Professional Expertise Distilled

Getting Started with Oracle Event Processing 11*g*

Create and develop real-world scenario Oracle CEP applications

Alexandre Alves
Robin J. Smith
Lloyd Williams

[PACKT] PUBLISHING
enterprise
professional expertise distilled

Getting Started with Oracle Event Processing 11*g*

Create and develop real-world scenario Oracle CEP applications

Alexandre Alves

Robin J. Smith

Lloyd Williams

[PACKT] PUBLISHING enterprise professional expertise distilled

BIRMINGHAM - MUMBAI

Getting Started with Oracle Event Processing 11*g*

Copyright © 2013 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the authors, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: March 2013

Production Reference: 1150313

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-84968-454-5

www.packtpub.com

Cover Image by Artie Ng (artherng@yahoo.com.au)

Credits

Authors

Alexandre Alves

Robin J. Smith

Lloyd Williams

Reviewers

Jeffrey A. Myers, Ph.D.

Ahmet Fuat Sungur

Prakash Jeya Prakash

Acquisition Editor

Grant Mizen

Lead Technical Editor

Dayan Hyames

Technical Editors

Vrinda Amberkar

Dominic Pereira

Project Coordinator

Leena Purkait

Proofreader

Samantha Lyon

Indexer

Hemangini Bari

Graphics

Sheetal Aute

Valentina D'Silva

Aditi Gajjar

Production Coordinators

Nitesh Thakur

Prachali Bhiwandkar

Cover Work

Nitesh Thakur

Prachali Bhiwandkar

About the Authors

Alexandre Alves has over 12 years of experience in software development working for large companies, such as IBM and Oracle. He has worked with network management, CORBA, JEE, web services, OSGi, BPEL, CEP, and middleware technologies in general. He is the co-author of the WS-BPEL 2.0 specification, co-author of BPEL for Java specification, author of the *OSGi in Depth* book, and a steering committee member of the Event Processing Technical Society (EPTS).

I would like to thank my family for giving me the support I needed to continue my work regardless of all other problems that life throws at you. I would like to thank my sons, Gabriel and Lucas, for providing for the fun-filled book-writing breaks, and understanding when I was in the book-writing, no-breaks (as they saw it) mode. I would like to especially thank Juliana, my wife-to-be, for her unyielding support, her caring, and especially for her lifelong understanding. For you, all is worth. Words put into a book are everlasting, so is our love.

Finally, I would like to thank my excellent co-authors and colleagues at Oracle for giving me the material and the experience I needed for writing this book.

Robin J. Smith, as a Product Management/Strategy Director at Oracle Corporation, is responsible for the Event Driven Architecture and Complex Event Processing technologies, focused on the evolution and delivery of the award winning and innovative Oracle Event Processing product, a corner-stone technology of the Oracle Event Driven Architecture strategy. Previously at BEA Systems, he successfully delivered the BEA WebLogic Event Server, the industry's first and only EDA CEP Java Application Server based on an exposed customized OSGi™ framework. At Sun Microsystems, as a software Product Line Manager for 8 years, he focused on the product management and marketing for the core SOA technologies, Netscape Process Manager and the award-winning Sun Java™ Studio Enterprise, a visual development and infrastructure environment focused on SOA, UML design tools and Java application profiling techniques. Over his career, Robin has worked in all of the major computing domains acquiring expertise as an architect for a leading Universal Content Management System and designed, engineered and implemented unique performance and systems management software for the Java Platform, AS/400, and VM Operating systems that have been used worldwide.

My deepest thanks to Phil Wilmshurst, who after a chat in the Bowlers Arms in Margate recommended me for my first computing job, starting a career at a young age which has now taken me around the world and to my computing successes in Silicon Valley, California. To Mike Leamer, who as a friend and manager motivated me to learn more and guided me to excel in my programming efforts in London. To the team at VM Software Inc., who gave me my "Famous for Fifteen Minutes" time when they purchased my unique VMMonitor product and finally, my family that inspires me to leap out of bed each morning and enjoy my continuing computing days of adventure, at my office in Redwood Shores, just south of the beautiful San Francisco.

Lloyd Williams has over 17 years of experience in the software development and IT industry. Lloyd graduated from Memorial University of Newfoundland in 1995 with a Bachelor of Commerce (Honors) specializing in Management Information Systems and Operations Management. He then moved to California to start consulting in the telecommunications industry. Since then, he has worked with numerous Fortune 500 companies around the globe in every industry. Lloyd's experience ranges from large telecommunications and automotive projects working with global systems integrators to leading the development of small event-driven RFID solutions at a small start-up.

He is currently an outbound product manager working for Oracle within the Business Integration team of the Oracle Fusion Middleware product family. He works with customers around the globe developing solutions that integrate Oracle Event Processing with SOA and BPM solutions.

I would like to thank my friends and family for their support, patience and help in producing this book as well as during many late nights and weekends working on many software development projects. I would like to thank my managers throughout the years who have provided me with opportunities to learn new skills and take on challenging tasks, as well as many clients and colleagues whom have provided invaluable opportunities for me to expand my knowledge and shape my career.

About the Reviewers

Jeffrey Myers holds a Ph.D. in Physics from the University of Michigan, where he studied energy transfer mechanisms in proteins and developed new experimental techniques in ultrafast optics. He has over 10 years of experience in experimental design, algorithm development, and data analysis. In his professional career, he has utilized relational databases and complex event processing to provide innovative analytic solutions. Dr. Myers currently works as an engineer with Northrop Grumman. His technical interests include pattern recognition, machine learning, sensors, and Big Data analytics.

Ahmet Fuat Sungur has 6 years of experience in working with Oracle products. Since 2008 he has been working in Telecommunication Industry. In his professional career, data processing technologies are his favorite subjects. He participated in several business intelligence-oriented applications, which was developed by using Java and Oracle technologies. Software architecture, distributed processing, Big Data and NoSQL databases are his other main interests. He has attended many national and international technical congresses as a speaker.

He is currently working for Turkcell, which is the biggest telecommunication company in Turkey, third in Europe. Also he holds a degree in computer engineering.

Prakash Jeya Prakash is an Oracle Certified SOA Expert and SOASchools certified SOA professional.

He started his career as a Java developer with TechMahindra and after a couple of years his career shift towards SOA started. Since then he has been working on the Oracle middleware stack. From 2008 to 2010, he worked as Tech Lead for BSS productized solution development at Nokia Siemens Networks, Bangalore, India. In July, 2010, he moved to UK and started his own company as a freelancer SOA consultant. Since October, 2011, he has been working as a Lead SOA consultant at Logica, UK.

www.PacktPub.com

Support files, eBooks, discount offers and more

You might want to visit `www.PacktPub.com` for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at `www.PacktPub.com` and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at `service@packtpub.com` for more details.

At `www.PacktPub.com`, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.

`http://PacktLib.PacktPub.com`

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- Fully searchable across every book published by Packt
- Copy and paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at `www.PacktPub.com`, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Instant Updates on New Packt Books

Get notified! Find out when new books are published by following `@PacktEnterprise` on Twitter, or the *Packt Enterprise* Facebook page.

Table of Contents

Preface

Events are everywhere. Events can have either positive or negative impacts on our lives and affect important business decisions. These events can impact a company's success, failure, and profitability.

Getting Started with Oracle Event Processing 11g will allow you to be benefited from the skills and years of experience from the original pioneers who were the driving force behind this immensely flexible, complete, and award-winning Event Stream Processing technology. It provides all of the information needed to rapidly deliver and understand Event Driven Architecture (EDA) applications.

After an introduction to the benefits and uses of Event Stream Processing, this book uses tutorials and practical examples to teach you how to create valuable and rewarding event-driven foundational applications. This book will provide a unique perspective on product creation, evolution, and a solid understanding of how to effectively use the product.

What this book covers

Chapter 1, *An Overview of Complex Event Processing*, provides an overview of the event processing technology, including the event processing language, the event processing network, and event-driven architectures.

Chapter 2, *An Overview of Oracle Event Processing*, provides an overview of the Oracle Event Processing, including the Eclipse-based design time, the management console, and other tools.

Chapter 3, *Adapting Events for OEP*, describes how to adapt external events into an OEP event, and how to convert back OEP events into external events through the use of the adapter SDK.

Chapter 4, *Assembling and Configuring OEP Applications*, describes how to assemble an event processing network together as an OEP application and how to configure its components.

Chapter 5, *Coding with CQL,* describes Oracle's event processing language, called CQL, and how it can be used to filter events, correlate events, aggregate events, and perform several other event processing tasks.

Chapter 6, *Managing and Monitoring Applications,* teaches you to perform management and monitoring tasks, such as deploying OEP applications, configuring work-managers, and using the logging service.

Chapter 7, *Using Tables and Caches for Contextual Data,* explains how to use data residing in tables and caches as contextual data when processing events.

Chapter 8, *Pattern Matching with CQL,* teaches you to pattern match events using CQL, a very powerful feature that can be used to find missing events, and other complex patterns.

Chapter 9, *Implementing Performance Scaling, Concurrency, and High Availability for Oracle Event Processing,* explores several mechanisms to improve performance of OEP applications and how to set up a OEP cluster supporting high availability.

Chapter 10, *Introducing Spatial: A Telemetric Use Case,* walks you through a real-world event processing case study, which makes extensive use of spatial features and telemetric.

Chapter 11, *Extending CQL with Spatial and JDBC,* teaches you to make use of geometry types in CQL using the Spatial cartridge, and how to invoke arbitrary SQL using the JDBC cartridge.

Chapter 12, *Looking Ahead: The Future of Oracle Event Processing*, takes a candid look at the future of event processing, including emerging topics such as event processing in Big Data, machine-to-machine architectures, and event intelligence.

What you need for this book

To make full use of this book, you need to install Oracle Event Processing 11*g*, which is available at Oracle Technology Network website, `http://www.oracle.com/technetwork/middleware/complex-event-processing/overview/index.html`. Select the 11*g* version, as this book is targeted toward this particular version.

Some examples make use of the Oracle Database 11*g* Release 2, which likewise can be found at `http://www.oracle.com/technetwork/database/enterprise-edition/overview/index.html`.

Who this book is for

This book is aimed for both developers as well as architects that need to learn about event processing, stream processing, and the event-driven architecture. Having some background knowledge of Java and SQL will help, but is not a must.

Conventions

In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.

Code words in text are shown as follows: "By using this method, you can define event types as a Java bean, `java.util.Map`, or tuple."

A block of code is set as follows:

```
<event-type-repository>
    <event-type name="Customer">
        <property name="name" type="char"/>
        <property name="address" type="Address"/>
    </event-type>
    <event-type name="Address">
        <class-name>postal.Address</class-name>
    </event-type>
<event-type-repository>
```

Any command-line input or output is written as follows:

```
com.bea.wlevs.adapters.jms;version="11.1.1.7_0",
com.bea.wlevs.adapters.jms.api;version="11.1.1.7_0",
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "From within the **EPN Editor** screen, right-click and select **New** and then **Adapter**".

Warnings or important notes appear in a box like this.

Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.

To send us general feedback, simply send an e-mail to `feedback@packtpub.com`, and mention the book title via the subject of your message.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, see our author guide on `www.packtpub.com/authors`.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

Downloading the example code

You can download the example code files for all Packt books you have purchased from your account at `http://www.PacktPub.com`. If you purchased this book elsewhere, you can visit `http://www.PacktPub.com/support` and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting `http://www.packtpub.com/support`, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from `http://www.packtpub.com/support`.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at `copyright@packtpub.com` with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at `questions@packtpub.com` if you are having a problem with any aspect of the book, and we will do our best to address it.

1
An Overview of Complex Event Processing

In this chapter, you will be introduced to the basic concepts of **Complex Event Processing** (**CEP**), its impact today on businesses across all industries, and the key artifacts that together constitute an **Event-Driven Solution Platform**. Some of the topics we will cover are as follows:

- What is event processing
- Relating this to a business in computing terms
- Use case: A solution for customer problems
- Key elements of event stream processing
- Event processing languages and extensibility
- Holistic event-driven and service-orientated architectures
- Predicting an event

What is event processing?

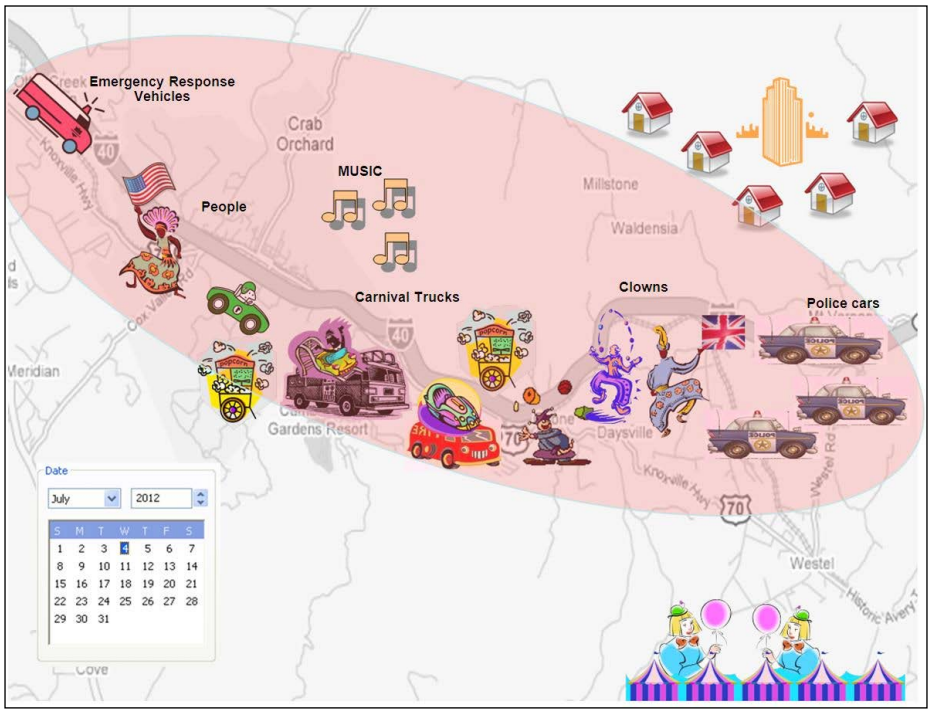
In the world around us, every second of every minute of every hour, the human brain is bombarded with a limitless number of things that happen either at the same time or sequentially, or in a totally and seemingly erratic way that may not make sense immediately but as more of these things happen, we can start to understand their relevance and importance.

For example, we hear cheering in the distance, we see balloons flying in the air, music starts to play, police cars and trucks appear pulling brightly covered trailers with puppets and people waving on them, followed by ambulances, and today's date is July 4th. Individually, these events could mean anything, but together? It's probably an Independence Day Carnival Parade!

Our brain can easily determine this fact in the blink of an eye" and while not overly simple to define in computing terms, we could describe a "Parade Event Pattern" as follows:

One (or more) police cars + followed/preceded by, or adjacent to + one (or more) carnival trucks + followed/preceded by, or adjacent to + one (or more waving people) + followed/preceded by, or adjacent to + one (or more emergency vehicles) + where music can be heard + and today's date is 4th July

Your brain is not restricted to sending information and just waiting until there is a response, or forced into following a series of fixed steps to get something done. As with this example, it is able to take the events happening now, their relevance to additional external factors such as today's anniversary date and understand a "parade" event pattern.



So as you learn more about Complex Event Processing, we focus on how this technology can take continuously flowing, never-ending information, from a potentially unlimited number of different places, and immediately understand how it relates to things happening right now and in the very near future, commonly known as **Real-Time Situation Awareness**.

Relating this to a business in computing terms

The problem now in the world of computers is the proliferation of data. Information arrives from many different systems, in vast quantities, at different times, at different speeds, some of importance now to certain other systems, people or processes, and some stored for later recovery and determination. Why the proliferation now?

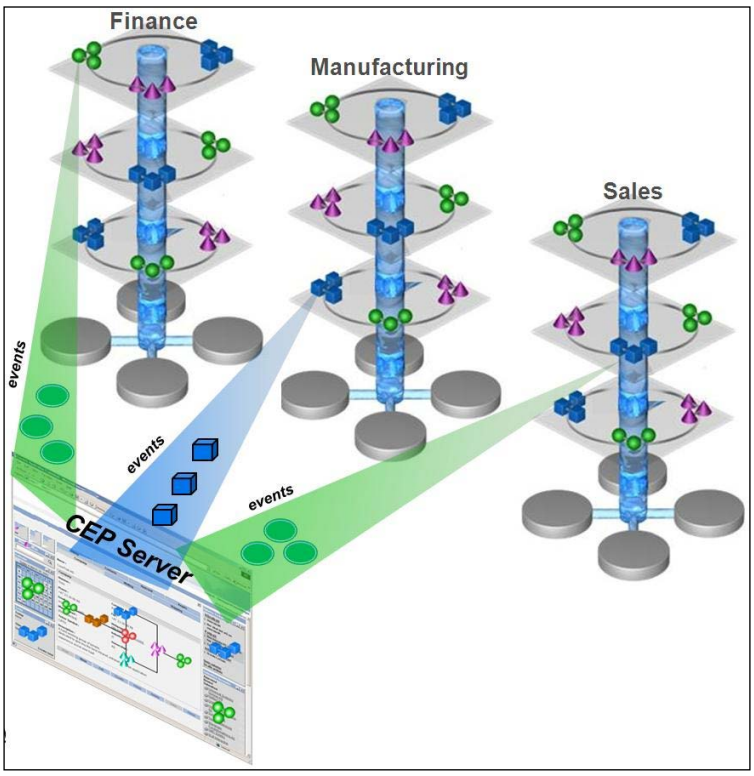
There are many issues involved, but here are just a few major ones:

- The cost of computer power and sophisticated environmental sensor devices has become less expensive
- Networking capacities increase and become more intelligent
- The many different functional computing silos (finance systems, manufacturing systems, sales systems, and so on) are broken down, rewritten, enabling processes that can span more and more business demands
- New computer solution demands expand beyond the enterprise to include partners, customers so more and more data sources and other inputs are brought online
- Computing technology architectures such as **Service Orientated Architecture** (**SOA**) becomes increasingly successful, resulting in an ever more elaborate ecosystem of re-usable services
- A **Big Data** explosion, a term now used widely for information that arrives in high volumes, with extreme velocity, and in a wide variety of mostly unstructured formats emanating from social media sites, cell phones, and many other sources
- A growing demand from businesses that expect their Information Technology (IT) teams to respond to market situations much more effectively in real time

As we evolve and the complexity of these systems "pour" more and more huge volumes of information at computer applications, we are reaching a "tipping point" where traditional point-to-point or request-reply-based solutions of the world break down and become unmaintainable and not extendable.

A company business can be influenced instantaneously from things (events) that can happen, not only in the "cozy" understandable world within its own environment but also from activities (events) from beyond, such as from "the Internet of things"—real-time sensor device that can measure and report on a multitude of situations, including "the impending danger from a sudden rise in temperature in a food storage facility" or "the global positioning system location of a shipping container which is having an unauthorized opening with movement detection sensed from within".

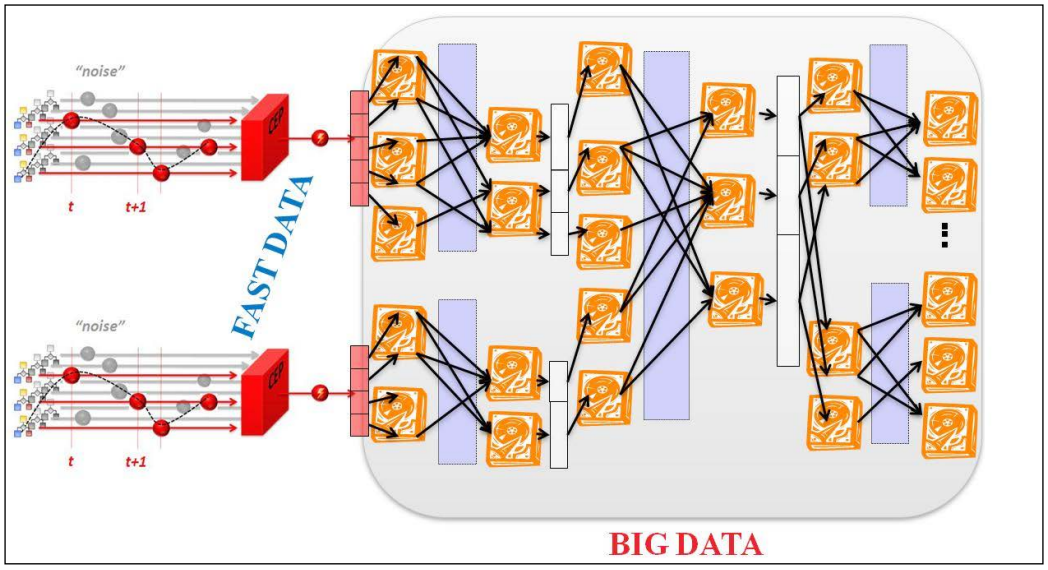
Immediate impact to a company's business can also come appear "out of nowhere" emanating from a change in global business conditions indicated from the ever-expanding social media outlets, for example, Twitter, instant messaging, and so on. Millions of people at the same time can all comment on the poor condition of a new product, highlighting an immediate need to change a product design. This will inevitably affect profits and will probably significantly affect the value of the business. So companies are now inevitably being manipulated by a wide range of both understood and misunderstood events.



In the past, probably going back over 15 years ago, business applications have had to conform to the methodologies, structure, and interfaces from the then available computing technologies (such as databases) where information must be inserted and statically placed. Only after this can users then analyze and respond. Traditional JEE Application Servers were generally implemented, expecting a client application to send an initial request and will then only process that request through, in most cases a significant amount of logic code, before it can respond back to the client. While these technologies enable, and will continue to provide benefit in more batch-orientated, less real-time approaches, newer lower latency and faster in-memory middleware products are now available.

Event-Driven (Architecture) based systems are intrinsically smarter, or better "equipped" to handle these types of situations, processing an entire business infrastructure as events that can be immediately interpreted and handled, spanning across the many departmental "silos" such as finance, manufacturing, and sales. These types of systems are also context aware and execute when they detect changes in the environment or business world, rather than occurring on a predefined (nightly) schedule or requiring someone to initiate an execution.

As the problems associated with Big Data grow substantially over the coming years in terms of the capture, management, and the ability to process the information within a tolerable amount of time, Event-Driven technologies (specifically Complex Event Processing) can provide **Fast Data** capabilities to apply a greater level of "intelligence" and decisioning to the originating data streams much closer to the "point of occurrence".

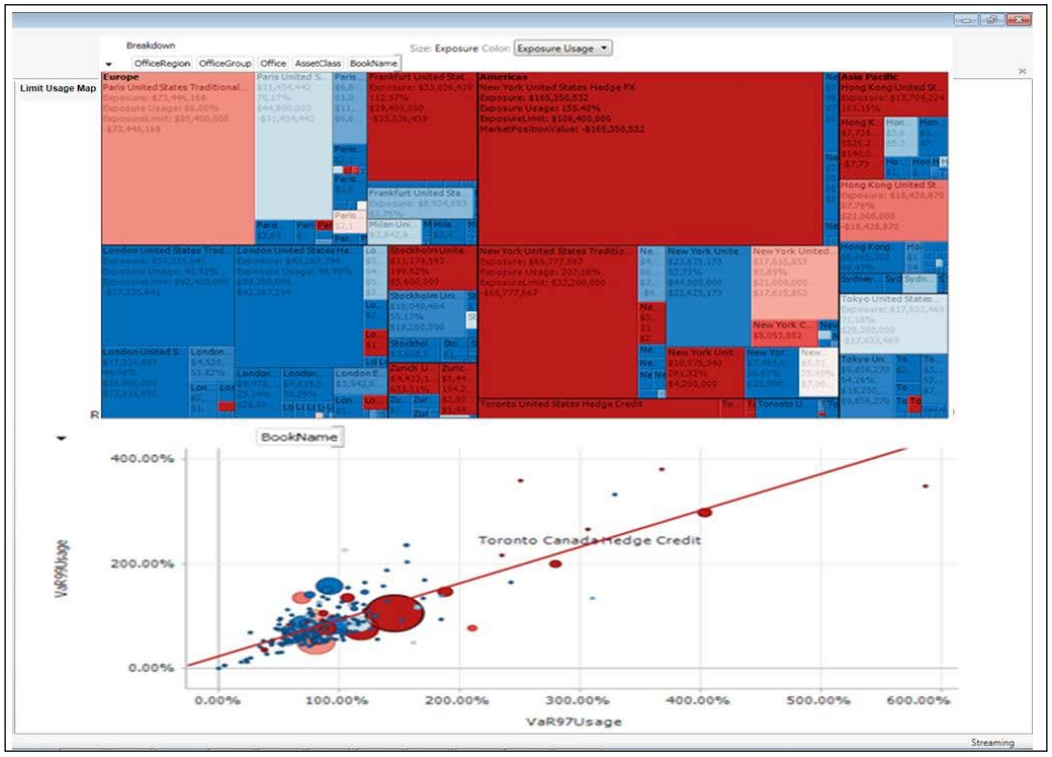


So the benefits of an Event-Driven technology approach is to turn that proliferation of data into real-time knowledge by firstly representing events (things that happen from anywhere) in standard ways, providing an ability to factor out events, route events, filter events, aggregate events, and correlate events intelligently, so that in most cases fragmented events can be evolved into holistic, solid, understandable business events, enabling the business to better view, control, and adapt to situations relatively instantaneously.

Use case: A solution for customer problems

So how are Complex Event Processing Platforms used now to solve business problems? Certainly over the past few years, this technology is being used across most, if not all, of the different types of industries.

The financial services capital markets companies are using this technology for real-time algorithmic trading and real-time risk management types of solutions. As the stock markets stream their endless financial instrument data with values which can instantly fluctuate, there is an ever growing need to effectively handle this huge volume of information, understand its impact and potential risk, and then react as quickly as possible. The better the capability to evaluate and predict the consequences of the information, and the quicker the ability to respond to the results of this analysis, the more successful the business and the more money that can be made with less exposure to business risks and threats. This type of real-time trading information can be usually visualized using **heat maps** and **scatter charts**.



In the Electricity industry, customers are using the Complex Event Processing (CEP) platform for many new types of applications, which include Smart Meter, Smart Grid, and outage detection monitoring solutions. Sophisticated **Demand Response** (**DR**) solutions bring together system operators and the power generation companies, who contract with energy management and monitoring companies to provide energy usage load reduction services on demand. These technology companies that are using CEP-based applications contract with commercial and industrial businesses that are large consumers of energy, whom agree to curtail energy usage on demand. Streaming event devices are installed at client locations to measure energy usage and, in some cases, proactively control the load using continuous energy demand and usage data at minute or, even second, intervals. The generated profit revenue received from system operators is then passed back to the clients, relative to the number of associated load reduction dispatches.

Handling real-time events has a long history in the telecommunications industry, such as those generated by the various devices on the network, events from mobile phones, or perhaps streaming **Call Detail Record** (**CDR**) events indicating the time of calls made and whether some of these calls failed. Complex Event Processing platforms provide the technology for many new applications and solutions in this domain. As in other industries, Event-Driven platforms have a broad base of possible implementations. Some businesses have created powerful network management and monitoring solutions, which can detect hardware failure-related events continuing over certain time periods, or situations where equipment has not been issuing events for some time and in these circumstances alert messages are distributed and escalated.

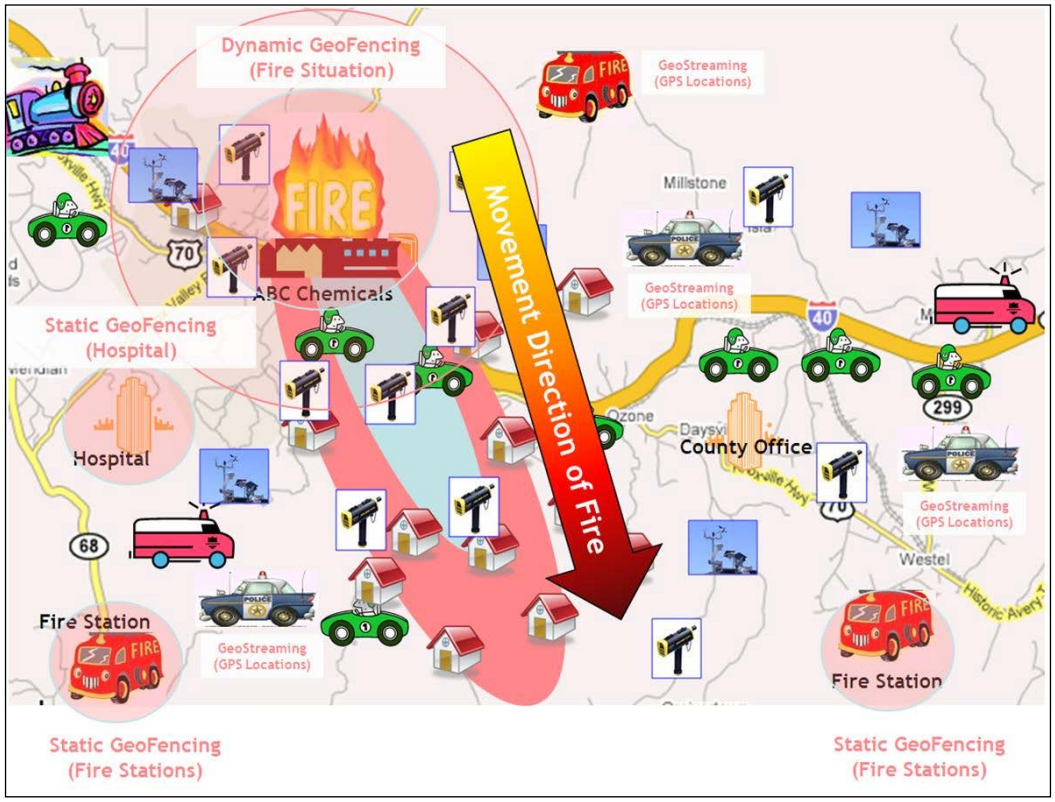
In the context of an enterprise-level mobile telecommunication IT infrastructure, there are many different applications coming from many different suppliers. When the overall performance is not immediately meeting expectations, it's not easy to identify which component is the offending issue in the supply chain. Therefore these next-generation management and monitoring applications (based on Complex Event Processing) provide the capabilities to show the complete, holistic "picture", providing full visibility to the situation of a business through flexibility and fully integrated features, enabling agility for the infrastructure to react quickly to changing scenarios, and providing full operability enabled by a solution designed to meet business needs.

A very powerful capability of Complex Event Processing platforms which is being leveraged in the Transportation, Telecommunications, and Public Sector domain is real-time integrated spatial analysis.

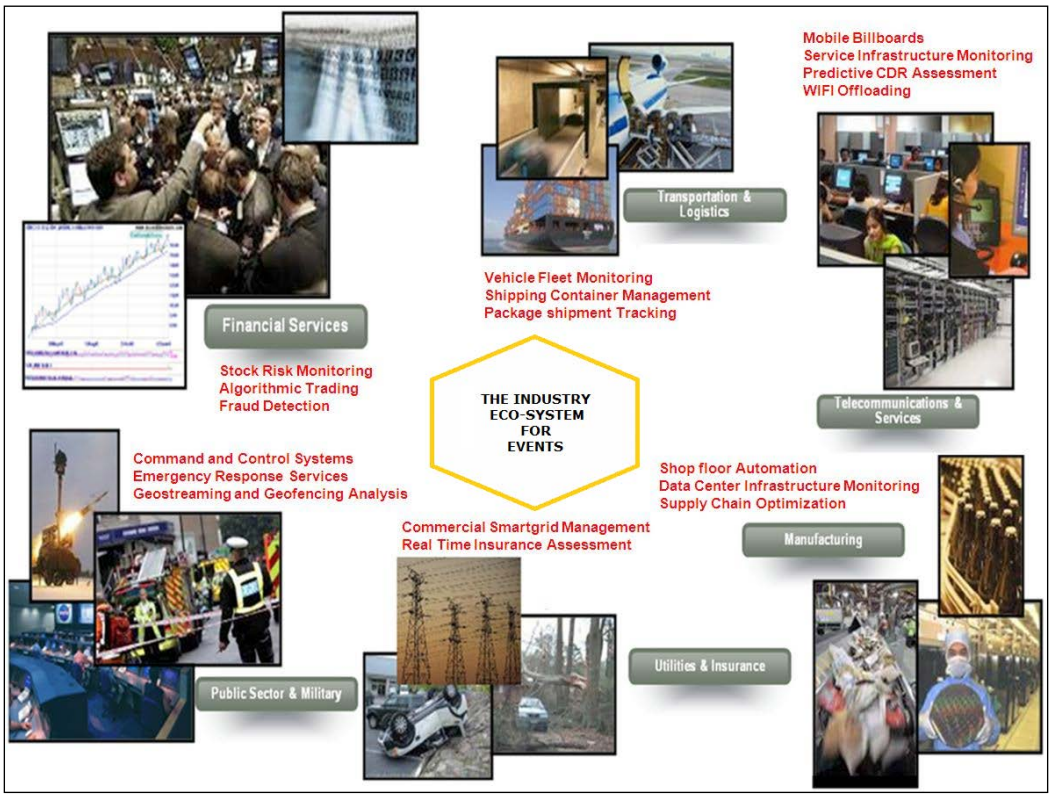
A business can use this technology in applications where there is the need to monitor the movements of its assets and resources. Using, for example, GPS (global positioning systems) the movement patterns of someone, or something can be tracked in real time as it passes through boundary points (such as security checkpoints in an airport) to identify its route and, to some extent, predict where this person or object may subsequently move next. Also, this capability can be used to analyze a current position and its relationship to geofenced areas. A geofenced area being the definition of a geographical shape (polygon) defined or declared by a series of spatial coordinates.

When a resource gets near, inside, or enters and exits the geofenced area, various actions can be immediately performed, such as a warning message of an imminent exposure to a dangerous natural disaster, or offering a big discount on a second coffee at the person's current location or soon to be, position, based on his or her current movement pattern.

First Responder emergency services solutions can use integrated spatial technologies to not only monitor a fire or hundreds of simultaneous fires, but also dynamically track the movement on the fire, affected by weather conditions (wind) or igniting hazardous materials. These types of systems can evaluate immediately the relevance, importance, and applicability of all of the related assets (fire engines, police vehicles, and so on) close to these areas. For example, if a fireman does not move in certain number of seconds when close to a fire, this could indicate a serious life threatening situation.



There are many other types of business solution implementations using Complex Event Processing platforms that range from online retail monitoring systems, real-time data center infrastructure management, fleet vehicle transportation monitoring, traffic flow monitoring with variable toll charging and speed control, oil fields and rig monitoring/automation, and a host of real-time sensing device opportunities, where these devices can monitor the environment inside shipping containers, or air pollution situations. The scope and different type of applications that can now benefit from using Complex Event Processing technologies are evolving just as quickly as the world is changing, with a growing need to predict and pre-empt and in, some cases, prevent situations from even happening.



Key elements of event stream processing

During the next few sections we will explore some of the basic principles and concepts commonly used in the creation of event-driven applications. These are the major "building blocks" for any solution that handles streaming event data.

An event

What is an event and how is it defined? Many people and technical societies define an event in many different ways, but in the context of this book, an event is an object that has a change in its state immediately, or over a period of time.

For example, let's take an everyday object, a house front door.

The door's "properties" is that it is made of wood, it has hinges, perhaps separate wooden panels, screws to keep it together, a handle or knob, and it has a color, blue. When the door opens, then it has changed its "state" and effectively an event has happened.

The door can have many event states: open, closed, opening, closing, and so on. It can even have a "non-event" state, for example, if somebody turns the door handle or knob, but the door does not open in 10 seconds, then this could be a situation when although the door should have opened it didn't in a certain time period, so this is an event that did not happen, but probably should have happened, based on the fact that the door handle did turn.

Anticipation or expecting some event to happen in a certain period of time is something that your brain can easily process but in computing terms it is something that is, on most occasions, difficult to program.

An event stream

Generated by hardware sensor devices, distributed anywhere from the "Internet of things", computer applications, database triggers, or generated from any of hundreds of different sources, events arrive for processing in an event stream or streams. Event streams can have events that are continuously flowing at high volumes or arrive in sporadic intervals, but the events never end and are always time ordered, just like in the real world.