Adnene Guabtni, Dennis Kundisch and Fethi A. Rabhi

A User-Driven SOA for Financial Market Data Analysis

This paper is concerned with an environment, referred to as Ad-hoc DAta Grid Environment (ADAGE), which facilitates the analysis of large financial datasets by expert end-users. The paper focuses on the design of a Service-Oriented Architecture (SOA) that makes it possible to define re-usable and interoperable software components as web services to manipulate entities of an underlying event-based data model. Such a model allows for a coherent representation of market activities as events, e.g., high-frequency market data like trade prices and quotes, and a subsequent analysis. The paper also describes an implementation of a user-driven composition tool based on the SOA which allows domain experts to conveniently compose services to execute individualised processes. The approach is illustrated on a case study about analysing the price setting behaviour of issuers in the market for structured products.

1 Motivation & Introduction

With the dramatic increase in the speed and availability of computer networks, a significant proportion of all economic activities is now conducted electronically. As a result, there has been a significant rise in the availability of 'ad-hoc data' i.e., semi-structured data for which useful data processing tools are not readily available (Fisher and Gruber 2005). Examples include Web logs, network traffic messages, periodic sampling of data from sensors and financial reports. In addition, ad-hoc data sources are often distributed over a network and accessed remotely with no integrated data model or the possibility to copy or integrate all data in a single repository for security, privacy or economic reasons. In recent years, such data have become valuable for many different communities of users with different needs. An end-user (typically a domain expert) performs data analysis in an exploratory way, often using several libraries or packages in an iterative trial-and-error fashion (Fischer et al. 2009). 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.

In particular, since the field of financial trading has seen an unprecedented increase in the number of participants and the volumes of trades conducted via electronic markets (Kin and Kaljuvee 2007; Weber 2006), high frequency data has become increasingly available for historical analysis by researchers in fields like econometrics, finance and accounting. For example, the Reuters Data Tick History (RDTH) system (Reuters DataScope Tick History (RDTH) system) allows access to intra-day trade and quote information for over 244 exchanges and Over The Counter (OTC) markets around the world. Users are provided with a Web interface for downloading market data according to different search criteria such as type of instrument, exchange, time period or frequency (e.g., intraday or interday data). Datasets are often stored in a format suitable for viewing as a spreadsheet. A row usually corresponds to a time-stamped piece of information such as the occurrence of a trade, a variation in an instrument's quoted price or an index, the publication of a news story or a market announcement etc. Analysing such datasets requires – at least – expert domain knowledge (e.g., in finance and microeconomics), experience and IT skills. Besides being able to identify suitable data sources and specify the right search criteria, users must be able to perform a wide range of analysis functions (statistical, data mining, language processing) and present results in a suitable form (e.g., through visualisation or report creation). Typically, analysis processes cannot be fully determined in advance as users tend to perform tasks in a piecemeal fashion. They use a dataset to generate some results, they combine results to build new datasets and the process may be repeated iteratively using datasets obtained with different search criteria (e.g., a different time period or different securities of the same product type). Users also tend to use a variety of tools such as Microsoft Excel, SPSS, SAS, and MATLAB to store results and perform routine calculations. When the type of analysis is complex, users spend a lot of effort cleaning, reading and interpreting the data, converting datasets from one format to another, copying some results from one file to another, merging datasets with different semantics, which increases analysis time and the risk of errors.

The purpose of our research is to investigate tools that enable researchers and expert users to analyse ad-hoc data. Our most important application area is the study of data originating from financial communication networks and distributed to academics for research purposes. Since we believe that building bespoke tools is not feasible due to the nature of the research process described above, we have been investigating an open architecture based on Service-Oriented Computing (SOC) principles that allows researchers to explore, frame and solve problems (Rabhi et al. 2009a). The motivation is that SOC has the potential to detach business processes from specific technology solutions and allow users to weave together business services, with no knowledge of the underlying technical service implementations (Sprott 2004).

The rest of the paper is organised as follows. Section 2 gives some context to our paper as well as discusses related work. Section 3 describes our solution which revolves around a SOA designed for high frequency financial market data analysis. Section 4 presents a case study in the area of structured products in which an idealised analysis business process is described as a composition of several identified services. Through a prototype tool, we demonstrate the usefulness of our approach in allowing domain experts to combine services to build a user-driven, individualised (research) process. Finally, Sect. 5 concludes this paper.

2 Context & Related Work

Analysing large amounts of high frequency financial market data – just like other mass data stemming from areas like network traffic, eHealth, Web 2.0 communities – is a time consuming, individualised and processing power-intense endeavour. As the research process is highly individualised, there is a need for *user-driven services*. Moreover, expert domain knowledge and user input is necessary at nearly each single step in the process. There have been many proposed approaches dealing with ad-hoc data management (e.g., Fisher and Gruber 2005) but they do not offer a flexible architecture that supports user-defined analysis processes nor the explicit integration of existing software packages. To address these needs, this research advocates the use of a Service-Oriented Architecture (SOA) based on the *input-processing-output* event streams paradigm. Most SOA research efforts have concentrated on technology implementations and enterprise issues like service composition, orchestration, and management, and there are few studies that attempt to understand the practical challenges to implementing SOA (Luthria and Rabhi 2009). A critical study of the technical constraints faced by developers attempting to develop SOA-based systems finds that a general lack of awareness of the critical aspects of service design poses significant hurdles to the development of these systems (Luthria and Rabhi 2008; Saunders et al. 2006). This has led to an increasing interest in service-oriented software development processes and techniques (Keith et al. 2009) and their application in different application domains, e.g., digital libraries (Baruzzo et al. 2009) and e-government (Álvarez Sabucedo et al. 2009).

Defining SOAs goes hand in hand with two important considerations. The first one is providing an underlying data model that represents domain abstractions and gives coherence between the different services. According to Carey (2008), 'Services provide operations that are akin to verbs. Missing are the nouns – the data entities'. Regarding this issue, our contribution to providing a data model is to adapt the *Common Base Event* (CBE) *Model* (Ogle et al. 2004), promoted by IBM, originally intended to facilitate the analysis of message-like information produced by networked computer systems and captured in log files. This data model can represent both market and news information and has been continuously evolving from previous work discussed in Mangkorntong and Rabhi (2009), Mangkorntong and Rabhi (2007), and Rabhi et al. (2009b). Event-based data models have been also proposed in the area of geographical information systems (Peuquet and Duan 1995) and e-government (Álvarez Sabucedo et al. 2009). The second consideration is the need to support user-defined composition. Our approach in this paper is to give complete control of invoking services to the user but existing solutions for composing web services (Milanovic and Malek 2004), particularly those focusing on user-oriented composition tools (Elgayyar et al. 2008; Gavran et al. 2006), could have been used as well.

3 ADAGE framework

Our solution – referred to as ADAGE (Ad-hoc DAta Grid Environment) – comprises the following elements:

- From a conceptual point of view, we model financial market data as streams of events which can be mapped into an event-based data model. This is described further in Sect. 3.1.
- From an architecture point of view, we provide a SOA that defines services for analysing and manipulating entities in the data model. Some services will be responsible for querying market and news data from existing repositories (thus acting as event sources), some will implement event processing functions (e.g., filtering, aggregation), some will calculate measures about events etc. This is described further in Sect. 3.2

3.1 Event-based modelling of financial market data

In our conceptual model, financial electronic markets (e-markets, Bakos 1991) can be thought of as distributed event systems producing different types of events such as market events or news events. Market events capture data attributes such as bid/ask, types of products being traded, volume/number of products etc. News events capture data related to particular news stories being published by news organisations such as Reuters.

A financial e-market system may operate in different phases – pre-trade, trading and post-trade (Gomber 2000; Harris 2006). The pre-trade phase involves the submission of buy/sell offers to the market, with the market generating trades during the trading phase. Besides clearing and settlement, the post-trade phase involves analysing the trades that have taken place, and understanding whether market rules have been followed, and signalling any illegal behaviour. Each of these phases generates different types of market events, and produces different event patterns. In this paper, we are only considering trading events originating from exchanges as they represent the type of datasets that are widely available for researchers. The ADAGE data model has been designed as a reference model for all the

ADAGE services. As mentioned earlier, this data model has been adapted from the CBE Model (Ogle et al. 2004) promoted by IBM. In our case study, we are using the following entities from the data model:

- *Event source*: The event source is the primary source of high frequency data. They can be either log files, databases, available web services, information portals etc. The main instance of an event source in our case study is SIRCA's RDTH system.
- *Event*: An event is the base entity from which other types of event entities are derived. The most important attributes in an event are the timestamp and the financial product concerned by the event. In our case study, the three types of events are *Trade* (representing the occurrence of a trade), *Quote* (representing the broadcast of a quote) or *Measure* (representing a snapshot of one of the market measures like volume-weighted average price (VWAP)).
- *Product*: A product (tradable or non-tradable) is uniquely identified through some identification code. In our case study, products are named using the Reuters Identification Code (RIC) and the two different types of products used are *Index* (non-tradable) and *Tracker Certificates* (tradable).

The proposed event-based data model makes it possible for several services to process data in a consistent way by sharing a common data reference model. Since this contribution focuses on the SOA aspect of the ADAGE framework, additional details of the event-based data model for financial market data can be found in Rabhi et al. (2009b).

3.2 ADAGE SOA

The ADAGE Service Oriented Architecture is designed according to user-driven requirements allowing execution of processes to be piloted by users at any stage of their execution. Three main categories of services have been identified: *Event Import Services* for providing market and news events, *Event Processing Services* for processing financial market and news events, and *Event Export Services* for extracting added value information from financial market and news events.

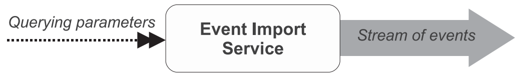

Figure 1: Event Import Service category

Event Import Services for providing market and news events (see Fig. 1): An Event Import Service, which is used to construct events from a data repository, has querying parameters as input and a stream of events as output. The querying parameters are different from one Event Import Service to another and will be specific to the data repository involved. E.g. a service for providing market events that occurred between a date $d1$ and a date $d2$ will produce a set containing all market events between that two dates (providing the event source contains such events otherwise an empty set is returned).

An Event Import Service aims to map information about event attributes from data repositories to the event-based data model and build a result set of events. Therefore, an Event Import Service performs two tasks:

Task 1 – Querying/Filtering: Collecting pieces of information provided by a data repository depending on querying criteria.

Task 2 – Mapping: Associating (adapted) event attributes from the repository to the data model compliant output. Import services allow users to specify various transformation functions adapted to the repository and to transform the event attribute notation to our common event-based data model (e.g., the timestamp format may differ from one event source to another and is not always normalised to GMT so a function to normalise time is necessary).

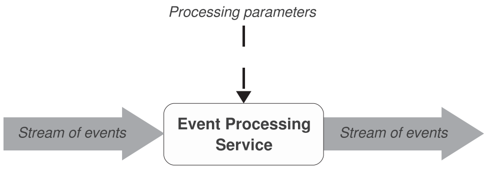

Figure 2: Event Processing Service category

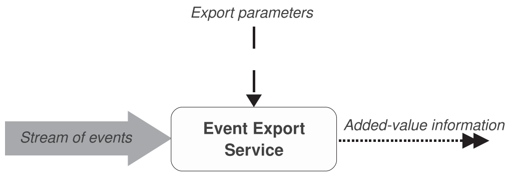

Figure 3: Event Export Service category

Event Processing Services for processing financial market and news events (see Fig. 2): An Event Processing Service has streams of events as both input and output. Input events can be generated from an Event Import Service or another Event Processing Service. An Event Processing Service also has a set of input parameters to guide the processing of the events. For example, a service for building time series will produce *Measure* events representing aggregations of *Trade* and *Quote* events. Such aggregations can be made over fixed time intervals and therefore, the size of the interval (e.g., 1 hour) should be part of the input parameters to the service.

Event Export services for extracting added value information from financial market and news events (see Fig. 3): An Event Export Service has a stream of events as input and provides added value information as output such as graphs or tables, e.g., in xml- or csv-format. As in an Event Processing Service, such a service requires a set of parameters to customise the output, e.g., type of events to include in a visualisation output.

The categorisation of the ADAGE services is justified in the next section. In order to ensure the usefulness and applicability of such a categorisation, we need to identify relevant services to implement and integrate to the ADAGE framework. Such identification is based on domain expert knowledge and case studies such as the one described in Sect. 4.

3.3 Service Composition and User Interaction

A service in a SOA is a self-contained piece of software providing for some functionality by processing a clearly defined input to provide for a clearly defined output without calls to other services embedded in it. Therefore, SOAs use the *input-processing-output* paradigm which has strongly influenced the traditional approach to system design and also service design. Such a paradigm makes services look as black boxes where user interaction is required to provide data only at the starting point of a process, and service composition, that needs service interoperability, is not explicitly supported. In fact, the output of a service is not guaranteed to be supported as input of other services. Enabling composition over interoperability is then an important requirement. In the context of our work, expert users are dealing with ad-hoc data where results of complex processes are not predictable and processes themselves are not expressed in advance. Depending on the results of the previous steps, the expert will choose the most appropriate alternative for the next step and invoke different services accordingly. User interaction is therefore an important requirement, too.

The two key requirements described above, enabling composition of services (over their interoperability) and enabling user interaction, have guided our architectural choice to limiting the categories of services to three, *Event Import Services*, *Event Processing Services* and *Event Export Services*, corresponding to the adapted *input-processing-output* paradigm visualised in Fig. 4.

- **Enabling user interaction**: It is possible to enable user interaction with all three categories of services over the parameters input in order to tune event import, processing or export.

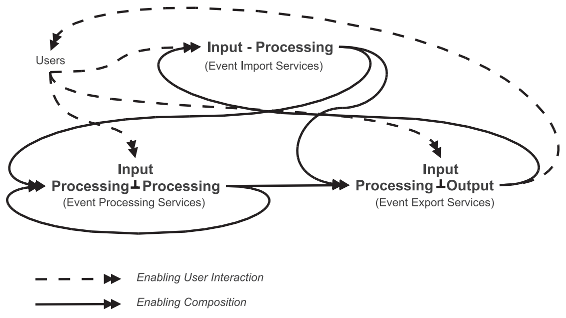

Figure 4: Adapted input-processing-output paradigm to enable user interaction and service composition

- **Enabling composition**: It is possible for *Event Import Services* to provide streams of events to *Event Processing Services* (the output of *Event Import Services* is a stream of events that can be used as input for *Event Processing Services*). Also, it is possible for several *Event Processing Services* to concurrently operate on different event streams and the output of one service to be used as input for another (the input and output of such a category of services is a stream of events). Finally, it is possible for *Event Export Services* to provide querying parameters (as resulting added value information) – but not a stream of events – to *Event Import Services* (the output of *Event Export Services* is a set of added value information that can be used as querying parameters for *Event Import Services*) as illustrated in Fig. 5.

Having presented the challenges in ad-hoc data analysis and briefly described an event-based data model as well as the associated SOA for analysing financial market events, the following case study identifies concrete services in order to illustrate the usefulness and applicability of the ADAGE framework.

4 Case Study

Research is about the testing of already established theories, refining existing theories, as well as developing and exploring new theories. This is a time-consuming, highly-individualised task

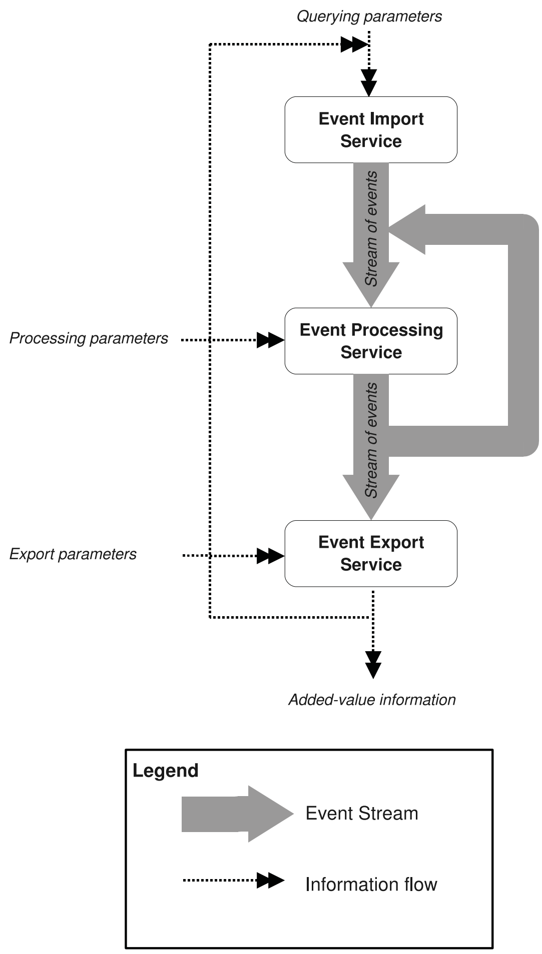

Figure 5: Combining different categories of event processing services

that is often – besides making use of an underlying and generally agreed on methodology – carried out in a trail-and-error fashion. If the presented ADAGE framework shall facilitate this process, the power to design and refine processes has to be shifted to the user in order to process mass data from financial markets. We use the example of the valuation of structured products in order to illustrate the ADAGE framework.

4.1 Structured products

From a legal point of view, structured products are nothing but a contract between the issuer and the buyer/holder of the structured product. This contract is described in the Securities Prospectuses, which is for instance in the EU harmonised according to Commission Regulation (EC) No. 809/2004 of 29 April 2004 (Prospectus Regulation). In this contract, the issuer describes the payoff structure of the product and the rights the issuer as well as the holder of the product possess. Generally, structured products are (unsecured) debt obligations with payments of interest and/or repayment of the principal contingent on one or several risk factors.[1] These cash flow characteristics can be exhaustively described by three objects (Cherubini and Della Lunga 2007):

Object 1: A set of maturity dates representing the due date of cash flow payments

$$\{t_1, t_2, \ldots, t_n\}$$

Object 2: A set of cash flows representing the interest payments on the capital

$$\{c_1, c_2, \ldots, c_n\}$$

Object 3: A repayment plan of the capital

$$\{k_1, k_2, \ldots, k_n\}$$

Building up a structured product amounts to setting rules allowing for univocal definition of each of these objects. All three objects may in principle contain deterministic or stochastic elements.

After the issuance of the products, they are continuously traded in the secondary market. Hence, prices for structured products can be obtained usually from one or several exchanges. The liquidity in these securities is compared to, say, blue-chip stocks, very low. Thus, it is a market maker, which in most cases is at the same time the issuer itself that provides for the necessary liquidity. By this the issuer constitutes one counterparty in almost all trades (Öchsner 2008). Consequently, analysing quotes and prices of trades in structured products means at the same time analysing the price setting behaviour of an issuer – and there is some evidence in literature that issuers use the price setting to their own advantage (Benet et al. 2006; Grünbichler and Wohlwend 2005; Stoimenov and Wilkens 2005; Wilkens and Stoimenov 2007; Wilkens et al. 2003).

We use the example of the price setting behaviour of an issuer for structured products in order to illustrate the framework for a number of reasons:

- Structured products have gained a lot of importance both in business practice (Wasescha 2008) as well as in research (see, e.g., Benet et al. 2006; Burth et al. 2001; Grünbichler and Wohlwend 2005; Klein and Kundisch 2008; Stoimenov and Wilkens 2005; Wilkens and Stoimenov 2007; Wilkens et al. 2003) in recent years.
- The valuation of structured products is not a trivial task but requires expert knowledge. A large amount of market, quoting and trading data has to be integrated and processed in order to be able to decide whether a structured product is priced fairly or not.
- Since a lot of private investors, both short-term and long-term oriented, have discovered structured products as an investment alternative in recent years, it is also an issue of consumer protection to shed some light on the issue of fair pricing of issuers.[2]
- A number of theories have been developed recently – mainly in finance such as the *life cycle*

[1] Accordingly, the SEC, for instance, defines structured products in its Rule 230.434 as 'Securities whose cash flow characteristics depend upon one or more indices or that have embedded forwards or options or securities where an investor's investment return and the issuer's payment obligations are contingent on, or highly sensitive to, changes in the value of underlying assets, indices, interest rates or cash flows.'

[2] This issue has gained increasing importance due to the bankruptcy of well-known issuers like Lehman Brothers during the global financial market crisis.

hypothesis of structured products (Wilkens et al. 2003) – that explain, e.g., the price setting behaviour of issuers of structured products. Still, compared to the number of products – e.g., in Germany far above 350,000 – and the fragmentation as well as lacking standardisation of the market, there is still ample need for further research.

Thus, structured products make apparently a compelling case to identify and implement the required services and at the same time illustrate the applicability of the proposed framework.

For the case study, we use the simplest possible product in the market of structured products, the so called *tracker* (or *index*) *certificate*. Tracker certificates belong to the group of *retail derivatives*. Retail derivatives are structured products that are engineered taking into account the specific (putative) needs of private investors. Moreover, they are – at least in Germany and Switzerland – sold via specific trading venues and with specific trading rules that are tailored to private investors.

Tracker certificates usually have the following characteristics: There are no interest payments (cf. object (2)) and they are issued as open-ended products, i.e., there is no fixed date of maturity (cf. object (1)). However, two cases can occur when a maturity date is set: The issuer draws the cancellation right or the holder draws a granted exercise option. In both cases the issuer has to provide for a cash settlement (cf. object (3)) usually according to the following formula, which can be found in the Securities Prospectus of the product:

$$\text{cash settlement} = \text{reference price of the index at the time of exercise or cancellation} \times \text{multiplier}$$

Determining and analysing the fair value of a tracker certificate continuously while it is being traded in the secondary market is straightforward (Klein and Kundisch 2008). Assuming the efficient market hypothesis Fama (1970) holds and abstracting from the credit risk – the issuer might go bankrupt leaving the buyer of its products with a total loss[3] – the current (inner or fair) value of a tracker certificate TC_t^{FV} at time t should equal the current value of the index I_t at time t times the tracker-specific multiplier r (e.g., 0.01[4]).

$$TC_t^{FV} = I_t \times r \tag{1}$$

For valuation purposes, usually, the time t_I where an index value was reported and the time the trade (or a quote) in a tracker certificate occurred t_{TC}, will not be exactly the same. Thus, it follows that:

$$min(t_I - t_{TC}) \text{ with } (t_{TC} - t_I) \geq 0 \tag{2}$$

At the same time, the time difference shall (or even must) not be too big, since then it might happen for instance that we compare the closing price of the index from yesterday with the opening price of the certificate today. Hence, the absolute difference between these two points in

[3]With the turmoil in the financial markets, it has become obvious that this risk should not be underestimated. Before it was taken over by JP Morgan, Bear Stearns was not able to set prices for its structured products for about a fortnight in March 2008. Moreover, investors in securities issued by Lehman Brothers have suffered severely since Lehman filed for Chapter 11 protection on 2008-09-15.

[4]The multiplier is used to make the products tradeable for private investors who usually invest small amounts. Take the example of the German blue-chip index DAX, which had an end of day index value of 5,574 points on 2009-09-09. Without a multiplier one unit of an index tracker on the DAX would cost around 5,574 Euros. Obviously most investors would not even be able to buy one single unit and it would not be possible to invest say 2,500 Euros. Thus, with a multiplier of, e.g., 0.01, the price for the index tracker that day will be around 56 Euros. By this, the granularity is appropriate for investors to invest nearly arbitrary small or big amounts of funds.

time should not exceed a predefined limit l.

$$|t_I - t_{TC}| \leq l \tag{3}$$

Equation (1) holds of course only true if both the certificate is traded as well as the index is calculated/determined at the time of the valuation. If the certificate is traded, but the index is not, we have to find another reference product (e.g., a future on the index that might still be traded) in order to evaluate the tracker certificate.

To analyse the price setting behaviour or the fairness of issuers, we can calculate the deviation D of the current price of the trade i at time t from the fair value as follows:

$$D_{t,i} = \frac{TC_{t,i} - (I_t \times r)}{TC_{t,i}} \tag{4}$$

To compare the price setting behaviour in specific time periods, we might define a period a and calculate, e.g., the mean or median of the deviations and the standard deviation of the mean. Moreover, other measures such as quartiles over the period a can be calculated and compared over time.

Before these kinds of calculations, comparisons and subsequent theory testing can take place, obviously, different streams of events (certificate trade events and index measure events) have to be integrated and matched on a time basis (cf. equations (2) and (3)). Moreover, issues such as missing data and wrongly reported raw data have to be tackled. This is usually a very time-consuming and ad-hoc process. In the following a number of services are described that were identified and implemented that shall facilitate this research process and at the same time illustrate the merits of the ADAGE framework.

4.2 Services to valuate tracker certificates

In order to analyse the price setting behaviour of issuers a number of services have been developed according to the RDTH archive. Figure 6 illustrates the overall business process to valuate tracker certificates using the ADAGE framework. White services correspond to Event Import Services, dark grey services correspond to Event Processing Services, while light grey services correspond to Event Export Services.

Broadly, the business process is described as follows: after selecting the tracker certificates that shall be analysed, the data query is generated and sent to the appropriate sources (RDTH Import Service). The result set will first be analysed at a high level to get an idea of what it contains, whether there are missing periods and other potential problems.[5] Consequently, the proper underlyings for the chosen tracker certificates are determined by the user. Usually the underlying will be an index such as DAX, SMI or Dow Jones. Again, a data query is generated and sent to the data repository (RDTH Import Service). 'Suspicious' data points need to be filtered and processed (e.g., deleted, modified, just flagged) as early as possible for both the certificates as well as the underlying(s) (Event Filtering Service). As soon as the data about the tracker certificate(s) and the proper underlying(s) have the quality the researcher is looking for, the valuation process can start (Event Merge Service followed by Time Series Building Service). The result set or a summarisation of it will be analysed to, e.g., test existing theories or develop new ones by visualisation or via external – usually statistical – tools (Visualisation and Download Services). In the following the mentioned services are described in more detail.

[5] This is performed by an *Event Preview Service* which belongs to the category of Event Export Services and is not further described in this paper.

4.2.1 Event Import Services

For financial market data, a market events import service (RDTH Import Service) locates quote and trading data according to some querying parameters in order to provide/generate corresponding events from the RDTH (Reuters DataScope Tick History (RDTH) system) data repository. However, to get information about structured products, there might also be other sources containing semi-structured meta-data about the products such as the website of an exchange. Moreover, not only data about the products themselves but also about issuers might be necessary for an analysis. Finally, the events may be timestamped either in local time or standardised to GMT. An Event Import Service always includes the standardisation to GMT since the valuation of retail derivatives will sometimes include data from different markets that are located in different time zones. Typical cases – also used in our analysis – in which the RDTH Import Service is invoked are illustrated in the following examples:

Example 1: Query on all market events occurred between two dates and concerning particular securities traded at a particular market, e.g., the two tracker certificates with the RICs 'DE709335.F' and 'DE543741.F' on the DAX index traded at the exchange Scoach from the date of issuance until today.

Figure 7 shows typical sample data of the tracker certificate with the RIC 'DE709335.F'[6] for less than 2 minutes in the widely used comma-separated values (CSV)-format. In addition data for the German blue-chip index DAX with the RIC '.GDAXI'[7] are shown. The data for the certificate contains both trade and quote data. Note that the timestamp for the certificate is in GMT while it is in local time for the index.

Example 2: Query on all market events occurred between two dates and concerning a particular security traded at different markets, e.g., the tracker certificate with the RICs 'DE709335.F' and 'DE709335.EW' on the DAX index traded at the Frankfurt-based exchange Scoach and at the Stuttgart-based exchange EUWAX from the date of issuance until today.

4.2.2 Event Processing Services

Event Merge Service:
This service matches timestamps of quotes or trades with according timestamps of the underlying(s) (for the determination of the proper underlying see Sect. 4.2.3) and puts a flag at a trade/underlying-tuple and quote/underlying-tuple, respectively, if the time difference exceeds a user specified limit according to equation (3).

Time Series Building Service:
This service calculates a range of financial measures from trades and quotes, as specified by the user in its parameters list. In this case study, the main measure that needs to be determined is the deviation from the fair value according to equation (4). The output of this service is a set of measure events describing the basic deviation between trades or quotes of a structured product and the measure events of its corresponding underlyings.

Event Filtering Service:
This service is used to filter out some events according to some user-defined parameters. In this case study, it detects outliers that either point to reporting mistakes in the raw data or help a user

[6] RDTH uses the RIC (Reuters Instrument Code) to uniquely identify securities traded at specific markets. The RIC 'DE709335.F' relates to the security with the International Securities Identification Number (ISIN) 'DE0007093353' traded at the Frankfurt-based exchange Scoach Germany ('.F').

[7] The RIC '.GDAXI' is associated with the ISIN 'DE0008469008' representing the index DAX determined based on the stock data received from trades on the electronic system XETRA.

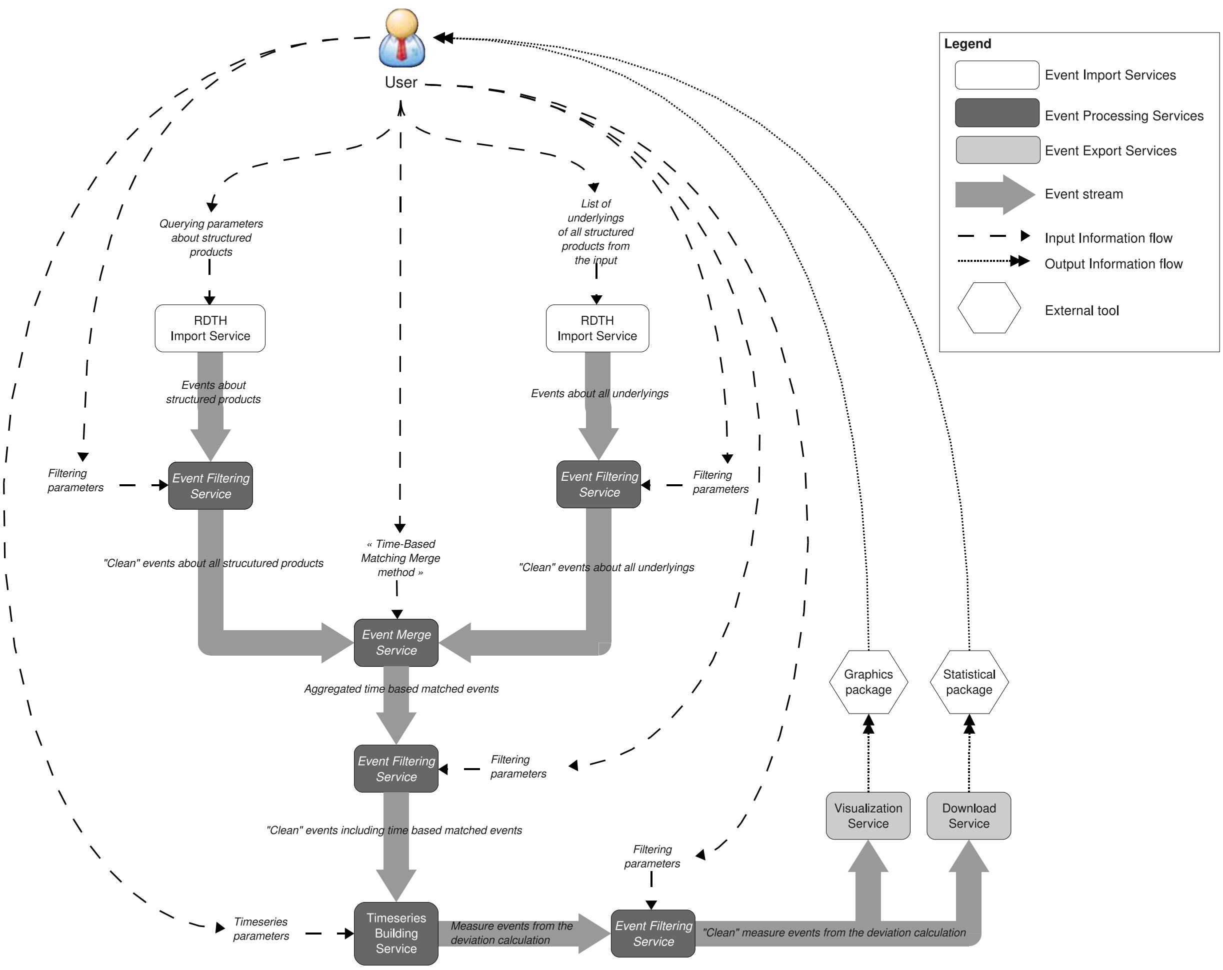

Figure 6: Business process to valuate tracker certificates using the ADAGE framework

```
#RIC,Date[G],Time[G],GMT Offset,Type,Price,Volume,Bid Price,Bid Size,Ask Price,Ask Size,
DE709335.F,04-AUG-2005,14:23:09.234,+2,Trade,48.79,400,,,,
DE709335.F,04-AUG-2005,14:23:09.625,+2,Quote,,,48.79,2000,48.79,2000
DE709335.F,04-AUG-2005,14:23:11.014,+2,Trade,48.79,400,,,,
DE709335.F,04-AUG-2005,14:23:11.604,+2,Quote,,,48.79,2000,48.79,2000
DE709335.F,04-AUG-2005,14:23:37.301,+2,Quote,,,48.8,,48.8,
DE709335.F,04-AUG-2005,14:23:54.522,+2,Quote,,,48.79,,48.79,
DE709335.F,04-AUG-2005,14:24:42.230,+2,Trade,48.78,600,,,,
DE709335.F,04-AUG-2005,14:24:42.730,+2,Quote,,,48.78,2000,48.78,2000
DE709335.F,04-AUG-2005,14:24:43.231,+2,Trade,48.78,600,,,,

#RIC,Date[L],Time[L],Type,Price,Volume
.GDAXI,04-AUG-2005,14:23:00.396,Index,4879.64
.GDAXI,04-AUG-2005,14:23:15.205,Index,4879.65
.GDAXI,04-AUG-2005,14:23:30.163,Index,4880.17
.GDAXI,04-AUG-2005,14:23:45.153,Index,4880.55
.GDAXI,04-AUG-2005,14:24:00.303,Index,4880.7
.GDAXI,04-AUG-2005,14:24:15.212,Index,4880.69
.GDAXI,04-AUG-2005,14:24:30.231,Index,4880.67
.GDAXI,04-AUG-2005,14:24:45.140,Index,4880.13
```

Figure 7: Data about certificate and underlying in CSV-format

to identify particularly interesting events. This is done after each processing step, e.g., on the output after invoking the RDTH Import Service or on the output from the Time Series Building Service either before or after using the Summarisation Service. The definition of an outlier is highly context specific and as such demands for expert domain knowledge as input.

In Tab. 1, an example of a rare but still typical outlier in raw event data is presented. The table shows consecutive trades in the afternoon of the 2007-09-26 in the tracker certificate with the RIC 'DEGS0J2F.F'. Most likely due to a reporting mistake, the price for the trades at 17:09 (= 5:09 PM) should be both 78.05 instead of 8.05.[8]

Table 1: Example for outliers in raw data

Time	Price	Volume
14:27:28	78.04	1000
17:09:18	8.05	2900
17:09:36	8.05	220
17:10:36	78.04	220
17:11:19	78.05	2900

4.2.3 Event Export Services

Visualisation Service:
This service may be used to visualise the resulting dataset from, e.g., the Time Series Building Service either before or after the usage of the Summarisation Service. Based on events processed using the ADAGE framework, Fig. 8 shows the development of the deviations for trades occurred in the security with the RIC 'DEWLB501.F' over time. Each dot corresponds to one trade in the security.

Obviously, the price setting behaviour of the issuer changed in that period. Thus, this visualisation may be a first and interesting step into

[8]Note that apparently no trade occurred between 2:27 PM and 5:09 PM.

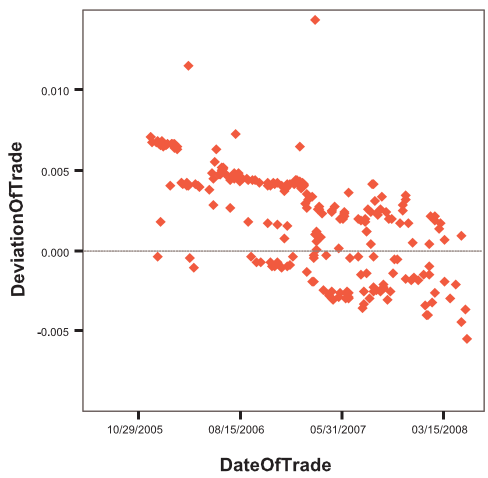

Figure 8: Deviation from fair value over time

the further – also statistical – analysis of the price setting behaviour of issuers of structured products.

Download Service:
This service allows users to download results in a format (e.g., CSV) suitable to importing into a statistical package. This way, existing tools can be integrated into the ADAGE framework. It is also possible to invoke this service from a package via an API or command line invocation.

4.3 Experiments

The user-driven ADAGE Graphical Composition Tool allows users to execute and compose the ADAGE services in a user-friendly manner. The three categories of services, which we described previously, are organised in three main panels (on the right of the user interface) using icons as shown in Fig. 9. An additional category called 'Other Services' is also available to provide for administrative services such as 'Trash Bin Service', 'Save Session' or 'Restore Session'. The left panel shows Event Result Sets resulting from Event Import Services or Event Processing Services as well as exported files resulting from Event Export Services. Executing a service is made possible by clicking on the corresponding icon. Then the user is asked to select event sets from the left panel to be used with the services as well as some input parameters over a form. For example, to execute the RDTH Import Service, the user is asked to specify the desired products, time period and event types required by the user. Such action makes the service executing the query and the resulting events are contained in an Event Result Set which is presented to the user as an icon on the left panel (see Fig. 9).

Using such an intuitive way of service invocation, the user can use event sets resulting from one service to be processed by other services in the two other categories provided by the ADAGE SOA (Event Processing Services and Event Export Services). In such a case, the user is asked to provide additional information as input parameters to perform the processing or the export. As a result, the user is defining queries for the planned analysis (using the services forms), feeding Event Result Sets into the right underlying Event Import Service, getting additional Event Result Sets, choosing additional processing to execute on them (cf. to the idealised process visualised in Fig. 6), etc. Thus, the analysis process is fully user-driven and the three categories of services help users to achieve their needs. At each step of the user-driven process, users can include invocation to the Event Preview Service. This service provides short but useful information on the content of an Event Result Set.

The flexibility and thus the advantage of this tool over traditional data processing and analysing approaches shall be illustrated by a small example concerned with outliers in the context of an analysis of so called *time-of-the-day effects*, a widely studied issue in financial markets (see, e.g., Ozenbas et al. 2002).

We used a sample dataset consisting of all trade events of eleven tracker certificates over eight years with the German blue-chip index DAX as underlying. Using an Oneway ANOVA test, we find no statistical significant differences in the hourly mean deviations[9] for this sample data when we include extreme outliers. By *extreme outliers* we mean price deviations from the calculated fair value – events that we get as output from the Time Series Building Service – that are above 10%. Usually, such (extreme) deviations point to some reporting mistake in the raw data rather than a stark mispricing by the issuer (see Tab. 1). When we exclude these outliers (just 18(!) among 23,031 data points) from the statistical test (see results reported in Tab. 2 below) all of a sudden, we find statistically significant differences in the mean hourly deviations. With the tool, these changes in the analysed event set may be accomplished quickly and conveniently: The (initial) stream of events as output from the Time Series Building Service is used as input to the Event Filtering Service. Based on the appropriate user input, the outliers are removed by this service. The initial event stream and this new 'cleaned' event stream may both be downloaded and input into a statistical package so that the type of test can be specified.

Table 2: Oneway ANOVA – Summarised Results (Extract)

Sample	F-Value	Lev. of Sig.
Incl. Outliers	1.163	0.311
Excl. Outliers	2.798	0.002

Moreover, the dataset may be split up easily and the tests may be performed for arbitrary subsamples and different timeframes according to the user's requirements.

Once more, this example illustrates the need for expert knowledge and user-piloting in the research process, since it is not at all clear upfront

[9]That is, all calculated deviations for trades in the certificates that occurred, e.g., in the time between 10 and 11 AM were summarised to an hourly mean deviation.

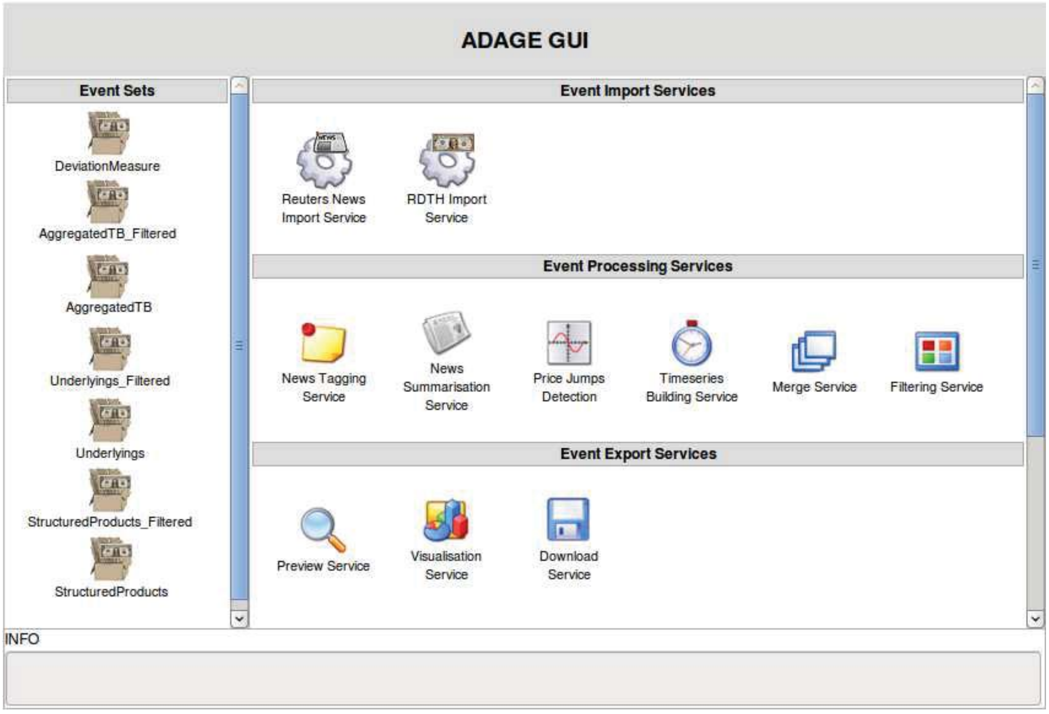

Figure 9: User-Driven ADAGE Graphical Composition Tool

whether something can be classified as an outlier due to false reporting rather than due to a questionable price setting behaviour of an issuer.

5 Conclusions & Future Work

In this paper, we propose a SOA that provides different categories of event-based services that assist users in defining complex analysis processes over heterogeneous and distributed ad-hoc data sources. The proposed framework helps researchers focusing on the research problem, the underlying theories and the correct statistical treatment instead of having them taking care of IT-related challenges. To illustrate this concept, a user-friendly ADAGE Graphical Composition Tool was developed to enable a number of pilot services to be invoked and applied to a complex case study involving structured products. Due to their granularity, the services may be re-used to analyse also the pricing behaviour of issuers with respect to other types of structured products than the discussed one in this case study.[10] In addition, most of these services will be also applicable in completely different domains as long as the underlying data model is event-based and as long as there are different data sources that have to be matched.

The flexibility of the user-driven service architecture in combination with the respective GUI allows for differing and highly individualised processes that are of utmost importance in the research process. Moreover, the user may decide individually after each step in the process, whether to invoke another service of the pre-defined set of services offered within the ADAGE framework or export the current result set for

[10] Apparently, there may be the need for some new services as well to perform an analysis, however, all the services presented in the case study above will be applicable and even needed. A categorisation of the different product categories and types of products may be found, e.g., at the website of the Swiss Structured Products Association (`http://www.svsp-verband.ch/home/produkttypen.aspx?lang=en&pc1=&pc2=`).

further analysis in other software packages (e.g., SPSS, SAS, MATLAB or STATA). It is important to note that by this, the suggested approach does not favour any theory or is limited to the testing of specific theories. In addition, the result sets calculated within the ADAGE framework may also be combined and enriched with other information, since there are also services available that facilitate the querying and analysis of news data (Rabhi et al. 2009a). Another advantage is that since the actual processing of the data is performed on a remote server, the performance of the client's machine is no bottleneck. Finally, the programming skills of the end-user are less important compared to the 'traditional' process of analysing the data, i.e., implementing small programs within the just mentioned software packages to perform the tasks described as services above.

We envisage that future work will be pursued in several directions. In the short term, we need to address some performance problems associated with the processing of very large data files[11]. Potential solutions include the usage of processing power grids and finding more efficient implementations of the services in terms of memory usage and algorithms. Existing techniques for mapping data grids with processing grids to avoid huge transfers of data from the data location to the processing location need to be adapted to this context (Foster et al. 2002).

Another important research area is addressing the issue of user-driven service compositions. Due to the exploratory nature of ad-hoc data analysis, defining static business processes in a service composition language is unlikely to work well in such a context of *emergent business processes* (Marjanovic 2005). Our belief is that the proposed ADAGE SOA can be better described as 'metadesign' framework i.e., 'a solution space in which users can create their own solutions to fit their needs' (Fischer et al. 2009). This school of thought emphasises not only the technical infrastructure but also the social infrastructures in which expert domain users actively participate in co-designing the infrastructure through collaboration. Therefore, one interesting avenue of research is to apply metadesign guidelines for sociotechnical systems (e.g., support human-problem interaction, promote mutual learning and support, providing incentives) in improving the usability of the system. For example, the data model needs to be able to evolve and still allow other developers to evolve existing services or implement new services. In the longer term, the basic idea of the architecture and data model may be transferred to other domains that are also characterised – at least partially – by time series analysis, e.g., economic analysis, health care, behavioural analysis.

[11]For instance DAX-tick-data just for the year 2006 adds up already to about 850 MB. Analysing tracker certificates, though, may include a time series of eight years for several certificates and underlyings at the same time.

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. Our thanks extend to Haresh Luthria, Martin Wagener, Christof Weinhardt, Aim Mangkorntong, Lawrence Yao and Kader Lattab for helping on different aspects of this work.

References

Álvarez Sabucedo L. M., Anido Rifón L. E., Míguez Pérez R., Santos Gago J. M. (2009) Providing standard-oriented data models and interfaces to eGovernment services. In: Computer Standards & Interfaces 31(5), pp. 1014–1027

Bakos J. Y. (1991) A strategic analysis of electronic marketplaces. In: MIS Quaterly 15(3), pp. 295–310

Baruzzo A., Casoto P., Dattolo A., Tasso C. (2009) A Conceptual Model for Digital Libraries Evolution. In: Filipe J., Cordeiro J. (eds.) WEBIST. INSTICC Press, pp. 299–304

Benet B. A., Giannetti A., Pissaris S. (2006) Gains from structured product markets: The case of reverse-exchangeable securities (RES). In: Journal of Banking & Finance 30(1), pp. 111–132

Burth S., Kraus T., Wohlwend H. (2001) The Pricing of Structured Products in the Swiss Market. In: The Journal of Derivatives 9(2), pp. 30–40

Carey M. J. (2008) SOA What? In: Computer 41, pp. 92–94

Cherubini U., Della Lunga G. (2007) Structured Finance: The Object Oriented Approach. Wiley Finance Series

Elgayyar M. M., Alda S. J., Cremers A. B. (2008) Towards a user-oriented environment for web services composition. In: WEUSE ’08: Proceedings of the 4th international workshop on End-user software engineering, Leipzig, Germany. ACM, New York, pp. 81–85

Fama E. F. (1970) Efficient Capital Markets: A Review of Theory and Empirical Work. In: The Journal of Finance 25(2), pp. 383–417

Fischer G., Nakakoji K., Ye Y. (2009) Metadesign: Guidelines for Supporting Domain Experts in Software Development. In: IEEE Software 26(5), pp. 37–44

Fisher K., Gruber R. (2005) PADS: a domain-specific language for processing ad hoc data. In: Sarkar V., Hall M. W. (eds.) PLDI. ACM, pp. 295–304

Foster I., Kesselman C., Nick J., Tuecke S. (2002) The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, open Grid Service Infrastruct WG, Global Grid Forum, June 22, 2002

Gavran I., Milanovic A., Srbljic S. (2006) End-User Programming Language for Service-Oriented Integration. In: Proceedings of 7th Workshop on Distributed Data and Structures (WDAS 2006). CA, USA, pp. 4–6

Gomber P. (2000) Elektronische Handelssysteme. Innovative Konzepte und Technologien im Wertpapierhandel. Physica, Heidelberg

Grünbichler A., Wohlwend H. (2005) The Valuation of Structured Products: Empirical Findings for the Swiss Market. In: Financial Markets and Portfolio Management 19(4), pp. 361–380

Harris L. (2006) Trading and Exchanges: Market Microstructure for Practitioners. Oxford University Press, New York

Keith M., Demirkan H., Goul M. (2009) Service-Oriented Software Development. In: Proceedings of the Americas Conference on Information Systems (AMCIS). Hamburg

Kin K., Kaljuvee J. (2007) Overview of Electronic and Algorithmic Trading. In: Electronic and Algorithmic Trading Technology. Academic Press, Boston, pp. 1–14

Klein C., Kundisch D. (2008) Index-Based Investment Vehicles – A Comparative Study for the German DAX. In: 32nd Annual Conference Advances in Data Analysis, Data Handling and Business Intelligence. Hamburg

Luthria H., Rabhi F. A. (2009) Service Oriented Computing in Practice – An Agenda for Research into the Factors in the Organizational Adoption of Service Oriented Architectures. In: Journal of Theoretical and Applied Electronic Commerce Research 4(1), pp. 39–56

Luthria H., Rabhi F. (2008) Organizational Constraints to Realizing Business Value from Service Oriented Architectures: An Empirical Study of Financial Service Institutions. In: Bouguettaya A., Krueger I., Margaria T. (eds.) Service-Oriented Computing – ICSOC 2008. Lecture Notes in Computer Science Vol. 5364. Springer, Berlin, pp. 256–270

Mangkorntong P., Rabhi F. A. (2007) A High-Level Approach for Defining & Composing Event Patterns and Its Application to E-Markets. In: Proceedings of 2nd International Workshop on Event-driven Architecture, Processing and Systems (EDA-PS’07). Vienna, Austria

Mangkorntong P., Rabhi F. A. (2009) A Domain-Driven Approach for Detecting Event Patterns in E-Markets. In: World Wide Web 12(1), pp. 69–86

Marjanovic O. (2005) Towards IS supported coordination in emergent business processes. In: Business Process Management Journal 11 (12), pp. 476–487

Milanovic N., Malek M. (2004) Current Solutions for Web Service Composition. In: IEEE Internet Computing 8(6), pp. 51–59

Öchsner T. (2008) Aktionärsschützer gegen Geschäftemacherei mit Zertifikaten. In: Süddeutsche Zeitung, 2008-02-09

Ogle D., Kreger H., Salahshour A. (2004) Canonical Situation Data Format: The Common Base Event V1.0.1. http://www.eclipse.org/tptp/platform/documents/resources/cbe101spec/CommonBaseEvent_SituationData_V1.0.1.pdf

Ozenbas D., Schwartz R. A., Wood R. A. (2002) Volatility in US and European Equity Markets: An Assessment of Market Quality. In: International Finance 5(3), pp. 437–61

Peuquet D. J., Duan N. (1995) An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. In: International Journal of Geographical Information Systems 9(1), pp. 7–24

Rabhi F. A., Rana O. F., Guabtni A., Benatallah B. (2009a) A User-Driven Environment for Financial Market Data Analysis. In: Aalst W., Mylopoulos J., Sadeh N. M., Shaw M. J., Szyperski C., Kundisch D., Veit D. J., Weitzel T., Weinhardt C. (eds.) Enterprise Applications and Services in the Finance Industry. Lecture Notes in Business Information Processing Vol. 23. Springer, Berlin, pp. 64–77

Rabhi F. A., Guabtni A., Yao L. (2009b) A data model for processing financial market and news data. In: International Journal of Electronic Finance 3(4), pp. 387–403

Saunders S., Ross M., Staples G., Wellington S. (2006) The software quality challenges of service oriented architectures in e-commerce. In: Software Quality Control 14(1), pp. 65–75

Sprott D. (2004) Service Oriented Architecture: An Introduction for Managers. http://www.eaktio.com/downloads/soa-cbdi-report-2004-july.pdf

Stoimenov P. A., Wilkens S. (2005) Are structured products ’fairly’ priced? An analysis of the German market for equity-linked instruments. In: Journal of Banking & Finance 29(12), pp. 2971–2993

The Securities Industry Research Centre of Asia-Pacific (SIRCA) (ed.) Reuters DataScope Tick History (RDTH) system. Last Access: 2009-08-05

Wasescha E. (2008) Die asiatischen Märkte ziehen den europäischen beim Volumen davon. In: Neue Züricher Zeitung, 2008-08-26

Weber B. W. (2006) Adoption of electronic trading at the international securities exchange. In: Decision Support Systems 41(4), pp. 728–746

Wilkens S., Stoimenov P. A. (2007) The pricing of leverage products: An empirical investigation of the German market for ‘long’ and ‘short’ stock index certificates. In: Journal of Banking & Finance 31(3), pp. 735–750

Wilkens S., Erner C., Röder K. (2003) The Pricing of Structured Products in Germany. In: The Journal of Derivatives 11(1), pp. 55–69

Dennis Kundisch

Information Management & E-Finance
Faculty of Business Administration
and Economics
University of Paderborn
Warburger Str. 100
33098 Paderborn
Germany
dennis.kundisch@wiwi.uni-paderborn.de

Adnene Guabtni, Fethi A. Rabhi

School of Computer Science and Engineering
University of New South Wales
Sydney 2052
Australia
{adnene.guabtni | f.rabhi}@unsw.edu.au