

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
28 February 2002 (28.02.2002)

PCT

(10) International Publication Number
WO 02/17125 A1

(51) International Patent Classification⁷: G06F 17/18

(21) International Application Number: PCT/CA00/00961

(22) International Filing Date: 21 August 2000 (21.08.2000)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant and

(72) Inventor: JOHNSON, Neil [CA/CA]; 116 LaScelles
Boulevard, Toronto, Ontario M5P 2E4 (CA).

(74) Agent: PILLAY, Kevin; Fasken Martineau DuMoulin
LLP, 4200 Toronto Dominion Bank Tower, Box 20,
Toronto-Dominion Centre, Toronto, Ontario M5K 1N6
(CA).

(81) Designated States (national): AE, AG, AL, AM, AT, AU,
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ,

DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR,
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,
NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM,
TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

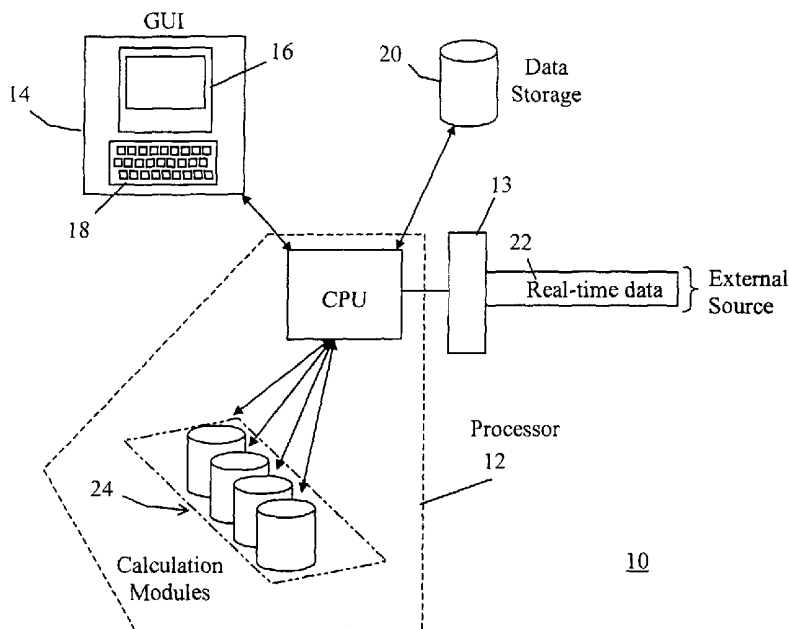
(84) Designated States (regional): ARIPO patent (GH, GM,
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European
patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE,
IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG,
CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHOD FOR PROCESSING TIME SERIES DATA



(57) Abstract: A method for modeling time series data, comprising the steps of: establishing a library of functions, each function having one or more attributes, the attributes determining the parameters of the function including its input and output values; representing ones of the functions and its associated attributes as icons on a graphical user interface; establishing a connection between at least one of a selected function and the time series data by a drag and drop operation performed on an attribute of the selected function; representing the connections graphically on the graphical user interface to create a model; and implementing the model in a processor.



WO 02/17125 A1

METHOD FOR PROCESSING TIME SERIES DATA

This invention relates generally to a system and method for analyzing time series data, and more particularly to a graphical user interface for implementing such a system.

5

BACKGROUND OF THE INVENTION

Time series data is typically comprised of discrete data values occurring at fixed intervals in time. Time series data can include measured data from the processes of an industrial plant, measured data of economic phenomena such as stock prices, volumes and the like. Many techniques are used in the analysis of time series data.

10

Depending on the type and frequency of the data, the analysis may be used for performance monitoring as well as the prediction of physical and business systems. Various types of software have been developed to help intelligently manage time series. Such software includes accounting packages, econometric modeling tools, statistical and neural network suites as well as dedicated programming languages. However, this current software does not allow users to manage time series and models in an intuitive and easily manageable way. This is especially true for particularly complex and interrelated data.

15

Managing high volumes of numbers and the complex relationships between them is a difficult and demanding task. Existing software often requires people who have an extensive understanding of complicated programming languages, years of experience working with the various algorithms involved, as well as an in-depth understanding of the field or business that has generated the data in order to complete such a task.

20

General-purpose tools such as spreadsheets are in wide use today to manage time series data. However, they are severely limited in several important aspects. First, they are unable to store the large volumes of data that are typically required and have severe limitations in their ability to handle real time streaming data. It is also a complicated process to estimate equations, use time series of different frequencies or manage complex models. Further, some programming specific functions between spreadsheet cells is required.

25

Specific purpose software tools exist that can perform complicated time series manipulation. These include programs such as Matlab™ and EPS™. However, they are intended for technical users and require learning a distinct programming language that few

30

people are familiar with. Accordingly, there is a need for an analysis tool that is both relatively easy to use without having to learn a specific programming language and which offer the ability to apply computer analysis to time series data.

5 It is an object of the present invention to obviate or mitigate some of the aforementioned problems.

SUMMARY OF THE INVENTION

In accordance with this invention there is provided a method for modeling time series data, comprising the steps of:

- 10 (a) establishing a library of functions, each for performing a computation or input data, and each function having one or more attributes, the attributes determining the parameters of the function including its input and output values;
- (b) representing ones of the functions and its associated attributes as icons on a graphical user interface;
- 15 (c) establishing a connection between at least one of a selected function and the time series data by a drag and drop operation performed on an attribute of the selected function;
- (d) representing the connections graphically on the graphical user interface to create a model; and
- 20 implementing the model in a processor.

BRIEF DESCRIPTION OF THE DRAWINGS

An embodiment of the invention will now be described by way of example only, with reference to the accompanying drawings in which:

25 **Figure 1** is schematic block diagram of a system according to an embodiment of the invention;

Figure 2 is a schematic block diagram of the functional blocks of the processing system according to an embodiment of the present invention; and

Figure 3 is a schematic view of the graphical user interface;

30 **Figure 4** is a graph illustrating the advantage of using a rollback;

Figure 5 is a schematic block diagram of general functional blocks of the system;

Figure 6 is a schematic block diagram of general functional blocks of a multi-tiered variation of the system illustrated in figure 5; and

Figure 7 is a schematic block diagram of an embodiment of the invention as it applies to an Internet application.

5 DESCRIPTION OF THE PREFERRED EMBODIMENT

For convenience in the following description, like numerals refer to like structures in the drawings.

A system according to a general embodiment of the invention is shown generally in figure 1 and represented by the numeral 10. The system 10 includes a processor 12 for receiving
10 an external data input 22 via an interface 13, a data storage 20, a graphical user interface (GUI) 14, coupled to the processor for displaying information to a user via a monitor 16 and a keyboard 18 or other input device for providing the user inputs to the system. The processor 12 includes calculation modules 24 for implementing computation and mathematical functions which may be applied to time series data and generated data. The calculation modules may be combined in a
15 manner to be described below for establishing complex rules and relationships between various time series calculations. The input data 22 may include both real-time and non real-time time series data. The processor 12 manipulates the data 22 according to the sequence determined by the arrangement of the calculation modules 24. The functions performed by the modules 24 are displayed graphically by way of icons on the monitor 16. Thus, for example, an averaging
20 function may be represented by a specific icon, while a variance function may be represented by another icon. Each function also includes a set of attributes, which are the input and output parameters of the function as well as preset values for a function. The icons provide a convenient graphical representation, which may be moved on the display screen in order to construct a variety of models. The icons provide the user with a method for selecting inputs for
25 the modules and for direction of the subsequent output. The system provides the ability to perform calculations simultaneously on independent sets of generated models, functions and equations. It makes use of computing platforms that use multiple threads and processors. The results are stored to data storage 20 and displayed to the user via the monitor 16.

The overall general functionality of the system is described with reference to figures 5
30 and 6. Typically, a user is presented with a graphical user interface. The user interface may be provided on a local network or on an Internet web site. Generally, it is desirable that the user

input a password in order to access one or more folders or portfolios in the user's account. In one embodiment, the portfolios or folders represent data files or objects, which have properties including names of securities, quantity of shares, and their current value.

Referring to figure 5, the user selects a number of the portfolios, which are combined into a combined portfolio by the system. Based on the type of information in the combined portfolio, an integrated data object is created. The integrated data object contains all the dynamic data variables that may be contained in the combined portfolio. For example, such variables include the price, volume, and trading history of a particular stock in the portfolio. The current values for the integrated data are then retrieved from a database. Data having different frequencies may be retrieved and converted automatically by specifying the correct options within the system.

The user may also select a valuation model, which is applied to the integrated data in accordance with a set of predefined rules. Typically, different rules are applied to different valuation models. The output of the valuation model is then passed to a ranking device. The ranking device sorts the portfolios based on the results of the valuation model and predetermined criteria, which is typically defined by the user. The ranking of the portfolios is then displayed to the user for making an informed decision. Alternately, the information may be sent to other applications, or stored in a database through the use of additional objects. The system may be accessed to set the frequency of refreshing the data as well as the frequency of invoking the retrieval, calculation, and ranking command.

Referring to figure 6, a multi-tiered system having a sophisticated set of commands is illustrated. The system acts in parallel on the combined portfolio for providing different rankings based on different factors. For example, as shown in figure 6, the combined portfolios are evaluated using macro factors and neural networks, as well as using a valuation model. Each of the different evaluations of the combined portfolio is then ranked based on predefined criteria and either stored, sent to a different application, or displayed to the user. Having simultaneous multiple analysis performed on the same combined portfolio provides the user with more information on which to make an informed decision regarding the portfolio. Furthermore, the system has the ability to compare and contrast different evaluation schemes.

Referring to figure 2, two of the major components of the processor 12, a user interface module 30 and a calculation or data module 32, are shown. The user interface 30 manages

external data feeds, transformations, models and outputs through a series of linked icons 34.

Underlying the user interface 30 is the calculation module 32 that is composed of:

1. Calculation modules 38 that each perform functions on data. These functions include simple numeric conversions to advanced analysis such as neural networks, pattern
5 recognition and simulations. The calculation modules are normally defined by a set of attributes, which determine the inputs, outputs and other parameters of the modules. For example if one of the modules implemented a running average the attributes may be a data input, a sample period and a running average output.
2. A calculation engine controller 36 manages inputs from the external data streams 22 via
10 the interface 13 and the flow of data between the various calculation modules 38.

The icons 34 in the user interface 30 can be adapted to present a systematic characterization of time series and models. The icons may be presented as individual icons representing arrays of data, or have specific designs to distinguish among different the classes of icons. For example, functions may be represented by a triangle, time series by a square and
15 output as a circle.

The icons 34 can be thought of as being hot-wired to an underlying complex of calculation modules 38, data sources 22 and outputs. This structure ensures that the most recent data is automatically generated and the integrity of the time series and predictions is maintained through a rigorous architecture that explicitly manages the correct sequencing of calculations and
20 interdependence among time series.

The results of calculations in the calculation modules 38 may be stored for later use, presented to the user via the GUI 14, or used in the calculation of another calculation module.

A more detailed description of the user interface module 30 is provided in the following paragraphs.

25 Figure 3 shows a screen shot 50 of the graphical user interface for one embodiment.

When the user is building a model 51 they will have at least two windows open, the Objects in Hierarchy window 52 and the Model window 54. The Model window 54 provides an area for the user to create the model 51. It also provides access to a set of functions 56 that can be used in the model. The Objects 58 in Hierarchy window 52 provides access to all the time series data
30 that can be used in the model. Individual data of a time series may contain a number of

attributes, or time series data. For example, the 'Intel' series 58 may contain the attributes of price 59, volume 60 and earnings 61, these attributes are displayed in a separate window 53.

The objects may also be represented in a graphical tree as shown in window 63. The data that is displayed in the Objects in Hierarchy window 52 can be referenced by a folder 62 or by individual series 64 in the tree display window 63. If the user drags an entire folder 62 such as 'My Portfolio' 62a over to the Model window 54 then all the series 64a in that folder 62a are added to the model 51. When the series are dragged to the Model window 54, a new icon 66a called 'Time Series' appears and the series' names are displayed in the top half of the icon 66a. The bottom half of the icon 66 displays an intersection of the attributes of the selected series 64a. If the folder 'My Portfolio' 62a changes, the model 51 is automatically updated to reflect those changes. For example, if the Oracle stock is added to the folder marked 'My Portfolio' 62a, then the set of series 62a would have Oracle added to it automatically and Oracle would appear in the top half of the time series icon 66a.

This process is particularly useful if a ranking object is used to generate a folder with ten stocks in it. If the stocks change their order, or are added and removed from the list of ten, the model is automatically updated, as are other objects that use that model or those series.

In a similar fashion to the creation of the first time series icon 66a, a second time series icon 66b can be created for various interest scenarios. The names of the interest scenarios appear in the top half of the icon 66b, while their attributes appears in the bottom half. The interest rate for each interest scenario is its only attribute.

Once a series 64 or set of series is added to the model 51 the user can drag in functions 56 to operate on those series 64. The functions 56 are categorized by use, with the most commonly used functions automatically placed in a special folder. When the user drags a function 56 into the Model window 54 a function icon 68 is displayed. The function icon has inputs and outputs. The inputs types can consist of various formats such as drop down boxes, tick boxes, date ranges or simple numbers. Inputs can also be attributes. For example, figure 3 shows a line connecting the price of the time series 66a to the input of a Percent Change function 68a. The connection implies that the user wants to perform a percent change calculation in the price of both Intel and nVidia that spans the previous ten days. The user via an input text box 70 specifies the number of days. The result of this calculation is a time series output. This input text box may also be

used to represent the type of interpolation/extrapolation the attribute goes through before it is used in a calculation.

In this example shown in window 54 there are two time series icons 66. The first icon 66a has two series, and the second icon 66b has three. The system deals with models as if there is only one series in each of those time series objects 66. Therefore the Model window 54 needs to create six unique combinations of interest rate scenarios and stocks internally. These combinations are:

1. Intel and Interest Scenario 1;
2. nVidia and Interest Scenario 1;
- 10 3. Intel and Interest Scenario 2;
4. nVidia and Interest Scenario 2;
5. Intel and Interest Scenario 3; and
6. nVidia and Interest Scenario 3.

Therefore there are six unique attributes for the predicted price, each slotted into a different folder in My Portfolio Model ## in the Objects in Hierarchy window 52. The attributes for the Intel folder in My Portfolio Model ## are the predicted prices for the Intel stock with each interest scenario. Similarly, the attributes of the nVidia folder are the predicted prices for nVidia for each interest scenario. Therefore each folder has three attributes.

The attributes for the Interest Scenario 1 folder in My Portfolio Model ## are the predicted prices for each of the Intel and nVidia stocks using interest scenario 1. Similarly, the attributes of the Interest Scenario 2 and Interest Scenario 3 folders are the predicted prices for each of the Intel and nVidia stocks using interest scenarios two and three respectively. Therefore each of the interest scenario folders has two attributes.

Any attribute that is displayed in the 'Objects in Hierarchy' window can be used as an input to another model, or it can be used to drive an output for the user. Therefore, the user can build up nested layers of models that depend on each other. In this example, the user could take the predicted price for nVidia and use it in another model. If the user developed a model that used the predicted price for nVidia, and then changed the model that generated that predicted price, the new model would automatically reflect those changes. This automatic linking ensures concurrent and consistent management of time series and models and enables real time applications. The icons enable the user to build very complex, nested models without the need

for coding or selecting columns and typing formulas. It also allows the user a simple view of what the model performs.

While the user interface module 30 provides the user with an easy drag and drop method for creating models, the calculation module 32 is responsible for providing the user interface 30 with all the data it requests. The calculation module 32 ensures that the data used is the most current available and that it is provided quickly and accurately. The user interface is connected to the calculation module via the Calculation Engine Interface 40 that allows it to describe precisely what calculations and models it requires. An example of how such a structure works is outlined in the example below, although other methods will be apparent to those skilled in the art.

The user builds a model in the model creation window 54. This process has already been discussed while describing the User Interface module 30. Once the model has been completed and the user is ready to make it active, the model window contacts the Calculation Engine Controller 36 via the Calculation Engine Interface 40. The Model window 54 describes the calculation, data, functions and the relationships between that have been specified in the model 51. The Calculation Engine Controller 36 builds a set of calculation components 38 that represents the model 51 as described by the Model window 54. The calculation components perform the prescribed functions represented by the function icons 68.

In the example illustrated in figure 2, the attributes numbered 1 through 6 require the acquisition of external information. The attributes have several unique characteristics. They are not set to any specific frequency and can deal with incoming data at random, changing frequencies. Also, the frequencies between attributes may differ. The time series represented by the attributes do not need to be contiguous. For example, if an attribute represents the Toronto Stock Exchange (TSE) price, it would only contain valid data for the hours that the exchange was open.

Since attributes typically do not have similar or even constant frequencies, an attribute set is required. The attribute set interpolates or extrapolates the attribute to a common, although not necessarily regular, frequency so that the calculation components 38 can operate on them. Since the technology is frequency independent the user does not need to be concerned with the fact that the data used for the same calculation may be updated daily or monthly.

To ensure that interpolation or extrapolation errors are not carried too far, the concept of a rollback is introduced. Referring to figure 4, a first time series A is collected at a frequency of 0.1 ms and a second time series B is collected at a frequency of 0.5 ms. In figure 4(a), the plot of the two different time series are shown at $t = 1.4$ ms. Therefore, the plot for time series A is accurate at $t = 1.4$ ms, since data points are collected every 0.1 ms. However, the most recent data point collected for time series B is at $t = 1.0$ ms. Therefore, the program has to extrapolate a data point using the previous data in order to determine where time series might actually be at $t = 1.4$ ms. However, this extrapolation process may not be accurate and, as shown in figure 6(a), it may even be possible to extrapolate the value of time series B to have a value below 0 at $t = 1.4$ ms, which is not a logical conclusion.

Furthermore, when the time reaches $t = 1.5$ ms, an accurate value for time series B is available and is plotted on the graph accordingly. However, due to the previous extrapolation error between $t = 1.0$ ms and $t = 1.4$ ms, the data illustrated between $t = 1.4$ ms and $t = 1.5$ ms, as shown in figure 6(b), is incorrect. This error is compounded when the incorrect information is used for extrapolating the data between $t = 1.5$ ms and $t = 2.0$ ms.

Therefore, as soon as the accurate value for time series B is available, the calculations are rolled back to the previously received accurate value and recalculated, as illustrated in figure 4(c). The rollback provides a more accurate estimate of the value of time series B between received data points. In the present embodiment, the rollback is accomplished because the program saves the data to a storage medium rather than simply displaying it. Thus, once the processor has the correct value for time series B at $t = 1.5$ ms, it can retrieve the value for time series B at $t = 1.0$ ms and interpolate the points in order to provide a more accurate graph to the user. The method of auto-correction also allows the processor to more accurately extrapolate the next point, $t = 2.0$ ms.

The calculation components 38 perform calculations using the attributes that are linked to them and they output other attributes. The output attributes can be used either as inputs to other functions or passed to the User Interface module for display and user interaction.

When the user wants to display a chart, a chart object asks the Calculation Engine Controller for a particular attribute. The Calculation Engine Controller provides an output cache where which the chart object can read and write data. It then copies all the data from the

attribute into the output cache. The Controller then calculates any changes in the data that have occurred since the last time the attribute was viewed.

When calculating a desired output, the Calculation Engine Controller 34 first determines what attributes from an outside source are required to perform the calculation. Once attributes
5 that require external data are determined, the Calculation Engine Controller 36 triggers the Data Stream Object to start grabbing data from the data server. The data is gathered starting from the most recently collected data point of the attribute. The system is capable of running in real time with a fully abstracted data layer. The data may be retrieved and refreshed at any frequency including real time. Therefore, the user does not have to adjust input sources or press a 'refresh'
10 button to update the system. Rather, while the system is operative, the most recent data is provided as soon as it becomes available. When the Data Stream Object adds a new data point to the attribute, the Calculation Engine Controller is informed. The Calculation Engine Controller then recalculates all the attributes that depend on the attribute that was updated.

The Calculation Engine Controller 36 also controls the order of calculation. In the above
15 example it is necessary to calculate Function 1 before it calculating Function 2, since Function 2 relies on the output of Function 1. It may occur that there are several functions that can be calculated simultaneously, that is if their inputs are independent of each other's output. This allows the Calculation Module to take advantage of the multithreading capabilities of the machine it is running on. When attribute number 9 is calculated, the Calculation Engine
20 Controller 36 moves the new data into the output cache and alerts the chart object that there is new data. This process is repeated when the Data Stream Object tells the Calculation Engine Controller that new data was added to one or more of the attributes that attribute number 9 requires for its calculation.

This process is real time, with both historical and new data being fed to the Data Stream
25 Object. Also, nothing is calculated unless it is needed for the model 51 specified in the User Interface Module 30.

This process continues until the User Interface tells the Calculation Engine Controller that it no longer needs a particular attribute. The Calculation Engine Controller stops the Data Stream Object from grabbing data for those attributes that are no longer needed for any
30 calculations.

In an alternate embodiment, the application of the system described above to an Internet based system is illustrated in figure 7. A user is coupled to the system via the Internet. Before being allowed access to the system the user is prompted for a user identification and password. Upon successful input of the user identification and password, the user can access his/her
5 portfolio by clicking a portfolio option. Selecting the portfolio option provides the user with access to the portfolio so that he/she may modify it as desired.

Once the user is satisfied with the portfolio, he/she can view the performance of the portfolio by selecting the performance option. Selecting the performance option provides the user with a list of list of possible performance formulae to select from. Having such a system on
10 the Internet allows users to have access to performance formulae from major financial institutions, providing the user with a broader knowledge base with which to make decisions. Once the user has selected an appropriate formula or formulae, the performance of the portfolio is analysed and the results are provided to the user.

Alternately, the user may predefine preferred formulae to apply to the portfolio. In this
15 case, the portfolio is analysed and the results are provided to the user once the user has selected the performance option, without having to make any other selections.

The user can then use the performance analysis for making an informed decision about purchasing and selling components of his/her portfolio. If the user decides to buy or sell a particular component of his/her portfolio, he/she selects the buy/sell option. The user is
20 prompted to enter the product to be purchased or sold as well as the desired quantity and price. The buy/sell offer is processed by the system and executed through a trading system, which is well known in the art.

Furthermore, the user has the ability to enter alerts into the system. If the user wants to be alerted of a predetermined change in a particular stock price, for example, the details of the
25 stock is provided to the system. If the change is realised, then the user is alerted by email or the like. Also, the user can set up an alert such that if the performance of a particular stock reaches a predefined threshold, then the system notifies the user via the performance analysis that the change has been realised.

It may be seen that the present invention provides a simple automated tool for
30 implementing and analyzing numerous data driven applications. These may include engineering applications to manage, analyze and model the behavior or performance of dynamic systems

including applications such as communications, chemical processes and other physical systems; business intelligence applications to analyze time series and accounting data to monitor both internal and external business environments and/or predict future business conditions; business modeling such as scheduling and strategic planning where arrays or data are managed in
5 conjunction with linked statistical or mathematical models. In such applications a custom user interface may also be implemented.

In summary, the invention provides for a compute program having an object based graphical interface which represents a system to control data retrieval from a plurality of data sources, initiate a plurality of statistical or other data analysis operations and a plurality of
10 generated outputs including data files, tables, graphs and other representations. The objects represented by icons may be linked through multiple connections which ensure the automatic or specified sending or retrieval of data among the data files, statistical operations and desired output formats. The output formats may be graphical, tabular, generated data files or other applications. The operations on time series may be characterized by a sequence of interactive
15 icons represented in flow models. Messages may be sent and received interactively by all types of objects. Each object has an associated set of properties. The links among the objects may impose a sequential order of operations or hierarchy of procedures which may be called directional links. A graphical and text interface will allow the user to monitor the properties of each object, congregate objects by shared properties and analyze the directional links and order
20 of operations. The systems of object directed operations may be replicated with the inherited properties of each object and the directional links. The systems are then amenable to modification or comparison with earlier systems.

Although the invention has been described with reference to certain specific
25 embodiments, various modifications thereof will be apparent to those skilled in the art without departing from the spirit and scope of the invention as outlined in the claims appended hereto.

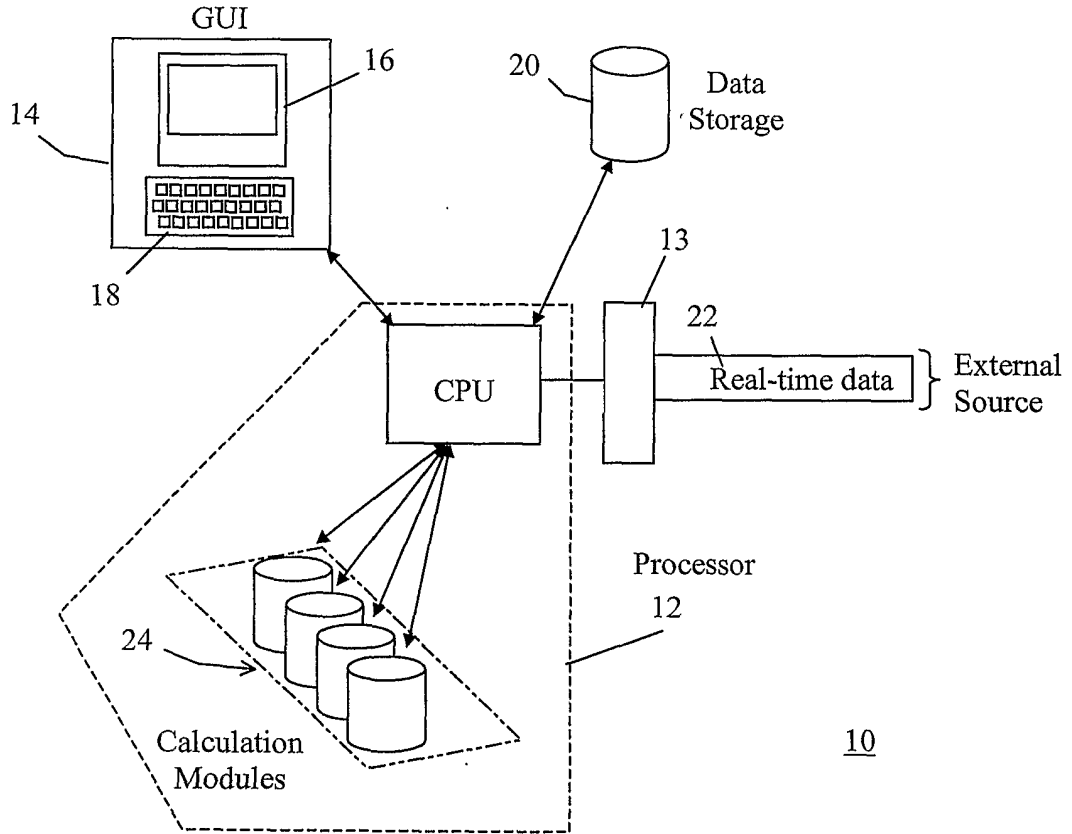


Figure 1

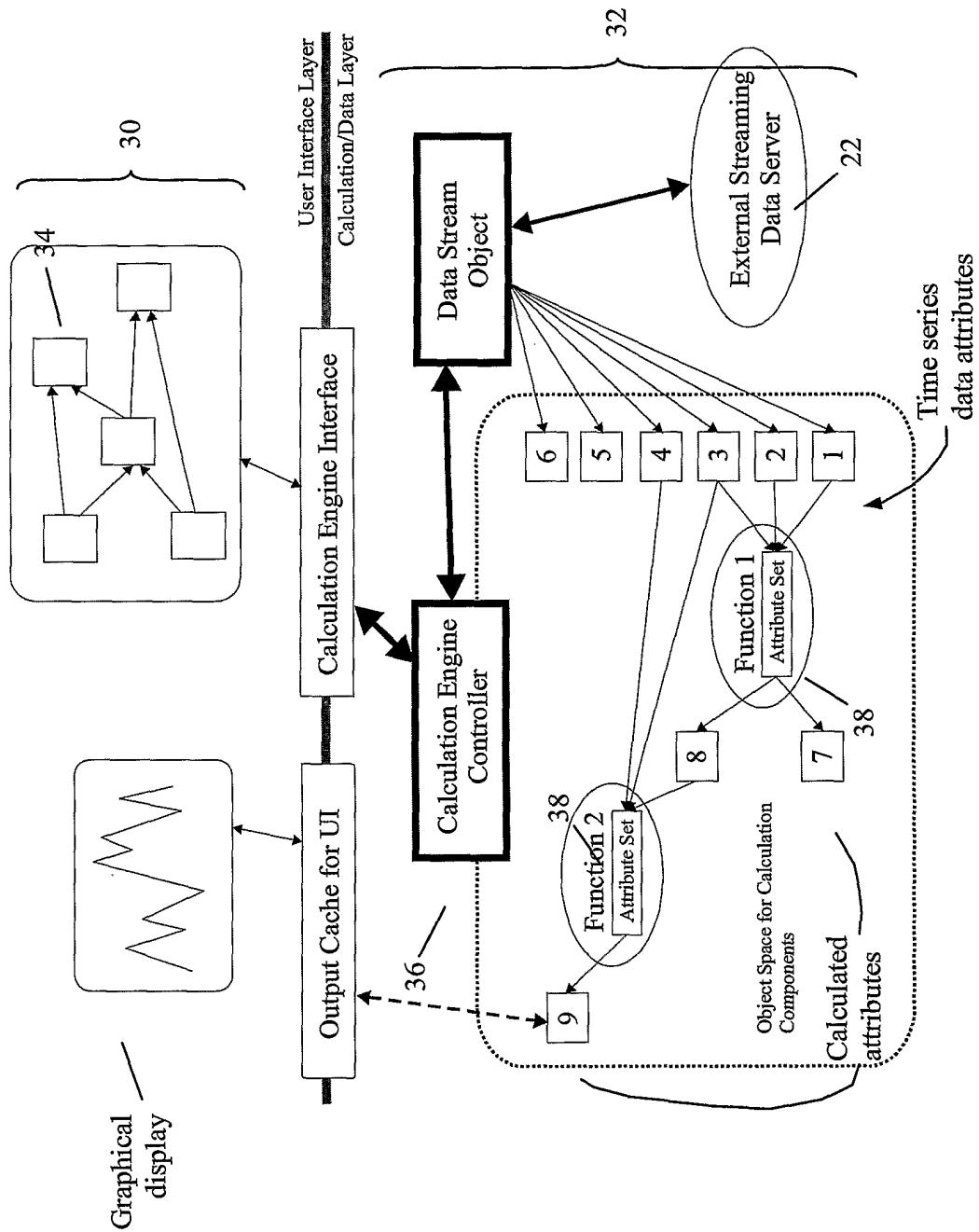


Figure 2

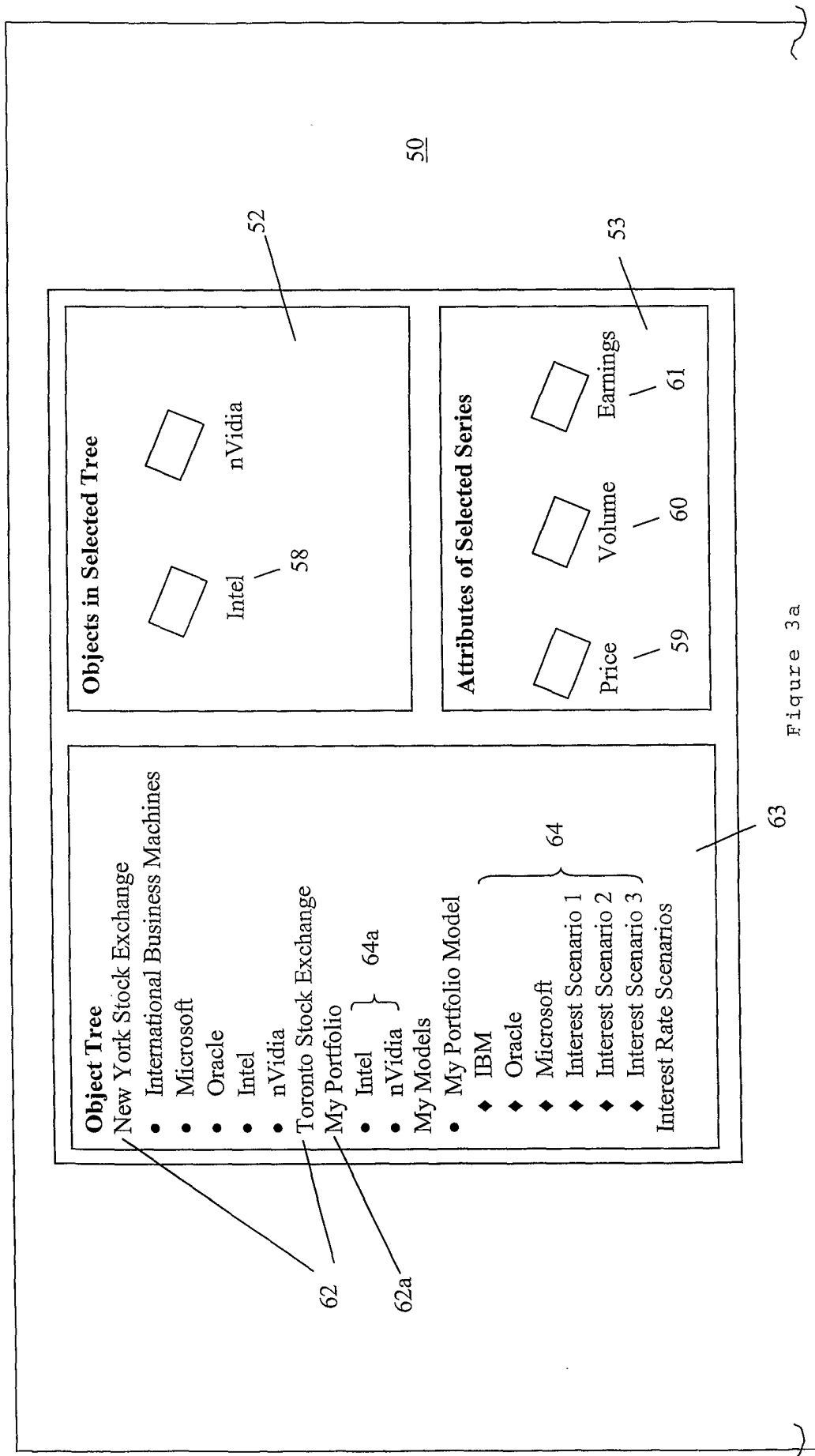


Figure 3a

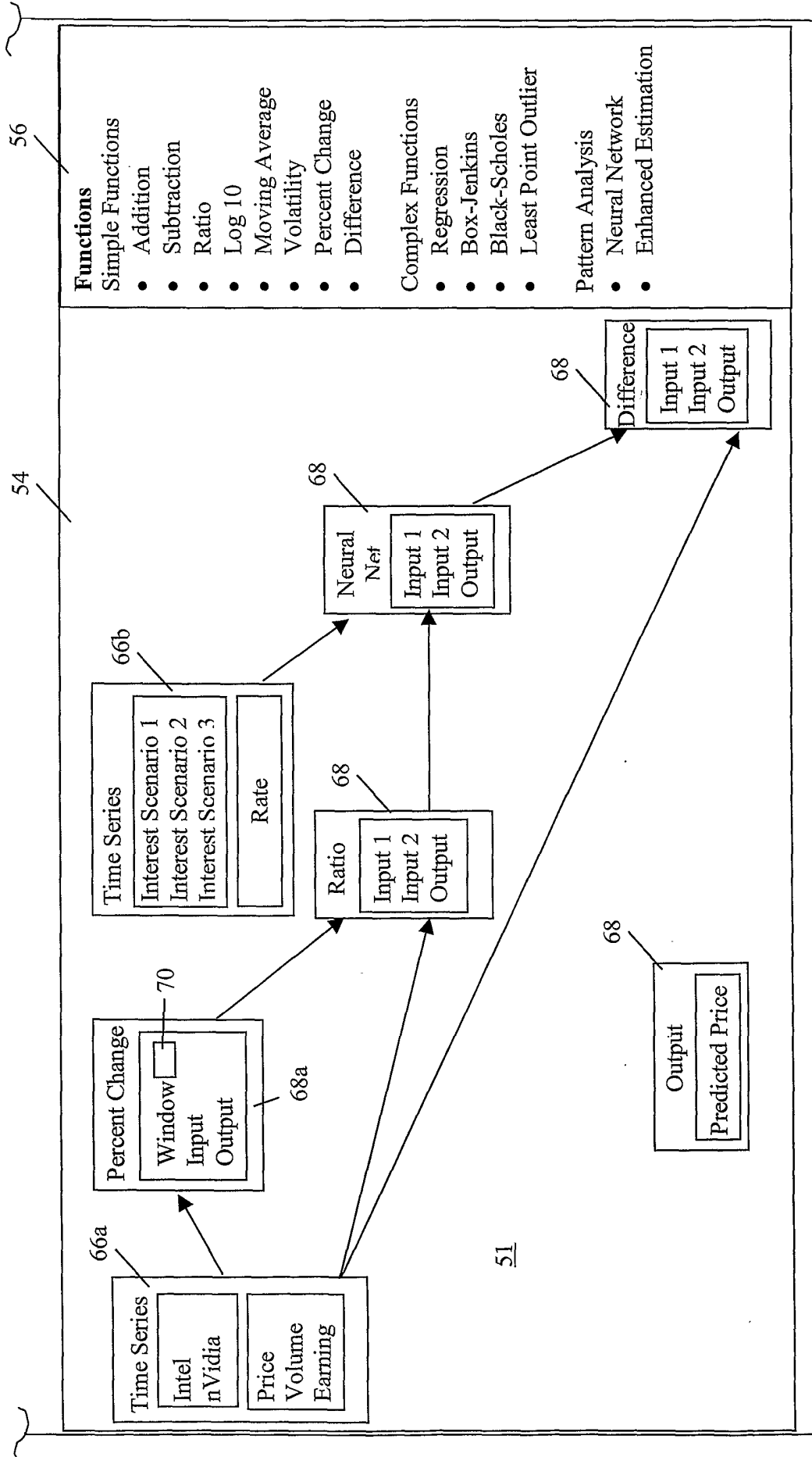


Figure 3 b

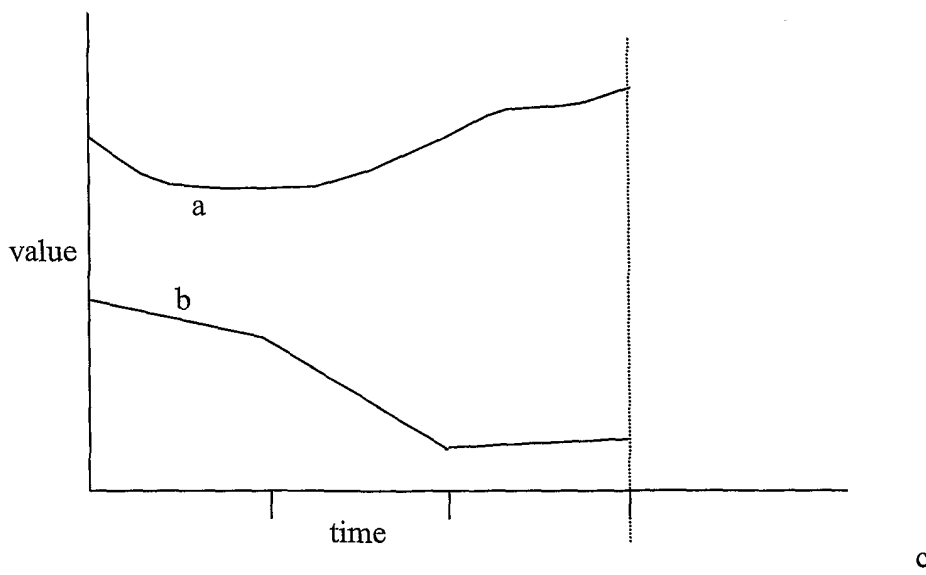
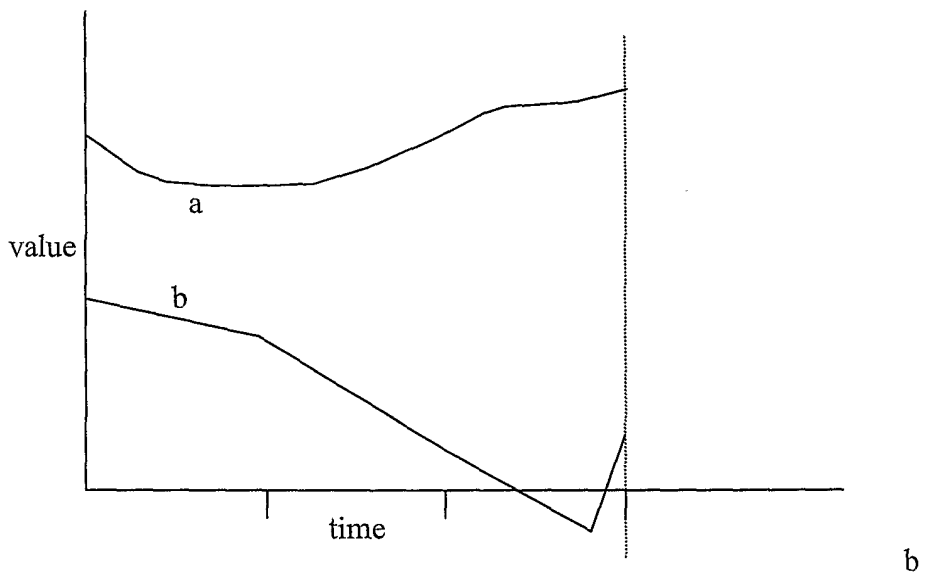
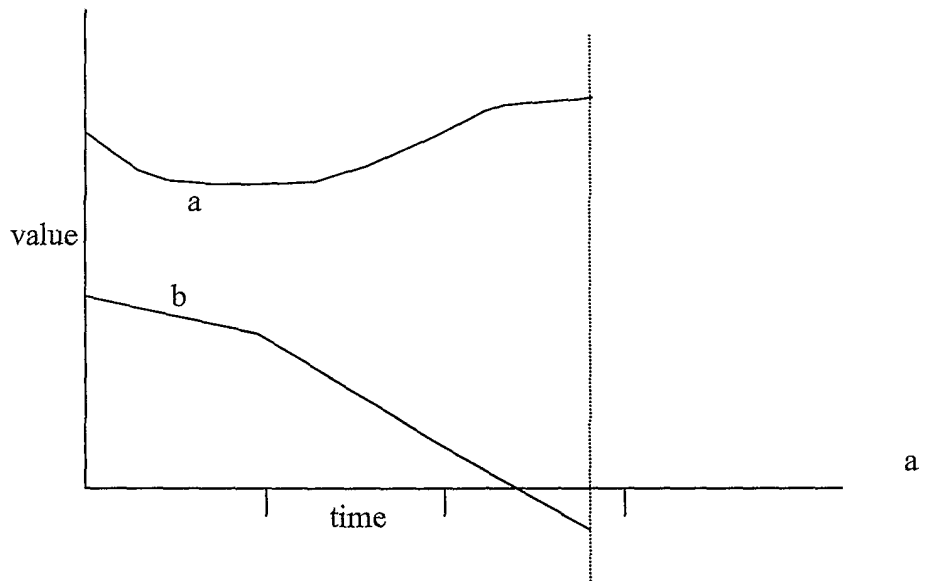


Figure 4

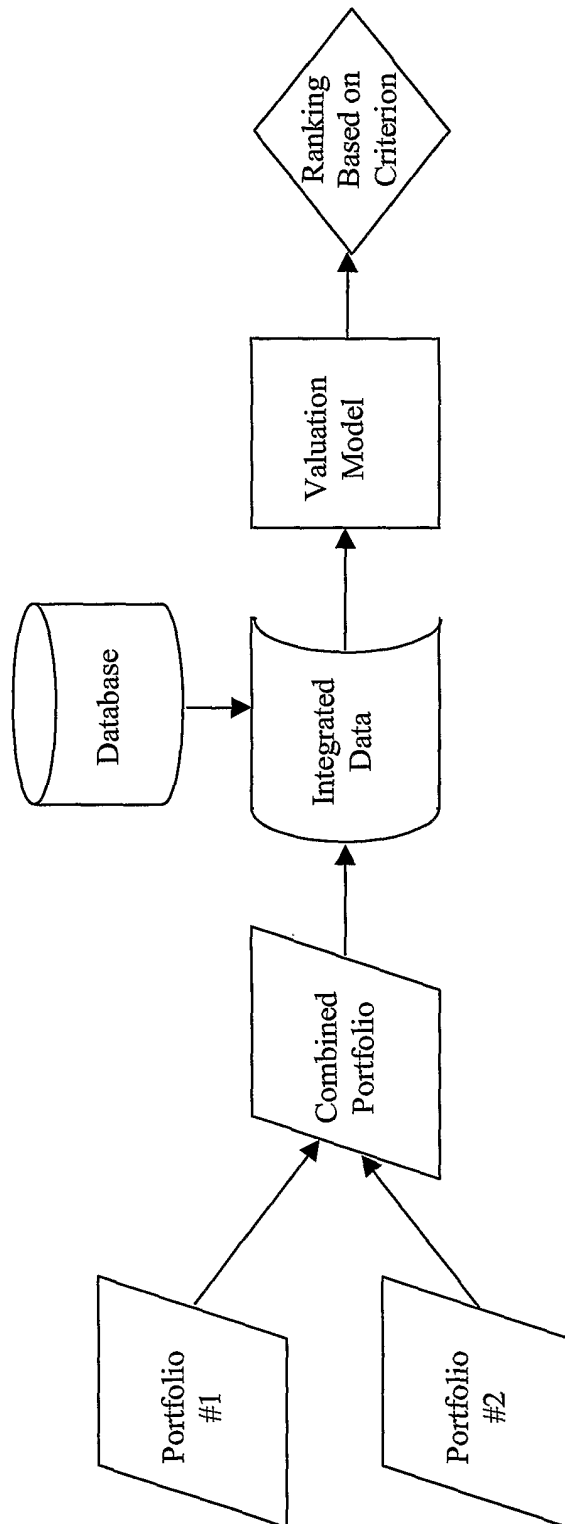


Figure 5

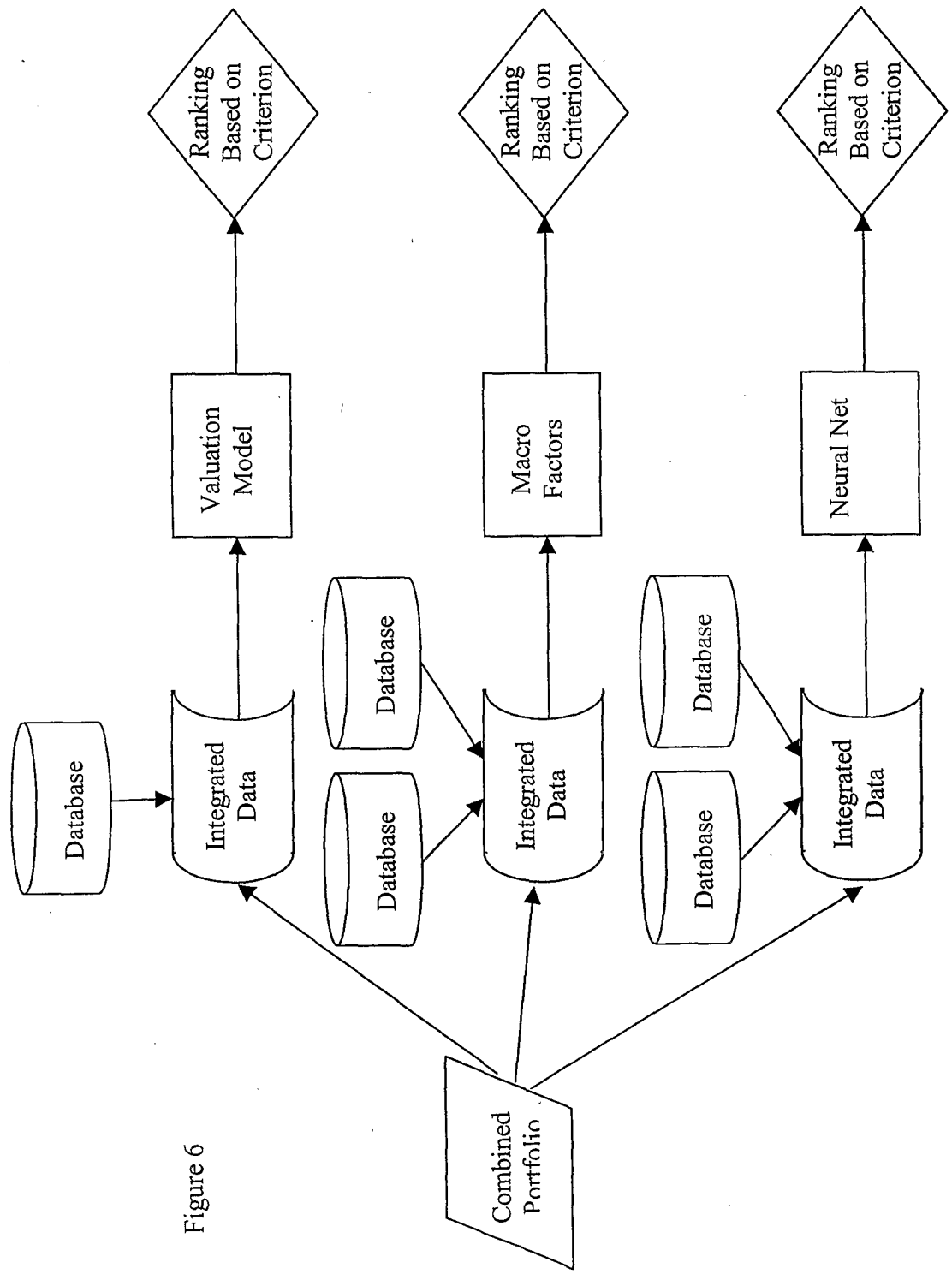


Figure 6

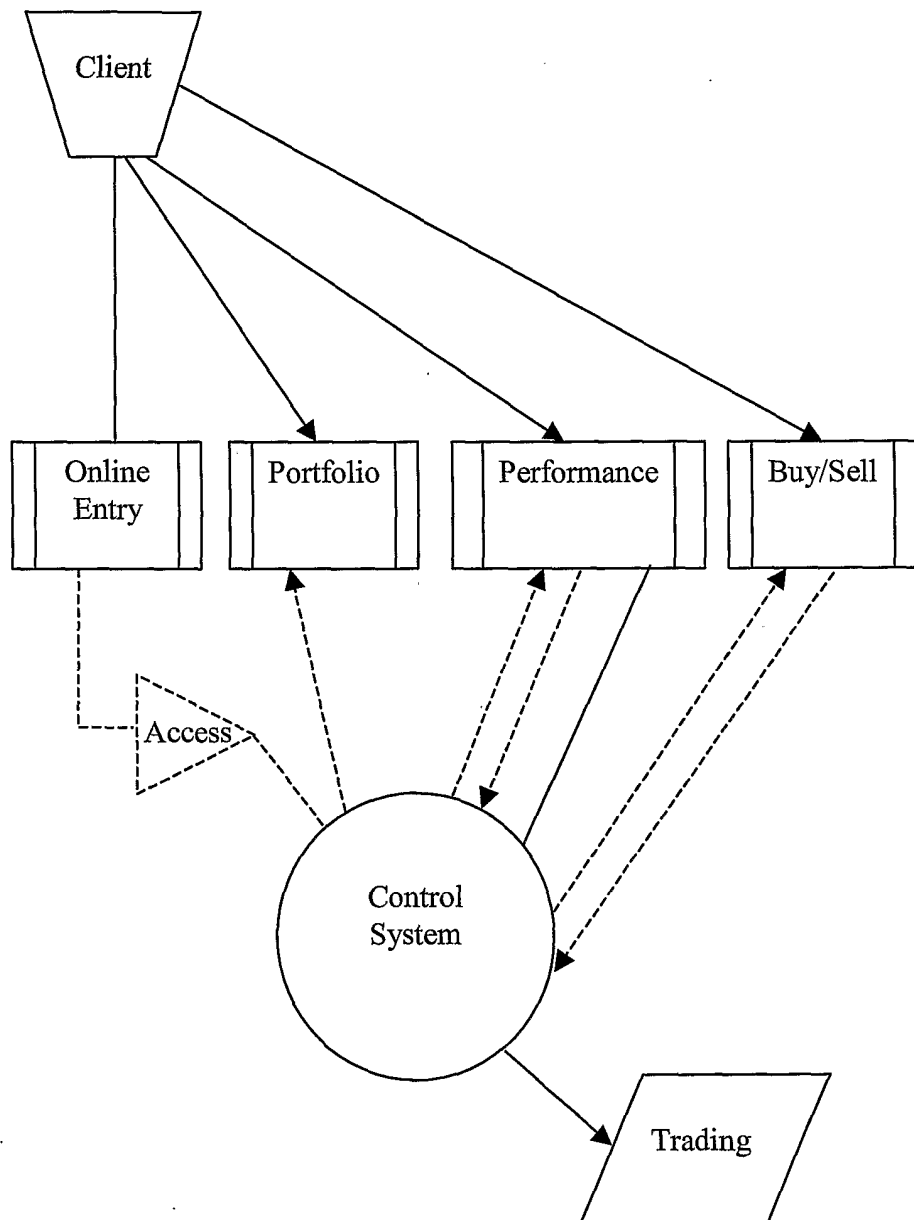


Figure 7

INTERNATIONAL SEARCH REPORT

International Application No
PCT/CA 00/00961

A. CLASSIFICATION OF SUBJECT MATTER IPC 7 G06F17/18				
According to International Patent Classification (IPC) or to both national classification and IPC				
B. FIELDS SEARCHED				
Minimum documentation searched (classification system followed by classification symbols) IPC 7 G06F				
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched				
Electronic data base consulted during the international search (name of data base and, where practical, search terms used) EPO-Internal, WPI Data, INSPEC				
C. DOCUMENTS CONSIDERED TO BE RELEVANT				
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
Y	US 6 061 058 A (JOHNSTON JR ROBERT GEORGE ET AL) 9 May 2000 (2000-05-09) claim 1	1-9		
Y	DE 196 43 918 C (SIEMENS AG) 5 February 1998 (1998-02-05) the whole document	1-9		
A	EP 0 453 371 A (AEROSPATIALE) 23 October 1991 (1991-10-23) claim 1	1-9		
--- -/--				
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C.				
<input checked="" type="checkbox"/> Patent family members are listed in annex.				
° <i>Special categories of cited documents:</i>				
<table style="width: 100%; border: none;"> <tr> <td style="width: 50%; border: none; vertical-align: top;"> <ul style="list-style-type: none"> *A* document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed </td> <td style="width: 50%; border: none; vertical-align: top;"> <ul style="list-style-type: none"> *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *Z* document member of the same patent family </td> </tr> </table>			<ul style="list-style-type: none"> *A* document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed 	<ul style="list-style-type: none"> *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *Z* document member of the same patent family
<ul style="list-style-type: none"> *A* document defining the general state of the art which is not considered to be of particular relevance *E* earlier document but published on or after the international filing date *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) *O* document referring to an oral disclosure, use, exhibition or other means *P* document published prior to the international filing date but later than the priority date claimed 	<ul style="list-style-type: none"> *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art. *Z* document member of the same patent family 			
Date of the actual completion of the international search <p style="text-align: center;">2 May 2001</p>	Date of mailing of the international search report <p style="text-align: center;">09/05/2001</p>			
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer <p style="text-align: center;">Pierfederici, A</p>			

INTERNATIONAL SEARCH REPORT

In nal Application No
PCT/CA 00/00961

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT		
Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>JENG L -G ET AL: "GRAPHICAL BLOCK-DIAGRAM BASED PROGRAMMING ENVIRONMENT FOR A DSP SILICON COMPILER" IEE PROCEEDINGS G. ELECTRONIC CIRCUITS & SYSTEMS, GB, INSTITUTION OF ELECTRICAL ENGINEERS. STEVENAGE, vol. 140, no. 5, 1 October 1993 (1993-10-01), pages 313-318, XP000400065 ISSN: 0622-0039 abstract page 315, left-hand column, paragraph 3 -right-hand column, paragraph 2 ---</p>	1-9
A	<p>NICOLAI M ET AL: "SLANG, a statistical language for descriptive time series analysis" COMPUTER SCIENCE AND STATISTICS: PROCEEDINGS OF THE 13TH SYMPOSIUM ON THE INTERFACE, PITTSBURGH, PA, USA, 12-13 MARCH 1981, pages 320-323, XP000997612 1981, New York, NY, USA, Springer-Verlag, USA ISBN: 0-387-90633-9 ---</p>	
A	<p>ZISSMAN M A ET AL: "A BLOCK DIAGRAM COMPILER FOR A DIGITAL SIGNAL PROCESSING MIND COMPUTER" INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH & SIGNAL PROCESSING. ICASSP, US, NEW YORK, IEEE, vol. CONF. 12, 1987, pages 1867-1870, XP000867153 ---</p>	
A	<p>BIER J C ET AL: "GABRIEL: A DESIGN ENVIRONMENT FOR DSP" IEEE MICRO, US, IEEE INC. NEW YORK, vol. 10, no. 5, 1 October 1990 (1990-10-01), pages 28-45, XP000170675 ISSN: 0272-1732 -----</p>	

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No PCT/CA 00/00961

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6061058 A	09-05-2000	NONE	
DE 19643918 C	05-02-1998	WO 9819252 A EP 0934567 A JP 2001502831 T	07-05-1998 11-08-1999 27-02-2001
EP 0453371 A	23-10-1991	FR 2661266 A CA 2040675 A DE 69129715 D DE 69129715 T ES 2119765 T US 5325534 A	25-10-1991 21-10-1991 13-08-1998 11-02-1999 16-10-1998 28-06-1994