Chapter 7

Distributed Modelling of Financial Instruments: An EM Approach

7.0 Overview

This chapter considers distributed support for financial instruments modelling. It argues that an adequate support for modelling financial instruments should take into account the interaction between human participants in the modelling activity. Section 7.1 considers the case study of an affine interest rate model developed by N. Webber[1] using a spreadsheet application. Section 7.2 overviews technology support for the considered case study: web-based support and distributed Empirical Modelling. Section 7.3 discusses the broad implications of distributed EM technology on modelling financial instruments. The chapter concludes with the prospects of Empirical Modelling technology applying in developing distributed environments for modelling financial instruments.

[1] A senior lecturer at Warwick Business School

7.1 The Case Study

The case study considered in this chapter is an Excel spreadsheet model developed by Nick Webber for an affine interest rate model detailed in [JW00]. This case study is selected for its simplicity (as a financial model) and because of the importance of interest rate models in pricing financial instruments. The original spreadsheet model was introduced for classroom use during a one week workshop on computational finance for MSc students in economics and finance at Warwick Business School. The use of the model by practitioners is discussed in this chapter, but the research work relating to this case study was conducted without any contact with industry.

7.1.1 About Affine interest rate models

Interest rate is often referred to as the time value of money - a dollar today is worth more in one-year's time because you can invest it and earn interest on it. Interest rates are associated with all borrowing activities, which are vital to the growth of the economy. However, interest rates are subject to fluctuation over time, which poses a risk for borrowers and encourages them to resort to hedging[2] strategies [JW00]. Moreover, new financial instruments based on interest rate have recently been engineered (e.g. interest rate option, swaps, etc). Their present valuations are computed, and their marked to market valuations[3] are tracked, thus widening the range of products in the interest rate market. This, in turn, influences the shape of the yield curve[4] in different markets as well as the efficiency[5] of these markets.

In the face of these recent developments in the interest rate market, models have been built to price and hedge interest rate instruments, to conduct risk management, and to better understand interest rate movement [JW00]. Interest rate models provide a quantitative framework for describing interest rate movements and valuing and hedging interest rate products. By fitting a model to available interest rate data, we can discover the dynamics of interest rates and the way that interest rate and derivative prices are related. This helps us in better understanding, pricing, and hedging interest rate products. James (et al, 2000) summarize the current status of interest rate models as follows: "although it is desirable to use

[2] Hedging a deal involves undertaking a similar or identical offsetting deal whose cashflow is opposed to that of the first deal [JW00].

[3] The marked to market value of a financial instrument or portfolio is its value when it is benchmarked to current market prices [JW00].

[4] The yield curve or term structure of interest rates is the set of interest rates for different investment periods or maturities [JW00].

models to predict future movement in prices or rates, it is somewhat unfortunate to realise that no interest rate model comes close to achieving this goal. Despite the enormous amount of research in banks and universities that has gone into interest rates, there is no reliable method of accurately predicting tomorrow or next week's interest rates. Interest rates models cannot be used to accurately predict daily ups and down, they can merely describe distributional properties of interest rate movements. Interest rate models are generally statistically based. Valuing and hedging financial instruments brings liquidity to the market enabling the financial industry to operate."

The finance literature describes a wide range of interest rate models based on different techniques, suitable for different purposes, and working with specific types of data sets. This variety of interest rate models makes it difficult to set comparative benchmarks. This is partly because the data that is available to test interest rate models is of poor quality, and because of the huge variability in market conditions. The priority when testing and using interest rate models is pricing and hedging. In this process the model is fitted to available current market data. Its floating parameters are adjusted until the model prices for various liquid market instruments match those seen in the market. It is essential that models give accurate prices for liquid market instruments. James (et al, 2000) put interest rate models into four categories: affine models, whole yield curve models, market models and price kernel models.

An interest rate model [JW00] is defined by its state variables and by their processes. The values taken by the set of state variables in a model completely determine the state of the system. The state variables processes determined how they change through time.

All stock prices and interest rate processes are stochastic processes. They change randomly over time, but the manner in which they change can be modelled. It is possible to divide the change in their values into two parts, the first is a non-random, deterministic component, called the drift of the process, and second is a noise term – the random part, called the volatility component of the process. For financial time series without jumps, noise is assumed to be a function of a Wiener process[6].

My case study focuses on the implementation of the Longstaff and Schwartz Two-factors affine[7] term structure model originally developed by N. Webber using a spreadsheet

---

[5] Market efficiency [6&7] is determined by the ability of prices to reflect present, past, public, and private information.

[6] a certain standardized stochastic process

[7] Affine interest rate models are the most popular models in the financial industry due to their tractability and flexibility. Affine models are widely studied in the finance literature, and are defined in terms of abstract state variables. They are separated into three main types Gaussian affine Models, CIR affine models, and three-factor affine family.

application. In affine models, spot rates are affine functions of the state variables, and the state variables can be taken to be spot rates of particular maturities. To ensure that these properties hold, the dynamics of the state variables is heavily restricted. The short rate is affine in the state variables.

The Longstaff and Schwartz Two-factors affine [JW00] model has two underlying state variables x_t and y_t . The processes for x_t and y_t are:

$$dx_t = (a - bx_t)dt + c\sqrt{x_t}\,dz_{1,t}$$

$$dy_t = (d - ey_t)dt + f\sqrt{y_t}\,dz_{2,t}$$

The short rate r_t is a linear function of x_t and y_t, and the short rate volatility v_t defined by $dr_t\, dr_t = v_t dt$ also turns out to be a linear function of x_t and y_t

$$r_t = \alpha x_t + \beta y_t$$

$$v_t = \alpha^2 x_t + \beta^2 y_t$$

For constants α and β the two previous equations can be inverted and the processes for r_t and v_t written down in terms of each other

$$dr_t = ((\alpha\gamma + \beta\eta) - \frac{\beta\delta - \alpha\xi}{\beta - \alpha} r_t - \frac{\xi - \delta}{\beta - \alpha} v_t)dt + (\frac{\alpha\beta r_t - \alpha v_t}{\beta - \alpha})^{\frac{1}{2}} dz_{1,t} + (\frac{\beta v_t - \alpha\beta r_t}{\beta - \alpha})^{\frac{1}{2}} dz_{2,t}$$

$$dv_t = ((\alpha^2\gamma + \beta^2\eta) - \frac{\alpha\beta(\delta - \xi)}{\beta - \alpha} r_t - \frac{\beta\xi - \alpha\delta}{\beta - \alpha} v_t)dt + (\frac{\alpha^3\beta r_t - \alpha^3 v_t}{\beta - \alpha})^{\frac{1}{2}} dz_{1,t} + (\frac{\beta^3 v_t - \alpha\beta^3 r_t}{\beta - \alpha})^{\frac{1}{2}} dz_{2,t}$$

For certain γ, δ, η and ξ, the Longstaff and Schwartz model can be solved for the prices of pure discount bonds. In terms of r_t and v_t, pure discount bond prices are:

$$B_t(\tau, r_t, v_t) = A^{2\gamma}(\tau)B^{2\eta}(\tau)\exp(\kappa\tau + C(\tau)r_t + D(\tau)v_t)$$

where τ=T-t and

$$A(\tau) = \frac{2\varphi}{(\delta + \varphi)(e^{\varphi\tau} - 1) + 2\varphi}$$

$$B(\tau) = \frac{2\varphi}{(\upsilon + \varphi)(e^{\varphi\tau} - 1) + 2\psi}$$

$$C(\tau) = \frac{\alpha\varphi(e^{\psi\tau} - 1)B(\tau) - B\psi(e^{\varphi\tau} - 1)A(\tau)}{\varphi\psi(\beta - \alpha)}$$

$$D(\tau) = \frac{-\varphi(e^{\psi\tau} - 1)B(\tau) + \psi(e^{\varphi\tau} - 1)A(\tau)}{\varphi\psi(\beta - \alpha)}$$

$\upsilon=\lambda+\xi$, $\varphi=\text{sqrt}(2\alpha+\delta^2)$, $\psi=\text{sqrt}(2\beta+\upsilon^2)$, and $\kappa=\gamma(\delta+\varphi)+\eta(\upsilon+\psi)$
and λ is a market price of risk.

The process of fitting an interest rate model to historical and current market data is known as estimation. It is also called calibrating a model if the parameters of the models are being estimated from current market data [JW00]. All yield curves models have a number of parameters that are floating (they may be adjusted until the model fits the available data within reasonable limits).

Nelson and Siegal curves are used to fit yield curves non-parametrically. The Nelson and Siegal curve is:

$$f_0(\tau) = \beta_0 + (\beta_1 + \beta_2\tau)e^{-\kappa\tau}$$

Nelson and Siegal curves can be used to model either the forward rate curve or the spot rate curve. The spot rate curve is derived as:

$$r(\tau) = \beta_0 + (\beta_1 + \frac{\beta_2}{\kappa})\frac{1 - e^{-\kappa\tau}}{\kappa\tau} - \frac{\beta_2}{\kappa}e^{-\kappa\tau}$$

The implementation of Longstaff and Schwartz and Nelson and Siegal models (calibration, documentation, extension, re-use, and result reporting) is discussed in the following sections.

7.1.2 The spreadsheet model implementation

The spreadsheet developed by Nick Webber, implements the Longstaff and Schwartz and Nelson and Siegal models. The Longstaff and Schwartz model is calibrated to fit term structure data generated by the Nelson and Siegal model. No real term structure data (interest rate against maturity) is used. The Nelson and Siegal model term structure simulates the real data. Parameters of both models can be varied. The success of the fitting of the Longstaff and Schwartz term structure data with that of Nelson and Siegal term structure data (that simulate real term structure data) is assessed by monitoring the root-mean square difference between the data of the two models. Curves for the term structure data generated by the two models are plotted, and parameters are varied until the two curves nearly fit each other. The following picture is a snapshot of the term structure fitting model.

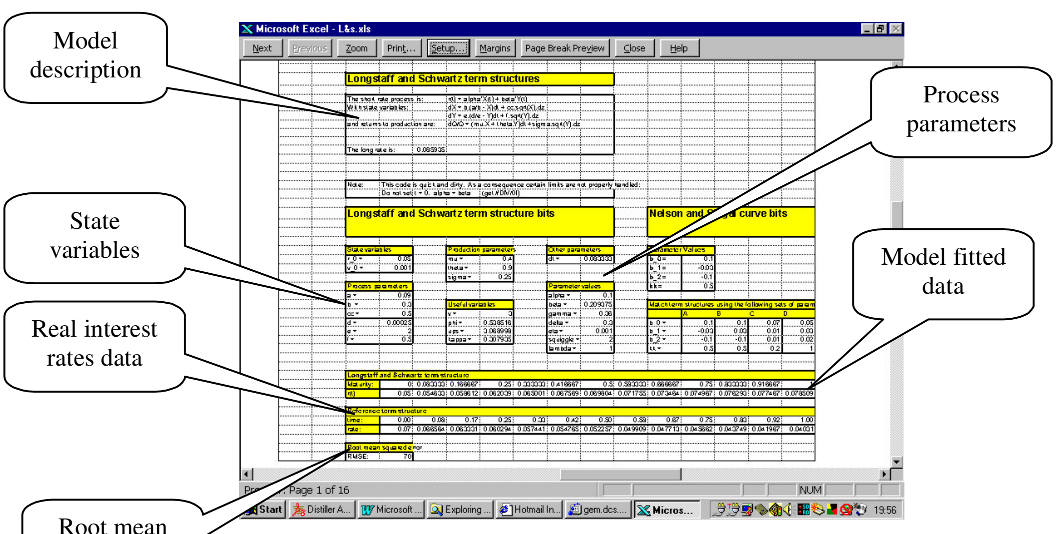


Figure 7.1 Spreadsheet implementation of the affine term structure model

The spreadsheet in Figure 7.1 is used to implement the two models. It clearly shows two parts: one part related to the implementation of the Longstaff and Schwartz model, and the other part related to the Nelson and Siegal model. The models are used to generate two interest rate series r(t) for the same set of generated maturities. Maturities are generated in consecutive spreadsheet cells starting from zero and incrementing by dt for the next cell (dt is a parameter set by the user). The spreadsheet gives a brief description of the Longstaff and Schwartz model in terms of state variables and process parameters and returns production process. The state variables and process parameters of the Longstaff and Schwartz model are each located in a spreadsheet cell. Also, the parameters of the Nelson and Siegal curve (used to simulate real term structure data) are located in their own spreadsheet cells. The figures below show a plot of the Longstaff and Schwartz and the Nelson and Siegal term structure data, and the macros used to implement the Longstaff and Schwartz model.

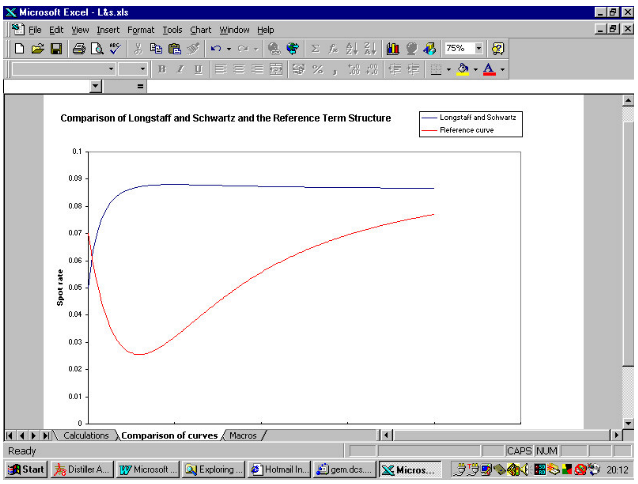


Figure 7.2 Comparative graph of the real and fitted term structure data

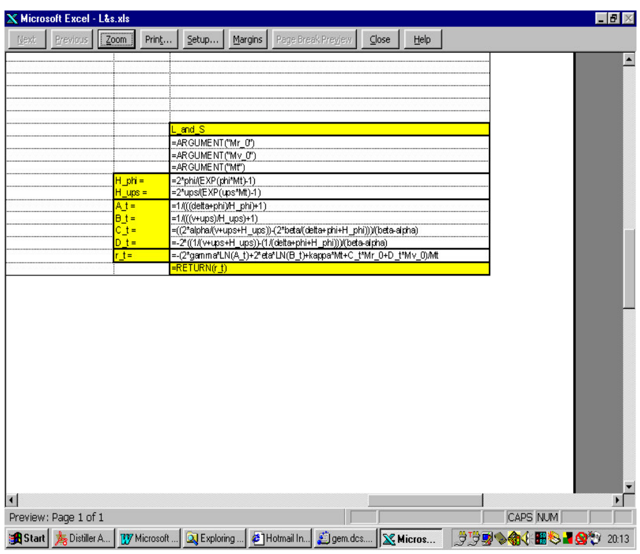


Figure 7.3 Macros used in the model

7.1.3 Classroom Observation

The spreadsheet described above was introduced in the computer laboratory for a one week workshop of computational finance given to MSc students in economics and finance at Warwick Business School.

In the computer lab, the students were supposed to calibrate the parameters of the affine interest rate model to fit Nelson and Siegal term structure data (simulating real data). The spreadsheet model estimates the error between the fitted data and the simulated real data. The supervisor introduced the spreadsheet to students by describing it on the blackboard. The students, each sitting on separate computer workstations, started varying the parameters of the model to get the best fit. They used direct face to face collaboration to discuss their results and share understanding of the model.

Some problems were encountered by students when working with the spreadsheet implementation of the two models. This was mainly due to the lack of a medium to store the result of their trials, to share the understanding of the model, and to report the results not only of the latest saved calibration of the model but of the history of their work. Sometimes good values of trial parameters were subsequently replaced by poor ones. These problems subverted the learning objective of the spreadsheet model and tended to subvert students' understanding of the relationship between the model parameters and the fitting error.

7.1.4 The use by practitioners

In a discussion with Nick Webber about the use of the considered model by practitioners to help in the hedging and pricing of financial instruments activity, the following static workflow model emerged (cf. Figure 7.4). This documentary workflow model describes the sharing of an interest rate model among the different departments of a financial institution. The sales department captures the specification of a financial product requested by a client. The term financial product used in this context refers to a financial contract (a hedging instrument, a stock, a bond, a derivative instrument). The sales department communicates the client's request to the quantitative analysis, the trading, and the risk and measurement department. These departments share and discuss different interest rate models to come up finally with the product requested by the client. Once developed, the financial product is delivered to the customer. This workflow description is not tailored to a particular financial institution. It just gives a general idea of the business process model used to develop a financial product and deliver it to a customer. At the heart of this business process model are financial instrument models similar to the ones considered in the case study.

The development of a financial product resembles to some extent the development of an engineering product. Financial experts share their knowledge to craft a product customized to the client's needs.

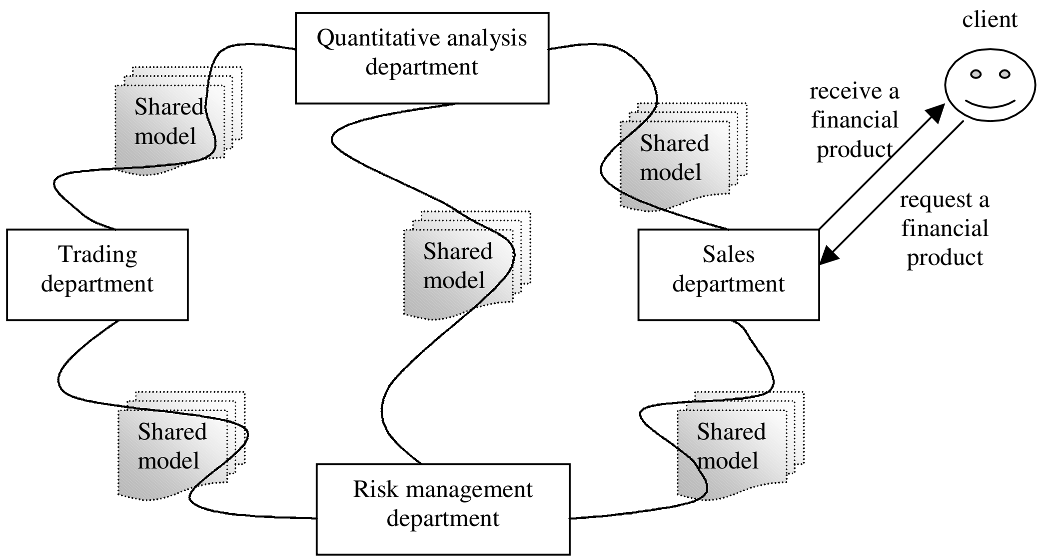


Figure 7.4 The flow across the financial enterprise of a shared model of a financial product requested by a client

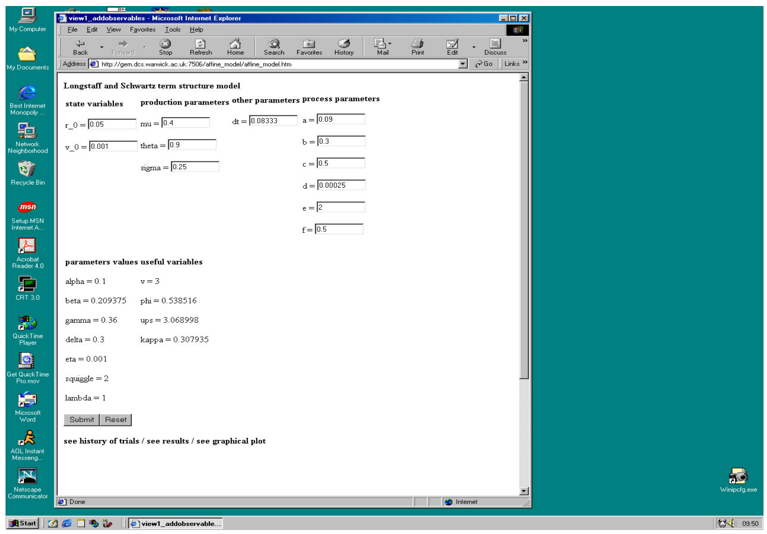
7.2 Re-engineering The Spreadsheet Model

This section presents two models developed by the author for the case study considered: a web-based model, and an EM model.
Note that the analysis in this section does not tackle financial issues related to the development of the affine interest rate model but is confined to discussing the computer-based implementation.

7.2.1 The web-based model

The first viable alternative to a single user spreadsheet model is a web based model. The author developed a web based model using (CGI) Common Gateway Interface programming written in C with the aid of some web-based libraries: CGIC (common gateway interface C library) and GD (a graphics library).

Screen shots of the web pages of the model are depicted in the figure below. The importance of the web based implementation is the rich documentation that can be supplied with the model and that can assist the user in understanding the model. The web model consists of a fill-in form to calibrate the parameters of the model. Once the user clicks on the submit button a common gateway application is invoked to evaluate the term structure data of the Longstaff and Schwartz and the Nelson and Siegal models and to estimate the root-mean-square difference between the term structures data series generated from the two models.



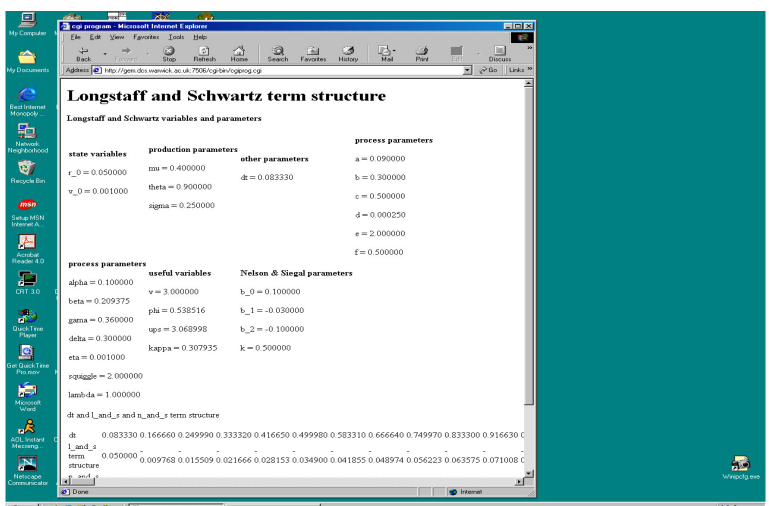


Figure 7.5 The web-based model

7.2.2 The dtkeden model

The dtkeden model was developed in an attempt to overcome the limitations of the spreadsheet and web based models described above. The qualities of this model include: i. support for interpersonal communication; ii. the de-centralisation of the modelling activity; iii. computer support for playing the roles of agents; iv. means for managing the history of model development and use (management of different trials of parameters); v. scope for enhanced visualization and exploration of output results.

i. *support for interpersonal communication:* The use of the model in a classroom or by practitioners demands support for various forms of interpersonal communication depending on the context and uses of the financial model. Interpersonal interaction is very important in the development of software systems including systems supporting the financial engineering[8] activity. This is widely recognized in [Sal87, Bos89, VF87]. The development of systems meeting various objectives and purposes relies on group social activity that involves various roles and modes of interaction among participants [AC98, Flo87, XIA98]. The research of Sonnenwald (1993) on 'communication in design' reveals that many design situations involve a group of participants interacting with each other in developing an evolving prototype of a system. Sun (1999) proposed dtkeden as a collaborative tool that facilitates interpersonal interaction and knowledge creation amongst participants. However, he stressed that computer-based interpersonal interaction should not be regarded as a replacement for conventional means of communication but as an important enhancement of existing ones.

ii. *Decentralisation of the modelling activity:* In a single user environment, the modelling activity is centred around one person. In order to derive sensible meaning from the model, the modeller needs to be able to play various roles and for this many different kinds of specialist expertise are required. For example, in the practitioner's scenario described in section 7.1.4 (cf. Figure 7.4), in the course of engineering a new financial instrument tailored to the need of a client, the modeller has to play the role of the client, the salesperson the risk analysis manager, the quantitative analyst, and the trading expert. Similarly, in the classroom scenario the modeller has to play the role of the teacher and student interchangeably. This creates a heavy load on the

[8] In this chapter, the terms 'financial engineering' and 'financial instruments modelling' are used interchangeably. Financial engineering is considered as the application of engineering principles in modelling financial instruments.

modeller and may result in an individual bias in the modelling activity. This individual bias is considered in Gruber and Sehl's shadow box experiment described in [Goo90]. This motivates the decentralisation of the modelling activity to alleviate the load on the modeller and to redress the individual bias.

iii. *Computer support for playing the roles of agents:* There is a benefit in the modeller being able to appreciate the roles of all agents participating in the financial engineering activity at first-hand. The dtkeden model can be constructed in such a way that each client can be configured to reflect the working environment of a participant (cf. the railway model). The modeller can also act in the role of a super-agent sited at the server and capable of gaining insight into the actions of participants through simulating changes to the environment, setting up scenarios, changing agent privileges, and monitoring and resolving conflicts in interaction as and when arising. Agency control is targeted with privileges for access and agency action set up by an external observer (the super-agent at the server) and enacted by internal actors (participants at the clients). In DEM, participants partake in the modelling activity by playing their own role and by collaboratively resolving conflicts in their shared understanding.

iv. *Management of the modelling activity*: Management and documentation of the modelling activity are an important consideration for tracking the history of the model development and directing its future evolution. Management of the modelling activity can take various forms. For example, managing a paper based modelling activity involves arranging the piles of papers describing the model according to various criteria. In computer-based modelling, the management activity is more elaborate and takes the form of storage, retrieval, saving, and documentation. The management of a model becomes more and more important with time-lapse and model evolution. In dtkeden, shared management of the modelling activity is possible through the communication of definitive scripts that involve file manipulation.

v. *Visual exploration of the resulting output*: tabulated results and 2D visualization give little support for the exploration and the identification of hidden pattern of relationship between data. Today, advanced visualisation techniques including the use of self organized maps [DK98], attribute explorer[9], and virtual reality [Car00, Car99], promise a richer medium to convey properties of the input and output data and the

[9] Originally developed by IBM, UK due to an idea of Bob Spence, Imperial College, London. The Attribute Explorer is implemented using EM tools by Roe (2000).

pattern of their relationship as the parameters of the model varies. Despite the lack of sophistication in visual effects in the dtkeden tool, a rich potential for interaction with the visual representation of observables in distributed views is possible (cf. The EM Attributed Explorer).

The following paragraphs describe the design and implementation of the stand alone affine interest rate model and its distributed variant for future use in a classroom context. Although the features of the distributed model are not limited to the interface but can be extended in an open ended way by sharing definitive scripts across the network, the interface design is essential to convey the features of the model. The interface is split into three parts the stand-alone, the toolbox, and the distributed part. The standalone part is the same for both the teacher and the student. It includes the parameters of the model and the graph plot. The toolbox is also the same for the teacher and student and it includes the option of updating the parameters of the model, saving the current trial of parameters, loading previous trials of parameters, and printing out the tabulated results of the fitted model. The distributed part differs for the teacher and student. The difference arises from the various roles attributed to the teacher and the student. The teacher, at the server, plays the role of the external observer in the DEM who sets the context of interaction of various internal agents (the students). The context of interaction is created by setting privileges for access assigned to various students, located each at a client workstation. The teacher can i. select a student, ii. enable / disable server propagation, iii. add / substract handles and iv. send messages to this student:

i. The teacher can select a student by typing his / her name in the `user` or `send to` input text boxes.

ii. Enable / disabling server propagation is used in the client-server architecture of dtkeden to enable / disable the propagation from the server to the client of any change made to observables at the server. By default, any change to the model introduced by the teacher is propagated to all students. If the teacher wishes to work on his/her own then there should be a mechanism that stops all propagation to the students. This involve Enabling / disabling server propagation.

iii. Adding / subtracting handles involves giving permission to a student to change or see the value of an observable (observables can be parameters in the Longstaff and Schwartz or Nelson and Siegal models).

iv. Sending messages to the students can take place by typing the name of the student in the `send to` text box and including the definition / statement to send in the send message area. Similarly, the teacher can receive message from students.

The student can select another student to send to / receive from messages. He / she can also send / receive messages from the teacher.

The following snapshot shows the interface of the teacher view in the distributed model.

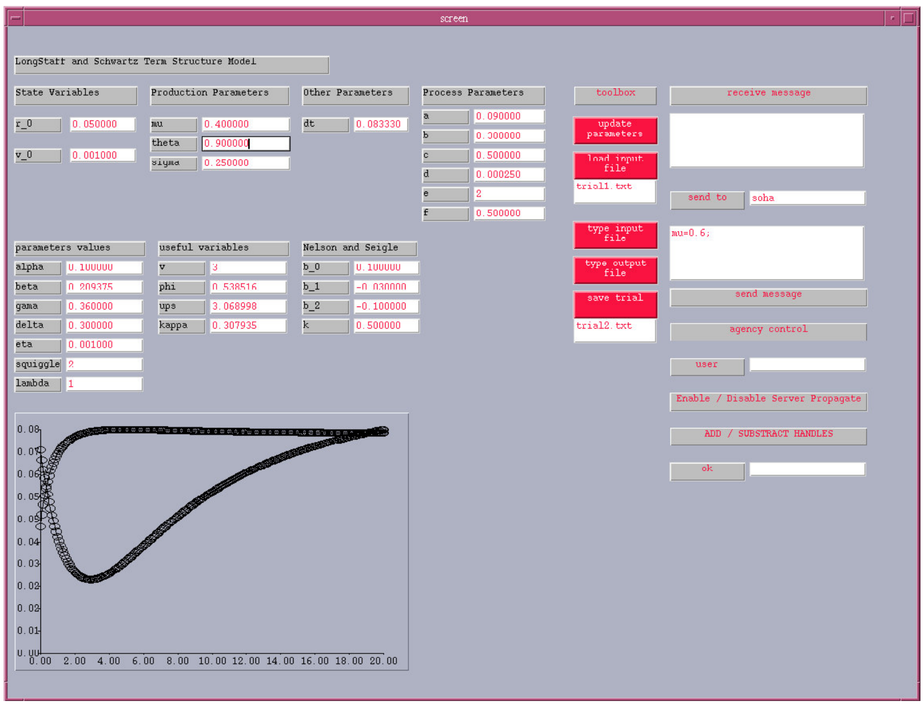


Figure 7.6 A snapshot of the teacher view in the distributed model

The interface of the teacher and student differs in the communication part as follows.

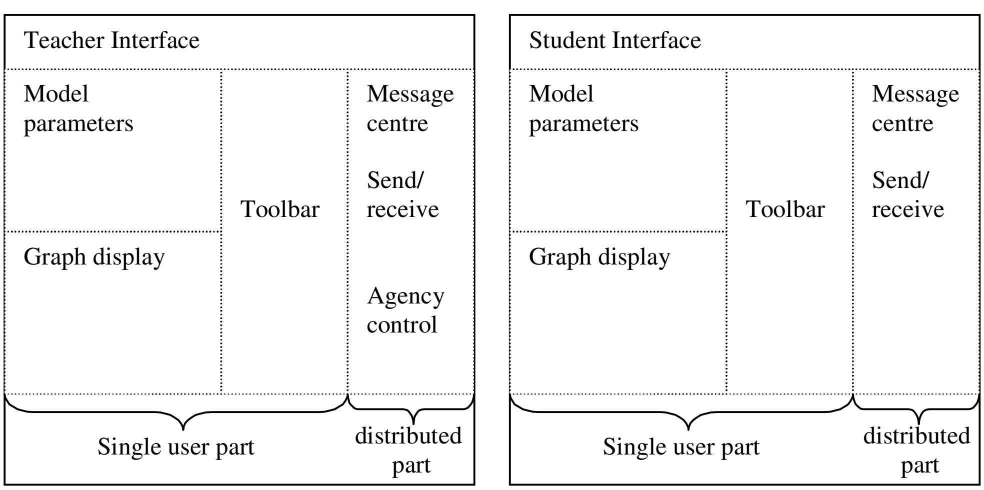


Figure 7.7 Views of participants in the distributed model

The development of the tkeden single user model is very straightforward and takes a short time once the developer has some familiarity with the tool. The interaction with the model is not limited to the visual interface. The model can enter new definitions for parameters, alter existing ones, and assign new values to parameters. This is facilitated by the tkeden window.

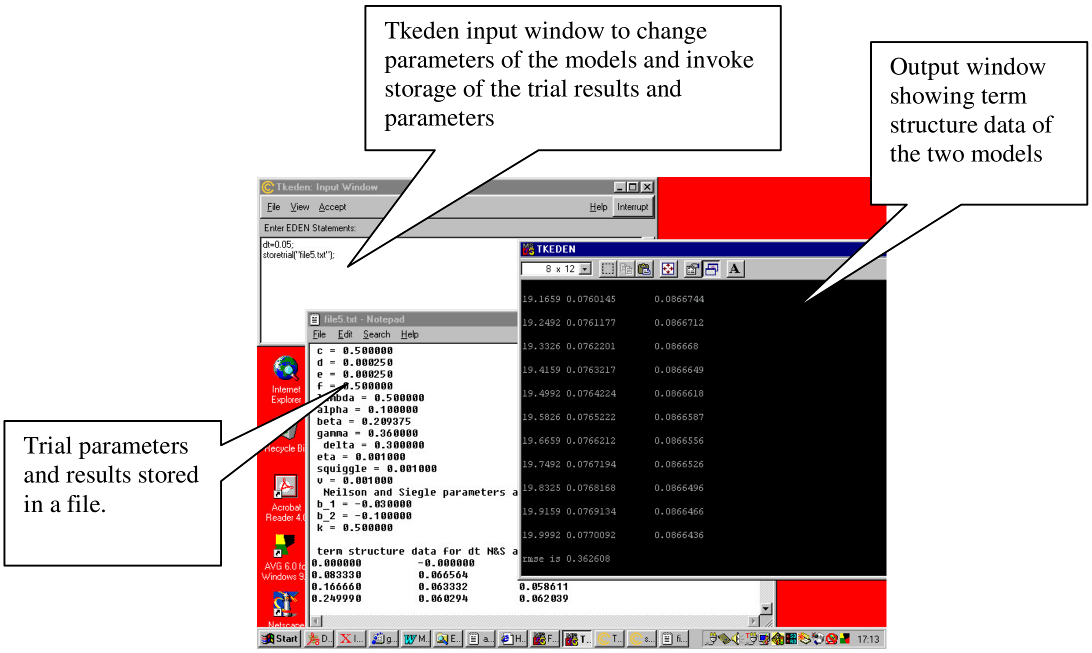


Figure 7.8 Open ended interaction with the model beyond the visual interface

Any parameter of the Longstaff and Schwartz or the Nelson and Siegal model can be changed and consequently their term structure data is changed. This is enabled through an event triggered action (cf. table below) that updates term structure data following a change in a parameter model value.

```
proc update:r_0,v_0, mu, theta, sigma, dt, a,b,c,d,e,f,lambda,b_0,
b_1, b_2, k {
 for (i=1;i<242;i++) {
   if (i==1){
   A_t[i]=0;
   B_t[i]=0;
   C_t[i]=0;
   D_t[i]=0;
   hphi[i]=0;
   hups[i]=0;
   taw[I]=0;
   nands_rate[i]=b_0+b_1;
   lands_rate[i]=r_0;
   Writeln(taw[i],"\t",nands_rate[I],"\t",lands_rate[i],"\n"); }
  Else
  {taw[I]=taw[i-1]+dt;
```

```
 nands_rate[i]=b_0+(b_1+b_2/k)*((1-exp((-1)*k*taw[i]))/(k*taw[i]))
              -(b_2/k)*exp((-1)*k*taw[i]);
 hphi[i]=2*phi/(exp(phi*taw[i])-1);
 hups[i]=2*ups/(exp(ups*taw[i])-1);
 A_t[i]=1/(((delta+phi)/hphi[i])+1);
 B_t[i]=1/(((v+ups)/hups[i])+1);
 C_t[i]=((2*alpha/(v+ups+hups[i]))-(2*beta/(delta+phi+hphi[i])))
        /(beta-alpha);
 D_t[i]=-2*((1/(v+ups+hups[i]))-(1/(delta+phi+hphi[i])))
        /(beta-alpha);
 lands_rate[i]=-(2*gamma*log(A_t[i])+2*eta*log(B_t[i])
              + kappa*taw[i]+C_t[i]*r_0+D_t[I]*v_0)/taw[i];
 writeln(taw[i],"\t",nands_rate[i],"\t",lands_rate[i],"\n");}
 rmse=rmse+(lands_rate[i]-nands_rate[i])
     *(lands_rate[i]-nands_rate[i]);
  }/*end for*/
 writeln("rmse is ",rmse);
}
```

The term structure data for the Longstaff and Schwartz and the Nelson and Siegal model is stored in a file. Results of different trials can be saved in separate files.

The multi-user model can be automatically established from the single user model by starting a server and including the teacher script at this server. Several clients can be connected to the server (within network capacity) each including the student script.

```
For (i=1;i<=CLIENT_LIST#;i++)
{sendClient(CLIENT_LIST[i],"include(\"studentscript\");");}
```

In the above script, CLIENT_LIST refers to the list of all connected clients (students) to the server (teacher). Modellers at different client workstations can change the parameters of the model in their own view and share their results by communicating their experience in interaction with the model. This communication is enabled through the send/receive feature in the message centre in the visual interface. The communication between the modellers at each client workstation is open ended. There is no pre-conceived workflow for the communication between different modellers. This makes the distributed version of the model open for use at a classroom or institutional level. The distributed model could be used to identify structured patterns of communication between modellers. Once this pattern can be repeated reliably, a more circumscribed model can emerge fulfilling a particular objective: educational, or decision support for financial product development.

The teacher (at the server) creates the context of interaction between students by adding / subtracting handles from each student and by enabling or disabling server propagation.

```
/* to enable / disable the propagation of any change to the model at
      the server from reaching the student model, the variable
                 propagateType is set to 1 / -1 */
propagateType=-1;
  /* The context of interaction is set by the teacher at the server
   through the visual interface, or by direct script inclusion */
propagateType=1;
addAgency("soha", "oracle" , "mu");
addAgency("soha", "handle" , "mu");
/* this will add agency to the client soha. This agency enables the student, soha, at the client
to see and manipulate the parameter mu. this agency could be also introduced at the server
using the lsd notation */
%lsd
      agent soha
      oracle mu
      handle mu
```

Figure 7.9 depicts the client-server configuration of the distributed model. Modellers can share experiential knowledge about the interaction with the model by passing possible parameter changes to each other or by exchanging files storing parameters and results of previous trials. Privileges to change the parameters of the model can be assigned to each modeller by declaring shared observables as being oracles or handles to a modeller.

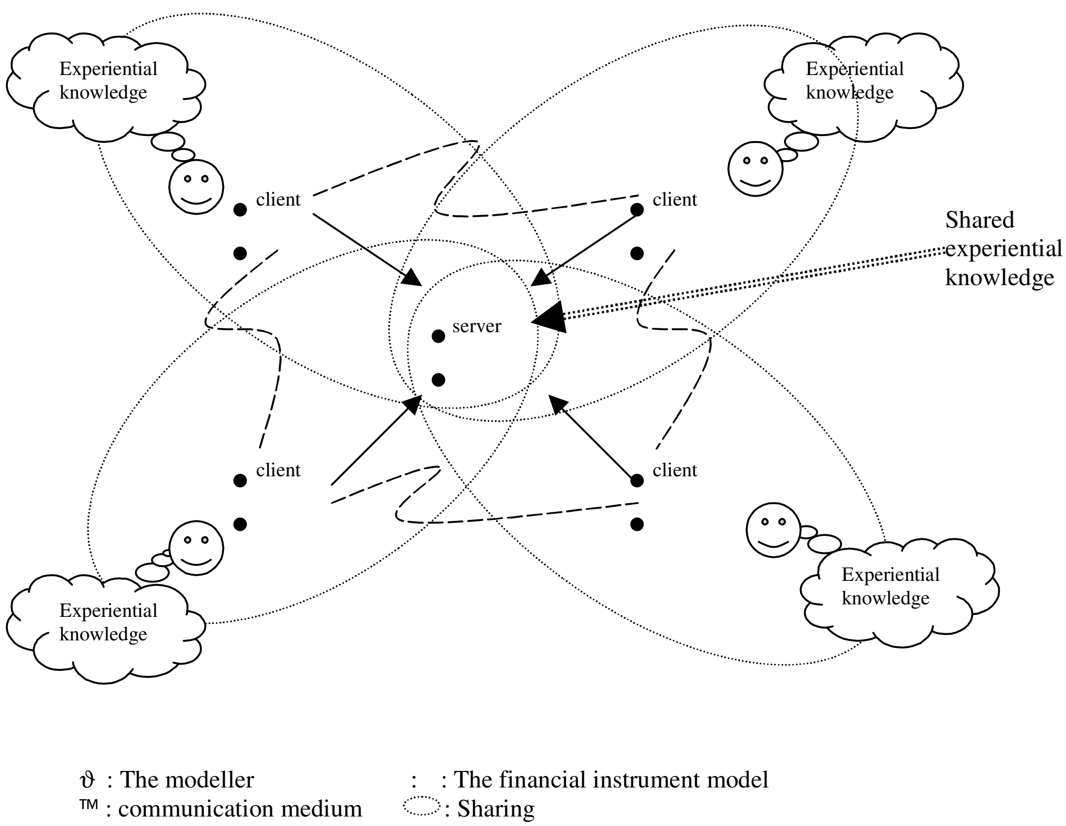


Figure 7.9 The distributed model for sharing experience

A possible communication between modellers might take the following form:

```
sendClient("soha"," dt=0.05; savetrial(\"file1.txt\");");
sendClient("meurig"," lambda=2;");
sendClient("jaratsri"," r_0=0.1;");
```

7.3 Distributed EM for Modelling Financial Instruments

Chapter 3 briefly introduced Distributed EM (DEM) enacted through the distributed EM tool dtkeden. DEM contributed to the motivation of a paradigm shift in Software System Development (SSD) and broad implications of EM in meeting the wider agenda of computing. The railway accident model, the virtual electrical laboratory (VEL), the warehouse model are case studies that reveal the potentially rich contribution of DEM to studies in computer mediated interpersonal interaction, computer-based group learning, and computer supported participative business process modelling. In these case studies, modellers are internal agents shaping the state of the model by interacting from within their own views and within their attributed agency created by an external modeller having a global view and superior agency power. Adopting a client server architecture, the internal modellers are at the clients and the external modeller is at the server.

This section discusses the claim that:

> ***"DEM has important features that motivate broad implications on principles and foundations for establishing virtual environments[10] for financial engineering[11]. This is supported by examining the wider needs of the financial engineering activity considered in its broader social context that involves a network of collaborating financial experts with various roles and privileges of action . The claim will be supported by referring to the role of DEM in the***

[10] The term 'environment', used above, designates a medium grouping a set of features and visited to achieve a particular objective. The attribute 'virtual' is used to inform that this environment is not a real physical environment but a computer-based one. This does not eliminate the correspondence between the computer-based environment and its real world referent. In the EM literature, such types of environments are often referred to as 'computer artefacts'.

[11] This would refer to the objective of distributed modelling of financial instruments.

affine interest model case study and various financial instruments modelling practices"

Distributing the modelling activity help in overcoming the limitations of centralized modelling, provides greater support for interpersonal communication and group learning and decision support, and redresses the individual modeller bias.

Distributed modelling relies on an established framework and a supporting technology for enacting this framework. Such a framework could be based on: i. a social science theory; ii. a view on the mode of interaction and role of different participants in multiagent systems; iii. the type of relationship between modellers; and iv. an appropriate definition of the role of agency in the modelling activity. The supporting technology for enacting the framework of distributed modelling relies on an appropriate network communication supporting different modes of interaction.

i. The DEM framework [Sun99] is based on the theories of distributed cognition[12] [Hut95] and ethnomethodology[13] [Gar67].
ii. Modellers collaborate as internal agents shaping the state of the model through views and privileges of actions. These privileges are set by an external modeller that creates the context for interaction.
iii. DEM supports a collaborative relationship between modellers, where individual modellers share insight in an open ended environment for collaboration. This environment is not constrained by a rigid mode of preconceived interaction.
iv. Agency gives modellers privileges to view observables in the model (oracle) and exert agency actions that introduce state change to observables in the model (handles).

The DEM framework is enacted using the dtkeden tool that features:

- Distributed communication of definitive scripts
- The support of various modes of interaction (the broadcast mode[14], the private mode[15], the interference mode[16], and the normal mode[17])

---

[12] The concept of distributed cognition represents a synthesis of cognitive, anthropological and social scientific approaches to the study of collaborative works. Its central theme involves locating cognitive activity in context, where context is not a fixed set of surrounding conditions but a wider dynamical process of which the cognition of an individual is only a part.

[13] The empirical investigation ('ology') of the methods ('method -') people (ethno) use to make sense of and at the same time accomplish communication, decision making, reasonableness, and acctions in every day life.

[14] This is the most primitive style of interaction between modellers [Sun99]. Messages are broadcast from the external modeller, at the server, to all internal modellers, at the client. This style of interaction resembles electronic group meeting without video or audio.

- Client server architecture for network communication

As discussed in chapter 4, DEM provides principles that support situated context dependent computer mediated collaboration. In the finance context, collaboration in modelling financial instruments is rich in interaction and features specific context dependent characteristics, patterns of interaction, and modes for sharing insight. This is best motivated by considering the rich level of interaction invoked in modelling financial instruments using the computer-based language developed by Jones et al (2000):

> *"a formalized language to describe financial contracts[18] is being developed with the aim of facilitating back office contract execution, accelerating in a substantial way risk analysis, graphically representing a contract as a decision tree, reminding the front-office about any potential exercise decision that has to be taken, and generating intuitive simulations for the marketing departments of investment banks". Jones et al (2000)*

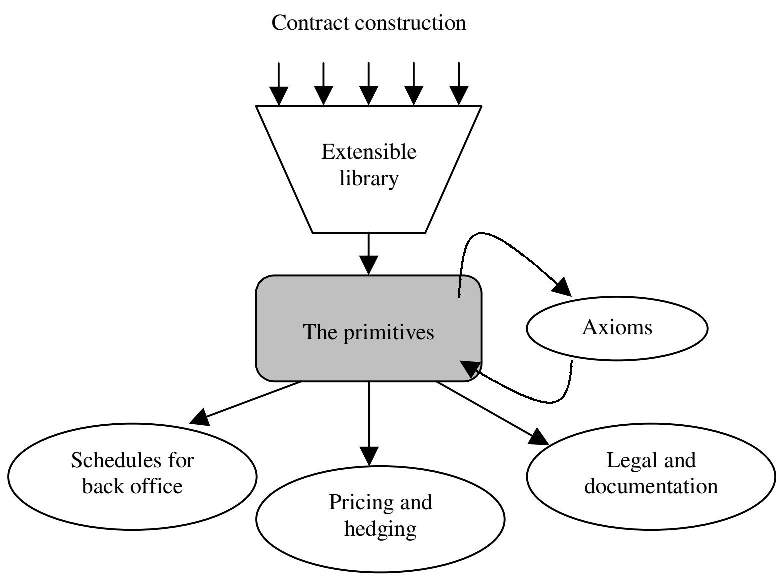


Figure 7.10 Features of the formalized language to describe financial contracts proposed by Jones et al (2000)

[15] Supports a one-to-one interaction between the external agent at the server and an internal agent at a client [Sun99]. One-to-one interaction plays an important role for managing different perspectives in a group [Son93].
[16] The interference mode [Sun99] concerns real world situations where many-to-many interactions are the norm and singular conditions require the intervention of a God-like superagent.
[17] Interpersonal interaction is mediated by the computer with reference to specified privileges of modellers to access observables.

Figure 7.10, reproduced from Jones et al (2000), illustrates the main features of the formalized language to describe financial contracts. The figure inspires a complex workflow of the financial instrument model between various departments in the financial institution (back office, front office, legal department, and pricing and hedging department).

This motivates further research and in the area of: i. computer-based interpersonal communication for modelling financial instruments; ii. modes of interaction in group financial engineering activity; iii. the role of agency in the distributed financial modelling activity; and (iv) computer-based group decision support for the financial instrument modelling activity.

Sonnenwald has raised many issues related to interpersonal interaction in design [Son93, Son99] that are of particular relevance to the DEM framework:

- The need for diverse modes of interaction between group members participating in a design process in order to develop a comprehensive understanding of the design and facilitate multiple exploration of knowledge.
- The need for diverse roles and interaction networks for inter-group and intra-group members and diverse styles of interaction between modellers in each stage of the design.
- The escalating complexity of the architecture of interaction among modellers from multiple disciplines, domains and backgrounds participating in large systems modelling and design.

The above issues raised by Sonnenwald reveal the challenge facing traditional computer-based support for interpersonal interaction that resorts to pre-conceived mode of interactions enabled through rigid interface design.

DEM considers the distributed modelling activity in its wider social context, where situated interpersonal interaction can be established at any instant through the communication of definitive scripts. The context of interaction is created through agency privileges set by an external modeller. Despite the limited modes of interaction supported by DEM, growing needs for distributed modelling can be cultivated through continuous interaction among modellers to identify a reliable pattern of interaction that may inform at a later stage traditional structured approaches to computer-based support of interpersonal interaction.

By referring to the simple case study of modelling the interest rate using the affine framework, considered in a classroom scenario, different lessons can be learned:

- An external modeller is needed to create the context of interaction. This could be the teacher in a classroom scenario or the general manager in the financial enterprise

[18] In the finance literature the terms ' financial instruments' and ' financial contracts' are used interchangeably.

scenario. This external modeller sets the privileges for interaction of various modellers in a situated way.

- Styles and modes of interaction among a group of participants are difficult to frame in prescribed actions. Various needs of interpersonal interaction can be supported within a flexible distributed modelling framework. This framework can accommodate various modes of interaction as and when they arise.
- Distributing the modelling activity serves a better educational, operational, and decision making objective in the finance context.

7.4 Summary and Outlook

This chapter considers the case study of an affine term structure spreadsheet model originally developed by Nick Webber. Two models are developed using web-based and distributed EM Technology.

Empirical Modelling technology addresses basic issues related to the requirements engineering of computer-based environments for modelling of financial instruments. Its basic contributions are in laying principles and foundations for a distributed framework for modelling financial instruments and in presenting a proof of concept technology for enacting this distributed framework. Such an environment would better support the financial instrument modelling activity and widen the contribution of the computing activity beyond the implementation of the mathematics of the financial model to providing group learning, group decision support and knowledge construction.