Using complex events to represent domain concepts in graphs

Riley T Perry, Cat Kutay, Fethi Rabhi

Computer Science and Engineering, The University of NSW

Abstract. We have developed an event based visualisation model for analysing patterns between news story data and stock prices. Visual analytics systems generally show a direct mapping from data to visualisation. We show that by inserting an intermediate step, which models an expert manipulating data, we can provide unique results that display patterns within the data being investigated and assist less expert users.

1 Introduction

As volumes of data grow the tools used to explore and analyse that data need to become more sophisticated. Some of the analysis process used by data experts can be incorporated into a visual model. Specifically we examine whether modelling financial data as events using Complex Event Processing (CEP) [2] techniques helps with data visualisation, analytics, and Exploratory Data Analysis (EDA).

An event can be defined as anything that happens, or is interpreted as happening at a particular time. Examples of events include banking transactions, financial trades and quotes, aircraft movements, updates in social media sites (e.g. Facebook), and sensor outputs [8]. EDA [1] is a term that describes techniques used to identify patterns and information in large amounts of data. Existing systems provide poor support for the grouping of data to extract conceptual patterns. This work proposes a visualisation model for presenting event data so as to incorporate expert techniques in data collation and pattern representation for event models.

The paper is structured as follows: Section 2 describes the proposed approach, introduces complex event processing concepts and contains definitions for the visualisation model. In section 3 the models and concepts are applied to financial data. Section 4 contains a brief evaluation of the visualisation. Finally, section 5 presents our conclusions and potential future work.

2 Proposed Approach

Our approach is to use a CEP model combined with an existing formal visualisation model (figure 1). Together these models allow us to visualise event occurrences on a graph, an example of which can be seen later in figure 5. The CEP and visualisation models are described next and the combined models are presented.

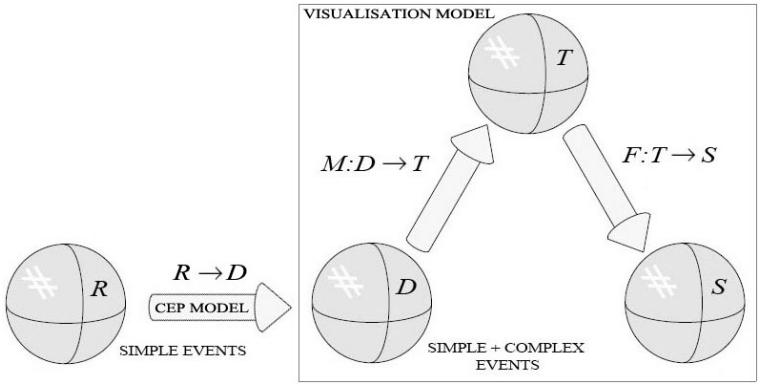

Fig. 1. The combined models

2.1 CEP Model

The role of the CEP model is to map *simple events* to *complex events* via pattern matching algorithms. Complex events are ultimately made up of simple events, which are usually *raw* events, a record of an occurrence in the real world. By utilizing a concept called CEP aggregation we can create event hierarchies to model higher level concepts.

Aggregation is a series of levels related to event complexity. Level one events are all simple events. All levels above level one are made up of complex events. The CEP model in this paper is inspired by previous work using Thomson Reuters stock market data feeds [6].

Events in a complex event are linked by relational mapping. Events can be related by *time*, *causality* and *aggregation*. E.g. events can be part of a price change and form a time sequence of changes in the price. Causality between events implies that preceding events had to happen to cause the latter event. The history of causes is called the *causal vector*. Aggregation can happen between events and they can be combined to form a more complex event. For example a news story from start to end may contain many individual news events. Formally, our events become complex when they involve aggregation.

Aspects An event is defined by the following aspects:

Form α - The main attribute store for an event.
Significance β - The type of activity the event describes within its form.
Relativity γ - A series of events that caused the event.

Formally, an event E occurred at time t and is uniquely identified by its identifier i. E is defined as a tuple: $E = (\alpha, \beta, \gamma)$. From left to right, $\alpha = (i, t, A)$. α contains the event identifier, timestamp, and n attributes where $A = \{a_1, a_2, \ldots, a_n\}$ and a_i is an attribute. β is single event type identifier which is usually the name of the event. γ is an ordered list of event identifiers: $\gamma = [i_1, i_2, \ldots, i_n]$ which are

identifiers for the events that directly caused E to occur. Events in γ may be complex or simple. If $\gamma = \emptyset$ then E is a simple event, otherwise E is complex.

When modelling an event in the visualisation we wish to retain the ability to decompose back to the component events. Hence when large scale patterns are identified we can examine the separate attributes of the complex event. Hence the CEP mapping must be reversible.

Event Patterns The purpose of defining event patterns is to use them to find patterns in event data and generate new events based on these matches. The event patterns can be implemented programmatically for flexibility or reduced to a set of simple set of primitive operations as part of a pattern language.

2.2 Visualisation Model

To make meaningful statements about, and compare and contrast various visualisation techniques we are using a model of the visualisation process. The model gives us a language for talking about the differences between visualisations. The proposed model was created by Matthew Alexander Hutchins as part of his doctoral thesis [3]. Here we consider visualisation spaces as algebras and mappings between them are morphisms. His model a formal model with data (D), task (T), and scene (S) spaces. His model can be seen in figure 1 and is the part of the figure contained within the box. The model translates data to a format that can be directly mapped to display elements on the screen. Space D contains data in its original format. Space T is the mapping from D to form an algebra suitable for visual representation and S is the data display on the screen. In Hutchins' model the mapping of data to the task space ($D \rightarrow T$) automatically incorporates the data manipulation techniques used by experts in that particular domain. Hence we extend the model with a pre-mapping to incorporate CEP as representing export pattern matching.

Spaces D, T, and S are defined below. These definitions are taken from [9].

Definition 1 A space is an algebra. The space is represented by $\{\Sigma, G, K\}$ Where Σ is a collection of sets, G represents functions with domains in Σ and co domains in $\Sigma+$ and K represents constants from sets in Σ. $\Sigma = \{H_1, H_2, \ldots, H_n\}$ where n = the number of sets. $G = \{G_1, G_2, \ldots, G_n\}$ where n = the number of functions. $K = \{K_1, K_2, \ldots, K_n\}$ where n = the number of constants.

Definition 2 A morphism between algebras $A = \{\Sigma_A, G_A, K_A\}$ and $B = \{\Sigma_B, G_B, K_B\}$ is a set of functions P with domains in ΣA and co domains in ΣB.

Definition 3 A function p is a relation from set H_1 to set H_2 if for every $x \in H_1$ there is a unique $y \in H_2$ such that $(x, y) \in p$. This can be represented by $p : H_1 \rightarrow H_2$.

An instance of the model is a tuple (d, t, s) where $t = M_p(d), s = F_p(t)$ Where d, t, and s represent the entire snapshot of values for spaces D, T, and S respectively.

2.3 Combined Model

We propose an extension to the visualisation model where data is event based and D is made up of simple and complex events. The raw data, which is made up of simple events, is contained in the space R. CEP is used between R and D to add complex events to D (also shown in figure 1). The conceptual understanding of entities and relationships in the semantic structure of the data is modelled using CEP structures.

3 Application to Financial Data Analysis

The combined models are now applied to financial data and suggested interface is presented.

3.1 Using the CEP Model: $R \rightarrow D$

R contains *Price Updates* and *News Updates*, which are simple events. The translation from R to D runs all simple events through a series of event pattern instances. Complex events will be created by these pattern instances. Simple events are then mapped directly to D and any complex events created are also mapped to D. These patterns are described next as part of an event hierarchy.

Level One Events

News Update A simple *news update* event ($\boldsymbol{NU}$) is an indivisible news related message. Each message consists of various codes including a story identifier called the *PNAC*. Simple news updates, when combined, form a partial or complete *News Story*. News stories, described next, are complex events.

Price Update A simple *Price Update* event ($\boldsymbol{PU}$) contains the trading price, volume, and trading timestamp (or period) for a particular security.

Level Two Events Events above level one are complex events. We describe here the type of complex events and how they are formed.

News Story A *News Story* (figure 2), $\boldsymbol{NS}$ with news events $\boldsymbol{NU_1}, \boldsymbol{NU_2}, \ldots, \boldsymbol{NU_n}$ can be represented by a simple tree diagram like this: $\boldsymbol{NS}$ sits in a two level *abstraction hierarchy*. $\boldsymbol{NU_1}, \boldsymbol{NU_2}, \ldots, \boldsymbol{NU_n}$ all share the same *PNAC* but may be different occurrences of same news story. Attributes of $\boldsymbol{NS}$ are drawn or derived from this pool. CEP is a realtime technology used to make decisions quickly. A set of definitive Pattern rules for $\boldsymbol{NS}$ can be hard to determine as a story can span several days with many events, or just be comprised of an instantaneous single event.

Price Jump $\boldsymbol{PJ}$ (figure 3) is a *Price Jump* which contains positive price update events $\boldsymbol{PU_1}$ and $\boldsymbol{PU_2}$. An example of a price jump for the running example is a price increase of over 2 points for a certain stock.

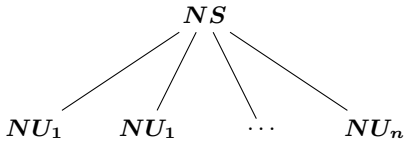

Fig. 2. A news story hierarchy

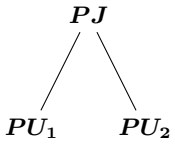

Fig. 3. A price jump hierarchy

Level Three Events We then have higher level hierarchies of aggregation, when combined event stories with pricing, that can have attributes of their own. This allows us to build higher level concepts which can in turn be metrics for analysis and further aggregation.

News Story Plus A *News Story Plus* (figure 4), $\boldsymbol{NS+}$, event is generated when there is a price jump within (and related to) the time period for a news story. Given a news story $\boldsymbol{NS}$ and a price jump $\boldsymbol{PJ}$ extra information can be gained from combining and doing calculations on the attributes of both. The concept of a news story plus (shown below) sits in three level CEP event abstraction hierarchy. Timestamps for $\boldsymbol{NU_1}, \boldsymbol{NU_2}, \ldots, \boldsymbol{NU_n}$ always

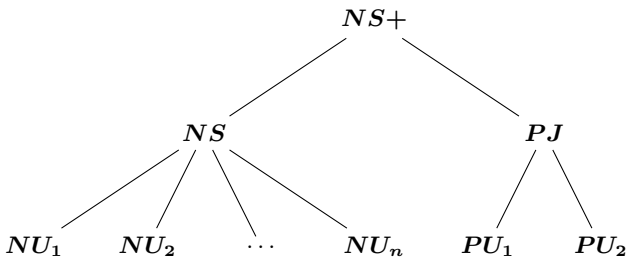

Fig. 4. A news story plus hierarchy

fall within the boundaries of $\boldsymbol{PU_1}$ and $\boldsymbol{PU_2}$, i.e.: $\boldsymbol{NU_1}^{(t)} >= \boldsymbol{PU_1}^{(t)}$ and $\boldsymbol{NU_n}^{(t)} <= \boldsymbol{PU_2}^{(t)}$. Preferably timestamps should be at regular intervals and should match exactly. i.e. $\boldsymbol{NU_1}^{(t)} = \boldsymbol{PU_1}^{(t)} \ldots \boldsymbol{NU_n}^{(t)} = \boldsymbol{PU_2}^{(t)}$.
It's useful at this point to introduce a variable, $\epsilon =$ event level-1, to represent the event level for display purposes. A news story plus for example has an ϵ of 2.

3.2 Using the Visualisation Model

The mapping of CEP (complex and simple) events $D \rightarrow$T can be described as mapping from a pool of events to a three dimensional space of discrete points. $T \rightarrow S$ is realised by filtering/zooming via the interface, which maps a subset of these points to a 2 dimensional plane. When visualizing an event on a two-dimensional plane t (time) will usually be mapped to the x axis. There are two distinct classes of significance of simple event for visualisation purposes:

2D One of which will apply to both the x (time) axis and a single form attribute for the y axis. An example is a price update.
1D One of will apply to the x axis. An example is an individual news update.

The mappings between visualisation spaces are now described in detail.

3.3 Mapping $\boldsymbol{D \rightarrow T}$

The data space of CEP events D is then mapped onto the Task space T. The set of mapping functions is given as $M : D \rightarrow T$.

From Section 2.2, $D = \{\Sigma_D, G_D, K_D\}$ and $T = \{\Sigma_T, G_T, K_T\}$, where $\Sigma_D = \{d_1, d_2, \ldots, d_n\}$ and $d_1, d_2, \ldots, d_n$ are simple or complex events with the exception that relativity (γ) is separated into a set of functions $G_D = \{g_1, g_2, \ldots, g_n\}$. $M = \{m_1, m_2, \ldots, m_n\}$ where $m_1, m_2, \ldots, m_n$ are mapping functions from D to T. $K_T = \{k_1, k_2, \ldots, k_n\}$ where $n =$ the number of constants and $k_1, k_2, \ldots, k_n$ are task attributes x and y. Each event has an x, y position on a cartesian plane relative to the smallest x and y attributes as atomic units in a meta space of planes. Each ϵ (event level) in E is mapped to a different plane. This is based on the hierarchy level determined by the relativity. Events of the same type are always on the same plane.

Simple Events $\boldsymbol{D \rightarrow T}$ A set of functions M maps data (events) to the task space. The mapping is a homomorphism represented by functions: $D = \{e_1, e_2, \ldots, e_n\}$ where each entry is a set of attributes for an event E. Below an instance of an event in D is referenced with D_e.
$T = \{(e_1, x_1, y_1, z_1 = 0), (e_2, x_2, y_2, z_2 = 0), \ldots, (e_n, x_n, y_n, z_n = 0)\}$

The mapping for a news update is $M_{\boldsymbol{NU}} : D \rightarrow T$ and $(x, y, E_{root}) \in M_{\boldsymbol{NU}}$ where $T_{e.x} = M_1(D_e) = (e.t/t_{min})$. To work out x position we must work out the minimum non zero timestamp increment t_{min} beforehand by iterating through all elements of D. i.e. the smallest non zero difference between all timestamps in D. y is always 0.

The mapping for a price update is $M_{\boldsymbol{PU}} : D \rightarrow T$ and $(x, y, E_{root}) \in M_{\boldsymbol{PU}}$ where $E_{root} = \boldsymbol{PU}$ (Price Update) and, $T_e.y = M_2(D_e) = (D_e.price/price_{min})$. To work out x position we must work out the minimum non zero price change $price_{min}$ beforehand by iterating through all elements of D. i.e. the smallest non zero difference between all prices in D.

If the event is the *Root Event* (E_{root}), a price update, then the event is mapped to a 2D plane, otherwise all planes are 1D and x based.

Complex Events $D \rightarrow T$ A z axis is introduced where each entry is a set of attributes for an event E.

$T = \{(e_1, x_1, y_1 = 0, z_1), (e_2, x_2, y_1 = 0, z_2), \ldots, (e_n, x_n, y_n = 0, z_n)\}$

The basic mapping for all complex events is the same. $M_{CE} : D \in T$ and $(x, z, E_o) \in M_{CE}$ where $T_{e.x} = M_{CE}(D_e) = (e.t/t_{min})$; $T_e.z = M_{CE}(D_e) = (z = \epsilon)$; where ϵ determines the z order of the plane.

Again, to work out x position we must work out the minimum non zero timestamp increment t_{min} beforehand by iterating through all elements of D. i.e. the smallest non zero difference between all timestamps in D.

UI Mapping $T \rightarrow S$ Again based on the definition from [9], for $F : T \rightarrow S$, $S = \{\Sigma_S, G_S, K_S\}$ and $F = \{f_1, f_2, \ldots, f_n\}$ where $f_1, f_2 \ldots, f_n$ are functions from T to S. $K = \{k_1, k_2, \ldots, k_n\}$ where $n =$ the number of constants and $K_S = \{s_1, s_2, \ldots, s_n)$ where $s_1, s_2, \ldots, s_n$ are scene attributes.

S represents direct visual elements on a display device. All events and 3D coordinates are stored in a 3D matrix in T (represented by x, y, and z components).

These provide the components for the mapping from the events to the visualisation space which provides the homomorphisms between the task and visual spaces.

The proposed user interface has 3 distinct sections. They are: An *Attribute Panel*, *Display Panel* (figure 1), and a *Filter Panel*. The display panel is the main graph and displays S_{win}, where the root event sets up the x, y plane. Events are then stacked based on their Z-order (z). There is always a selected event, E_{sel}. The attributes in the attribute panel are those of the selected event and are drawn from K_T. The filter panel allows the removal of members of T and the display panel displays all, or part of the filtered members of T depending on zoom and stacking controls which are based on relativity, i.e. G_T. Of particular interest is filtering out x, y ranges (zooming), i.e. $S_{win} = \{$ a window in $T_x, T_y\} = \{x_{start}, x_{end}, y_{start}, y_{end}\}$ and showing only events within the selected event's relativity.

With the display panel in figure 5 when we want more information on the news story plus event we would simply click on the event, in this case a large square in the top left corner. What appears then, in the panel below the arrow, is just the news story plus event and its constituent events. These parent events, and in turn, their parents, are shown here.

A news story plus occurs when a news story is generated that contains a positive price jump. You can see that the price jump (the two price updates) did indeed occur over the life of the news story, which started with news update 1 and was finalised with news update 2. Under the news story plus event there are three distinct events: news story, news update 2, and price update 2.

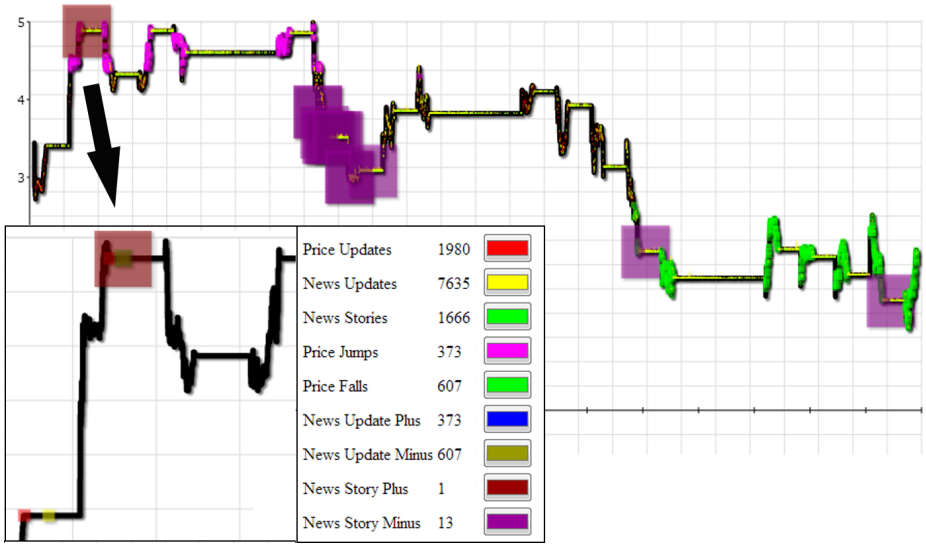

Fig. 5. The display panel

4 Visualisation Evaluation

A key method by which we may validate a process in science is the principle of falsifiability, championed by Karl Popper [4]. According to [5] a falsifiable images is a "valid pictorial representation of the truth". Falsifiability of a visual representation of data involves establishing the necessary and sufficient condition of the validity of the mapping between image and data, in the following format:

Consistent The data is consistent with the image through a homomorphism, and

Representative The image is representation of the data without distortion

The visualisation model should be such that you can detect in the image patterns that represent patterns that can then be verified in the data. Since mappings $D \rightarrow T$ are structure-preserving homomorphisms, the visualisation model mapping is falsifiable, or, the events are detectable in the final visualisation are present in D.

Other visualisation systems generally only show data representations of the R space. The event based version shows higher level entities and a hierarchy of those entities which represent the output of CEP patterns.

5 Conclusions and Future Work

By thinking of the domain conceptually and using these concepts within a visualisation framework we can enhance more simplistic visualisation techniques.

Future work would include building different CEP models, the usability of a prototype developed from this model against multiple visualisation systems, and developing an iterative model to change or add CEP patterns on the fly.

Another promising area for future research is in formal validation and visualisation falsifiability. A potential way to analyse the value of modeling tool is to verify that the visualisation is falsifiable. For this we are developing an algebraic formalism could be used to describe the CEP process from $R \rightarrow D$.

References

1. Tukey, J. A.Exploratory Data Analysis, Addison-Wesley (1977).
2. Luckham, A.D.C.: The Power of Events: An Introduction toComplex Event Processing in Distributed Enterprise Systems. Addison-Wesley, Reading (2002).
3. Hutchins, M.A. (1999) Modelling Visaualization Using Formal Algebra, A thesis for Doctor of Philosophy in The Australian National University, CSIRO ICT Centre, Canberra, Australia.
4. K. Popper, The Logic of Scientific Discovery, Routledge/Taylor & Francis e-Library, 2005.
5. Andrew J. Hanson, Putting Science First: Distinguishing Visualizations from Pretty Pictures. Computer Graphics and Applications, IEEE, 34, 4. 2014.
6. Calum S. Robertson, Fethi A. Rabhi, Maurice Peat. A (2012) Service-Oriented Approach towards Real Time Financial News Analysis University of New South Wales, Smart Services CRC, Sirca, University of Sydney.
7. Reuters NewsScope Archive v2.0 User Guide v2.4 Date of Issue: 29th May 2008.
8. W. Chen and F.A. Rabhi, An RDR-Based Approach for Event Data Analysis, In J.G. Davis, H. Demirkan and H.R. Motahari-Nezhad (eds), Service Research and Innovation, Lecture Notes in Business Information Processing Volume 177, 2014, pp 1-14.
9. T. Mala*, P. Bhargavi and T.V. Geetha, GVP model based temporal visualisation of user-centric data, International Journal of Metadata, Semantics and Ontologies Volume 3, Issue 4, 2008, pp 305-317.