Sense & Response Service Architecture (SARESA): An Approach towards a Real-time Business Intelligence Solution and its use for a Fraud Detection Application

Tho Manh Nguyen
Institute for Software Technology and Interactive System
Vienna University of Technology
Favoritenstrasse 9-11/188 A1040
tho@ifs.tuwien.ac.at

Josef Schiefer
Institute for Software Technology and Interactive System
Vienna University of Technology
Favoritenstrasse 9-11/188 A1040
js@ifs.tuwien.ac.at

A Min Tjoa
Institute for Software Technology and Interactive System
Vienna University of Technology
Favoritenstrasse 9-11/188 A1040
amin@ifs.tuwien.ac.at

ABSTRACT
The dynamic business environment of many organizations require massive monitoring of their processes in real-time in order to proactively respond to exceptional situations and to take advantage of time-sensitive business opportunities. The ability to sense and interpret events about a changing business environment requires an event-driven IT infrastructure for pwerforming fast and well-informed decisions and putting them into action. However, traditional Business Intelligence (BI) and Data Warehousing technologies do not directly address time sensitive monitoring and analytical requirements. We introduce an enhanced BI architecture that covers the complete process to sense, interpret, predict, automate and respond to business environments and thereby aims to decrease the reaction time needed for business decisions. We propose an event-driven IT infrastructure to operate BI applications which enable real-time analytics across corporate business processes, notifies the business of actionable recommendations or automatically triggers business operations, and effectively closing the gap between Business Intelligence systems and business processes. A scenario from the area of mobile phone fraud detection was chosen for building a prototype that illustrates the proposed approach by using current available IT technologies.

Categories and Subject Descriptors
H.2.7 [**Database Management**]: Database Administration– *data warehouse and repository.*

General Terms
Management, Measurement, Design, Reliability.

Keywords
Real-time Business Intelligence, Event Sense & Response. Data Analysis, Real-time Data Warehousing and OLAP.

DOLAP'05, November 4–5, 2005, Bremen, Germany.

1. INTRODUCTION
Advances of modern technologies in various domains has accelerated the intensity of competition, increased the volume of data/information available, and shortened decision-making cycles considerably. More and more information is managed and stored electronically, and the data is getting increasingly complex in both structure and semantics. Consequently, strategic decision makers are being exposed to the huge inflows of data and information from their various resources and they are under rigid time constraints to make the right decisions.

The requirements of monitoring and reacting to enterprise- or organization-level events which cannot be detected by a single operational system become more essential in the competitive business market. Monitoring vast amounts of data is more than just providing real-time alerts. It is about saying "There is something going on here and we are continuously receiving more data. Here is all the information available at this moment that will help you to find the needle in the haystack and to make a well informed timely decision". Companies need to track business processes, such as order processing, quality assurance, inventory, logistics, compliance, etc. in near real-time, to improve operational efficiency as business events are happening. They are also looking for answers to questions such as: What characteristics do our most profitable customers share, and how can we serve them better? Executives are more and more challenged to make quick, well informed decisions that address growing business issues and regulatory standards. To answer these questions, companies need a window into the current fitness of the organization and the tools to act.

However, with traditional Data Warehouse and Business Intelligence techniques, still a gap exists between the time the operational data is created and the time the analysis information is becoming available. Based on the traditional assumption that business decisions do not require most actual information but rich amount of historical data for tactical decision making, the Data Warehouse refresh process is typically performed in large batches (e.g. once a week or over-night) thus potentially causing a considerable delay when the information from operational systems is reflected in the Data Warehouse. Out-dated warehouse information is therefore not appropriate for applications which require analysis at the speed of the business. Applications in the finance or telecommunication sector often requires real-time

analytical functionalities to detect suspicious activities in customer accounts in a timely manner before it results in financial damages. As a result, deployment of business intelligence solutions for both operational and tactical decision-making is becoming an increasingly significant use of information assets.

In this paper, we propose an event-driven Business Intelligence solution that integrates a real-time closed loop decision-making (Sense & Respond loop). This loop is a dynamic discovery process which continuously 1) observes and collects events from a business environment, 2) converts the event data into meaningful business information, 3) discovers and analyzes business situations and exceptions, 4) automatically selects the most appropriate actions for a response to the business environment, and finally 5) executes the business actions based on the decision that has been made. Based on these characteristics, we develop a Sense & Response Service Architecture (SARESA) as an infrastructure for building real-time business applications. To illustrate real-time sensing of events, real-time data analysis and instant proactive response a Fraud Detection application prototype implementation using existing OLAP and Data Warehousing technologies is described.

The remainder of the paper is organized as follow: Section 2 reviews the backgrounds and related work. In Section 3, we introduce the Real-time Business Intelligence requirements and discuss the state of the aert Real-time Business Intelligence architecture. This architecture includes also the Real-time Data Warehouse component. Section 4 introduces a novel architecture for an enhanced Real-time Business Intelligence called SARESA (Sense And REspond Service Architecture) which uses a Sense & Response approach for overcoming the real-time business analysis requirements. The prototype implementation of SARESA architecture applied in the Fraud Detection application is described in Section 5. The paper concludes with an outlook for future research (Section 6).

2. RELATED WORK

Since the Gartner group coined the term “Zero Latency” in 1998 [13], a lot of related terms such as Zero-Latency Enterprise [11, 18], Active Warehousing [5], Real-time Analytics [25], Real-time Warehousing [15, 21, 27, 32], Real-time Decision Support [24] have appeared in literature.

The basic idea behind a zero-latency enterprise strategy is to speed up the flow of information and business processes to achieve a competitive advantage. This implies that all parts of the enterprise can respond to events as soon as they become known to any part of the enterprise. The HP ZLE framework [18] is a data-oriented architecture that concentrates on a real-time ZLE data store as its new construct forming a “hot cache” of data from across the enterprise.

Active Data Warehousing [28] combines active mechanisms based on ECA (event-condition-action) rules which are known from active databases with the integrated analysis capabilities of Data Warehouse environments to extend (passive) systems with reactive capabilities. An active Data Warehouse (ADWH) is event-driven, reacts in a time frame appropriate to business needs, and will make tactical decisions or cause operational actions rather than wait to produce periodic reports [5]. Therefore, the design of an ADWH has to consider technical aspects : scalability, high availability, frequent (i.e. just-in-time or continuously) data loading, mixed workload, etc. as well as the integration of active mechanisms which deal with the two sorts of propagation delays in Data Warehouse environments: a) delays in capturing real-world events by the operational systems, and b) delays in loading and integrating data into the Data Warehouse.

In [15, 27,29] Real-time Data Warehousing (RTDW) is introduced referring to the technical aspects that timely perform automatic updates in a Data Warehouse . A strict definition of real-time implies that any data change occurring in a source system is automatically and instantaneously reflected in the Data Warehouse. Ideally this implies that all changes in the Data Warehousing environment take place simultaneously with the change in the source system. RTDW concepts include physical modifications to the database schema and the database environment, movement of data across the enterprise, ETL processes, modification of downstream processes, alerts, creation of extracts, cubes and data marts, and the whole new methodology for designing and implementing RTDWs.

Real-time analytics [25] encompasses the ability to use all available resources in an organization to improve the operations and quality of service, at the moment they are neededr. If a piece of information is created or modified in an operational system, it is sensed and acted upon by an analytical process. Real-time analytics complement real-time operational systems. Agile organizations will need to measure, evaluate and react to events with a closed-loop of telemetry-like information, rules, decisions and triggers, all in real-time.

So far, most of existing data analytical solutions operates only on traditional (stored) data sets. However, in recent applications, data more and more originates from data streams [1, 31] rather than from finite stored datasets. This especially applies for manufacturing processes, click-streams, and call detail records in telecommunication [10, 26].

So far, research results have been reported for modeling and handling data streams including algorithms for data stream processing to fully-fledged data stream systems. In continuous query processing, several approximation methods are used for data reduction and synopsis construction such as sketches [12], random sampling, histograms [22], and wavelets [8]. Some other approximate methods are applied to tackle the blocking operator such as Sliding Window [9], load shedding [2], punctuation [30]. k-Constraints [3] are used in clustering and monitoring data stream. Other research topics cover data stream management system models, architectures and related issues e.g. memory minimization, operator scheduling, query opti-mization, multiple query, distributed query processing etc. [1, 31]. Conventional OLAP and data mining models have been extended to tackle data streams, such as multi-dimensional analysis [16], frequent item sets and association rules [17], clustering [14] and classification, decision tress [19]. However, it must be stressed that most of previous approaches on data stream processing focus on approximate methods based on statistical estimations due to the limited storage and computing resource

Recently researchers have tried to build an entire data stream management system (DSMS; instead of DBMS - database management system) from scratch [1, 4]. An alternative approach can be imagined in the modification of existing DBMS by extending the envisioned functionality.

3. REALTIME BUSINESS INTELLIGENCE

3.1 Real-time BI Requirement

Many essential operational decisions (e.g. promotion effectiveness, customer retention, key account information [20] need some actual yet integrated and subject-oriented data in or near real-time [33]. However, the direct real-time operational/tactical decision support is not achieved by traditional Business Intelligence Systems. These types of analytical applications are generally completely disconnected from operational IT systems. The decisions are executed by communicating them as a command or suggestion to humans, thus always cause latency. The real-time analysis requirements demand a set of service levels that go beyond what is covered by typical of a traditional Business Intelligence System:

- **Data freshness**: The need for data freshness escalates significantly, because particular data have to be updated more frequently in order to improve decision-making.support for various near real time requirements (high/low priority data).
- **Continuous data integration** [7] enables (near) real-time capturing and loading from different operational sources and event-based triggering of actions even during data integration. This sort of data integration results in an increasing number of late-arriving data (e.g. due to propagation delays), because timeliness becomes more important.
- Highly available analytical environments based on an **analysis engine** that can consistently generate and provide access to current business analyses at any time not restricted by loading windows typical of the common batch approach.
- Recommendations on discovered situations or exceptions: which require **Active decision engines** that can trigger (rule-driven) tactical decisions for “routine-type tasks” encountered in an analytical environment.
- Changes of a business process or settings in the operational environment must not disrupt the interoperability with the event stream processing. An **adaptive platform** for the **event stream processing** is required to deal with the changes of the operational environment.
- **High availability** and **scalability** are indispensable criteria. since the number of users and performance requirements for a real-time Data Warehouse will increase by orders of magnitude with the deployment of analytic applications enabling tactical decision support.

While a complete real-time enterprise Business Intelligence System might still constitute a future challenge, some yet feasible approaches may well enable Data Warehouses to react "just-in-time" [2] on changing customer needs, supply chain demands, and financial concerns. In our vision, the Real-time Business Intelligence steers towards the goal of timeliness to its logical extreme of **instantaneous awareness** and **appropriate response** to events captured in the business environment.

3.2 Real-time BI Architecture

To satisfy the requirements mentioned above, it is necessary to enable a complete business intelligence process to observe, understand, predict, react to, reorganize, automate and control the feedback loops in real-time. Therefore, the Real-time Business Intelligence in our vision must include *analytical services* which are continuously fed with data from the operational environment and can be directly invoked by other systems. The objective of analytical services is to handle real-time data and to cope with continuously updated data. The fresh data for the continuous analysis requests is provided by the Real-time Data Cache.

In the classical three-tiered Data Warehouse architecture [6], data from data sources is extracted, transformed and loaded into the Data Warehouse via ETL components (tier 1). The warehouse storage (tier 2) manages huge of detailed and pre-aggregated data which is available for complex multi-dimensional analytical query from the OLAP server and other reporting tools (tier 3). In this architecture, the ETL technology is designed for batch updates while the warehouse system is offline, and hence is not suitable for real-time processing. Therefore, besides the traditional ETL components, specialized real-time ETL components are necessary in tier 1.

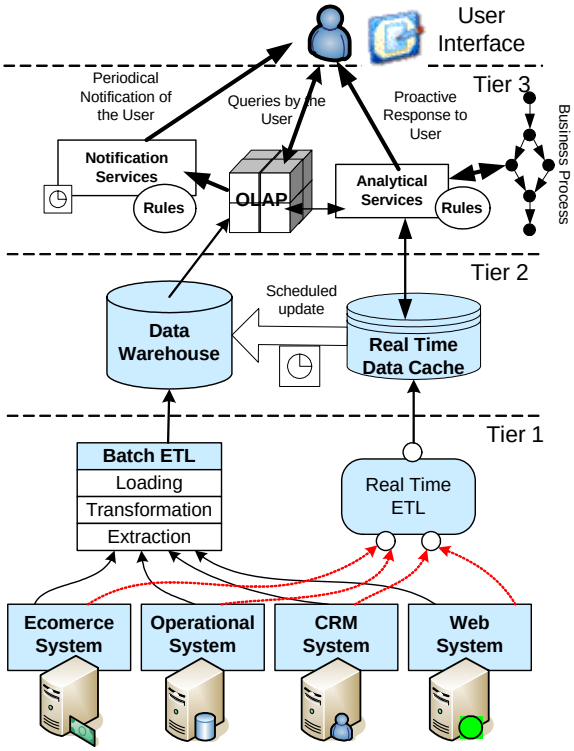

Figure 1: Traditional Real-time BI Architecture.

The continuous updates to a Data Warehouse would reveal new problems such as interference with complex analytical queries, materialized aggregates, sophisticated index structures, views, and the costly maintenance of cubes. In tier 2, a traditional Data Warehouse storage system needs to be extended by a Real-time Data Cache which serves as a staging area for managing real-time updates and periodically batcheupdates to the Data Warehouse.

In tier 3, the analytical services retrieve their data from the Real-time Data Cache as well as from the OLAP Cube (which built upon the Data Warehouse) as soon as analytical requests are required by a business process. The analytical services continuously analyze the data patterns and discovery of situations and exceptions. The rule engine assists the services to recognize

certain situations and exceptions as well as generates an appropriate response. Therefore, by continuously observing and analyzing data, the analytical services can propose proactive responses to optimize a business process and adapt the business environment. The notification service also analyzes data and sends relevant notifications to the user in the periodical manner.

One major problem of existing BI architectures as shown in Figure 1 is the lack of integration between OLAP systems and the operational business environment. As shown in Figure 1 in traditional BI architectures data for OLAP purposes are periodically refreshed. In the following section, we propose an architecture that uses OLAP as analytical service with current business data and that allows business processes to take full advantage of OLAP capabilities during the process execution.

4. SENSE & RESPONSE SERVICE ARCHITECTURE (SARESA)

The main objective of SARESA is to continuously receives, processes and augments events from various source systems, and transforms in near real-time these events into performance indicators and intelligent business actions. It automatically discovers and analyses business situations or exceptions and can create reactive and proactive responses such as generating early warnings, preventing damage, loss or extensive cost, exploiting time-critical business opportunities, or adapting business systems with minimal latency.

4.1 Sense & Response Loops

The data processing in SARESA is controlled by "Sense & Response loops" which can be divided into 5 phases. Table 1 describes the function of each phase in detail.

Table 1. Phases of Sense & Respond Loops

Phase	Designation	Function in SARESA
Sense	Which is the current state of the business environment?	Events are continuously captured and transmitted to SARESA where they are initially unified before the actual data processing begins.
Interpret	What do the captured data indicate? What do the data mean for the current situation of the organisation?	Transformation of the captured events (raw data) into business information, such as key performance indicators, business situations and exceptions.
Analyse	Which business opportunities and risks can arise? What are the possibilities to improve the current situation of the organisation?	Analysis of key performance indicators and determination of root causes for business situations and exceptions. Prediction of the performance and assessment of the risks for changing the business environment.
Decide	Which strategy is the best to improve the current situation of the organisation? What are the actions required to successfully put a decision into action?	SARESA proposes the best option for improving the current business situations and determines the most appropriate action for a response to the business environment. This step can be automated with rules or by involving persons.
Respond	Who has to implement the decision? How can the decision be put into action?	Response to business environment by communicating the decision as a command or suggestion (e.g. by e-mail) or by directly adapting and reconfiguring business processes and IT systems.

4.2 SARESA Architecture

In the recent years, Service Oriented Architecture (SOA) [23, 24] has gained popularity as new software engineering paradigm. It arose from the necessity of creating components providing clearly defined small pieces of functionality that later on can be assembled into complex applications. Using the SOA approach, we model Sense & Response Service Architecture (SARESA) as a pool of services (system services and Sense & Respond services) and establish the infrastructure that enables a robust communication and interaction between them (Figure 2).

The underlying infrastructure offers many system services, which can be universally used by the Sense & Respond services. The system services fulfill basic tasks such as event correlation, event synchronization, logging, thread management, exception handling and centralized configuration management. The Event Service Bus provides the core infrastructure that enables a robust and flexible communication between Sense & Respond services.

Each phase in the Sense & Respond loop introduced in the previous section is supported by special Sense & Respond services which can flexibly interact with the Event Service Bus. For instance, the Event Transformation services include Event Adapters to receive events from a business environment in order to transform them into a standardized format (→ Sense phase).

Additionally, they include components to manage metrics such as the calculation of performance indicators (→ Interpret phase). The remaining Sense & Respond services correspond to the phases of Sense & Respond loops. With SARESA it is possible to include user defined services for various tasks such as discovering situations, a third-party analysis tool as an analytical service or an external rule engine for making automated decisions in Sense & Respond loops.

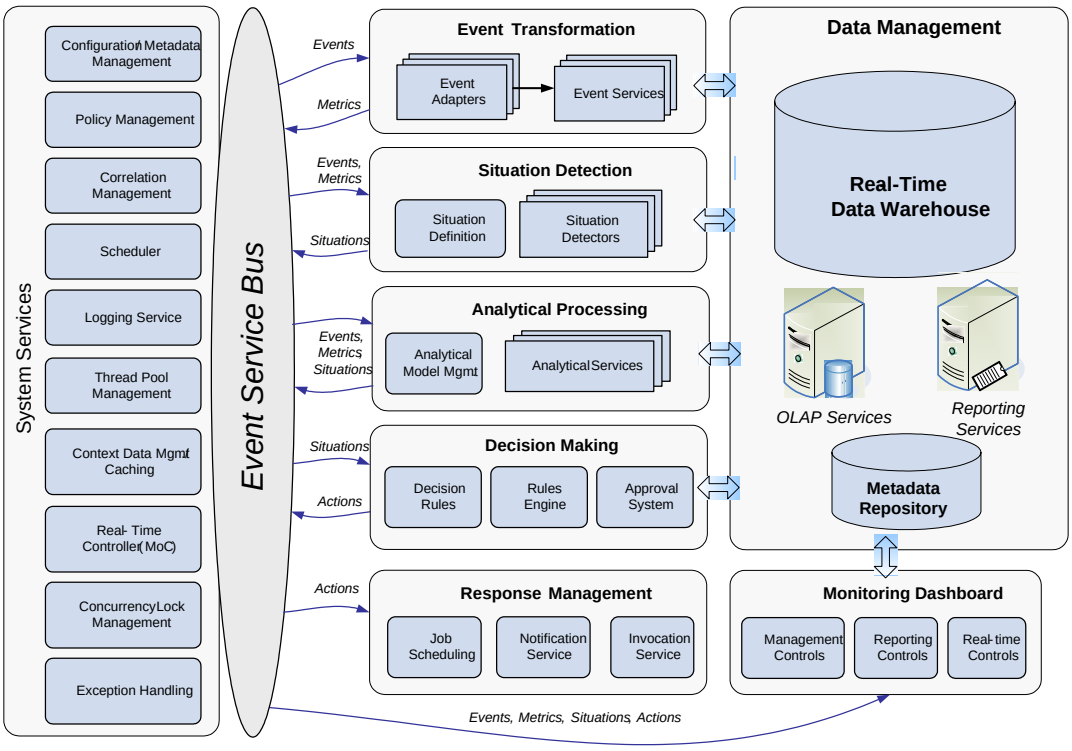

Figure 2: SARESA Architecture

For data management, we distinguish different types of data: historical data, real-time data and metadata. The real-time Data Warehouse provides a single view of historical data and real-time data. For multidimensional data analysis, SARESA obviously supports services for Online Analytical Processing (OLAP) and reporting. All metadata of the SARESA system is stored in a separate metadata repository.

Finally, SARESA includes a monitoring dashboard which provides a user interface for the administrator. It gives an overview of the current status of the event processing during the execution of Sense & Respond loops. With the monitoring dashboard, the administrator can easily recognize overloading situations in order to reconfigure the system. If failures arise during event processing, the administrator can intervene immediately and fix the problem.

4.3 Event Processing Model (EPM)

The processing steps, their relationships to each other, as well as the parameters of the analysis and data transformation processes can be individually defined for every organisation. SARESA uses an event processing model (EPM) for modelling Sense & Respond loops. Similar to a construction kit, the EPM offers various building blocks for Sense & Respond services which can be used to construct a Sense & Respond loop. Dependent on the requirements and the business problem, these building blocks can be flexibly conjoined or disconnected. Links between the building blocks represent a flow of events from one service to the next. The EPM provides:

- Definition of the structure for the processed events and data
- Interfaces to external systems for receiving data (Sense) and also for responding by executing business transactions (Respond)
- Data transformations, data analysis and persistence
- Definition of situations and exceptions in data to which a response should be triggered
- Modelling the data and control flows for the Sense & Respond loop (e.g. calculation of a metric always occurs before data analysis)
- Declaration of Sense & Respond services for processing steps including their input and output parameters

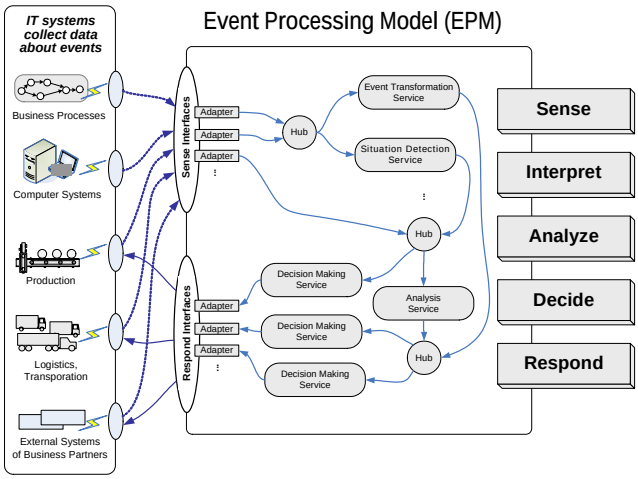

Figure 3: Event Processing Model

- Definition of the relationships and dependencies of Sense & Respond services and event data (e.g. data that has to be correlated before the processing starts)

Figure 3 shows the EPM of SARESA. Data is collected and received from source systems and continuously processed. The EPM shows the event-flows of Sense & Respond processes and visualizes complex processing steps. For Sense & Respond loops it is vital that the processing steps work seamlessly together in such a way as to enable a continuous and efficient execution of all processing phases; the EPM offers this capability. The EPM provides the following capabilities:

- **Flexible control of Sense & Respond loops**: Complete Sense & Respond processes can be modeled and user friendly visualized.
- **Capturing and integration** of events from various source systems.
- **Adaptiveness**: The processing steps of a Sense & Respond loop can be flexibly changed and updated.
- **Business Intelligence**: The processing steps can be combined with data analysis.
- **Real-time processing**: All processing steps are executed straight through without any delays.

5. REAL-TIME BI IMPLEMENTATION

5.1 Mobile Fraud Detection Scenario

A Mobile Fraud Detection scenario is chosen to illustrate the real-time event sense and response requirement. In this scenario, a timely analysis and response to prevent fraudulent activities is required. The most prominent fraud detection methods are based on the analysis of the usage patterns of mobile users. Call Detail Records (CDRs) are gathered as events and analyzed in order to generate business data such as calling time, geographic position of the mobile devices, call duration, and call frequency to recognize individual caller patterns of normal or fraudulent behaviour. In our application, it is a requirement that no CDR is lost and a fraud should be countered as soon as it is detected.

Fraud is detected by checking some pre-defined rules. The mining approach to generate these rules is considered in a working paper [35]. The rules can be of a complex nature such as “an international mobile call from Austria to China of a certain customer lasts over 30 minutes will not be considered as a fraud if its duration is not over 1.5 times of his/her average call duration from Europe to Asia within the last 3 months, otherwise, it will be considered as a fraud and should be stopped immediately when it reaches such a threshold”. The rules use aggregates which are provided and managed by an OLAP server (e.g. average call duration from Europe to Asia within the last 3 months for a certain user).

Figure 4 illustrate the Normal Call and Fraud Call from Austria to China of a certain customer. A phone call starts with the PhoneCallStarted event and ends with PhoneCallHungUp event. In the case of Normal Call, the HungUp occurs 8 minutes after the Start event happens. Because the fraud threshold (which is calculated by the Analysis Service based on the historical CDR data of this customer) is 15 minutes, the 8 minute call is considered as a normal call. However, a Fraud Call which last longer than 15 minutes will be stopped immediately by the system when it reaches the threshold. The PhoneCallHungUp in this case will never occur.

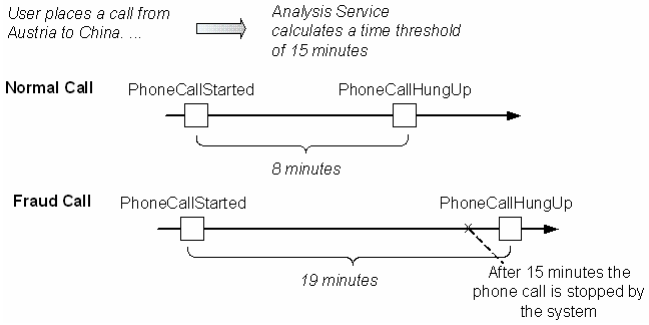

Figure 4: A fraudulent phone call from Austria to China scenario

5.2 SARESA System Deployment

We have implemented the SARESA prototype and have used it to discover the fraud patterns for mobile phone calls. The prototype is implemented using Visual Studio.Net 2005 beta 2 and MS SQL Server 2005 beta 3.

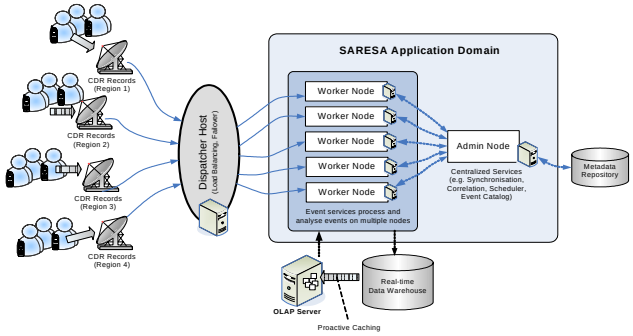

Figure 5: SARESA System Deployment

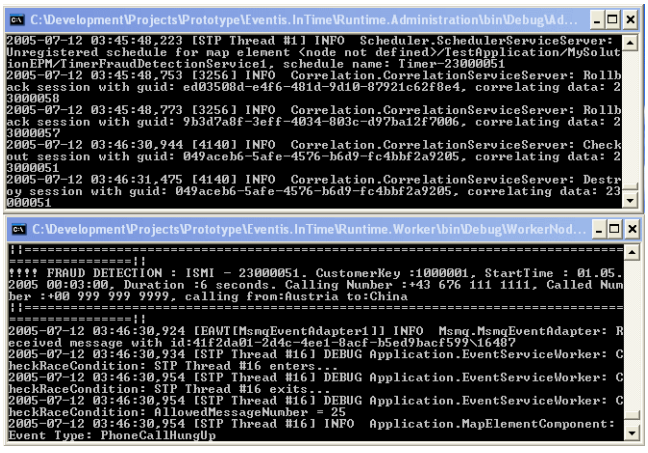

Figure 6. Fraud Detection Service running at the Worker and Admin Node

Events (CDR records) from heterogeneous sources are collected, normalized and dispatched to multiple process nodes (Worker

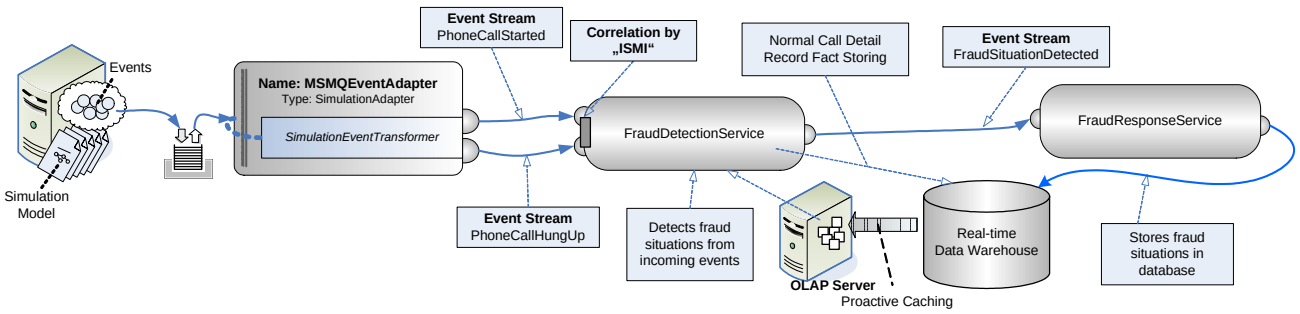

Figure 7. Fraud Detection Event Processing Model (EPM)

Nodes) by a centralized service (dispatcher host). On each worker node, an instance of a SARESA application is running to process and to manage the Sense & Response loops. Some universal services such as synchronization and event correlation must be provided by a central coordination node (Admin Node) which is called by worker nodes.

A central server (Dispatcher Host) receives the events from various event sources and distributes them to computer nodes (Worker Nodes). The dispatcher host thus controls the load balancing among the worker nodes and assures the failover and recovery in case a worker node fails. Each worker node is hosting an application instance which processes events of Sense & Response loops. A central admin node hosts centralized system services such as event correlation and synchronization (Figure 5).

Figure 6 shows the console of the worker and the admin node. The admin console shows the correlation session management and timer management activities since these services are executed centrally on the admin node. The worker console shows the results of the event processing such as the detection of a fraud.

The Event Process Model (EPM) which controls the event process order in Sense & Response loops is described in an XML document. It includes all application settings and parameters that are required for the event stream processing. In the EPM, event processing components are used as elements of a map which can be linked together. First, all elements of a map must be listed. This is necessary since an application component could be potentially used multiple times in a single map or in multiple maps. Each map element has its own name and can have its own set of parameters. The following example shows how to define an MSMQ EventAdapterComponent as map element. The MSMQ EventAdapter element has some parameters for indicating the queue from which messages (=events) should be received.

```
<EventProcessingMap name="MySolutionEPM">
<EventAdapterMapElement   name="MsmqEventAdapter1"
componentName="MsmqEventAdapter">
        <InitParameters>
        <Parameter name="QueueName">
                <Value>.\Private$\TestQueue</Value>
        </Parameter>
        <Parameter name="Mode">
                <Value>Transactional</Value>
        </Parameter>
        </InitParameters>
</EventAdapterMapElement>
```

Figure 7 shows a graphical representation of the EPM for the fraud detection. The current prototype supports EPM as an XML-file which must be conform to epm.xsd schema. However, the visual graphical EPM tool which allow user to browse and select components for mapping is taken into consideration in our ongoing work. Further information about this EPM configuration could be found at our technical report [36].

In the prototype, we use a simulator (Figure 8) to generate sample phone call events which are published to a message queue. An event adapter in the EPM receives the sample phone call events from that message queue. In the EPM the event adapter has multiple connections with the FraudDetectionService which represents event flows.

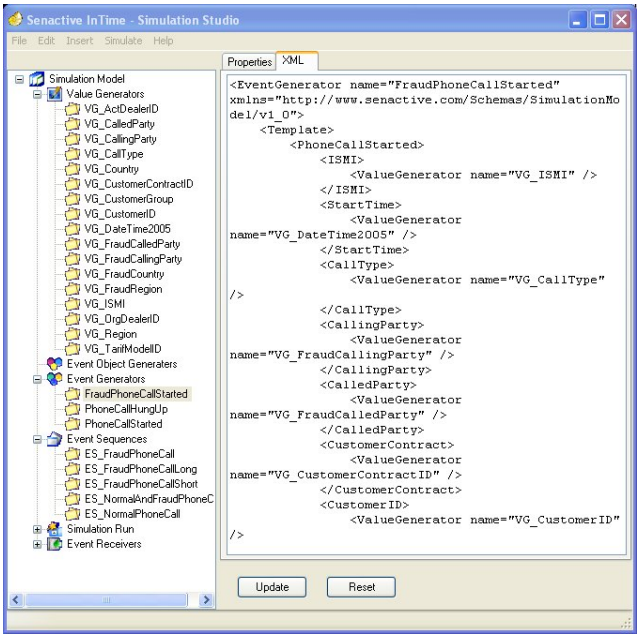

Figure 6.11. Simulation Studio

The FraudDetectionService is a service to analyze the phone call and to decide whether it is fraudulent or not. The fraud detection policy and its logical process are conducted by this service. It receives PhoneCallStarted and PhoneCallHungUp events, conducts the multi-dimensional analysis, decides whether a phone call is a fraudulent call and raises in a fraud case the FraudSituationDetected event. Correlation sessions are used for correlating event pairs [34] of PhoneCallStarted and PhoneCallHungUp to set a time threshold which indicates a fraud

case (e.g. 30 minutes). If a PhoneCallHungUp event is not received within the time threshold, a FraudSituationDetected event is raised. A time threshold is determined based on a multi-dimensional analysis of historical records of the user. In a fraud case, the FraudResponseService receives the FraudDetection Situation event and issue the relevant response such as sending alarms to the customer, or stopping the phone call.

5.3 Real-time Analysis Service

The current version of SQL Server 2005 Beta 3 supports Real-time OLAP. For SARESA, we use these real-time analysis capabilities and deploy and manage the OLAP cubes with the MS Analysis Service which is part of SQL Server 2005 (see Figure 8). The real-time data cache and refresh of the OLAP cube is automatically managed by MS SQL Server 2005. The appendix shows the cube structure and the cube browser of the FraudDetectionDW. The analytical service for the fraud detection is implemented as an event service which is running at the worker nodes. For each event PhoneCallStarted, the analytical service executes MDX queries and calculates the time threshold for a potential fraud case, i.e. 1.5 * average phone call from caller's continent to receiver's continent in the last 3 months. This time threshold is used to set a timer to prevent fraudulent phone call. Figure 9 shows the MDX query in SQL Server Management Studio to calculate the time threshold.

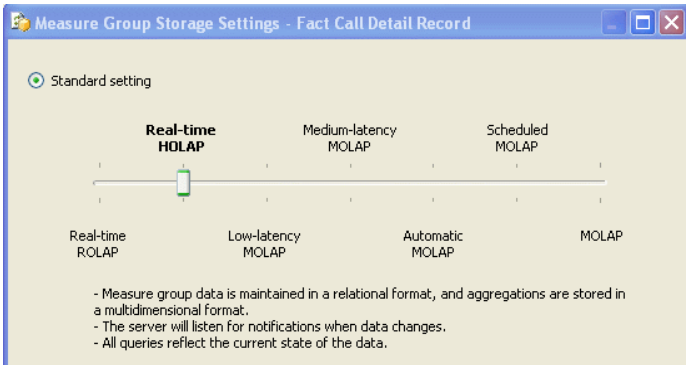

Figure 8: Real-time HOLAP Cube Structure in MS Analysis Server

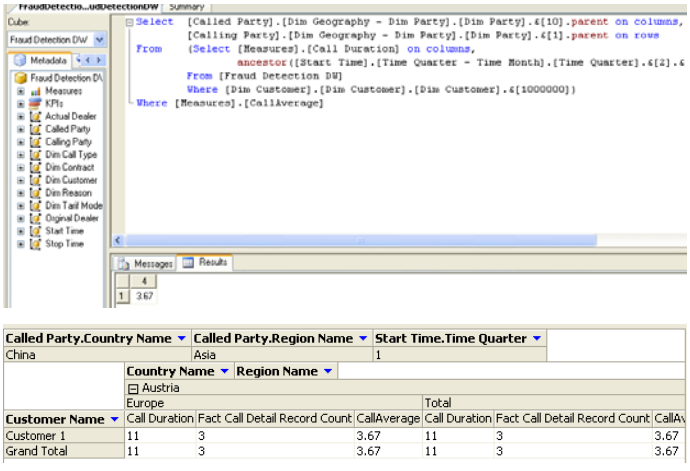

Figure 9: MDX query and on FraudDetectionDW cube

6. CONCLUSION

In this paper we presented a real-time Business Intelligence architecture called SARESA with the aim of providing continuous, real-time analytics in order to enable proactive responses to a business environment for effectively managing and controlling time-sensitive business processes. We introduced Sense & Respond loops and a service-oriented architecture that is able to detect situations and exceptions, perform complex analytical tasks and reflect on the gap between current situations and desired management goals. Traditional Business Intelligence architectures lack in the support of real-time BI and closed-loop decision making. A major goal of SARESA is to tightly integrate Business Intelligence with business processes by monitoring various IT system or other observables and generating reactive and proactive responses such as generating early warnings, preventing damage, loss or extensive cost, exploiting time-critical business opportunities, or adapting business systems with minimal latency. We presented our mobile phone fraud detection prototype implementation on Visual Studio.Net 2005 and MS SQL Server 2005. The work presented in this paper is part of a larger, long-term research effort aiming to develop a service-oriented Business Intelligence platform for supporting time-sensitive business processes. Future research and development efforts will focus on enhancing the SARESA system with advanced analysis and decision-making capabilities such as data mining on event streams and tightly integrating OLAP with rule engines.

7. ACKNOWLEDGMENTS

The first author has been supported by a Technology Grant South East Asia (No. 322/1/EZA/2002) of the Austrian Council Research and Technology and the ASEA-UNINET.

8. REFERENCES

[1] BABCOCK, B.; BABU, S.; DATAR, M.; MOTWANI, R; WIDOM, J., Models and Issues in Data Stream Systems, Proc. of the 2002 ACM Symp. on Principles of Database Systems, June 2002.

[2] BABCOCK, B; DATAR, M; MOTWANI, R, Load Shedding for Aggregation Queries over Data Streams, Proc. of Intl. Conf. on Data Engineering (ICDE 2004), 2004.

[3] BABCOCK, B; OLSTON, C, Distributed Top-K Monitoring, Proc. of the ACM Intl. Conf. on Management of Data (SIGMOD 2003), June 2003.

[4] BABU, S; WIDOM, J, Continuous Queries over Data Streams, ACM SIGMOD Record, Vol. 30(3), Sept. 2001.

[5] BROBST, S; BALLINGER, C, Active Data Warehousing, Whitepaper EB–1327, NCR Corporation, 2000.

[6] CHAUDHURI, S.; DAYAL, U.: An overview of Data Warehousing and olap technology, SIGMOD Record, 26(1):65–74, 1997.

[7] BRUCKNER, R; LIST, B; SCHIEFER, J, Striving Towards Near Real-time Data Integration for Data

Warehouses..Proc. of the 4th Intl. Conf. on Data Warehousing and Knowledge Discovery (DaWaK 2002), Springer LNCS 2454, pp. 317–326, Aix-en-Provence, France, Sept. 2002.

[8] CHAKRABARTI, K et al, Approximate query processing using wavelets, The VLDB Journal vol. 10, 2001

[9] CHANDRASEKARAN, S.; FRANKLIN, M., Streaming queries over streaming data, Proc. 28th Intl. Conf. on Very Large Data Bases, Aug. 2002.

[10] CHEN, O; HSU, M; DAYAL, U, A Data-Warehouse / OLAP Framework for Scalable Telecommunication Tandem Traffic Analysis. Proc. 16th Intl. Conf. on Data Engineering (ICDE), IEEE CS Press, San Diego, CA, Mar. 2000.

[11] COMPAQ CORP, Compaq Global Services - Zero Latency Enterprise, http://clac.compaq.com/globalservices/zle/

[12] DOBRA, A et al, Processing complex aggregate queries over data streams, Proc. of the 2002 ACM SIGMOD Intl. Conf. on Management of Data, 2002.

[13] GARTNER GROUP, Introducing the Zero-Latency Enterprise, Research Note COM-04-3770, June 1998.

[14] GUHA, S; MISHRA, N; MOTWANI, R; O'CALLAGHAN, L, Clustering Data Streams, Proc. of the 41st IEEE Annual Symposium On Foundations of Computer Science, pp. 359–366, Redondo Beach, CA, Nov. 2000.

[15] HAISTEN , M, Real-time Data Warehousing Defined, Library article from BetterManagement.com, 2002

[16] HAN J. et al, Multi Dimensional Regression Analysis of Time-Series Data Streams, Proc. of the 28th VLDB Conf. Hong Kong, 2002.

[17] HIDBER, C, Online Association Rule Mining, Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, pp. 145–156, Philadelphia, PA, June 1999.

[18] HEWLETT-PACKARD, Zero latency enterprise architecture, White paper, June 2002.

[19] HULTEN, G.; SPENCER, L.; DOMINGOS P., Mining Time-changing Data Streams, Proc. of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD01), CA, 2001.

[20] INMON, W., Building the Operational Data Store, 2nd edition, Wiley: New York et al. 1999.

[21] LANGSETH, J, Real-time Data Warehousing: Challenges and Solutions, Article published at DSSResources.COM, 02/08/2004

[22] MUTHUKRISHNAN, S ;STRAUSS, M, Maintenance of Multidimensional Histograms The 23rd Conference FSTTCS 2003, India, 2003.

[23] NATIS, Y, Service-Oriented Architecture Scenario, Gartner Research, ID ID Number: AV-19-6751, 16 April 2003

[24] PALLOS, M, Service-Oriented Architecture: A Primer , eAI Journal , December 2001

[25] RADEN, N, Exploring the Business Imperative of Real-time Analytics, Teradata white paper, October 2003.

[26] STREAMBASE, StreamBase Systems (2005) The World's first Stream Processing Engine, http://www.streambase.com/

[27] TERR, S, Real-time Data Warehousing 101, Article published at DataWarehouse.com, March 29,2004

[28] THALHAMMER,T.; SCHREFL, M., Realizing active Data Warehouses with off-the-shelf database technology, Softw. Pract. Exper. 2002.

[29] THO, N.; TJOA A., Zero-Latency Data Warehousing: Continuous Data Integration and Assembling Active Rules, 5th Intl. Conf. on Information Integration and Web-based Applications and Services (IIWAS2003)

[30] TUCKER, P.; MAIER, D.; SHEARD, T., Applying Punctuation Schemes to Queries Over Continuous Data Streams, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, March 2003.

[31] WIDOM J. et al, Query processing, approximation, and resource management in a data stream management system, Proc. First Biennial Conf. on Innovative Data Systems Research (CIDR), Jan. 2003.

[32] WHITE, C, Intelligent Business Strategies: Real-time Data Warehousing Heats Up, DMReview Publication, August 2002

[33] SCHULTE, W., Application Integration Scenario: How the War is Being Won, in: Gartner Group (Ed.): Application Integration – Making E-Business Work, London, 6-7 September 2000.

[34] SCHIEFER, J., MCGREGOR, C., Correlating Events for Monitoring Business Processes, International Conference on Enterprise Information Systems, Porto, 2004.

[35] THO, M.N, SCHIEFER, J., TJOA M., ZELESSA (Zero-Latency Event Sensing and Responding): An Enabler for Real-time Business Intelligence, Technical Report 123/IFS/2005 (submitted to OeNB project proposal).

[36] THO, M.N, Zero-Latency Data Warehousing: Toward a Zero Latency Event Sensing and Responding Data Warehousing, PhD Thesis, Vienna University of Technology, August 2005.

Appendix

FraudDetectionDW cube structure and the cube browser