Decision Support Systems 52 (2012) 869–883

Contents lists available at SciVerse ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

# A platform for situational awareness in operational BI

Malu Castellanos *, Chetan Gupta, Song Wang, Umeshwar Dayal, Miguel Durazo

*HP Labs, United States*

ARTICLE INFO

Available online 29 November 2011

*Keywords:*
Information extraction
Correlation
Unstructured data
Streaming data
Real-time
Business intelligence
Situational awareness

ABSTRACT

Enterprises are being swamped with data, and much of it is unstructured in origin. As these data volumes for unstructured data increase, there is a need to extract more value from them. For the purpose of gaining business insight, besides traditional text mining, we need capabilities to correlate unstructured data emanating from different sources. An important instance of this is the capability to correlate streaming unstructured web data with internal document data in near real time, which can give enterprises significant tremendous competitive advantage by enabling them to be aware of external events that can affect their business operations. This situational awareness gives business managers the opportunity to make informed operational decisions before it is too late. SIE-OBI is a platform being developed at HP Labs that responds to this need. In this paper, we describe SIE-OBI and illustrate its use via an application that provides awareness of events described in news articles that could affect the contracts of an enterprise. We present the main components of the platform architecture and illustrate their functionality to our contractual situational awareness scenario.



## 1. Introduction

In the context of an enterprise, the capability to extract valuable information from all sorts of data (i.e., structured, semi-structured and unstructured) and act instantly, either reactively or proactively, provides a tremendous competitive advantage. In particular, identifying external events and information that may affect the enterprise operations, objectives or decisions is essential to an intelligent enterprise. Today when so much of this data is available at our fingertips, we fall rather short in our capability to consume it at the pace with which it is generated. This is particularly true of unstructured data. Enterprises today not only have mountains of internal semi-structured and unstructured data in the form of contracts, release documents, customer reviews, etc., but also endless external data from the Web in terms of news articles, product reviews, etc. Very often, the external data is available as soon as it is generated. External sources such as social media in the form of news feeds, tweets and blogs, among others, provide the opportunity to get informed instantly, but only if we can quickly figure out the relevance of the information contained in them to the enterprise. One way to figure out the relevance is to find correlations between the external and the internal data.

Typically, enterprise data warehouses (EDW) serve the vital function of being a single version of the truth for historical business reporting, but they often fall short when it comes to providing a real-time view for operational decision making. Moreover, in general, EDWs are fed from structured sources like relational databases and ignore unstructured sources like document repositories and social media feeds. As a consequence, the opportunity to gain competitive advantage from correlating unstructured historical data with the unstructured streaming data from the web in a timely manner is lost.

Large enterprises have thousands of customers and partners all over the world and a myriad of legacy contracts of a variety of types with them. Data buried in the legalese of contracts is not being used to make business decisions upon the occurrence of world events that may affect contractual relationships. Numerous examples of such events and their potential impact on business operations exist: political instability in a country (what contracts exist with suppliers based in this country?), significant fluctuations in currency values (what contracts are denominated in this currency?), changes in commercial law (how does the change affect the risk in each contract?), mergers and acquisitions (what contracts exist with the parties involved in the merger?), a natural disaster in a region (what contracts exist with providers in that region?). If business managers were aware of events like those mentioned above and the contractual relationships that they affect, they would have the opportunity to take immediate action. For example, in case of a typhoon in the Pacific region where a manufacturing company has its main suppliers, the ability of extracting and correlating this information from news feeds with the suppliers' contracts in near real time would alert the business managers of a situation that may affect the business operations that depend on those suppliers. It is easy to see the complexity and infeasibility of manually correlating news feeds with contracts. Not only is the volume of news articles massive (and the number of contracts might be quite large as well) but the stream arrival rate is too fast to cope with. Furthermore, it is practically impossible to keep track of the many details in the

* Corresponding author.
*E-mail addresses:* malu.castellanos@hp.com (M. Castellanos), chetan.gupta@hp.com (C. Gupta), songw@hp.com (S. Wang), umeshwar.dayal@hp.com (U. Dayal), miguel.durazo@hp.com (M. Durazo).

0167-9236/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2011.11.011

contracts to identify the news articles that might affect them, especially to do so for every contract and every news article. This is what we call contractual situational awareness [6].

As another example, for the same manufacturing company, it is important to know what customers are saying about its new products or its competitors' products so that it can intervene in blog discussions or provide immediate feedback to the development team to make its products more competitive. That is, once a product is announced in some news site, people start commenting on its features, on blogs, tweets, review sites, etc. Analyzing these comments gives valuable insight into the customers' experience with a particular product. This is what we call sentiment awareness, and it is a very important aspect of operational customer intelligence which nowadays is a key capability that companies must have to get insight into the voice of their customers, and adapt to their needs.

In our investigations and to our surprise, we found out that scenarios like the two above are a common reality in many different domains in industry. An important requirement of situational awareness applications is to provide timely insight into these situations so that appropriate action can be taken in real time. This motivated us to develop a novel platform called SIE-OBI (Streaming Information Extraction for Operational Business Intelligence) that integrates the required functionalities to exploit *relevant* fast streaming information from the web by correlating it with internal (historical or slow streaming) data sources to alert business managers of situations that can potentially affect their business. In this paper, without loss of generality of the functional principles of the platform, we elaborate on its application to situational awareness in the contracts scenario. This will help illustrate the platform's capabilities.

In a nutshell, SIE-OBI situational awareness applications use novel techniques to extract relevant information from two or more disparate sources of unstructured data, typically one internal slow text stream and one external fast text stream, and to determine which elements (i.e., documents) in one stream are correlated to which elements in the other stream. Notice that our goal in developing SIE-OBI is to create a framework to facilitate the development of situational awareness applications. SIE-OBI reduces the time and effort to build data flows that integrate structured and unstructured, slow and fast streams, and correlates and analyzes them in near-real-time. As part of this effort we are developing algorithms for different functions, including information extraction and analytics, to be wrapped as operators of a library used to build data flows. In addition, we are developing techniques to optimize these flows with respect to different quality metrics in addition to performance (we call this, QoX-based optimization) [7].

In this paper we first give an overview of the architecture of SIE-OBI in Section 2. Then, in Sections 3 and 4, we describe the functionality of some of its components for information extraction and hybrid query processing, respectively. In Section 5, we illustrate the use of the SIE-OBI platform for the contractual situational awareness application. In Section 6, we describe a prototype implementation of the application and show experimental results on real data. Finally in Section 7, we pinpoint some of the main differences with related work, and in Section 8, we conclude with our plans for ongoing and future work.

## 2. Architecture

Fig. 1 shows the proposed framework that is the basis of the SIE-OBI platform. It provides the application developer with a unified interface to build situational awareness applications, and a platform on which the applications execute. One of the design goals of SIE-OBI is to facilitate the declarative specification of the application logic as data flow graphs, and let the framework build the appropriate physical data flow (i.e., execution plan). In this way the application developer doesn't need to worry about optimizing data flows and is only required to specify the corresponding QoX objectives and tradeoffs [7] that he is willing to make (e.g., he may prefer to gain more accuracy even at the cost of lower performance). Then, the optimizer takes the QoX objectives into account when coming up with an execution plan (see Section 5). The SIE-OBI framework also provides runtime components to support adaptive optimization, runtime statistics gathering and monitoring, and an execution engine.

To create an application, which is the main objective of this framework, the developer needs to specify the following items through the application development interface:

1. *Data Sources*: Two or more data sources may be specified.
2. *Processing Logic*: The logical operations are expressed in the form of a data flow graph, which includes the extraction operations, and the queries that specify the streaming correlation and other processing required.
3. *QoX objectives and tradeoffs*: The QoX specification for each logical operation or a set of logical operations in the data flow can include accuracy, response time, throughput, freshness etc.

For the kind of situational-awareness applications described in this paper, where the data sources are textual, the main data flow containing the processing logic can be separated into the two stages shown in Fig. 2:

- Information extraction (IE) (processed by the information extraction component in Fig. 1)
- Hybrid query processing (processed by the hybrid SQL query processing component in Fig. 1).

To transform the structured or unstructured data sources into concept vectors required as input for the correlation, various information extraction methods are provided in a library. The library of operators is one of the main components of the SIE-OBI platform. It contains an extensible set of operators that are required for the specification of data flows. The application developer can specify which operators to use according to the nature of the application. The hybrid SQL query library also provides SQL-like operators over both streaming and static data. This includes the correlation operations, for which one implementation is the hierarchical neighborhood tree (HNT) based implementation. A metadata manager includes the schemas for the concepts that need to be extracted, and also the definition of the concept hierarchies that are used both by the extraction operations and the HNT based correlation operations.

From the application logic (i.e., logical data flow), the optimizer will construct a physical plan with optimizations such as: (1) selection of physical operators for the logical ones; (2) setting of QoX knobs in some physical operators; (3) data partitioning to enable parallel execution; (4) pipelining of data flow and assignment of operations to physical processing nodes; (5) runtime adaptation of the physical data flow with current statistics; (6) QoX-based execution control through scheduling. Once this optimized data flow has been obtained, the execution engine executes the flow.

Notice that there is a distinction between logical operators that are used to specify logical data flows and their corresponding physical operators (i.e., implementation methods) that the optimizer uses to translate the logical flows into physical ones. In general, there are one or more implementations for each logical operator and it is the task of the optimizer to choose which ones to use. For example, a logical hierarchical similarity-based join could be translated into a physical HNT-based approximate join, or an entity extraction logical operator could be mapped to a genetic-based entity recognizer implementation.

In this paper we do not describe the optimizer and scheduler or the run time monitor. Instead, we focus on the information extraction and hybrid SQL query processing components in Fig. 1 and in particular, on the operators available for the corresponding dataflow stages, shown in Fig. 2, of the kind of situational awareness applications addressed in this paper.

Next, we discuss the two different kinds of operators: operators for information extraction (Section 3) and operators for hybrid SQL