

Complex event processing

Event processing is a method of tracking and analyzing (processing) streams of information (data) about things that happen (events), and deriving a conclusion from them. **Complex event processing**, or CEP, is event processing that combines data from multiple sources to infer events or patterns that suggest more complicated circumstances. The goal of complex event processing is to identify meaningful events (such as opportunities or threats) and respond to them as quickly as possible.

These events may be happening across the various layers of an organization as sales leads, orders or customer service calls. Or, they may be news items, text messages, social media posts, stock market feeds, traffic reports, weather reports, or other kinds of data. An event may also be defined as a "change of state," when a measurement exceeds a predefined threshold of time, temperature, or other value. Analysts suggest that CEP will give organizations a new way to analyze patterns in real-time, and help the business side communicate better with IT and service departments.

The vast amount of information available about events is sometimes referred to as the event cloud.

Conceptual description

Among thousands of incoming events, a monitoring system may for instance receive the following three from the same source:

1. church bells ringing.
2. the appearance of a man in a tuxedo with a woman in a flowing white gown.
3. rice flying through the air.

From these events the monitoring system may infer a *complex event*: a wedding. CEP as a technique helps discover complex events by analyzing and correlating other events:^[1] the bells, the man and woman in wedding attire and the rice flying through the air.

CEP relies on a number of techniques,^[2] including:

- Event-pattern detection
- Event abstraction
- Event filtering
- Event aggregation and transformation
- Modeling event hierarchies
- Detecting relationships (such as causality, membership or timing) between events
- Abstracting event-driven processes

Commercial applications of CEP exist in variety of industries and include algorithmic stock-trading,^[3] the detection of credit-card fraud, business activity monitoring, and security monitoring.^[4]

History

The CEP area has roots in Discrete event simulation, active database area and some programming languages. The activity in the industry was preceded by a wave of research projects in the 1990s. According to the first project that paved the way to a generic CEP language and execution model was the Rapide project in Stanford University, directed by David Luckham, in parallel there have been three other research projects: Infospheres in California Institute of Technology, directed by K. Mani Chandy, Apama in University of Cambridge directed by John Bates, and Amit in IBM Haifa Research Laboratory, directed by Opher Etzion. The commercial products were dependents of the concepts developed in these and some later research projects. Community efforts started in a series of event processing symposiums organized by the Event Processing Technical Society, and later by the ACM DEBS

conference series. One of the community effort was in producing the event processing manifesto ^[5]

Related concepts

CEP is used in Operational Intelligence (OI) solutions to provide insight into business operations by running query analysis against live feeds and event data. OI solutions use real-time data to collect and correlate against historical data to provide insight into and analysis of the current situation. Multiple sources of data can be combined from different organizational silos to provide a common operating picture that uses current information. Wherever real-time insight has the greatest value, OI solutions can be applied to deliver the information and need.

In network management, systems management, application management and service management, people usually refer instead to event correlation. As CEP engines, event correlation engines (*event correlators*) analyze a mass of events, pinpoint the most significant ones, and trigger actions. However, most of them do not produce new inferred events. Instead, they relate high-level events with low-level events.^[6]

Inference engines, e.g. rule-based reasoning engines typically produce inferred information in artificial intelligence. However, they do not usually produce new information in the form of complex (i.e., inferred) events.

Example

A more systemic example of CEP involves a car, some sensors and various events and reactions. Imagine that a car has several sensors—one that measures tire pressure, one that measures speed, and one that detects if someone sits on a seat or leaves a seat.

In the first situation, the car is moving and the pressure of one of the tires moves from 45 psi (pound per square inch) to 41 psi over 15 minutes. As the pressure in the tire is decreasing, a series of events containing the tire pressure is generated. In addition, a series of events containing the speed of the car is generated. The car's Event Processor may detect a situation whereby a loss of tire pressure over a relatively long period of time results in the creation of the "lossOfTirePressure" event. This new event may trigger a reaction process to note the pressure loss into the car's maintenance log, and alert the driver via the car's portal that the tire pressure has reduced.

In the second situation, the car is moving and the pressure of one of the tires drops from 45 psi to 20 psi in 5 seconds. A different situation is detected—perhaps because the loss of pressure occurred over a shorter period of time, or perhaps because the difference in values between each event were larger than a predefined limit. The different situation results in a new event "blowOutTire" being generated. This new event triggers a different reaction process to immediately alert the driver and to initiate onboard computer routines to assist the driver in bringing the car to a stop without losing control through skidding.

In addition, events that represent detected situations can also be combined with other events in order to detect more complex situations. For example, in the final situation the car was moving normally but suffers a blown tire which results in the car leaving the road and striking a tree and the driver is thrown from the car. A series of different situations are rapidly detected. The combination of "blowOutTire", "zeroSpeed" and "driverLeftSeat" within a very short space of time results in a new situation being detected: "occupantThrownAccident". Even though there is no direct measurement that can determine conclusively that the driver was thrown, or that there was an accident, the combination of events allows the situation to be detected and a new event to be created to signify the detected situation. This is the essence of a complex (or composite) event. It is complex because one cannot directly detect the situation; one has to infer or deduce that the situation has occurred from a combination of other events.

Types

Most CEP solutions and concepts can be classified into two main categories:

1. Aggregation-oriented CEP
2. Detection-oriented CEP

An aggregation-oriented CEP solution is focused on executing on-line algorithms as a response to event data entering the system. A simple example is to continuously calculate an average based in data on the inbound events.

Detection-oriented CEP is focused on detecting combinations of events called events patterns or situations. A simple example of detecting a situation is to look for a specific sequence of events.

Currently many application use hybrid of the two approaches.

Integration with business process management

Of course, rarely does the application of a new technology exist in isolation. A natural fit for CEP has been with business process management, or BPM.^[7] BPM very much focuses on end-to-end business processes, in order to continuously optimize and align for its operational environment.

However, the optimization of a business does not rely solely upon its individual, end-to-end processes. Seemingly disparate processes can affect each other significantly. Consider this scenario: In the aerospace industry, it is good practice to monitor breakdowns of vehicles to look for trends (determine potential weaknesses in manufacturing processes, material, etc.). Another separate process monitors current operational vehicles' life cycles and decommissions them when appropriate. Now one use for CEP is to link these separate processes, so that in the case of when the initial process (breakdown monitoring) discovers a malfunction based on metal fatigue (a significant event) an action can be created to exploit the second process (life cycle) to issue a recall on vehicles using the same batch of metal discovered as faulty in the initial process.

The integration of CEP and BPM must exist at two levels, both at the business awareness level (users must understand the potential holistic benefits of their individual processes) and also at the technological level (there needs to be a method by which CEP can interact with BPM implementation).

Computation-oriented CEP's role can arguably be seen to overlap with Business Rule technology.

For example, customer service centers are using CEP for click-stream analysis and customer experience management. CEP software can factor real-time information about millions of events (clicks or other interactions) per second into business intelligence and other decision-support applications. These "recommendation applications" help agents provide personalized service based on each customer's experience. The CEP application may collect data about what customers on the phone are currently doing, or how they have recently interacted with the company in other various channels, including in-branch, or on the Web via self-service features, instant messaging and email. The application then analyzes the total customer experience and recommends scripts or next steps that guide the agent on the phone, and hopefully keep the customer happy.

Another example of CEP in practice is in the healthcare industry. The HyReminder system, developed by the Worcester Polytechnic Institute and UMass Medical School, continually tracks healthcare workers for hygiene compliance (e.g. sanitizing hands and wearing masks), reminding them to perform hygiene when appropriate to prevent the spread of infectious disease. Each worker wears an RFID badge that displays a green (safe), yellow (warning) or red (violation) light, depending on what behavior the RFID chip has observed.

In financial services

The financial services industry was an early adopter of CEP technology, using complex event processing to structure and contextualize available data so that it could inform trading behavior, specifically algorithmic trading, by identifying opportunities or threats that indicate traders (or automatic trading systems) should buy or sell. Algorithmic trading is already a practice in stock trading. It is estimated that around 60% of Equity trading in the United States is by way of algo trades. CEP is expected to continue to help financial institutions improve their algorithms and be more efficient.

Recent improvements in CEP technologies have made it more affordable, helping smaller firms to create trading algorithms of their own and compete with larger firms. CEP has evolved from an emerging technology to an essential platform of many capital markets. The technology's most consistent growth has been in banking, serving fraud detection, online banking, and multichannel marketing initiatives.

Today, a wide variety of financial applications use CEP, including profit, loss, and risk management systems, order and liquidity analysis, quantitative trading and signal generation systems, and others.

Integration with time series databases

A time series database is a software system that is optimized for the handling of data organized by time. Time series are finite or infinite sequences of data items, where each item has an associated timestamp and the sequence of timestamps is non-decreasing. Elements of a time series are often called ticks. The timestamps are not required to be ascending (merely non-decreasing) because in practice the time resolution of some systems such as financial data sources can be quite low (milliseconds, microseconds or even nanoseconds), so consecutive events may carry equal timestamps.

Time series data provides a historical context to the analysis typically associated with complex event processing. This can apply to any vertical industry such as finance^[8] and cooperatively with other technologies such as BPM as described elsewhere in this document.

Consider the scenario in finance where there is a need to understand historic price volatility to determine statistical thresholds of future price movements. This is helpful for both trade models and transaction cost analysis.

The ideal case for CEP analysis is to view historical time series and real-time streaming data as a single time continuum. What happened yesterday, last week or last month is simply an extension of what is occurring today and what may occur in the future. An example may involve comparing current market volumes to historic volumes, prices and volatility for trade execution logic. Or the need to act upon live market prices may involve comparisons to benchmarks that include sector and index movements, whose intra-day and historic trends gauge volatility and smooth outliers.

References

- [1] D. Luckham, "The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems", Addison-Wesley, 2002.
- [2] O. Etzion and P. Niblett, "Event Processing in Action", Manning Publications, 2010.
- [3] Complex Event Processing for Trading (<http://fixglobal.com/home/secrets-revealed-trading-tools-uncover-hidden-opportunities/>), FIXGlobal, June 2011
- [4] Details of commercial products and use cases (<http://www.complexevents.com>)
- [5] <http://drops.dagstuhl.de/opus/volltexte/2011/2985/Mani> K. Chandy and Opher Etzion and Rainer von Ammon(eds),10201 Executive Summary and Manifesto -- Event Processing, Dagstuhl seminar Proceedings 10201, ISSN 1862-4405, 2011
- [6] J.P. Martin-Flatin, G. Jakobson and L. Lewis, "Event Correlation in Integrated Management: Lessons Learned and Outlook", Journal of Network and Systems Management, Vol. 17, No. 4, December 2007.
- [7] C. Janiesch, M. Matzner and O. Müller: "A Blueprint for Event-Driven Business Activity Management", Lecture Notes in Computer Science, 2011, Volume 6896/2011, 17-28,
- [8] "Time Series in Finance", Retrieved May 16, 2012 (<http://cs.nyu.edu/shasha/papers/jagtalk.html>)

External links

- Complex Event Processing & Real Time Intelligence (<http://www.complexevents.com>)
 - Separating the Wheat from the Chaff (<http://www.rfidjournal.com/article/view/1196>) Article about CEP as applied to RFID, appeared in RFID Journal
 - The Event Processing Technical Society (<http://www.ep-ts.com>)
 - EsperTech (<http://esper.codehaus.org>) - Esper is a component and platform for complex event processing (CEP) for Java for .NET, see (<http://esper.codehaus.org>)
 - EVAM - Event & Action Manager is business oriented CEP solution (<http://www.evam.com>)
 - Complex event processing: still on the launch pad in Computerworld (<http://www.computerworld.com.au/index.php/id;918976660;fp;4;fpid;1398720840>)
 - Processing Flows of Information: From Data Stream to Complex Event Processing (<http://home.dei.polimi.it/margara/papers/survey.pdf>) - Survey article on Data Stream and Complex Event Processing Systems
 - Real-Time Business Insight - Event Processing in Practice (<http://www.rtinsights.com>) Online Magazine
 - The ACM conference on distributed event-based systems (<http://www.debs.org>) - Research articles on innovations in event-based systems
 - Big Data For Finance (<http://www.bigdataforfinance.com/bigdata/2012/05/what-is-time-series.html>) - CEP pattern matching and historical time-series creates relationships to explain future developments
-

Article Sources and Contributors

Complex event processing *Source:* <http://en.wikipedia.org/w/index.php?oldid=586977477> *Contributors:* ACalleM, Aaron Kauppi, Amniarix, Anandnalya, AndrewHowse, Andy Flury, Andy Payne, Anthony Appleyard, Apuneet, AtticusX, Babkin, Bartschouw, Betacommand, Bharath r, Bkonrad, Bonhomie, Cander0000, Charleyrich, Chrismartins, CitizenB, Compfreak7, Darp-a-parp, Decoy, DhruvaB, Differo, Dougher, Download, Eboily, Echidnaslumber, Elonka, EoGuy, Erianna, Euchiasmus, Eudymon, Eugene-Elgato, Expertease, Ferran.cabrer, Flameviper, Fritzibunwalla, Fullings, Geeker87, Giant fan, Gsaurer, Harshana05, Hemphilldavid, HighKing, Hu12, Ihalilaltun, Isvana, J. J. F. Nau, JP.Martin-Flatin, Japinder, JeffAdkins, Jmcenaney, Joedoe, Joel7687, Jtq4u, Julianclark, KarlHeg, Katharineamy, Kevin S Pool, Kirilian, Kku, Leo gan 57, Link1983Link, LouisLovas, MC Wapiti, Mark Renier, Mefisk, Mindmatrix, Modify, Mofojackson, Mohit.thatte, Mrkwpalmer, Muloem, Neilmegovern, Nixeagle, Northorangebear, Ozge barbaros, PhnomPencil, Pratyeka, Priyankpiyush, Radagast83, Raymondwin, RichardVeryard, Rjwilmsi, Ronnotel, Sandeepbishnoi.wiki, Sergeman, Shalin420, Sjkim1971, Slade8181, Smartse, Tayloj, Tentinator, The Thing That Should Not Be, Themfromspace, Theodolite, Tikiwont, Tvorozil, Winterst, Wireless friend, Wmahan, Ycbrad, Ygokirmak, 216 anonymous edits

License

Creative Commons Attribution-Share Alike 3.0
[//creativecommons.org/licenses/by-sa/3.0/](http://creativecommons.org/licenses/by-sa/3.0/)