/repaccounts.aspx.

[10] A. Chhabra, P. Thulasiraman and R. K. Thulasiram, *FLEET: A Architecture for evaLuating European options in parallEl and disTributed environment*, Intl. J. of High Performance Computing and Applications, (under review).

[11] S. Barua, R. K. Thulasiram, and P. Thulasiraman, *High Performance Computing for a Financial Application using Fast Fourier Transform*, Proc.

European Parallel Computing Conference, (EuroPar 2005), Lisbon, Portugal, Aug.30-Sep2, 2005; (LNCS Vol.3648; pp.1246-1253).

[12] S. Rahmail, I. Shiller and R. K. Thulasiram, *Different estimators of the underlying asset's volatility and option pricing errors: Parallel Monte- Carlo simulation,* Proc. Intl. Conf. on Computational Finance and its Applications (ICCFA), pp.121-131, Bologna, Italy, April 21-23, 2004.

[13] R.Thulasiram, L. Litov, H. Nojumi, C. Downing and G. Gao, *Multithreaded Algorithms for Pricing a Class of Complex Options*, Proceedings (CD-RoM) of the IEEE/ACM Int. Parallel and Distributed Processing Symposium (IPDPS - April 2001).

[14] J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, 7:229–263, 1979.

[15] D. A. Anderson, J. C. Tannehill, and R. H. Pletcher. *Computational fluid dynamics and heat transfer.* Hemisphere Publishing Corporation, New York, USA, 1984.

[16] D. Tavalla and C. Randall. *Pricing financial instruments: The finite difference method.* John Wiley and Sons, New York, 2000.

[17] R. K. Thulasiram, C. Zhen and A. Gumel, " *A second order L_{0} stable algorithm for evaluating European options*", Proc. Intl. Symp. on High Performance Computing Systems and Applications, May 2004 Winnipeg, MB, Canada, pp.17-23.

[18] H. Kargupta, B. Park, S. Pittie, L. Liu, D. Kushraj, and K. Sarkar. *Mobimine: Monitoring the stock market from a PDA. SIGKDD Explor. Newsl,* 3(2):37–46, 2002.

[19] J. Ajenstat and P. Jones. *Virtual decision making for stock market trading as a network of cooperating autonomous agents.* In *Proc. (CDROM) 37th Annual Hawaii International Conference on System Sciences* (HICSS 04), Big Island, Hawaii, Jan. 2004.

[20] J. Roche. *Mobile trading comes of age.* In *United Nations Conference on Trade and Development*, September 2002.

[21] J. Roche. *The global trading village, mutual benefits for the international online trading community and developing countries.* In *UN Conf. on Trade and Development*, April 1999.

[22] J. Roche. *Web to wap: For a real start of the new millennium.UN Conf. on Trade and Development*, Oct. 2000.

[23] FinTools. *Options calculator.* http://www.fintools.com, March 2000. Montgomery Investment Technology.

[24] Windale Technologies. *Option Pricing Analysis X Software.* http://windale.com/optionsx.php3, June 2005

[25] FIS Group. *Option calculator.* www.fis-group.com, 2003.

[26] CommSec. *Keep up the market, wherever you are.* CommSec,http://www.satisfac.com.au/convCommSecsharetrading.htm, Sep. 2003.

[27] QuoTrek. *Real time on-line data.* http://www.quotrek.com /default.asp, Feb. 2003.

[28] Microsoft *Mobile Solution Partner. Pit trader work wirelessly with pockets pcs and solution from phatware*, leapfrog technologies. *In Mobility Partner Council*, San Diego, CA, 2003.

[29] A. Web. *The tools of the trade.* In *Hedge Fund Review*, London, 2004. Incisive Media Investments Limited.

[30] E. Chang and K. Cheng. *Stock options to complement stock treading.* Technical report, Hong Kong Exchanges Clearing Limited – Educational Article: Issue No. 18, October 2003.

[31] Thinkorswim Inc: stock option trading: *online brokerage services*:http://www.thinkorswim.com/tos/displayPage.tos?webpage=softwareLobby. March. 2006.

[32] R. Kosala and H. Blockeel. *Web mining research: A survey.* SIGKDD Explor. Newsl., 2(1):1–15, 2000.

[33] B. Liu and C. Kevin. C. Chang. "Editorial: Special Issue on Web Content Mining" *SIGKDD Explorations special issue on Web Content Mining*, Dec, 2004.

[34] N. Kushmerick, Wrapper Induction: Efficiency and Expressiveness. *Artificial Intelligence*, 118:15-68, 2000

[35] K. Chang, K C.-C., He, B., Li, C., Patel, M., Zhang, Z. *Structured Databases on the Web: Observations and Implications. SIGMOD Record*, 33(3), 2004

[36] S. Brin, & L. Page, The Anatomy of a Large-Scale HyperTextual Web Search Engine. *Proceedings of the Seventh International World Wide Web Conference*, Brisbane, Australia, 1998.

[37] S. Brin. Extracting patterns and relations from the World Wide Web. In *WebDB Workshop*, pp.172–183. Springer-Verlag, 1999.

[38] J. Srivastava, R. Cooley, M. Deshpande, P-T. Tan. *Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data.* To appear in SIGKDD Explorations, (1) 2, 2000

[39] R. Cooley, B. Mobasher, and J. Srivastava. Web Mining: Information and Pattern Discovery on the World Wide Web. In *8th IEEE Intl. Conf. on Tools with AI*, 1997

[40] M. Mallick. *Mobile and Wireless Design Essentials*. John Wiley & Sons, Inc., 2003.

[41] J. White. *An introduction to java 2 micro edition (j2me); java in small things*. In ICSE '01: Proceedings of the 23rd International Conference on Software Engineering, pages 724–725, Toronto, Ontario, Canada, 2001.IEEE Computer Society.

[42] R. Eckstein K. Topley. *J2ME in a nutshell.* O'Reilly & Associates, Inc., 2002.

[43] N. Kaisa, V. Mattila, and S. Ruuska. *Design: Designing mobile phones and communicators for consumer needs at nokia*. Interactions, 6(5):23–26, 1999.

[44] W. Itani and A. Kayssi. *J2me application-layer end-to-end security form-commerce*. Journal of Network and Computer Applications, 27(1):13–32, 2004.

[45] Y. Wang and J. Hu. *A machine learning based approach for table detection on the web*. In WWW '02: Proceedings of the 11th international 29 conference on World Wide Web, pages 242–250, Honolulu, Hawaii, USA, 2002. ACM Press.

[46] Y. Yang and W. Luk. *A framework for web table mining*. In WIDM '02: Proceedings of the 4th international workshop on Web information and data management, pages 36–42, McLean, Virginia, USA, 2002. ACM Press.

[47] H. Chen, S. Tsai, and J. Tsai. *Mining tables from large scale html texts*. In Proceedings of the 17th conference on Computational linguistics, pages 166–172, 2000.

[48] K. Kola and R. K. Thulasiram, UNNATI: *Ubiquitous option pricing - a fiNaNce ApplicaTIon framework*, Proc. 13th Intl. Conf. on Advanced Computing and Communications, Coimbatore, India, Dec. 2005.

[49] R. Thulasiram, L. Litov, H. Nojumi, C. Downing and G. Gao, *Multithreaded Algorithms for Pricing a Class of Complex Options*, Proceedings (CD-RoM) of the IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS - April 2001).