

EXCERPT

Complex Event Processing Opportunity Analysis and Assessment of Key Products (Excerpt from IDC #215785)

Maureen Fleming

Jeff Silverstein

IN THIS EXCERPT

The content for this excerpt was taken directly from the IDC Customer Needs and Strategies Report, Complex Event Processing Opportunity Analysis and Assessment of Key Products, by Maureen Fleming and Jeff Silverstein (Doc # 215785). All or part of the following sections are included in this excerpt: IDC Opinion, Situation Overview, and Vendor Profile. Also included is Table 1 and Figure 2 & 16.

IDC OPINION

Most applications depend on users to ask an application to perform a task. With the widespread collapse of our economy, it is clear that this approach is no longer valid for many types of mission-critical applications that involve managing risk and for important areas of customer service. The new paradigm requires an event-driven approach that tells users when they need to take action. Complex event processing (CEP) software will take on the application server role for these new event-driven systems. Other findings include:

- ☒ We estimate that CEP software will grow 65% in 2008 to approximately \$140 million, making CEP the fastest-growing segment of middleware for the second year in a row.
- ☒ The small size of this market disguises its growing importance as the anchor product in multimillion-dollar event-driven infrastructure deals.
- ☒ Depending on the application requirements and the role and skills of the most active group of users, the importance of certain capabilities over others drives the selection process. There is no single best-of-breed CEP product in every important area. To one buyer, a capability may be mission critical, while to another buyer, that same capability may be viewed as needlessly costly and complex.
- ☒ While CEP is covered most heavily in relationship to financial trading systems, this covered only 30% of total spending in 2007 and 2008. There is also significant coverage of stream processing vendors, yet enterprises spent significantly more on rules- and model-oriented CEP engines in 2007 and 2008.

SITUATION OVERVIEW

CEP: The Brains of an Event-Driven System

In many ways, a CEP engine can be likened to the brains of an event-driven system. It receives messages from an MOM, relates the new message to what is already in its term memory, quickly identifies problems and opportunities, and sends instructions to other systems to take action.

In the mid-1990s, academic research began at Cal Tech (Mani Chandy), Cambridge University (John Bates), and Stanford University (David Luckham) in the field of event processing, with the intent of developing a new approach to real-time identification of sequences of related events.

At the time, systems were capable of processing a single event, such as a stock price, but there was no automatic way to rapidly relate that event to others without extensive coding. The larger the code, the more challenging it was to make changes. Those groups that invested in the code were left with brittle, nondynamic systems.

This problem resulted in research and product development to build engines that could receive, normalize, and correlate streams of data against models of conditions — or predefined event patterns — that needed monitoring.

In a 2002 book by Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*, the concept of an "accessible information gap" was described as "the difference between the level at which we can monitor activities in a system and the level of activities we want to know about."

From that perspective, CEP closes the gap between low-level system activities and higher-level information to which people and systems can respond. As the brains of an event-driven infrastructure, CEP is equipped with the ability to detect patterns and to send instructions to other systems to effect appropriate rapid responses.

Key CEP Definitions

For purposes of this study, we are adopting the glossary and definitions developed by the Event Processing Technical Society (www.complexevents.com/?p=361) and will try to adhere to it as much as possible as we compare different vendor offerings. While there are many terms in the glossary, the terms and explanations needed for this study are as follows:

- ☒ **Complex event:** A complex event is an event that is an abstraction of other events called its members. In thinking about this from the perspective of using a CEP engine, a complex event is the discovery of a fact derived by correlating multiple events and applying rules to that set of events
- ☒ **Event pattern:** An event pattern is, in essence, the model that describes a complex event. Event patterns contain a template that describes the complex event, its rules, and variables as well as relational operators. An event pattern matches sets of related events by replacing variables with values. We believe

that the term "condition" is synonymous with event pattern as is "complex event model," and use these terms interchangeably in this study.

- ☒ **Event pattern instance:** Event pattern instance is the state of an event pattern where some of the pattern member variables have been replaced by values. This can also be thought of as the state of an event pattern. Each event pattern can have "n" number of event pattern instances.
- ☒ **Event channel:** An event channel is a conduit in which events are transmitted from event sources (emitters) to event sinks (consumers). An event channel may be message-oriented middleware that carries the event from source to consumer, along with adapters that translate the event into a form consumable by the processing engine.

Differences in CEP Capabilities

Event-driven applications are useful in a broad spectrum of areas. As the brains of an event-driven application, CEP usage is very broad. The day-to-day use of CEP also varies. In some cases, the users are IT groups and in others, the users are non-IT professionals who are subject matter experts.

Depending on the application requirements and the role and skills of the most active group of users, the importance of certain capabilities over others drives the selection process. The following sections list the five core capabilities of a CEP engine, keeping in mind that buyers view these capabilities in a spectrum. To one buyer, a capability may be mission critical, while to another buyer, that same capability may be viewed as needlessly costly and complex.

Correlation Sophistication

There are many approaches to describing a pattern that triggers a complex event. Regardless of whether the condition is described in Java, in SQL, in a model, or as a sequence of rules, there are basic capabilities that need to be present as well as value-added capabilities that buyers should look for, depending on their use case.

At a basic level, a CEP engine should be able to allow a user to describe the following:

- ☒ A particular pattern that consists of a sequence of expected events or the absence of an expected event
- ☒ How the condition should be updated by a new event, including the ability to apply mathematical operations to the event and apply rules to it
- ☒ Rules to test if the condition is matched to determine whether a complex event should be created and fired off

A simple example is in trading, where a new price for a security is received by a CEP engine. Each of the conditions that subscribe to the "Security N" topic then performs an update and tests whether there is a match. If so, a complex event is triggered and

may be published out to an MOM or receiving system or used in some other part of the CEP engine.

More complicated examples are found in financial middle and back offices, transportation, logistics, warfare, and promotions areas. In these cases, an event — such as the arrival of a particular flight — will have several conditions that are being monitored and many dependencies. Similar to the change in price event, a new event may need to update an existing data field or object.

For example, a flight departure event will change the state of the flight status. This may trigger instructions to recalculate predicted arrival time. If the calculation indicates that the flight will be late, this event will be published out and trigger calculations for many other conditions, such as baggage handling scheduling, crew scheduling, and gate scheduling. In other words, there is a chain reaction that has to be managed within the CEP engine.

While it is fairly straightforward to handle a single flight and that flight's chain reaction, an airport or airline has to handle that particular late flight, all of the impacts on the late flight, plus any other late flight and those impacts. That means the design has to account for a fairly large system and system impacts.

The type of CEP solution that handles these complex scenarios must be able to do so where modelers and developers don't get lost in the code and where the process of adapting the conditions and system of conditions can quickly identify what needs to be changed.

Types of Matches

It is a given that any CEP solution can monitor groups of events over a given period of time where there are only a few members of the group. A condition should be able to model or describe a match or nonmatch pattern, as illustrated in Figure 2.

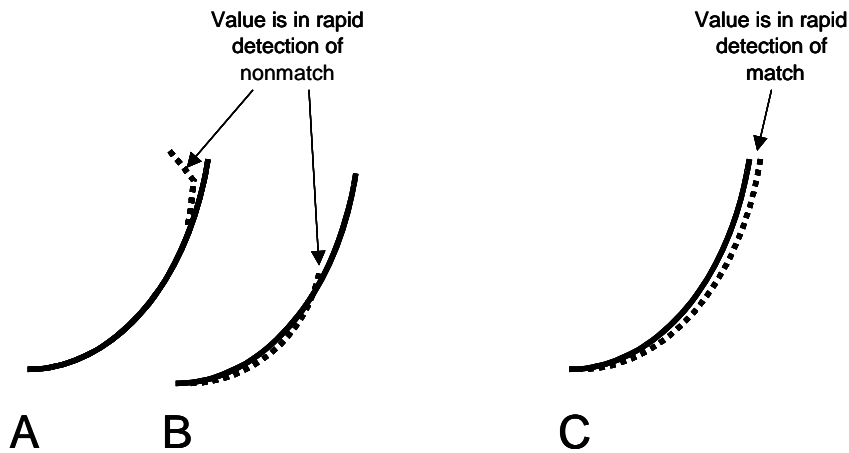
In match type A, for example, a condition would describe a normal sequence of events and monitor for abnormalities. In trading, the sequence of events might be normal expectations for a group of securities, with any deviation triggering the publication of a new event.

In match type B, again, a normal sequence is described, with a trigger when the pattern stops before completing within a certain amount of time. This is the type of monitoring IT uses for systems performance monitoring.

In model type C, the CEP engine is looking for a match, and when found, a new event is published. A use case for this may be in a warfare scenario that describes what a missile attack pattern looks like. The match of the pattern triggers a new event.

FIGURE 2

Complex Event Detection



Source: IDC, 2008

To satisfy an enterprise's requirements for condition modeling, a CEP engine should be able to:

- Describe a particular pattern that consists of a sequence of occurring — or not occurring — events, with the ability to handle an arbitrary number of events within the pattern without sacrificing performance of the system
- Apply mathematical operations to an event, a set of events, or an event pattern
- Apply rules to an event, a set of events, or an event pattern
- Monitor groups of events over long periods of time
- Allow the output of one condition model to be used as input for another
- Handle spatial logic
- Look up data by querying another system
- Create conditions (which include the events in the pattern, operations, and rules) and easily relate those conditions to others
- Visualize conditions and relationship of conditions to others
- Automatically assess impact of change to one or multiple conditions
- Simulate a condition or sets of conditions

Business User Control

CEP tools are classified as middleware, which is typically sold to developers. However, in some cases, the involvement of lines of business in CEP tool selection is following a pattern that more closely resembles business process management (BPM) suites.

In a BPM suite, the line of business is frequently involved with the tool selection and has a strong day-to-day role in improving process performance. Products are increasingly selected based on how easy it is to model a process, how quickly a process activity can be reconfigured, or how easily a transaction route or process flow can be changed, particularly without IT involvement. In other words, the ability of a BPM suite to demonstrate alignment with business cycle times is becoming an important selection point.

We have also found this to be true for many CEP use cases, with similar separations between IT and business users. IT is concerned with implementing the CEP solution and monitoring system performance. The changes IT is involved with are around the addition of events from new channels, system upgrades, and the development of other application-specific capabilities that are outside the domain of determining what the condition models are.

Business is concerned with the conditions, testing them, and putting them into production. Features supporting business user control include:

- Ability for trained business users to describe conditions and make changes to them (That means they need an interface that requires training, but not programming, ability.)
- Changes to the system affecting agility that are made quickly and cleanly by a developer (That means the developer is able to use a tool that rapidly identifies where the change must be made and does a dependency analysis to determine what will be affected by the change.)
- Ability for business users to change policies or policy thresholds that are applied to the conditions
- At a system level, changes to a condition model that do not require recompiling code or any type of IT-driven change management cycle
- Rapid testing to ensure the creation and change of logic doesn't introduce errors
- Ability to test a new condition model or scenario in a production environment and compare results of the test against the baseline model

High-Performance Data Management

There are variations in requirements around data management. At the most complex, the system must be able to manage a large number of active condition instances, where the condition model is long lived and where there may be multiple conditions and events in the model. Certain types of real-time risk management, real-time multichannel marketing, logistics, and fraud detection are examples that fit the high end of this category.

To guarantee low-latency performance, active data must be managed in memory. Applications that have a large number of instances and large volumes of data and require low-latency processing speeds and reliability in terms of capturing all of the data of every transaction are likely to require grid-based data caches to achieve desired results.

Low Latency

Low latency is measured by:

- The speed by which incoming events can be correlated, matched, and updated and a new event published out
- The volume of event streams that can be processed through the engine in a given amount of time

All of the CEP engines claim they are the fastest, but latency is a function of the complexity of the conditions that drive the use of the engine, how the event data is stored, and the core capabilities of the runtime engine.

Organizations that are planning to purchase CEP software may find that this is an area where the use case will determine the performance. A product that works really well in rapidly and efficiently processing high volumes of transactions in short time windows with simple condition models may not work at all in a use case that requires highly complex condition models in long-lived time windows.

Deterministic Behavior

Performance must be predictable and repeatable. The system must be able to produce output that is repeatable (i.e., the same sequences of events will always produce the exact same results).

Comparison of CEP Products

At this point, most of the CEP products are fairly complete in that they have a development environment, runtime, data caching, and so forth. However, there is no one CEP product that is best of breed across all of the five areas of CEP capabilities. Over time, we expect to see products mature to the point where capabilities are sophisticated across the full spectrum.

We have seen a good amount of work occur over the past several months on user interface and overall usability.

In addition, we are also seeing an increasing number of vendors offer event-driven applications built on their respective CEP engines. Many of the vendors have announced CEP applications, solutions, or frameworks.

Table 1 offers our assessment of the strengths of individual CEP products across each of the five categories of correlation sophistication (listed below as condition complexity), business user control, high-performance data management (listed below as data management), low latency, and deterministic behavior.

TABLE 1					
Comparison of Complex Event Processing Product Capabilities					
	Condition Complexity	Business Control	Data Management	Low Latency	Deterministic
Progress Apama	4.0	5.0	3.5	5.0	5.0

Source: IDC, 2008

* Table 1 include only vendor Progress Apama

The higher the score, the more a product is optimized toward that specific CEP capability. We are not totaling scores because that would not select the best product. It would be a mistake, for example, for a buyer to short-list two products that have opposite strengths simply because they received the two highest scores.

Vendor Profile

Progress Apama

In April 2005, Progress entered into the CEP market with the acquisition of British start-up Apama. In 2007, Progress Apama was the second-largest CEP vendor, growing 36% to an estimated \$17 million in revenue. Revenue is slightly understated relative to number of customers because Apama is frequently sold under a term license in trading markets. We expect Progress Apama to grow in low triple digits in 2008 and maintain second position.

Apama is a highly sophisticated CEP offering, with high scores in all categories. Apama was early to recognize the importance of a non-IT professional development environment. As a result, the product is both technologically strong as well as useable by trained business users.

John Bates, who cofounded Apama and was a pioneer in complex event processing, manages this business for Progress. The other cofounder, Giles Nelson, also remained with Progress and is director of technology for EMEA operations and remains actively involved with Apama.

CEP Strategy

Progress has a multipronged CEP strategy:

- ☒ Focus on its primary financial markets with commercial CEP trading applications and solution accelerators to trading groups that, with training, will be able to create, adapt, and deploy event patterns without the involvement of developers
- ☒ Sell regulatory and compliance solutions in cooperation with partners (For example, Apama is used by Turquoise for compliance and the Financial Services Authority, based in the United Kingdom, to spot insider trading patterns. Both of these relationships were sold through Progress' relationship with Detica.)
- ☒ Continue offering a robust generalized CEP platform that can be sold in all vertical markets as a platform for event-driven applications
- ☒ Incorporate Apama as a component of a larger event-driven architecture in conjunction with other Progress products in the Progress SOA portfolio, or sell into other SOA architectures

Products

Apama Event Processing Platform

Apama is a model-driven, object-oriented CEP engine. It has a strong user base among non-IT professionals, who use the modeling capability to build and adapt event patterns.

In spring 2008, Progress introduced Apama 4.0, which unified all of the subcomponents of the offering into a single, installable version for simple deployment. It also significantly increased throughput and reduced latency.

Apama Studio

The Apama Studio consists of an EPL that can be used to build a complete CEP application or used by developers to create objects, called SmartBlocks, that are then used within the modeling environment to develop separate event patterns.

Apama's EPL also has a Java interface. In addition, Studio has a modeling environment for nondevelopers to assemble event patterns.

Dashboard Studio is the user interface development environment for an event-driven application. It is based on an OEM license from SL Corporation's RTView.

Research Studio is used to simulate new event patterns or trading strategies against select time periods.

Apama Correlator

This is the Apama CEP execution environment.

EventStore

EventStore maintains an event history for auditing and use in Research Studio. EventStore is a time series database based on Progress' ObjectStore.

Other

Apama supports a broad range of adapters covering the capital markets in addition to common messaging middleware.

Capital Markets Framework

Built on the Apama platform, the Capital Markets Framework was developed by a dedicated financial markets engineering team to lower the implementation cost of using CEP for trading and market surveillance. It consists of several components that can be used out of the box or customized to specific trading environments. The framework consists of:

- ☒ Prepackaged algorithms for use as-is or for customization in proprietary strategies
- ☒ Backtesting to evaluate new strategies and analyze existing ones
- ☒ An array of packaged connectivity adapters with tools for extension to customer-specific environments
- ☒ Solution accelerators, including smart order routing, market surveillance, and real-time pricing
- ☒ Risk Firewall, which includes prepackaged patterns for common trading mistakes in addition to the ability to set trading risk policies and ensure that trades don't violate those policies (A firewall, in general, is a hardware or software solution to enforce security policies. With Apama's Risk Firewall, the policies are set in design time, and all trades run through the Risk Firewall to ensure compliance.)

Customers and Partners

Customers include ABN Amro, Aspect Capital, Bank of China, Bank of Montreal, BGN, Deutsche Bank, Finamex, Financial Services Authority, ING Wholesale Banking, JPMorgan, Leading Investment and Security (Korea), NYSE Euronext, Petro Canada, Premier Manufacturing, SEB, and Turquoise.

Partners include Detica, Dow Jones, Fxall, Koscom, Manuvis, and Ullink.

Strengths and Challenges

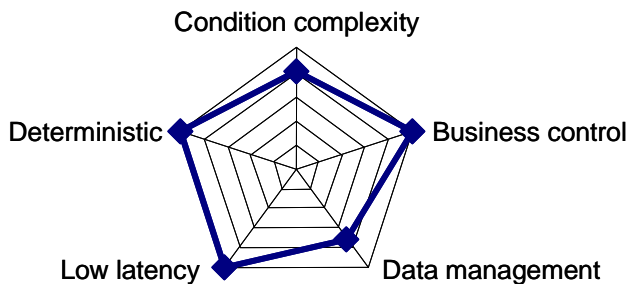
Strengths

- ☒ Apama's design and packaged approach will play well in the current financial market, where they need to continue trading, and trading effectively, but with diminished IT support staff.

- ☒ Apama's approach also makes sense where there is a need for trading sophistication in smaller firms, such as hedge funds, that do not have an appetite for large IT staffs to support highly customized projects.
- ☒ Apama has been successful selling into financial surveillance markets, with deals including Turquoise and FSA. In the later stage of the forecast period covered in this study, we are likely to see regulations emerge around better financial risk management. Whether customers preemptively choose to implement event-driven risk systems ahead of the regulations or following them, Apama is a logical choice.
- ☒ Progress has a strong partner infrastructure and expertise in going to market with partners. As event-driven systems and applications grow in importance, we expect to see partners from nonfinancial markets leverage the Apama platform as a key building block. (Figure 16 aligns Apama's capabilities with CEP requirements. There are few usage scenarios that are not supported by Apama capabilities.)

FIGURE 16

Alignment of Progress Apama with Complex Event Processing Capabilities



Source: IDC, September 2008

Challenges

- ☒ Because Apama is so closely identified with the financial markets, the current downturn is likely to negatively impact Apama's opportunity and growth prospects in the near term. Therefore, it is incumbent that Progress figure out how to cost effectively apply this technology to new markets with better short-term growth prospects.

Market Leaders in Innovation

Risk Firewalls

In 2008, when Apama introduced its Capital Markets Framework, it introduced a Risk Firewall, which includes prepackaged patterns for common trading mistakes in addition to the ability to set trading risk policies and ensure that trades don't violate

those policies. A firewall, in general, is a hardware or software solution to enforce security policies. With Apama's Risk Firewall, the policies are set in design time, and all trades run through the Risk Firewall to ensure compliance.

Given the strong demand in multiple markets for improved governance around risk, we believe the notion of a risk firewall is critical. From that perspective, Apama has one of the stronger points of innovation in CEP that will be increasingly important in 2009 and beyond.

Copyright Notice

This IDC research document was published as part of an IDC continuous intelligence service, providing written research, analyst interactions, telebriefings, and conferences. Visit www.idc.com to learn more about IDC subscription and consulting services. To view a list of IDC offices worldwide, visit www.idc.com/offices. Please contact the IDC Hotline at 800.343.4952, ext. 7988 (or +1.508.988.7988) or sales@idc.com for information on applying the price of this document toward the purchase of an IDC service or for information on additional copies or Web rights.

Copyright 2009 IDC. Reproduction is forbidden unless authorized. All rights reserved.