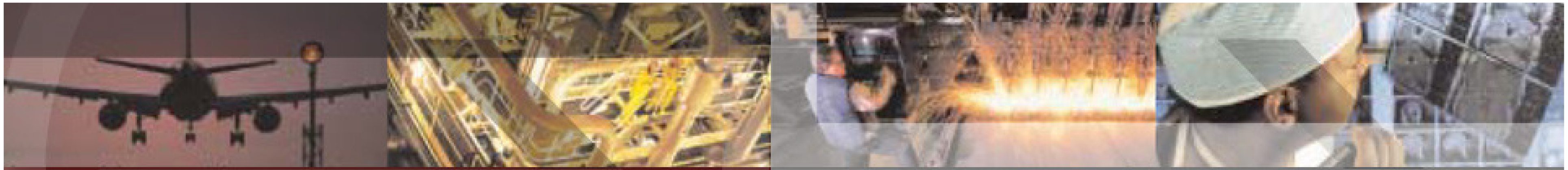
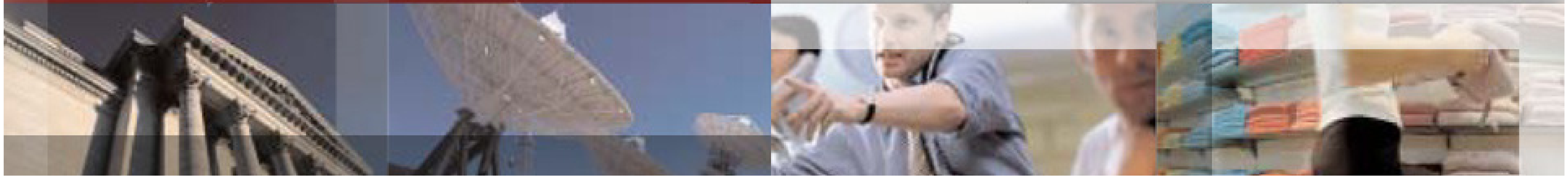


Complex Event Processing:

EPTS Glossary Tutorial

TIBCO®
The Power of Now®

Paul Vincent
TIBCO Software Inc.



EPTS

- Event Processing Technical Society
- Vendor group + interested parties (e.g. David Luckham, Gartner)
- Website: http://www.ep-ts.com
- Glossary: http://complexevents.com/?p=361

Following definitions are (re-ordered, mostly without change) from the EPTS glossary web site

Event

- **Event***:* anything that happens, or is contemplated as happening.
  - *Examples:*
    - a financial trade,
    - an airplane lands,
    - a sensor outputs a reading,
    - a change of state in a database, a finite state machine,
    - a key stroke,
    - a natural occurrence such as an earthquake,
    - a social or historical happening, e.g., the abolition of slavery, the battle of Waterloo, the Russian revolution, and the Irish potato famine.
- **Event** (also **event object**, **event message**, **event tuple**): an object that represents, encodes or records an event, generally for the purpose of computer processing.
  - *Examples:*
    - a purchase order (records a purchase activity),
    - an email confirmation of an airline reservation,
    - stock tick message that reports a stock trade,
    - a message that reports an RFID sensor reading,
    - a medical insurance claim document .

Notes on Events

- Events are **processed** by computer systems via their representations as "**event objects**".
  - The same activity may be represented by more than one event object
  - Each event object might record different attributes of the activity.
- In many event processing systems, for example simulation systems, events are **immutable**.
  - A modification or transformation of an event must be achieved by creating a new event object
    - not by altering the original event.
  - Deletion would entail removing an event from further processing.
- ***Overloading****:* event objects contain data. The word "event" is overloaded so that it can be used as a synonym for event object.
  - The term "**event**" is used to denote both the everyday meaning (anything that happens) and the computer science meaning (an event object or message). The context of each use indicates which meaning is intended.

Event Types and Attributes

- **Event type** (also **event class**, **event definition**, or **event schema**): An event type is a class of event objects.
  - *Examples:*
    - the type of all price quotations,
    - the type of all sensor readings for any kind of sensor.
  - *Notes:*
    - All events must be instances of an **event type**.
    - An event has the structure defined by its type.
    - The structure is represented as a collection of event **attributes**.
    - Event types should be defined within the type definition system of a modern strongly typed computer language such as XML Schema or Java.
- Events will usually specify certain standardized data attributes, such as:
  1. a unique event identifier by means of which the event can be referenced,
  2. the type of the event,
  3. its creation time stamps,
  4. its source of creation.
- **Event Attribute** (also called **event property**): a component of the structure of an event.
  - *Note:* An attribute can have a simple or complex data type.

Simple and Raw Events

- **Simple event***:* an event that is not an abstraction or composition of other events.
- **Raw event**: an event object that records a real-world event.
  - *Note:* A raw event may represent a simple real-world event (e.g. the phone rang) or a complex real-world event (e.g. the stock market crash of 1929 was a complex real world event that can be recorded as a “complex raw event”).

Instantaneous and Virtual Events

- **Instantaneous event:** an event whose duration is less than the granularity of any clock that is applied in the system.
  - has a single timestamp signifying when the event happened.
  - the time interval of the event has a measured length of zero (i.e. the start and end times are the same).
  - *Note*: an instantaneous event may have other attributes as timestamps for other information, e.g., arrival time.
- **Virtual event***:* an event that does not happen in the physical world but appears to signify a real world event; an event that is imagined or modeled or simulated.
  - *Examples:*
    - instruction executions modeled by a hardware design simulation,
    - events predicted by a weather simulation,
    - events modeled by a war game,
    - events that take place in a dream (”these dreams of you, So real and so true” — Van Morrison),
    - events in Virtual Reality.

Event Relationships

- **Cause**: an event A is a cause of another event B if A had to happen in order for B to happen.
  - *Examples:*
    - the birth of a father and the birth of a son of the father,
    - sending an email and a reply to that email.
  - *Note:* this is a definition of *computational causality*. It requires A to be necessary for B to happen. Other definitions of causality are possible, e.g., *probable cause*.
- **Abstraction**: an event is an abstraction of a set of events if it summarizes, represents, or denotes that set of events.
- **Related events**: events are related by *time, causality, abstraction* and other relationships.
  - Time and causality impose partial orderings upon events.

Derived and Composite Events

- **Derived event** (also **synthesized event**): an event that is generated as a result of applying a method or process to one or more other events.
  - *Example*:
    - A “Company B is a good buy” event may be derived from “Company B price has dropped where Company A price has risen within 5 minutes”
  - *Note*: The absence of an event, say in a given time interval, can lead to a derived event for the missing event.
- **Composite event**: a derived, complex event that is created by combining base events using a specific set of event constructors
  - Operators include disjunction, conjunction, sequence, etc.
  - A composite event always includes the base (member) events from which it is derived.
  - *Note:* A derived event may not be a composite event if its method of derivation lies outside a specified set of allowed constructors.
- *Note*: The term ***aggregate event*** is sometimes used for some forms of composite or derived event.

Complex Events

- **Complex event**: an event that is an abstraction of other events
  - Abstracted events are the *members* of the complex event
  - *Examples:*
    - the 1929 stock market crash (an abstraction denoting many thousands of member events),
    - the 2004 Indonesian Tsunami (an abstraction of many natural events) ,
    - a completed stock purchase (an abstraction of the events in a transaction to purchase the stock).
    - a successful on-line shopping cart checkout (an abstraction of shopping cart events on an on-line website).
  - *Note:* a complex event *denotes* or *signifies* the set of its member events. This may or may not include references to individual member events.
- A composite event or a derived event is a **complex event**. The converses are not necessarily true.

Timestamps

- **Timestamp**: a time value attribute of an event, recording the reading of a clock in the system in which the event was created or observed.
  - *Examples:*
    - creation time: the time interval or time at which an event was created,
    - arrival time: the time at which an event arrived at a point of observation.
  - *Note:* An event can contain timestamps according to one or more clocks. For example it could contain both its creation time according to a clock where it was created and its arrival time at a system location according to a clock at that location.
- In systems with **multiple clocks**, the issue of clock **synchronization** is an ongoing topic of research
  - *Example:* clocks in a distributed computer system
- Not all time attributes are "timestamps"
  - *Example:* Derived events' times may be derived from their source events.

Sources and Sinks

- **Event source** (also **event emitter** or **event producer**): an entity that sends events.
  - *Examples:*
    - a software module,
    - sensor,
    - clock.
- **Event sink** (also **event consumer**): an entity that receives events.
  - *Examples:*
    - a software module,
    - database,
    - dashboard,
    - person
- **Event channel** (also **event connection, event pathway, event topic**): a conduit in which events are transmitted from event sources (emitters) to event sinks (consumers).
  - *Notes:*
    - A channel can carry events of multiple types.
    - An event channel may be public (without access restrictions), or controlled.
    - An event channel is a medium for delivering one or more event streams.
    - A single event channel may be consumed by multiple event consumers.

Streams and Clouds

- **Event stream**: a linearly ordered sequence of events.
  - *Notes:*
    - Usually, streams are ordered by time, e.g., arrival time.
    - An event stream may be bounded by a certain time interval or other criteria (content, space, source), or be open ended and unbounded.
    - A stream may contain events of many different types.
- **Window**: a bounded portion of an event stream.
  - *Example:* The events in the last ten minutes - i.e., a ten-minute moving window.
  - *Notes:* Windows define sub-sequences of an event stream typically to focus the event processing on specific data, or to improve event processing performance, but may also have other uses.
- **Event cloud**: a partially ordered set of events (poset), either bounded or unbounded, where the partial orderings are imposed by the causal, timing and other relationships between the events.
  - *Notes:*
    - Typically an event cloud is created by the events produced by one or more distributed systems.
    - An event cloud may contain many event types, event streams and event channels.
    - The difference between a cloud and a stream is that there is no event relationship that totally orders the events in a cloud. A stream is a cloud, but the converse is not necessarily true.
  - CEP usually refers to event processing that assumes an event cloud as input, and thereby can make no assumptions about the arrival order of events.

Event Processing

- **Event processing**: computing that performs operations on events, including reading, creating, transforming and deleting events.
  - *Note:* The overloaded meaning event object processing is intended in this context.
- **Complex-event processing (CEP):** Computing that performs operations on complex events, including reading, creating, transforming or abstracting them.
  - *Note:* CEP results in creating complex events even if some or all of the source events are simple events
- **Event Stream Processing (ESP):** Computing on inputs that are event streams.
  - *Examples:* Applications that use stock market feeds as inputs and process events in their order of arrival to compute running average stock prices, volume weighted average prices over time windows, etc.
  - *Notes:* ESP has its origins in Active Databases and Data Streams Management. The terminologies ESP and CEP are conceptual classifications. They can be useful in delineating philosophies of event processing and intended applications, but do not specify precisely the underlying capabilities of event processing engines.

Event Processing Patterns

- **Event template**: an event form or descriptor where some of its parameters are variables. An event template matches single events by replacing the variables with values.
  - *Example:* Send of Any Msg: String Msg; Send(John, Msg)
- **Event pattern**: a template containing event templates, relational operators and variables. An event pattern can match sets of related events by replacing variables with values.
  - *Examples:*
    - a pattern of events defining those sets of events in a completed sales transaction.
    - a pattern of events in an email correspondence:
String Msg, Time T1, T2 ; Send(John, Msg, T1) and Receive(John, Msg, T2);
    - a pattern defining the events in any successfully resolved customer complaint.
Customer C, Agent A, Problem P, Time T1, T2, T3; Complain(C, P, T1) -> Engage(A, C, T2) -> Resolved (P, T3);
  - *Note:* event patterns can often be specified graphically.
- **Pattern Instance** (also **event pattern instance**): a set of related events resulting from an event pattern by replacing the variables by values.
  - *Examples:*
    - Send(John, “See the NYT today”, 15.00 EST) and
Receive(John, “See the NYT today”, 12.05 PST);

Event Processing by Rules

- **Constraint** (also **event pattern constraint**): A Boolean condition that must be satisfied by the events observed in a system.
  - *Examples:*
    - A service level agreement limiting the time taken to complete a mortgage transaction from the time an application is received.
- **Rule** (in event processing): A prescribed method for processing events.
  - *Examples:*
    - Whenever three timeouts have happened send an alert to the network manager.
    - If more than ten shopping carts have been active for more than five minutes then activate the website reaction time monitor and display an amber alert on the dashboard.
    - Whenever IBM trades 2% above its 1 hour VWAP and then within 15 minutes trades 5 points below then buy 1000 shares IBM.
  - *Notes:* Event processing rules may be prescribed in many different ways, including by finite state machines, activity diagrams, Java code, SQL code, ECA (event-condition-action) rules or reactive rules that are triggered by event patterns.
- **Event pattern triggered reactive rule**: a rule that prescribes actions to be taken whenever an instance of a given event pattern is detected.

Event Processing Languages, Agents, Networks

- **Event Processing Language (EPL)**: a high level computer language for defining the behavior of event processing agents.
- **Event Processing Agent (EPA)** (also **event processing component**, **event mediator**): a software module that processes events.
  - *Notes:* Event source and event sink are roles that an EPA may play. One EPA could act in both roles - it could be an event source at one moment, an event sink at another time.
- **Event Processing Network (EPN)**: A set of event processing agents (**EPA**s) and a set of **event channels** connecting them.
  - *Notes:*
    - the set of EPAs can be dynamic, i.e., EPAs can be created and destroyed
    - the set of channels can be dynamic, i.e., channels can be created and destroyed
    - dynamic behavior may controlled by patterns of events occurring in the network
    - an EPN need not be an acyclic directed graph, e.g., feedback loops (cycles) are possible
    - the runtime deployment of an EPN may be distributed across multiple physical networks, computers and software artifacts.