

# Time Series Prediction Based on Event Driven Business Process Management

Eva Zámečnicková<sup>1</sup>, Jitka Kreslíková<sup>2</sup>

*1 Ph.D. student at FIT BUT, Brno, Czech Republic*

*2 Associate professor at FIT BUT, Brno, Czech Republic*

## **Abstract:**

*In this paper we would like to focus on time series prediction for building trading systems based on unification of the seemingly antagonistic concepts of Complex Events Processing (CEP) and discrete, legacy business logic. In the realm of the high frequency market-making trading systems, the incoming stream of data from multiple asset sources and venues shall be processed primarily by a real-time CEP engine till the point when the statistical characteristics of the continuous results cross into the "fat-tail" distribution region, when a set of stringent risk-management business rules kick in.*

**Keywords:** Business Rules, Complex Event Processing, Event-Driven Architecture, Financial Markets, High Frequency Data, Prediction, Trading.

## I. INTRODUCTION

Financial markets generate high frequency data in large quantities. The smallest data unit used for analysis is called a *tick*. In most cases, the incoming data is inhomogeneous and contains outliers and noise. For processing of this data, complex-event processing (CEP) platform can be used. Today's systems are trying to achieve the ability for the timely reaction to the occurrence of real-world situations. In the system environment this property has become a fundamental requirement. This applies to many different applications including fully automated financial trading and time series prediction. There is no ideal CEP architecture - the optimal solution requires a number of technologies, e.g. distributed computing, service oriented architecture (SOA) and high speed processing.

In the following sections we will describe both approaches – complex-event processing as business process management (BPM) and we will point out their main advantages considering the properties which can be useful for high frequency data processing. Both platforms use events and rules for data processing. Although it's possible to only focus on CEP or, alternatively, only BPM for data processing, the better way is to work with "hybrid" architecture which uses the benefits of both platforms. The systems complement each other – CEP is better for preprocessing and correlation of high volume input data and on the other hand BPM can generate automated decisions based on incoming data.

The approach of using the advantages of discrete event processing and business process management is also mentioned in [1], where the designed systems are called "Event-Driven Business Process Management". The term "Event Driven Business Process Management" was first used in June 2003 by Bruce Silver. The term was understood as a synthesis of workflow and Enterprise Application Integration (EAI). A concept of event processing and real-time business activity management (BAM) was described as single event processing without knowing anything about CEP.

## II. COMPLEX-EVENT PROCESSING

Information in this section is based on [2], [3] and [4]. An *event* is a component of data and represents that something happened in the real world. It may signify a problem, an opportunity, a threshold, or a deviation. In business terms an event is composed of two elements – an event header and event body. The event header contains elements describing the event occurrence, such as the event specification ID, event type, event name, event timestamp, event occurrence number and event creator. The event body must fully describe what happened, so any interested party (subscriber to the event) can process the information without having to go back to the source system. A complex event is a composite event that is an abstraction of two or more events.

This data flows in streams. An example of event can be e.g. financial market event – a completed stock purchase (an aggregation of the events in a transaction to purchase the stock). Complex Event Processing (CEP) is a set of techniques and tools used for setting systems that react to events in real time. We find this property very useful for market data processing. By preprocessing of data we can detect a specific pattern which repeats itself in source data. Patterns are used to express situations that have to be detected. In a pattern, events are correlated with each other by means of utilizing operators. The moment we recognize a pattern, we can determine appropriate next action at the right moment in real time.

Algorithmic trading applies CEP by calculating complex algorithms that indicate when to sell or buy based on real-time processing. Market data (including the rise and fall of stock prices during the day) can be viewed as events. This data needs to be analyzed in real time in order to identify the trends in data and to react to them automatically. Financial-trading institutions could monitor worldwide stock, commodity, and other markets to recognize threats related to current holdings or potential new opportunities. For example, if a system detects a rise or fall in a stock price, it should be able to identify whether the change is a regular, periodic event or represents a fleeting market opportunity or risk [6]. Today a lot of firms participating in market trading are using event processing as their technological framework for algorithmic trading. CEP is useful in applications which deal with streaming event data and need low-latency, adaptive decisions in response to changing conditions reflected in those events. By using event pattern matching rules, CEP is able to infer causal connections between seemingly disparate events.

CEP software technology enables the complex events detection in real time. This makes the CEP challenging, because general software tools for event-analysis don't work in real time. Real-time analysis is becoming nearly a fundamental requirement – the systems process real-time data to make immediate decisions in diverse matters such as stock trading, fraud detection etc. A CEP system is a software infrastructure that typically consists of a layer of adapters that connect directly to data sources and forward the information into the analysis engine. A CEP server evaluates the business logic, then runs event-stream-analysis algorithms and delivers the results continuously to users [6]. Essentially, there are two types of CEP toolsets - stand-alone software engines or embedded CEP services. This technology allows applications to identify complex sequences of events, like specifying the order of events. These complex patterns of events can have temporal constraints (within specified short time interval) or spatial constraints (within certain given distance) or some other causal relationship. These constraints can be expressed by using Complex Event Processing Language (EPL). The data can be evaluated by using *event queries* which are evaluated continuously while the events happen. Among the application areas belong, for instance: business activity monitoring, sensor networks, market data analysis.

Data Stream Management (DSM) is something like event database. This system allows events to be stored or replayed in real time. In DSM system, the derived data can be also stored, so it's possible to use the data to simulate the behavior of new event scenario on historical data. Historically recurring event patterns can be identified in data streams. Event stream processing (ESP) refers to database techniques of processing streams that assume that events arrive in a specific order to the stream processing computing engine. ESP is considered a subset of CEP, which does not assume events arrive in a specific order.

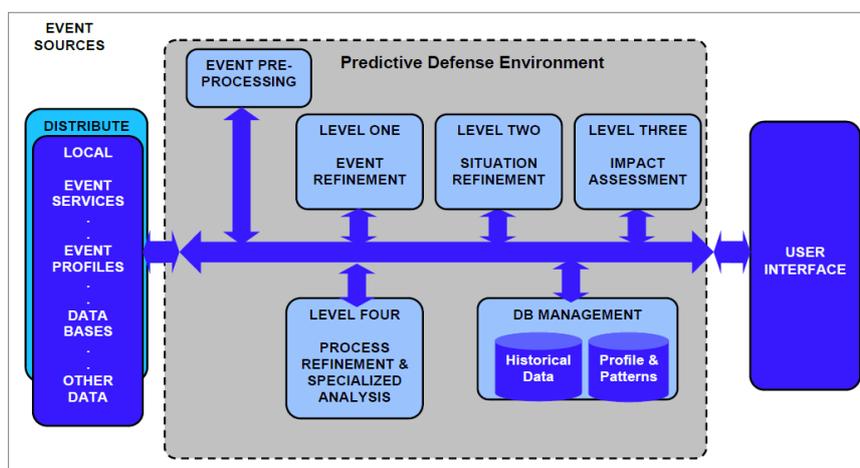


Figure 1 Complex Event Processing reference architecture - adapted from [5]

As Figure 1 shows, CEP is composed of several levels which conform to desired level of inference. At the lowest level, the event preprocessing runs – during this phase we clean the input data stream to produce some understandable data. On the next level, the events that were detected in input data are refined and subsequently initial decisions and correlations are done. The main challenge is to find relevant data. Then, situation refinement and impact assessment follows. At the level of impact assessment, we may predict the intentions of subject or to estimate potential losses or opportunities. At the end, the process refinement is done. All the results of event processing and operational visualization at all levels are summed up in a human readable format via user interface.

**2.1. Differences between traditional computing and event processing**

The main difference lies in the kind of data each of these two systems use. Traditional approach uses static data. Usually the data is stored in a database and a static data analysis is performed on it. So we use the data to query or to mine the information about the history.

On the other hand, event processing requires streaming data. With event processing, a business can identify patterns online and make decisions while the detected information is still relevant. The basic pattern-matching algorithm is implemented by way of looking for a potential sequence among the collected events. The algorithm then determines whether the pattern exists, what pattern it is and which events belong to it. An example might be an attempt to execute a credit card transaction from different places in a short time interval. Based on this data the decision to reject the next transaction might be made.

**2.2. Event-Driven Architecture**

Event-Driven Architecture (EDA) is an architecture devoted to designing and implementing systems that enable a business to respond to events in a real time. Because the events are complex, accordingly the systems have to be robust and scalable. This software infrastructure is very loosely coupled and highly distributed. The source of the event is only aware of the emergence of the event. The creator has no knowledge of the event’s subsequent processing, or the interested parties (event subscribers). The traceability of an event through a dynamic multipath event network can be difficult. EDA is best used for asynchronous flows of work and information. The main idea behind EDA is that a large software system consists of many small components that all have their own function. The communication between the components is done by using events. An event can be a notification, which tells other components that a certain action is done. Because events are very important within an EDA, the handling and routing of events is very important, too. Figure 2 shows a model of EDA components.

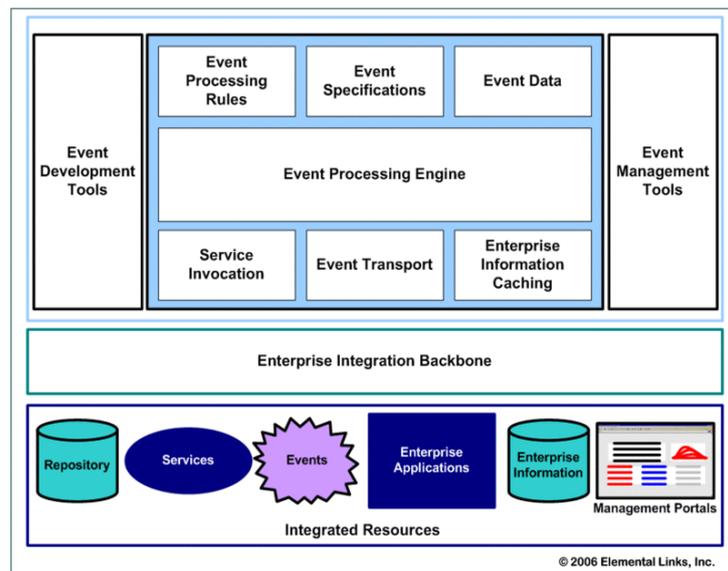


Figure 2: Event-Driven Architecture Components (taken from [4])

### 2.3 Business Activity Monitoring

One important aspect of CEP is business activity monitoring (BAM). What is BAM and how does it relate to event processing? BAM is defined as a process that provides a real-time access to key business performance indicators. It represents the last step in event processing as mentioned earlier in section 2. Examples of BAM applications also include algorithmic trading. End users perceive BAM by means of notifications and dashboards, but BAM systems can also trigger automated external processes. BAM applications monitor raw events as well as the real-time decisions made by event scenarios. BAM monitoring implementation is composed of three main steps:

- 1) gathering of data in sufficient quantity so that its processing can provide meaningful result,
- 2) processing of data considering relevant factor and based on this, categorizing and identifying the data,
- 3) clearly displaying the result of analysis in a user-friendly way.

### 2.4 Event scenario

An event scenario is a pattern of rules that can indicate a complex event is occurring. Event Scenario is not a standard formal notion. An example of an event scenario used in an algorithmic trading application is shown below on Figure 3. The scenario has a series of states. Each state can have one or more rules that control the flow. Let's analyze the example in Figure 3. In "start" state we begin to look for a pattern initially triggered by a rule that matches a set of filter criteria. In this case, it's looking for changes in stock prices on two ticker symbols. In the "Check Quantities" state we check for these price changes. After detecting a price change, the spread, or difference between the spreads, is calculated. If the spread does not reach a certain trading condition, the scenario returns to the "Wait for spread" state. If it does reach a buy or sell condition, the scenario transitions to the "Issue Orders" state, where it takes action — buying or selling — based on complex series of events.

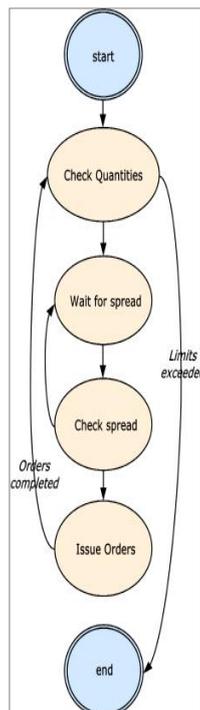


Figure 3 Example of event scenario [3]

## III. BUSINESS PROCESS MANAGEMENT

Business Process Management (BPM) is a concept that intersects the fields of management and information technology. BPM covers the realm of business processes that, among other entities, consist of organizations, humans, and systems. This system is known and widely used since the early 1900's. BPM includes three activities: process design, execution, and monitoring. Business process instances are running concurrently and completely independent from each other. There are many definitions of the "business rule" term. According to [6], the business rule is a statement that defines or constrains some aspect of the business process which is intended to assert business structure, or to control or influence the behavior of the business.

A business rule cannot be decomposed further into more detailed business rules. The main characteristics of business rules include, among others, the following statements:

- Business rule classify, compute, compare and control data to direct the flow in a business process.
- Business rules can certify the data in the business forms.
- Business rules are not processes.
- Business rules provide criteria for making decisions.

A business rule may be described formally, e.g. by using the BPEL language (Business Process Execution Language) or decision trees. A business rule can also be described informally in plain language. Formalization of business rules lies in identifying the atomic statement as the definition of a term, fact, constraint, or derivation. Terms, facts, and some of the constraints can be represented as graphical models. Constraints and derivations must be translated into another formalism. One of the possible applications of business rules in CEP platforms is in event patterns description. By using the rules we can recognize certain patterns in the input events stream. Business rules might be very helpful during the decision making process. Business rules can be defined as restrictions, guidelines, computations, inferences, timings and triggers; the last two are especially fit for event processing frameworks. Business rules drive process definitions as well as the decisions made within business processes. The mapping between rules, processes and decisions is considerably easier if done from an event perspective, where the logic is defined in a precise rule set. BPM is appropriate for fairly predictable processes, while CEP is best suited for responding to events.

### 3.1 BPEL

BPEL (Business Process Execution Language) is an XML-based language that enables task-sharing and serves for processes automation. By using BPEL, a business process is formally described. Business processes can also be described by Business Process Modeling Notation (BPMN), but BPMN is based on graphical representation, so for the high volume data processing, BPEL is more efficient. Smart methods exist which transform BPMN into BPEL and vice versa.

BPEL allows definition of business processes that call external services using Web Services. It provides functions for data manipulation and allows user to define process variables. Using this data, the course of the process can be influenced. This model supports business process lifecycle management. The process can be stopped, paused, resumed, etc. BPEL also supports long-running transaction model that facilitates the definition of transaction boundaries and compensatory actions in case of a failed transaction. Support for business process events (support for asynchronous call model) is also defined.

Example of a simple process in BPEL:

```
<process name="order"
  targetNamespace="http://jbpm.org/examples/hello"
  xmlns:tns="http://jbpm.org/examples/hello"
  xmlns:bpel="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/">
  <import ... > <!-- here is usually imported some external source - eg.wsdl -->
  <partnerLinks>
    <!-- establishes the relationship with the caller agent -->
    <partnerLink name="caller" partnerLinkType="tns:Greeter-Caller"
myRole="Greeter" />
  </partnerLinks>
  <variables>
    <variable name="request" messageType="tns:nameMessage" />
    <variable name="response" messageType="tns:greetingMessage" />
  </variables>
  <correlationSets>
    <correlationSet name="CorrelationSet1" properties="ns1:orderId">
  </correlationSets>
  <sequence name="MainSeq">
    <!-- contains the main body of the process -->
  </sequence>
</process>
```

### 3.2 XBRL

In relation to CEP, XBRL might be used for source events description. We might benefit from the fact that BPEL is also a XML-based language, thus we can analyze the input data in a more structured way.

Extensible Business Reporting Language (XBRL) is an XML-based markup language for electronic transmission of business and financial data. This language is an established standard for data exchange between different platforms. It provides major benefits in the preparation, analysis and communication of business information. It is an open standard and it is already utilized in a number of countries. The introduction of XBRL tags enables automated processing of business information by computer software, cutting out labour intensive

and costly processes of manual re-entry and comparison [7]. XBRL can handle data in different languages and accounting standards. It can be adapted to meet different requirements in a flexible way. Data can be transformed into XBRL by appropriate mapping tools. The benefits are clearly visible in automation, cost saving, faster, more reliable and accurate data handling, improved analysis and in better quality of information and decision-making. The last property might be very useful for financial markets' complex events processing. XBRL is a powerful and flexible version of XML which has been defined specifically to meet the requirements of business and financial information interchange.

The language dictionary is defined in XBRL Taxonomies. These are the categorization schemes which define the specific tags for individual items of data. By using XBRL, stock exchanges, investment analysts, advisers and other users can benefit from greater data transparency and robustness, clarity and consistency of company financial data and the possibility to utilize more powerful software tools for data analysis. Figure 4 shows an example of an asset, represented in XBRL (left) and in human readable form (right).

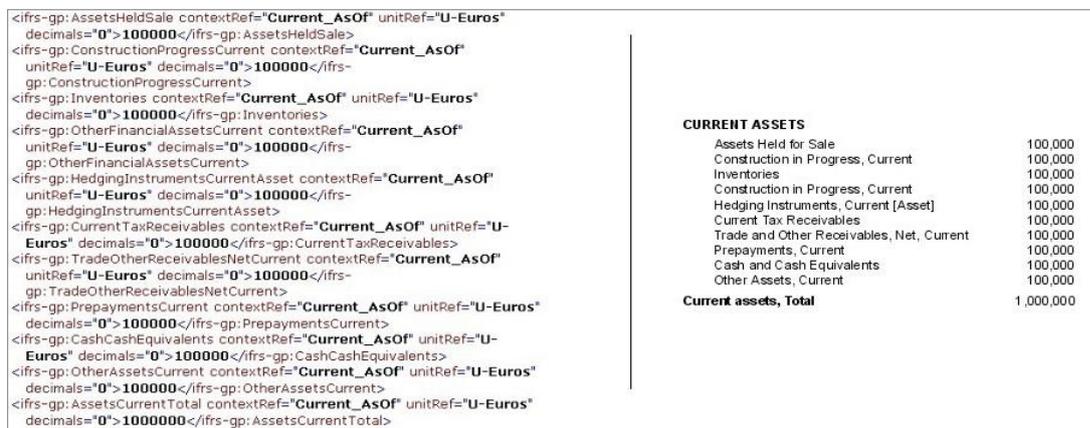


Figure 4 Example of asset described in XBRL and in human readable format [7]

#### IV. CONCLUSION

The decision making process is the key element of complex event processing. We would like to focus on this part of the process and use the rules specified by BPM for the decision making. In CEP, the decisions are made when the data stream reaches the system. If trading decisions should be made, the processing speed of the system is crucial. In this case we use data-driven decision making model which requires both historical and real time data. Using a stand-alone CEP solution is quite expensive, but if we contribute to processing with a BPM solution, we might speed up the whole process and take advantage of utilizing both the CEP and BPM domains at the same time.

The future research will be focused on design of algorithmic models to effectively process high volume market data. We will use advantage of CEP platform for its processing in real time and we will try to integrate business rules to control the flow of events prediction.

#### ACKNOWLEDGEMENT

This work was partially supported by the BUT FIT grant FIT-S-14-2299, "Research and application of advanced methods in ICT".

#### REFERENCES

- [1] Rainer, A., Emmersberger, C., Springer, F., Wolff, C.: Event-Driven Business Process Management and its Practical Application Taking the Example of DHL. In: 1st International workshop on Complex Event Processing for the Future Internet, 28.09.-30.09.2008, Vienna, Austria.
- [2] Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems, Addison Wesley Professional, May 2002, ISBN: 0201727897.
- [3] Palmer, M., Džmuráň, M.: An Introduction to Event Processing, Powering Real-Time, Intelligent Business Applications.
- [4] Michelson, B.: Event-Driven Architecture Overview, 2006, [online, visited March 2014] <<http://www.elementallinks.com/2006/02/06/event-driven-architecture-overview/#sthash.8OHnhIDm.dpbs>>.
- [5] Steinberg, A., & Bowman, C.: Handbook of Multisensor Data Fusion, JDL Laboratories, CRC Press, 2001.
- [6] Leavitt, N.: Complex-Event processing poised for growth, Industry Trends, IEEE Computer Society, 2009.
- [7] Financial Report Foundation, [online, visited March 2014] <<http://www.ifrs.org/Pages/default.aspx>>.