Event Processing Technical Society

Event Processing Glossary - Version 1.1

Editors: David Luckham, Roy Schulte

July 2008

Permission to copy and display this glossary in any medium without fee or royalty is hereby granted, provided that you include the copyright notice as shown herein and link or URL to the Material on the EPTS website.

DISCLAIMERS:

THIS MATERIAL IS PROVIDED "AS IS," AND THE EVENT PROCESSING TECHNICAL SOCIETY (EPTS) AND ITS MEMBERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS MATERIAL ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW, THE EPTS AND ITS MEMBERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THIS MATERIAL OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

No other rights are granted by implication, estoppel or otherwise.

© Copyright 2008 Event Processing Technical Society, All rights reserved.

Preface

Purpose

The purpose of the EPTS glossary of terms is to facilitate industry use of event processing technology by providing a common language for developing applications and software infrastructure that use event processing concepts. The event processing glossary has three goals:

- Accelerate the learning of the event processing concept
- Further community communication by enabling practitioners to utilize common concepts and terms
- Provide a foundation for analysis and the development of best practices, publications, and industry standards

Organization

The glossary is presented according to the logical order of the terms. An index with an alphabetical listing is available at the beginning of the document for convenience.

Content

This glossary covers a small set of basic terms related to event processing. It will be frequently updated with additional terms in response to suggestions from the event processing community for improvements and additions.

Our approach is to define each term independently of any particular implementation, product, or domain of application. So, for example, the term *event object* has popular meanings as a tuple, a vector, a row, etc. These are all realizations of events in particular approaches and products. Even the most basic term, *event*, is problematic. Essentially there are two distinct meanings:

- An activity that happens
- Something that represents that activity in a computer system

It is tempting to introduce two separate terms such as *event* and *event object*. However, in any discussion longer than a paragraph or two, this becomes intolerably clumsy and one finds the distinction being misused, forgotten or dropped altogether. For example, using the two separate terms would dictate that —*event processing* (see below) should be —*event object processing*. The best solution is to overload the word —*event*. The context of each use becomes the indicator of which meaning is intended. This has been standard practice in the field of event-driven simulation for the past thirty years. It was the approach taken by the physicists of the early 20th century in discussions of relativity where —event also has two meanings. We have chosen to follow their example in the knowledge that it did not lead them into ambiguity problems.

Alphabetical Index of Glossary Terms

Abstraction
Architecture
Architecture style
Cause
Clocks
Complex event
Complex event processing (CEP)
Composite event
Constraint (event pattern constraint)
Derived event (synthesized event)
Event
Event (event object, event message, event tuple)
Event attribute (event property)
Event channel (event connection, event pathway, event topic)
Event cloud
Event-driven
Event-driven architecture (EDA)
Event pattern
Event pattern triggered reactive rule
Event processing
Event processing agent (EPA) (event processing component, event mediator)
Event processing language (EPL)
Event processing network (EPN)
Event sink (event consumer)
Event source (event emitter or event producer)
Event stream
Event stream processing (ESP)
Event template
Event timing (timing)
Event type (event class, event definition, or event schema)
Granularity (chronon)
Instantaneous event
Pattern instance (event pattern instance)
Raw event
Relationships between events
Rule (in event processing)
Simple event
Time interval
Timestamp
Virtual event
Window

Glossary of Terms

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 or 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 (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:

1. Events are processed by computer systems by processing 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. In such systems, a modification or transformation of an event must be achieved by creating a new event object and not by altering the original event. Deletion would entail removing an event from further processing.
2. Overloading: Event objects contain data. The word —event is overloaded so that it can be used as a synonym for event object. In discussing event processing, the word —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.

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 type (event class, event definition, or event schema)

A class of event objects.

Examples:

- The type of all price quotations
- The type of all sensor readings for any kind of sensor

Notes:

1. 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 (below).
2. *Event types* should be defined within the type definition system of a modern strongly typed computer language such as XML Schema or Java. Any standard for representing events will usually specify certain predefined data, examples of which might be:

   - A unique event identifier used to reference the event
   - The type of the event
   - The time stamps of the event's creation
   - The source of creation for the event

Event attribute (event property)

A component of the structure of an event.

Notes:

An event attribute can have a simple or complex data type.

Event processing

Computing that performs operations on events, including reading, creating, transforming and deleting events.

Notes:

The overloaded meaning event object processing is intended in this context.

Clocks

A process that creates an ordered ascending sequence of values of type Time with a uniform interval between them. Each value is produced at a tick (or clock tick).

Granularity (chronon)

The length of the interval between clock ticks.

Event timing (timing)

The time value attributes of an event.

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

Notes:

1. An *event* can contain *timestamps* according to one or more *clocks*. For example, it can 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.

2. In systems with multiple *clocks*, the issue of clock synchronization is an ongoing topic of research. Not all *timing attributes* are timestamps. *Timing* in *derived events*, for example, may be derived from *timing* of the source *events*.

Time interval

A period of time bounded by two timing attributes called the interval's start time and end time.

Instantaneous event

An event whose duration is less than the granularity of any clock that is applied in the system. An instantaneous event object will have a single timestamp signifying when the event happened. This is the special case where the time interval of the event has a measured length of zero (the start and end times are the same).

Notes:

An *instantaneous event* may have other *timestamps* with differing significances, e.g., arrival time.

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

Notes:

This is a definition of *computational causality*. It requires A to be necessary for B to happen. For example B's father is a cause of B, but so is B's mother. Other definitions of causality are possible, e.g., *probable cause*. The meaning and definitions of *intentional* or *philosophical causality* have been debated in countless books on philosophy.

Abstraction

An event is an abstraction of a set of events if it summarizes, represents, or denotes that set of events.

Complex event

An event that is an abstraction of other events called its members.

Examples:

- The 1929 stock market crash – an abstraction denoting many thousands of member events, including individual stock trades)
- The 2004 Indonesian Tsunami – an abstraction of many natural events
- A CPU instruction –an abstraction of register transfer level (RTL) 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

Notes:

A complex event denotes or signifies the set of its member events. It is tempting to say that a complex event references the set of its members, the implication being that the event contains a reference. In many cases this is true. But in general, reference may be too strong a requirement since it implies that the members can be determined from the reference. For example, there is no accepted agreement as to which events are members of the 1929 stock market crash.

Derived event (synthesized event)

An event that is generated as a result of applying a method or process to one or more other events.

Examples:

- An event reporting that company B has entered the bidding to take over A with probability 0.9, might be derived from an event reporting that the price of company A's stock has jumped 10% in 5 minutes.
- The absence of an event, say in a given time interval, can lead to a derived event reporting that the first event did not happen.

Composite event

A derived, complex event that is created by combining base events using a specific set of event constructors such as disjunction, conjunction, sequence, etc. A composite event always includes the base (member) events from which it is derived.

Notes:

A derived event is not necessarily composite if its method of derivation lies outside a specified set of allowed constructors. The terminology composite and constructor are from the Active Database terminology.

Relationships between events

Events are related by time, causality, abstraction, and other relationships. Time and causality impose partial orderings upon events.

Notes:

1. Regarding the relationships of composite, derived and complex events: A composite event or a derived event is a complex event. The converses are not necessarily true.
2. The term aggregate event is sometimes used for some forms of composite or derived event.

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.

Notes:

A raw event may represent a simple real-world event (e.g., the phone rang) or a complex real-world event. For example, the stock market crash of 1929 was a complex real world event that can be recorded by a complex raw event.

Complex event processing (CEP)

Computing that performs operations on complex events, including reading, creating, transforming, or abstracting them.

Notes:

CEP ultimately creates complex events even if some or all of the source events are simple events. See also the definitions for event stream processing (ESP), event streams, and event clouds, below.

Event source (event emitter or event producer)

An entity that sends events.

Examples:

- Software module
- Sensor
- Clock

Event sink (event consumer)

An entity that receives events.

Examples:

- Software module
- Database
- Dashboard
- Person

Event channel (event connection, event pathway, event topic)

A conduit in which events are transmitted from event sources (emitters) to event sinks (consumers).

Notes:

1. A channel can carry events of multiple types.
2. An event channel may be public (without access restrictions) or controlled.
3. An event channel is a medium for delivering one or more event streams.
4. A single event channel may be consumed by multiple event consumers.

Event template

An event form or descriptor, some of whose parameters are variables. An event template matches single events by replacing the variables with values.

Examples:

- Send of any message
- 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)

Notes:

Event patterns can often be specified graphically.

Pattern Instance (event pattern instance)

A set of related events resulting from an event pattern where the variables are replaced by values.

Examples:

- end(John, See the NYT today, 15.00 EST) and Receive(John, See the NYT today, 12.05 PST).

Constraint (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, UML diagrams, graphical methods, Java code, SQL code, ECA (event-condition-action) rules or reactive rules that are triggered by event patterns (below).

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 agent (EPA) (event processing component, event mediator)

A software module that processes events.

Notes:

1. Event source and event sink are roles that an EPA may play;
2. One EPA could act in both roles - it could be an event source at one moment and an event sink at another time.

Event processing language (EPL)

A high level computer language for defining the behavior of event processing agents.

Event stream

A linearly ordered sequence of events.

Notes:

1. Usually, streams are ordered by time, e.g., arrival time;
2. An event stream may be bounded by a certain time interval or other criteria (content, space, source), or be open ended and unbounded.
3. A stream may contain events of many different types.

Window

A bounded portion of an event stream.

Examples:

- The events in the last ten minutes – i.e., a ten-minute moving window.

Notes:

Windows define subsequences of an event stream typically to focus the event processing on specific data or to improve event processing performance; however, they may also have other uses.

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:

1. ESP has its origins in active databases and data streams management;
2. The terms 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 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:

1. Typically an event cloud is created by the events produced by one or more distributed systems.
2. An event cloud may contain many event types, event streams, and event channels.
3. 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.
4. CEP usually refers to event processing that assumes an event cloud as input, and therefore can make no assumptions about the arrival order of events.

Event Processing Network (EPN)

A set of event processing agents (EPAs) and a set of event channels connecting them.

Notes:

1. The set of EPAs can be dynamic, i.e., EPAs can be created and destroyed.
2. The set of channels can be dynamic, i.e., channels can be created and destroyed.
3. Dynamic behavior may controlled by patterns of events occurring in the network.
4. An EPN need not be an acyclic directed graph, e.g., feedback loops (cycles) are possible.
5. The runtime deployment of an event processing network may be distributed across multiple physical networks, computers, and software artifacts.
6. Various graphical representations of EPNs are possible.

Event-Driven

The behavior of a device, software module or other entity whose execution is in response to the arrival of events from external or internal sources.

Examples:

- A cell phone
- An event triggered rule
- An operating system
- A bank's trust department where the personnel spend their time putting out fires (i.e., event-driven rather than goal-driven or directed)

Architecture

(From IEEE) The fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.

Notes:

Other definitions of architecture: The conceptual structure and overall logical organization of a computer or computer-based system from the point of view of its use or design; a particular realization of this.

Architecture style

(From Roy T. Fielding) *A coordinated set of architectural constraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style*.

Event-driven architecture (EDA)

An architectural style in which some of the components are event driven and communicate by means of events.

Notes:

This is a very encompassing definition of EDA in which systems that are only partially event driven are included. A purist would require all components to be event driven and all communication between them to be by events. However, some people hold that there are no real world business application systems in which all of the communication between the components is via event-driven relationships.

Glossary According to Lexicographic Order (definitions only)

Abstraction*: An event is an abstraction of a set of events if it summarizes, represents, or denotes that set of events*.

Architecture (*from IEEE****)****: The fundamental organization of a system embodied in its components, their relationships to each other and to the environment, and the principles guiding its design and evolution.*

Architecture style *(from Roy T. Fielding): A coordinated set of architectural constraints that restricts the roles/features of architectural elements and the allowed relationships among those elements within any architecture that conforms to that style*.

Cause: *An event A is a cause of another event B if A had to happen in order for B to happen.*

Clocks*: A process that creates an ordered ascending sequence of values of type Time with a uniform interval between them. Each value is produced at a tick (or clock tick).*

Complex event**:** An *event* that is an *abstraction* of other *events* called its *members*.

Complex-event processing (CEP)*: Computing that performs operations on complex events, including reading, creating, transforming or abstracting them*.

Composite event*: A derived, complex event that is created by combining base events using a specific set of event constructors such as disjunction, conjunction, sequence, etc. A composite event always includes the base (member) events from which it is derived.*

Constraint (***event pattern constraint***): *A Boolean condition that must be satisfied by the events observed in a system.*

Derived event **(*****synthesized event*****)****: An event that is generated as a result of applying a method or process to one or more other events.*

Event*: Anything that happens, or is contemplated as happening****.***

Event* (event object, event message, event tuple): An object that represents encodes or records an event, generally for the purpose of computer processing*.

Event attribute (***event property***)*: A component of the structure of an event.*

Event channel (***event connection, event pathway, event topic***)*: A conduit in which events are transmitted from event sources (emitters) to event sinks (consumers).*

Event cloud*: A partially ordered set of events (poset), either bounded or unbounded.*

Event-driven*: The behavior of a device, software module or other entity whose execution is in response to the arrival of events from external or internal sources.*

Event-driven architecture (EDA)*: An architectural style in which some of the components are event driven and communicate by means of events.*

Event sink (event consumer)*: An entity that receives events.*

Event source* (event emitter** or **event producer): An entity that sends events.*

Event processing*: Computing that performs operations on events, including reading, creating, transforming and deleting events*

Event stream*: A linearly ordered sequence of events.*

Event stream processing (ESP)*: Computing on inputs that are event streams.*

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.*

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 agent (EPA) (event processing component, event mediator)*: A software module that processes events.*

Event processing language (EPL*): A high level computer language for defining the behavior of event processing agents.*

Event processing network (EPN)*:* A set of *event processing agents (EPAs)* and a set of event channels connecting them.

Event template*: An event form or descriptor some of whose parameters are variables. An event template matches single events by replacing the variables with values.*

Event timing (***timing*)***: The time value attributes of an event.*

Event type (***event class*, *event definition*,** or ***event schema***)*: An event type is a class of event objects.*

Granularity (***chronon***)*: The length of the interval between clock ticks.*

Pattern instance* (*event pattern instance*): A set of related events resulting from an **event pattern** by replacing the variables by values.*

Raw event*: An event object that records a real-world event.*

Relationships between events*: Events are related by time, causality, abstraction and other relationships. Time and causality impose partial orderings upon events.*

Rule (in ***event processing***)*: A prescribed method for processing events*.

Simple event*: An event that is not an abstraction or composition of other events.*

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.*

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*.

Window*: A bounded portion of an event stream.*

Acknowledgements We are indebted to many correspondents who have made contributions, suggestions and comments over the past year. They are too numerous to mention individually, but we owe them our thanks.