2010

Modelling Exploratory Analysis Processes for eResearch

Lawrence Yao
University of New South Wales, l.yao@student.unsw.edu.au

Fethi A. Rabhi
University of New South Wales, f.rabhi@unsw.edu.au

Modelling Exploratory Analysis Processes for eResearch

Lawrence Yao
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
Email: l.yao@student.unsw.edu.au

Fethi A. Rabhi
School of Computer Science and Engineering
University of New South Wales
Sydney, Australia
Email: f.rabhi@unsw.edu.au

Abstract

Financial markets produce high-frequency data and analysing it involves transforming the data, detecting patterns and testing financial models. These actions or steps form an exploratory analysis process (EAP). ADAGE is an open SOA incorporating a BPMS that allows users to model EAPs by composing analysis services. A typical application scenario is used to evaluate ADAGE's ability to express an EAP as a business process. It is shown that current BPMS technology cannot satisfactorily represent EAPs as fully executable business processes. Using the theory of situation awareness, EAPs are shown to be dynamic in nature. Hence three extensions based on the late composition technique are proposed: (1) a dynamic process representation of EAPs; (2) a process execution model; and (3) process templates to automate repetitive steps of EAPs.

Keywords

Business Process Management, Service-Oriented Architecture, Financial data analysis, Situation Awareness

INTRODUCTION

eResearch is being adopted worldwide across all research disciplines, harnessing high-capacity and collaborative information and communications technology to improve the conduct of research and enable research that cannot be conducted otherwise. A number of national initiatives are taking place in the UK (under the name e-Science), the US (as Cyberinfrastructure) and Australia. eResearch is mainly driven by the availability of vast amounts of data from sources as diverse as Web logs, network traffic messages, scientific instruments, sensors and reports. An academic researcher (typically a domain expert) performs data analysis in an exploratory way, often using several libraries or packages in an iterative trial-and-error fashion. Analysis activities include browsing or visualising specific information from a data source, querying single or multiple data sources, transforming data (e.g. cleaning, enriching, aggregating, summarising), analysing correlations, detecting patterns and testing or discovering models.

For example, high-frequency financial data is increasingly becoming more available to both market participants and researchers. Tsay (2005) considers high-frequency data to be observations taken at daily intervals or at even finer time scales. When all transactions are recorded together with their associated characteristics, Engle (2000) calls this ultra-high-frequency data. Other descriptions have also been used such as "real time tick data" or "tick-by-tick data" (Sun et al. 2008). High-frequency data has been widely used by finance researchers to study market behaviour (Goodhart and O'Hara 1997). Studies have used high-frequency data to analyse many aspects of market microstructure including volatility and price jumps (Andersen 2000; Bollerslev et al. 2008; Engle 2000). Exchange rate trading behaviour has been linked to US macroeconomic data releases (Chaboud et al. 2008). Therefore, high-frequency data is valuable and has been used in many empirical works related to financial market research.

The purpose of our research is to investigate IT approaches to support researchers and practitioners in analysing vast amounts of data. In this paper, we focus on datasets originating from financial exchanges made available by third party information providers to the research community. For example, the Thomson Reuters Tick History (Thomson Reuters 2010) system allows access to intraday trade and quote information for over 200 exchanges and Over-The-Counter (OTC) markets around the world. There are many software tools and information systems available that help users look for statistical properties in high-frequency data, test financial models against available data etc. Some examples include S-PLUS, SPSS, SAS and MATLAB. However, many of these tools and systems do not support the definition and execution of complex analysis processes. Section 2 of this paper

gives some background on the ADAGE project (Guabtni et al. 2010; Rabhi et al. 2009b), where we have been investigating an open architecture based on service-oriented computing (SOC) principles. The motivation is that SOC offers the potential to detach the analysis process from specific technology solutions and allows users to weave together analysis services, with no technical knowledge required of the underlying service implementation. However, little evaluation has been conducted on the effectiveness of ADAGE in handling complex analysis scenarios. Hence section 3 presents a complex application scenario as a means to evaluate ADAGE's capabilities. From the evaluation a deficiency is found, and section 4 attempts to find explanations for the deficiency by using situation awareness principles. Extensions to ADAGE are then proposed to overcome the deficiency. Section 5 discusses related work and section 6 concludes this paper.

BACKGROUND ON ADAGE

This section provides a background overview on the ADAGE service-oriented architecture (SOA). The main elements of the ADAGE architecture are: (1) the steps carried out as part of high-frequency data analysis form a process, which is referred to as an *exploratory analysis process* (EAP); (2) steps of the EAP can utilise reusable computations or "building blocks", each is supported by software components referred to as *services*; (3) these services can be implemented using different programming languages or through existing packages but will have a Web-discoverable service interface so that they can be invoked locally or remotely; (4) there is an event-based data model which describes the different types of observations recorded in high-frequency data as events. In this paper, the types of events of interest are *Trade* (representing the parameters of a trade such as price), *Index* (representing the parameters of a market index), *Measure* (representing the parameters of a computed measure such as volatility) and *News* (representing news articles and reports). Such a data model is already published in Rabhi et al. (2009a) and (5) services will produce and consume events through a *shared event repository* that stores instances of the data model as *event sets*. Various software libraries implement the required functionality for accessing, creating and storing event sets.

The architecture of ADAGE is illustrated in Figure 1. The benefits of a service-based approach are many. It enables different technologies (e.g. Web services) to provide facilities such as run time messaging, service composition and choreography, process modelling, discovery, semantic support and run time management. The Event Sources layer of the SOA corresponds to the data repositories. Such sources of data are usually heterogeneous and can offer news and/or market data. The two layers on top of it consist of services that allow events to be imported, processed and exported to and from the shared event repository. The Business Process layer incorporates a Business Process Management System (BPMS), allowing advanced usage of the SOA such as discovery, aggregation, orchestration and choreography of the proposed Web services. This layer facilitates the combination of several calls to the services, grouping them into a business process that can be executed with the embedded Business Process Engine.

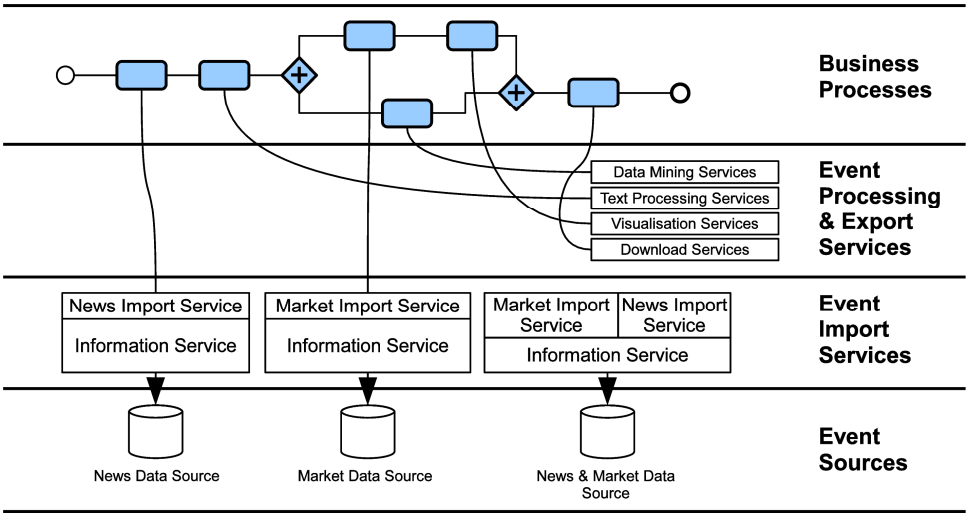

Figure 1: The ADAGE SOA for analysing news and market data.

Services are grouped according to their interactions with the shared event repository as follows:

- Import Services group: these services create event sets of interest from native data repositories (including Web repositories). The import services used in the application scenario are:
    - *TRTH Import Service:* imports events from the Thomson Reuters Tick History (TRTH) System (Thomson Reuters 2010) into the shared event repository. These events can cover company

stocks from every financial exchange worldwide, currency exchange rates, indices and interest rates. A selection of events can be imported using criteria such as period, exchange etc.

- *News Import Service:* imports news articles and reports from a news archive provider such as Thomson Reuters. Imported data are stored as *News* events.

- Processing Services group: these services transform event sets in various ways, e.g. to extract or add new information or combine some event sets together etc. The processing services used in the application scenario are:
  - *Time Series Building Service*: processes events to produce time series data sampled at equal time intervals. This service is highly customisable as it allows the sampling period to be modified and a number of measures (such as the return, spread and vwap) to be included in the time series. A time series event set is modelled as a series of *Measure* events in our data model.
  - *Merge Service*: combines different time series into a single time series. This service can produce a merged time series in which the trade price of a stock is associated with an index value at around the time of the trade.
  - *Abnormal Returns Service:* analyses trading price time series of a stock by using market level data (like index price and interest rate) to compute abnormal returns and therefore detect unusual price movements.

- Export Services group: these services convert event sets into a format suitable for further processing (e.g. CSV) or into a graphical format for viewing/interpreting by the user. The application scenario uses a single export service:
  - *Visualisation Service*: allows a time series to be visualised as a chart. This service does not create an event set.

A prototype implementation of these ADAGE services has been realised together with a simple Web Graphical User Interface (GUI) that enables users to interactively invoke services from a Web Browser. The Web GUI, illustrated in Figure 2, can be used by domain experts (e.g. financial data analysts) to supply parameters and event sets from the shared event repository to the ADAGE services. It is developed in a modular fashion that allows new services to be easily integrated into the system. However, ADAGE's capabilities in modelling and executing EAPs at the Business Process layer have not been evaluated, so it is looked at in the next section.

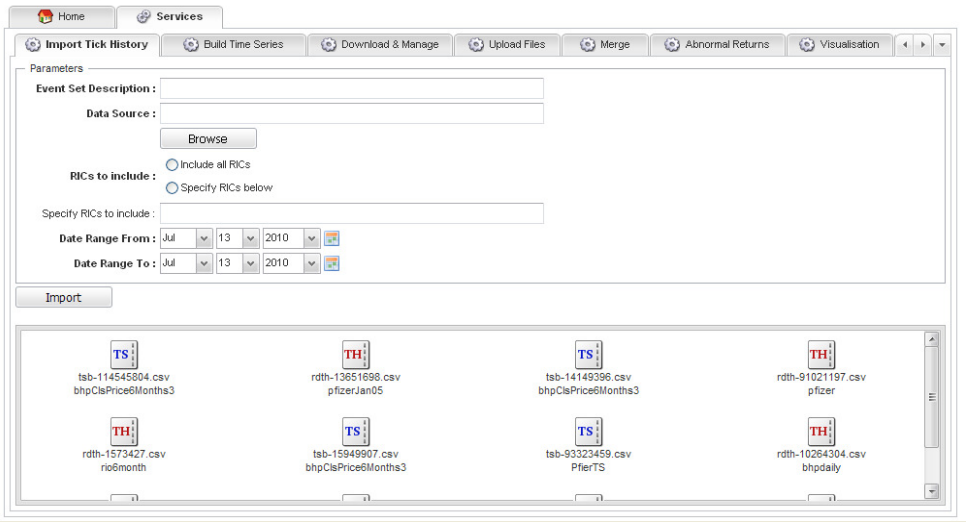

Figure 2: The ADAGE Web GUI. The icons towards the bottom of the figure represent event sets in the shared event repository.

MODELLING EXPLORATORY ANALYSIS PROCESS USING ADAGE

This section examines an application scenario that is representative of what domain expert users commonly do when analysing high-frequency data. The steps carried out in this application scenario form an EAP. The application scenario first describes how a user performs the analysis using the Web GUI of ADAGE. An attempt is then made to model this EAP as a business process so it can be execute at the Business Process layer of ADAGE. This section will conclude by showing that there are difficulties in constructing a satisfactory business process representation of the EAP.

This application scenario involves a fictional character Susan, an analyst who is looking for significant price movements of stocks in major pharmaceutical companies traded on the New York Stock Exchange (NYSE). Susan will start by looking at the data of Pfizer in the year 2004. Her aim is to find a significant price movement in Pfizer stocks and see if there are any correlation between the price movement and relevant information available from the news. Although the character Susan is fictional, real data is used to obtain the outcomes presented here. Note that due to limited space, several related steps are aggregated into a numbered item below.

1. To begin, Susan uses the TRTH Import Service to find the Reuters Instrument Code for Pfizer stocks on the NYSE. In TRTH, each company is uniquely identified by a Reuters Instrument Code (RIC). The RIC for Pfizer is found to be PFE.N. Susan then uses the service to import the daily closing trade price of PFE.N from 1 Jan 2004 to 31 Dec 2004. The imported data is stored in the shared event repository as an event set; this event set can be used as input to other services of ADAGE.
2. Susan finds the RIC for the S&P 500 index to be .SPXT. She then imports the daily closing .SPXT index data from 1 Jan 2004 to 31 Dec 2004 (the same period as above). This index data will be compared with Pfizer data to determine if there are any significant price movement.
3. Using the Time Series Building Service, Susan creates a time series using the daily PFE.N data. She then repeats this step and creates another time series using the daily .SPXT data. The result is two time series event sets been created in the shared event repository.
4. Susan uses the Merge Service to merge the two time series event sets, thereby creating a combined (PFE.N + .SPXT) time series event set.
5. Susan invokes the Abnormal Returns Service, giving it as input the combined time series event set. This service runs an algorithm on the given input events and produces a list of abnormal movements in PFE.N price relative to .SPXT value. The result of this service is an abnormal returns event set.
6. Using the Visualisation Service on the abnormal returns event set, Susan obtains a chart showing the significant price movements of Pfizer stocks relative to S&P 500 index in 2004. By examining this chart, Susan finds a significant price movement around 16th and 17th of Dec 2004.
7. She decides to take a closer look at those dates. So she uses the TRTH Import Service to import the high-frequency trade data for PFE.N and high-frequency index data for .SPXT for the 16th and 17th of Dec 2004.
8. Using the Time Series Building Service, Susan creates two hourly time series from the high-frequency data of PFE.N and .SPXT. She then combines the two time series using Merge Service, and feeds the results to the Abnormal Returns Service.
9. Using the Visualisation Service on the abnormal returns output from the previous step, Susan sees that there is indeed a significant price drop towards the end of 16th and early 17th of Dec 2004.
10. Susan forms the hypothesis that this price drop may be the result of a change (or the perception by the market that there is a change) in the fundamental values of Pfizer. She decides to check this hypothesis by looking for relevant news on the 16th or 17th Dec 2004.
11. Using the News Import Service, Susan imports all news articles that mention Pfizer in the month of Dec 2004. She then uses the Visualisation Service to read them, and finds that there are unfavourable reports on Celebrex – one of Pfizer's major drugs. On 17th Dec 2004 it has been reported that trials showed a link between Celebrex use and increased risk of heart attack, hence this is likely to have affected Pfizer's stock price.

Through this application scenario, it can be seen that EAPs in general tend to involve performing steps in a piecemeal fashion. Users use datasets to generate some results (e.g. Susan obtains a chart showing significant price movements using an abnormal returns event set), they combine results to build new datasets (e.g. Susan combines the data for Pfizer with the data for S&P 500), and the steps may be repeated iteratively using datasets obtained with different search criteria (e.g. Susan first looks at daily data in 2004, then repeats by looking at hourly data between 16-17 Dec 2004). Notably throughout the steps of the EAP, the user is in control and decides what to do base on the information available at the time.

Due to this user-driven nature, it is difficult to have EAPs modelled as business processes and executed at the Business Process layer of ADAGE. Most BPMS systems do not fully support the definition and execution of partial or incomplete business processes. For example, Figure 3 shows a business process model representing the steps Susan has taken to obtain a chart of significant price movement using daily data of PFE.N and .SPXT. A similar business process model can be created to represent the steps to obtain a chart using hourly data of PFE.N and .SPXT. However it is not known beforehand that Susan will want such a chart. For instance, if there are no significant price movements found using the daily data, then Susan will not have attempted to look at the hourly

data. It is also not known what Susan will do after the end of this application scenario. She may look at Pfizer's other significant price movements in 2004, or she may become interested in other drugs similar to Celebrex. For instance, Kolata (2004) reports on 1 Oct 2004 that Vioxx – a drug similar to Celebrex – is also linked to increased risk of heart attack and strokes. The article has also mentioned a significant drop in the stock price of Merck, the company that makes Vioxx, and Susan may want to look into Merck next. Therefore, the EAP described in this application scenario cannot be satisfactorily represented as a complete business process and executed as part of the analysis. Designing the business process representation as she carries out the analysis is inefficient and cumbersome with current BPMS technology.

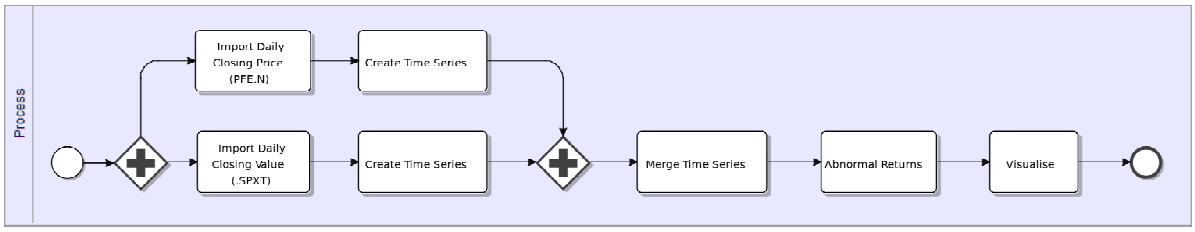

Figure 3: A part of the EAP represented as a business process using BPMN.

EXTENSIONS TO MODEL EXPLORATORY ANALYSIS PROCESSES

This section uses the theory of situation awareness as a means to study EAPs. The application scenario from previous section has shown that EAPs are dynamic in nature so effective process representation must also be dynamic. In addition, we introduce *process templates* as a dynamic and efficient way of creating a process representation of an EAP. Extensions to ADAGE are then proposed incorporating dynamic processes and process templates. Finally an example is presented to illustrate the added value of the proposed extensions.

Using Situation Awareness to Study EAPs

An effective understanding of EAPs is required to determine the challenges involved in representing EAPs as executable processes. This paper proposes the use of situation awareness (SA) theory (Endsley 1995) as an appropriate means to understand EAPs.

SA is used because we believe it is an appropriate fit with the method of high-frequency data research outlined in Dacorogna et al. (2001). Dacorogna et al. note that high-frequency data research consists of three steps. The first step involves data exploration to find the fundamental statistical properties or "stylized facts" exhibited by the data. The second step involves formulating financial models based on the stylized facts discovered in step one. The third and final step is the verification of these models; not only to see if they reproduce the stylized facts in the high-frequency data, but also to evaluate how well these models can predict future market behaviour. Essentially, users attempt to gain an understanding or awareness of market behaviour through financial models that are developed based on the properties of high-frequency data.

In simple terms, having a high level of SA means to be well aware of what is happening around you. Intuitively, people analyse high-frequency data to attempt to find out what is happening in the market. A similar correlation can also be found by looking at the general definition of SA (Endsley 1995; Endsley 2000), which consists of three levels:

- Level 1 SA: Perception. Involves gathering data through our senses and perceiving the relevant information. This correlates well with the first step of the method of high-frequency data research: finding stylized facts exhibited by high-frequency data.
- Level 2 SA: Comprehension. Processing and understanding what the information perceived at Level 1 SA means. This correlates well with the second step: formulating financial models based on stylized facts.
- Level 3 SA: Projection. The ability to predict what will happen in the future based on the available information. This correlates well with the third step: verification and evaluation of financial models to see how well they predict future market behaviour.

Due to the striking correspondence between carrying out high-frequency data research and gaining SA, we feel it is appropriate to understand and model the nature of EAPs through applying SA theory and concepts. Many cognitive processes influence how a person gains SA. The most interesting ones in the context of high-frequency data analysis are goals and mental models (see Figure 4). These are discussed in detail below.

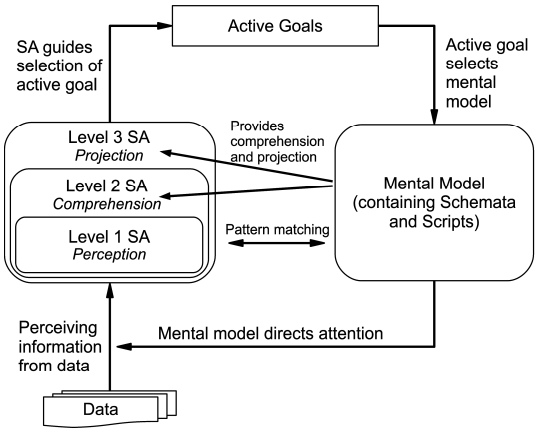

Figure 4: Role of mental models and goals in SA, derived from Endsley (2000).

Goals and the Dynamic Nature of EAPs

Users' actions are motivated by what they want to achieve based on their goals. In the price movement application scenario earlier, Susan's initial goal is to find a significant price movement in Pfizer stocks. This is Susan's initial goal because goals can change over time. When she has found a significant price movement, her new goal is then to see if there is any correlation between the price movement and relevant information available from the news. Endsley (2000) points out that goals affect SA from both a top-down and bottom-up angle. Top-down is also known as goal-driven information processing, in which the user performs tasks in order to achieve a high-level goal (e.g. the steps Susan performs to obtain a chart of Pfizer's significant price movements). Bottom-up is also known as data-driven information processing, in which perceived cues may affect the currently selected goal (e.g. when Susan notices a significant price movement between the 16th and 17th of Dec 2004, she then focuses her attention on these dates). Endsley notes that there can be multiple goals at any one time, but only a subset of active goals direct the user's attention. In addition, "the alternating between bottom-up and top-down processing is one of the most important mechanisms underlying SA" (Endsley 2000, p. 20). This mechanism leads to a dynamic EAP that can change as it is been carried out, e.g. due to changes in active goals and perceived information.

Mental Models and Process Templates

According to Endsley (2000), users develop mental models of how a system operates. Schemata of prototypical situations that users may recognise easily can be associated with these mental models. Users recognise prototypical situations by matching critical cues from the environment to such schemata in their long term memory. The schemata may also invoke scripts that indicate appropriate actions to take, perhaps based on training or experience. Hence users can achieve rapid decision making through the use of schemata and scripts.

In the context of EAPs, scripts of the mental model can give rise to *process templates*. A process template encapsulates a set of steps that a script will suggest as appropriate actions for a particular prototypical situation. The users with their understanding of the system and data (i.e. the mental model) will pick the appropriate process template to use (i.e. activate scripts of the proper actions to take) based on their understanding of the situation and available data (i.e. pattern matching against their schemata). These process templates can then be used as an efficient way of constructing representations of EAPs.

Extending ADAGE to model and execute EAPs

Three extensions are proposed to ADAGE: (1) a dynamic process model representing EAPs; (2) a process execution model; and (3) the use of process templates in the representation of EAPs. Note that the third extension is dependent on the first two extensions, and the second extension is dependent on the first extension. A dynamic process execution model is needed to implement process templates. To achieve dynamic process execution, it is first necessary to have a dynamic process model to represent an EAP. These extensions are built on the idea that process representation of the EAP is created (or modelled) and executed on-the-fly. There is no separate design mode and run mode, so domain expert users can focus on what they need to do next and see the results as soon as possible. The EAP process representation, once created and executed, becomes a record of their actions and common actions can be repeated by means of templates.

The term ADAGE Business Process (ABP) will be used to refer to an executable dynamic process representation of an EAP. This ABP representation is defined below.

ABP1 An **ABP** is a linear sequence of **process fragments**.

ABP2 A **process fragment** is a linear sequence of **tasks**.
ABP3 A **task** consists of a set of **parameters** and refers to an ADAGE **service**.
ABP4 An **ABP** has a minimum of zero **process fragments** and no maximum limit. This means it is possible to start with an empty ABP and have process fragments added later on.
ABP5 A **process fragment** has a minimum of one **task** and no maximum limit.
ABP6 A **task** has a minimum of zero **parameters** and no maximum limit.
ABP7 An **ABP** can have **process fragments** added to it at any time.
ABP8 A **process fragment** must always be added to the end of an **ABP**.

When an EAP is represented as an ABP, it can then be executed by a Business Process Engine that supports the dynamic execution model defined below.

BPE1 **ABP**, **process fragments** and **tasks** can be executed by a Business Process Engine in the *execution* state.
BPE2 When an **ABP** is executed, its **process fragments** (if any) are executed in sequence.
BPE3 When a **process fragment** is executed, its **tasks** are executed in sequence.
BPE4 When a **task** is executed, the referenced ADAGE **service** is invoked using the **parameters** (if any) of the **task** as input.
BPE5 When all of the **process fragments** of an **ABP** have been executed, execution moves from the *execution* state to the *suspended* state.
BPE6 If additional **process fragments** are added to an **ABP** and the Business Process Engine is in the *suspended* state, it moves to the *execution* state and executes the added **process fragments**.
BPE7 Execution of an **ABP** is completed when all (if any) **process fragments** have been executed and no more **process fragments** will be added to the **ABP**.
BPE8 When executing an **ABP**, a history is kept of the **tasks** that have been executed. This **task history** can be examined at a later time.

With the ability to execute ABPs, process templates can be introduced as an efficient means of constructing ABPs. Process templates are defined below.

PT1 A **process template** consists of a linear sequence of **tasks** and a set of **placeholder parameters**.
PT2 The **tasks** within a **process template** are defined in ABP3 with the exception that the **parameters** of the **task** must refer to the **placeholder parameters** of the **process template** it is in.
PT3 A **process template** can be customised into a **process fragment** by replacing the **placeholder parameters** with concrete values. Doing so will update the **parameters** of the **tasks** within the **process template**.
PT4 A **process template** has a minimum of one **task** and no maximum limit.
PT5 A **process template** has a minimum of zero **placeholder parameters** and no maximum limit.

Based on the definitions above, the relationships between the various concepts mentioned are now explained. A domain expert user performs data analysis as a sequence of *steps*; we call this sequence of steps the *EAP*. An *ABP* is a system "instantiation" of an EAP. A step represents a unit of work, and is "instantiated" as a *task*. A task is executed by invoking the referenced ADAGE service. A *process fragment* is a group of one or more tasks. It has been shown that the steps of the EAP cannot be fixed in advance. So at the beginning of the analysis, the user starts with an empty ABP. As each step of the EAP is performed, tasks are created and wrapped in progress fragments and then added to the ABP. A Business Process Engine executes the available process fragments within the ABP and pauses when all process fragments have been executed. As soon as a new process fragment becomes available (i.e. as a result of the user performing a step), the execution continues. Process templates can be used as a means to "replay" previous steps that have been carried out already. Process templates are created from tasks already executed. Once created, process templates can be customised into process fragments and added to an ABP.

Application Scenario Revisited

An example based on the price movement application scenario presented earlier is discussed here to better illustrate the use of process templates. The application scenario is represented as a single ABP, and each step carried out by Susan is wrapped in a process fragment that is added to the ABP and executed immediately. At item 7 in the application scenario, Susan decides to take a closer look at the 16th and 17th of Dec 2004. She intends to perform steps similar to what she has done using the daily data (items 1-6 in the application scenario). To save herself time, she creates a process template from the steps she has done earlier (items 1-6). This process template will have the following placeholder parameters: company's RIC, index's RIC, date range, and time interval. The steps it covers are mapped to the following tasks: (1) call TRTH Import Service with company's RIC and date range; (2) call TRTH Import Service with index's RIC and date range; (3) call Time Series Building Service with company data and time interval; (4) call Time Series Building Service with index data and

time interval; (5) call Merge Service with company time series and index time series; (6) call Abnormal Returns Service with the merged time series; and (7) call Visualisation Service with the abnormal returns event set. Susan then customises the process template with Pfizer (PFE.N) as the company, S&P 500 (.SPXT) as the index, 16-17 Dec 2004 as the date range and an hourly time interval. This customisation creates a new process fragment from the process template, and the process fragment is then added to her current ABP for execution. In addition, Susan can save this process template to a library of process templates for use later.

Figure 5 illustrates the extensions proposed in this section and the idea of process creation on-the-fly. In the figure, X' refers to a task based on X with placeholder parameters; X_Y refers to a task X that is customised based another task Y.

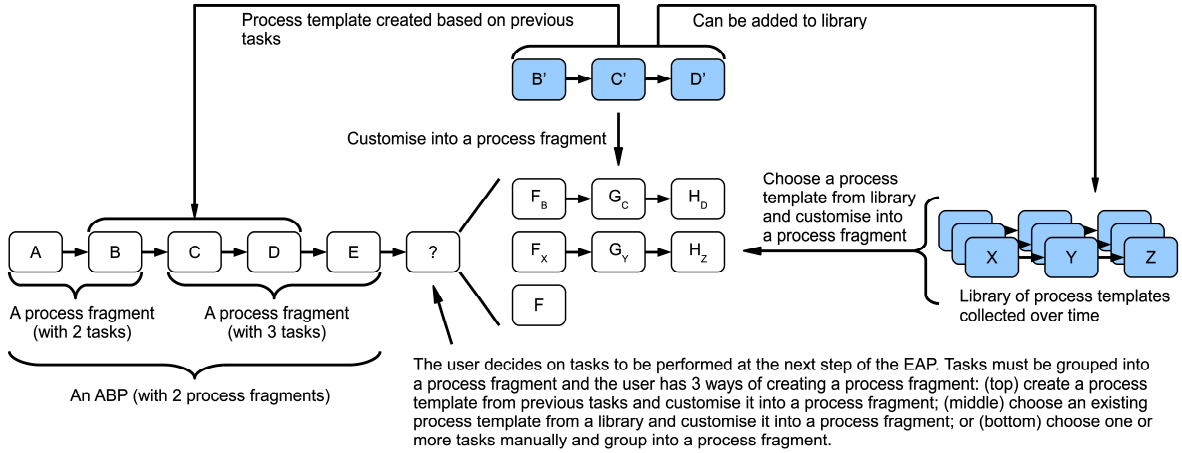

Figure 5: Modelling an exploratory analysis process (as an ABP) on-the-fly.

DISCUSSION AND RELATED WORK

EAPs have some similarities as well as differences when compared to business processes. Figure 3 shows it is possible to model an EAP as a business process where each analysis step corresponds to a task that invokes a service of ADAGE. However, whereas a business process involves elements of organisational structure, task ownership and customers (Davenport 1993); an EAP is a user-driven process that involves elements of the user's mental model and goals. Whereas a business process usually involves collaboration between people, an EAP as described in this paper involves an individual. Hence, many of the business process modelling primitives are not required when modelling EAPs; for example an analyst hardly ever performs multiple tasks in parallel. Reducing the primitives to simple sequences also makes it easy for our intended target audience of researchers and practitioners to construct processes as they analyse vast amounts of data.

Although this paper mentions business process execution in BPMS, it shall be noted that workflow technology has existed before it. In general, Workflow Management has focused on the execution aspect of business processes, while Business Process Management looks at how business is conducted and if the business processes are aligned with business objectives (Jablonski 1995). Workflow technology has been adapted as a means to control the interaction of computational components and provide a means to represent and reproduce these interactions in scientific workflow systems for grid computing such as Triana (Churches et al. 2006) and Taverna (Oinn et al. 2004). Yu and Buyya (2006) have surveyed many scientific workflow systems and note they lack standardisation in workflow specification syntax and semantics. To model and execute EAPs, Workflow Management Systems, BPMS and scientific workflow systems need to be flexible and support process change at run time.

There has been much research on flexible Process-Aware Information Systems (PAIS). The term PAIS covers systems that allow for separating process logic and application code such as Workflow Management Systems (Weber et al. 2008). Schonenberg et al. (2008) categorise run time flexibility approaches into *flexibility by deviation*, *flexibility by change*, and *flexibility by underspecification*. Flexibility by deviation is the ability to deviate from the prescribed process execution path without changing the underlying process model. Flexibility by change, which Weber et al. (2008) calls *adaptation patterns*, is the ability to structurally change the process model at run time and one or more executing process instances may then be migrated to the new process model. Flexibility by underspecification, which Weber et al. (2008) calls *patterns for changes in predefined regions*, consists of techniques such as *late binding*, *late modelling* and *late composition*. Late binding (or late selection) allows parts of the process model (the placeholder) to be left unspecified at design time; then during execution when the placeholder is enabled (or before it is enabled) it is replaced by a concrete activity or process fragment selected from a predefined repository based on rules or user decision. Late modelling also uses placeholders like

the *pockets of flexibility* in Sadiq et al. (2001). During process execution, these placeholders are replaced with process fragments either from a predefined repository like in late binding or created new as required at run time; therefore late modelling encompasses late binding. Late composition allows the user to select or create process fragments on-the-fly within constraints, hence supporting the ad-hoc definition of a process within limits. Late composition is essentially the approach proposed in this paper to model and execute EAPs, so concepts such as process fragments and execution on-the-fly are not new. Process templates have also been utilised to achieve process flexibility and re-use (Kumar and Yao 2009; Sadiq et al. 2001). However, Reichert et al. (2009) note the trade-off between constraints and flexibility: a total lack of constraints can defeat the purpose of PAIS support, while too many constraints can compromise flexibility. We believe the right balance of constraints and flexibility depends on the application domain. Hence the extensions proposed in this paper are an attempt to find that balance in the domain of financial data analysis and eResearch in general.

CONCLUSIONS

This paper has presented a representative application scenario used to evaluate ADAGE's capabilities in representing an exploratory analysis process (EAP) as an executable business process. It is shown that current BPMS technology cannot satisfactorily represent EAPs as fully executable business processes. By applying the theory of situation awareness, this paper shows that EAPs are dynamic in nature so effective process representation also needs to be dynamic. This paper proposes three extensions based on this dynamism: (1) ADAGE Business Processes (ABPs) to represent EAPs; (2) a dynamic execution model to execute ABPs; and (3) process templates to quickly perform repetitive or common steps in EAPs.

As this paper presents work-in-progress, more research is needed in formalising the ABP representation and evaluating the effectiveness of the representation as well as the execution model. Investigation is needed in integrating ABP execution with the ADAGE SOA and to examine what impact it has on the ADAGE architecture. Evaluation is also needed to determine the usability and usefulness of process templates from end-user perspectives.

REFERENCES

Andersen, T.G. 2000. "Some Reflections on Analysis of High-Frequency Data," *Journal of Business & Economic Statistics* (18:2), pp. 146-153.

Bollerslev, T., Law, T.H., and Tauchen, G. 2008. "Risk, Jumps, and Diversification," *Journal of Econometrics* (144:1), May, pp. 234-256.

Chaboud, A.P., Wright, J.H., and Chernenko, S.V. 2008. "Trading Activity and Macroeconomic Announcements in High-Frequency Exchange Rate Data," *Journal of the European Economic Association* (6:2/3), April/May, pp. 589-596.

Churches, D., Gombas, G., Harrison, A., Maassen, J., Robinson, C., Shields, M., Taylor, I., and Wang I. 2006. "Programming Scientific and Distributed Workflow with Triana Services," *Concurrency and Computation: Practice and Experience* (18:10), August, pp. 1021-1037.

Dacorogna, M.M., Gençay, R., Müller, U., Olsen, R.B., and Pictet, O.V. 2001. *An Introduction to High-Frequency Finance*, Academic Press.

Davenport, T.H. 1993. *Process Innovation: Reengineering Work through Information Technology*, Harvard Business School Press, Boston, Massachusetts.

Endsley, M.R. 1995. "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors* (37:1), pp. 32-64.

Endsley, M.R. 2000. "Theoretical Underpinnings of Situation Awareness: A Critical Review," in *Situation Awareness Analysis and Measurement,* M.R. Endsley and D.J. Garland (eds.), Lawrence Erlbaum Associates, Inc., Mahwah, New Jersey, pp. 3-32.

Engle, R.F. 2000. "The Econometrics of Ultra-High-Frequency Data," *Econometrica* (68:1), pp. 1-22.

Goodhart, C.A.E., and O'Hara, M. 1997. "High Frequency Data in Financial Markets: Issues and Applications," *Journal of Empirical Finance* (4:2-3), pp. 73-114.

Guabtni, A., Kundisch, D., and Rabhi, F.A. 2010. "A User-Driven SOA for Financial Market Data Analysis," to appear in *Enterprise Modelling and Information Systems Architectures*, 2010.

Jablonski, S. 1995. "On the Complementarity of Workflow Management and Business Process Modeling," *ACM SIGOIS Bulletin* (16:1), August, pp. 33-38.

Kolata, G. 2004. "A Widely used Arthritis Drug is Withdrawn," *The New York Times,* retrieved 1 July, 2010, from http://www.nytimes.com/2004/10/01/business/01drug.html?scp=2&sq=Vioxx&st=nyt

Kumar, A., and Yao, W. 2009. "Process Materialization Using Templates and Rules to Design Flexible Process Models," in *Lecture Notes in Computer Science* (5858), Springer, Berlin Heidelberg, pp. 122-136.

Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M.R., Wipat, A., and Li, P. 2004. "Taverna: A Tool for the Composition and Enactment of Bioinformatics Workflows," *Bioinformatics* (20:17), pp. 3045-3054.

Rabhi, F.A., Guabtni, A., and Yao, L. 2009a. "A Data Model for Processing Financial Market and News Data," *International Journal of Electronic Finance* (3:4), pp. 387-403.

Rabhi, F.A., Rana, O.F., Guabtni, A., and Benatallah, B. 2009b. "A User-Driven Environment for Financial Market Data Analysis," in *Enterprise Applications and Services in the Finance Industry,* D. Kundisch, D.J. Veit, T. Weitzel and C. Weinhardt (eds.), Springer, Berlin Heidelberg, pp. 64-77.

Reichert, M., Rinderle-Ma, S., and Dadam, P. 2009. "Flexibility in Process-Aware Information Systems," in *Lecture Notes in Computer Science* (5460), Springer, Berlin Heidelberg, pp. 115-135.

Sadiq, S., Sadiq, W., and Orlowska, M. 2001. "Pockets of Flexibility in Workflow Specification," *Lecture Notes in Computer Science* (2224), Springer, Berlin Heidelberg, pp. 513-526.

Schonenberg, H., Mans, R., Russell, N., Mulyar, N., and van der Aalst, W. 2008. "Process Flexibility: A Survey of Contemporary Approaches," in *Advances in Enterprise Engineering I*, J.L.G. Dietz, A. Albani and J. Barjis (eds.), Springer-Verlag, Berlin Heidelberg, pp. 16-30.

Sun, W., Rachev, S., and Fabozzi, F. 2008. "Long-Range Dependence, Fractal Processes, and Intra-Daily Data," in *Handbook on Information Technology in Finance,* D. Seese, C. Weinhardt and F. Schlottmann (eds.), Springer, Berlin Heidelberg, pp. 543-585.

Thomson Reuters 2010. "Thomson Reuters Tick History," Retrieved 1 July, 2010, from http://thomsonreuters.com/products_services/financial/financial_products/quantitave_research_trading/tick_history

Tsay, R.S. 2005. *Analysis of Financial Time Series*, John Wiley and Sons, Inc., Hoboken, New Jersey, 2nd edn.

Weber, B., Reichert, M., and Rinderle-Ma, S. 2008. "Change Patterns and Change Support Features – Enhancing Flexibility in Process-Aware Information Systems," *Data and Knowledge Engineering* (66:3), pp. 438-466.

Yu, J., and Buyya, R. 2006. "A Taxonomy of Workflow Management Systems for Grid Computing," *Journal of Grid Computing* (3:3-4), September, pp. 171-200.

ACKNOWLEDGEMENTS

The work described in this paper is a part of a large project called Ad-hoc DAta Grid Environments (ADAGE) which is funded by the Australian Government's DEST Innovation Science Linkage (ISL) Scheme. We are grateful to the Securities Industry Research Centre of Asia-Pacific (SIRCA) for providing the financial data used in this research on behalf of Thomson Reuters. Our thanks extend to Maurice Peat, Dennis Kundisch, Martin Wagner, Christof Weinhardt, Aim Mangkorntong, and Kader Lattab for helping on different aspects of this work.

COPYRIGHT