Why is event-driven thinking different from traditional thinking about computing?

Presenters: Opher Etzion and Jeffrey Adkins
A year ago, Roy Schulte from Gartner published a personal blog entitled “does anybody care about event processing.”

His observation is that 95% of the event processing market is not visible since it is home-built and not labeled as EP.

He admitted that his predictions about the actual size of the event processing market is smaller than predicted.

In this tutorial we will discuss what is event driven thinking and how it is possible to help organizations to help themselves in exploiting events.
Agenda

I. Introduction – Brief History of Event Processing in practice

II. The major differentiation factor of event-based thinking

III. The Ontology of event and event influence

IV. Anatomy of reactive systems

V. Pragmatics: A business oriented approach

VI. Summary
Topic I – Introduction & a brief history of event processing in practice
Event Processing History

First start-ups: Descendants of academic projects

Around 2000

Apama acquisition by Progress

TIBCO and Oracle announce products

2005

Streambase Coral8

2007

EPTS Established

Hitting the analysts hype...

2008

IBM Joins

2012

M&A: TIBCO/Streambase, Software AG/Apama

New players: SAS, Yahoo, Twitter

2013

EP at the height of BIG DATA hype Cycle

6/30/2013
Events, as “data in motion” is one of the fundamental ingredients in big data: event-driven analytics

Event-driven services / making events part of SOA

Event-based decision making and event driven Optimization

Event-driven processing as a backbone of next generation systems: event-based robotic, autonomic vehicles, human enhancement technology...

Where are event used today? Virtually everywhere
Event Processing in 2013

There is now an accelerated development of new event-based systems – many of the current trends in computing are event-driven.
Relatively new players in event and stream processing

Sas

S4

IFTTT

ON \{X\}

Google

Intelligent events

Storm
Big Data Hype Cycle 2012

Event processing
Again in the hype Cycle – in different context
Event Driven Applications follow the 4D paradigm

Detect → Derive → Decide → Do

Did Something Happen? → Yes! → What should we do about it?

I want to know about it immediately and react in the best possible way.
This is how the event-driven application market looks

Source: Event processing Manifesto

Build your Own

Use COTS

New segments

manual
0: unused
No event awareness

1: manual
Subscription to some events, manual handling

2: implicit
Events are stored in databases and are processed as part of process oriented

3: islands
Explicit event processing for some applications as islands; instrumentation and actions are hard coded and sporadic

4: integrated
Event processing is integrated with main business processes; instrumentation and actuators are well established

5: strategic
Strategic view of event processing across the enterprise
Barriers for Wider Adoption

Lack of standards: SOA took off only when WS standards were accepted

Lack of sufficient awareness and good ROI understanding: Need entry points and methodology about benefits to individuals, enterprises, packaged applications providers.

Luck of understanding of what is event-based thinking and how to translate it to implementation

Lack of skills – current tools require highly skilled developers to do tricky programming
Major Gap: Products in this area are geared towards IT Developers

A comprehensive user survey shows that 84% of the users wish that event rules could be defined by business users.

There is a gap

Current models: Implementation oriented

Business analysts oriented Modeling

Chart 16: Event Rule Definers
What makes it difficult to express requirements?

Develop correct application with the right semantics

Observation:
A substantial amount of effort is invested today in many of the tools to workaround the inability of the language to easily create correct solutions
Some Correctness Topics

The right interpretation of language constructs

The right order of events

The right classification of events to windows
Bid scenario - ground rules:

1. All bidders that issued a bid within the validity interval participate in the bid.
2. The highest bid wins. In the case of tie between bids, the first accepted bid wins the auction.

Race conditions:
Between events;
Between events and
Window start/end
A simple scenario to demonstrate complexity – why “native implementation” does not work?

Bid scenario- ground rules:
1. All bidders that issued a bid within the validity interval participate in the bid.
2. The highest bid wins. In the case of tie between bids, the first accepted bid wins the auction

Trace:

===Input Bids===
Bid Start 12:55:00
credit bid id=2, occurrence time=12:55:32, price=4
cash bid id=29, occurrence time=12:55:33, price=4
cash bid id=33, occurrence time=12:55:34, price=3
credit bid id=66, occurrence time=12:55:36, price=4
credit bid id=56, occurrence time=12:55:59, price=5
Bid End 12:56:00

===Winning Bid===
cash bid id=29, occurrence time=12:55:33, price=4

Race conditions:
Between events;
Between events and Window start/end
Positively looking: what is the main challenge to resolve in order to accelerate the use of events?

Advancing the technology is always helpful - it is happening, yet it is not the main challenge.

Organizations are not interested in technology for technology’s sake alone.

The next frontier: change the thinking start with the business need and then get to the IT side.
Summary of topic I:

In the business level – it is not well understood how to think in events and how to utilize events

In the application level – the life-cycle of event-based systems require skilled IT developers

Next - we explain what is different about event-driven thinking
Topic II – the major differentiation of event-based thinking
A simple example: event-based Anti money Laundering

An account is suspicious if any of the following patterns are satisfied:

1. Frequent big cash deposit
2. Frequent cases of a big cash deposit followed by transfer abroad
3. Lack of account activity
4. Increasing amounts of deposits
Characteristics of Event-Driven Scenarios

- Events trigger action
- Events influence logic for the results
- There may be multiple events whose combined content influences the results
- Temporal contexts (15 days) influence the results

A suspicious account is detected whenever there are at least three big cash deposits followed by transfers abroad in the last 15 days.
Traditional Thinking

Insert the event into a database; use periodic or on-demand queries to process the events

The processing may not be efficient – many of the requests will not yield results

However:

The processing may not be effective – the time to react may be missed
Process Oriented Thinking (EPMM Level 2)
Difficulties in expressing such scenario in traditional thinking

The event-driven vs. request-driven nature

Effectiveness and Efficiency issues

The hidden state handling

The temporal oriented behavior
Efficiency and effectiveness issues

The processing may not be efficient – many of the requests will not yield results

HOPEVER
If the processing is periodic or by request then:

The processing may not be effective – the time to react may be missed
Request driven vs. event driven thinking
In daily life we often react to events..
Traditionally programmers are trained to think in a request driven way.

Searching the web, database queries, use of web services, use of mobile applications.
<table>
<thead>
<tr>
<th>Question</th>
<th>Response Driven</th>
<th>Event Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Why is an action being taken?</td>
<td>As a response to a specific request</td>
<td>Triggered by the fact of a specific situation</td>
</tr>
<tr>
<td>When is an action being taken?</td>
<td>When the request is being processed</td>
<td>Determined based on the context of the situation</td>
</tr>
<tr>
<td>What happens when the request / event occurs?</td>
<td>A response is always produced</td>
<td>The event can be ignored, increment the state, trigger an internal derive event, or trigger a situation</td>
</tr>
</tbody>
</table>
Temporal consideration changes everything

The logic is sensitive to timing of events
A delivery should be confirmed by the deadline

The logic is sensitive to the order of events
The winner in the bid is the first one who made the highest bid

Determination by timing considerations
Driver ranking increase and decrease are determined every 20 assignments

Why? What? When?
Temporal consideration changes everything

The logic is sensitive to timing of events
A delivery should be confirmed by the deadline

The logic is sensitive to the order of events
The winner in the bid is the first one who made the highest bid

Determination by timing considerations
Driver ranking increase and decrease are determined every 20 assignments

© 2013, IBM Corporation
Logic sensitive to the timing of events’ occurrences

Events during rush hour are of interest, events outside rush hour are not

Events that occur or don’t occur relative to a deadline

Is the reported problem already solved, or is it still open?
Logic sensitive to the order of events’ occurrences

Who arrived first?

Has the bid arrived while the auction was still open?
The nature of situation is determined by timing considerations

Determine the status of a patient based on blood pressure measurements:

Every 8 measurements

Every 5 hours
Handling Hidden State

Pattern “event1 occurs after event2” requires keeping state of all unmatched instances of event1.
Summary of topic II:

In many cases – event driven functionality is expressed using the traditional request-response fashion.

Fundamental differences exist between the two paradigms, and benefits exist in using event-driven modeling and implementation for certain applications.

Next – drilling down to the essence of event driven thinking.
Topic III – The ontology of events and event influence
What is an event – three views

The happening view

An event is anything that happens, or is contemplated as happening.

The state change view

An event is a state of change of anything

The detectable condition view

An event is a detectable condition that can trigger a notification
Events in Linguistics thinking

**State Change**

I now pronounce you, husband and wife

To Be or Not To Be

---

Events that we want to know, can, and should, be first worked through as done as a sentence.

“Friends, you and me... you brought another friend... and then there were three...”

---

To Boldly go where no one has gone before
Events in Linguistics thinking

What’s The Story

Actor did action that changed noun from state to state because of a reason. Observer saw it, and told all.
Ancient Criterion of Change

An Object, $x$, changes if and only if

i. there is a property, $P$,

ii. there is an object, $x$,

iii. there are distinct times, $t$ and $t'$ ($t \neq t'$), and

iv. that $x$ has $P$ at $t$ and fails to have $P$ at $t'$

(Lombard)
Quality Space

Quality Spaces are sets ($S$) of simple, static properties \{P_1, P_2, \ldots, P_n\} which meet conditions:

(i) If at any time, $t$, any object, $x$, has $P_i \in S$ then

(ii) If at any time, $t'$, $x$ doesn’t have $P_i$, it will have $P_j \in S$ where $i \neq j$

We call this a Dimension. An object can have multiple dimensions. Could be represented by an ERD, enumeration
The Goal

It is the MOVEMENT along this quality space that constitutes an event.

The Goal: To become aware of these events, so that we may react to them.
The Awareness boundary represents the boundary of an ecosystem’s knowledge of events / situations. This is the knowledge that an ecosystem uses to understand situations, decide on a course of action and perform that course of action.
There are situations that occur outside of an ecosystem’s awareness boundary to which a reaction inside the ecosystem is warranted.
Some situations’ occurrences can be directly detected by sensors or come into an ecosystem through data feeds or some other instrumentation.
Once detected, a virtual representation of the situation, called an event, exists within the ecosystem. These event can be used as part of the work of the ecosystem.
Other situations that occur outside our awareness boundary can’t be directly detected, so we have to derive that it occurred.
These two situations are indicators that the third situation has occurred and since we have knowledge of their occurrence, we can use it to derive the third, non-detectable situation.
There is a relationship between the real world situation’s occurrence and the virtual representation. We propose calling it “a Luckham relationship”.

Propose calling relationship between real world event and derived awareness of event a Luckham Relationship

Because we can’t directly detect it

Situation occur outside of awareness

Derive Mechanism

Input

Situation occur out side awareness

Indicators

Indicated

Situation occur out side awareness

Indicators

6/30/2013
There are still other situations that cannot be directly detected nor derived. These situations require a human to observe it and enter it in the system.
This is actually the most common way situations become known. These too should generate an event inside the ecosystem to separate out awareness from reaction.
These two situations, when occurred together in a pattern, indicated that the situation on top has occurred.
There are different criteria of change that may play a part in this pattern indicating the situation on top occurred.
This pattern also has a probability associated with it which indicates the confidence that the situation on top occurred.
A particular situation may have more than one pattern that indicates to a certain level of confidence that it has occurred.
Object – Nouns of our story

Nouns

names a person, place or thing

The things we manipulate, collect, buy, sell, talk with....
It has state and dimensions
It has relations with other THINGS
Actors – The Hero, The Helper, and The Obstacle

All participates in the process. Actor (hero) is the one who performs the change. Helper assists “hero” and Obstacle inhibits the process. These can help tell about what is happening.
Observer – Reports what is seen

Observer is not part of the process

There are potentially multiple observers of the same events

Observer might have a subjective inaccurate perspective

Observers may not be familiar with the context
Processes

We have some noun / thing that is important to the business and we want to know when it changes.

Some Noun of Importance

This noun is something we care about
Processes

A process consumes that noun and transforms it into something more useful.

This noun is something we care about

A process is needed to alter the noun

Transforms

Some Noun of Importance but different

The noun is altered
Processes

A process is needed to alter the noun

Transforms

Gives off indicators of the change

Some Noun of Importance but different

The noun is altered

When a noun is changed, it gives off indicators that can be sensed.

This noun is something we care about
A process is needed to alter the noun.

Processes

Some Noun of Importance

This noun is something we care about

Some Noun of Importance but different

Transforms

Gives off indicators of the change

The altered noun.

The event is a logical representation of the indicators

Event “noun has changed”

An event is created inside the ecosystem that is the logical representation of indicators.
Process Ownership

Process can be owned by one of the five groups below. Which have several characteristics. The spectrum of each shows how much of the process and indicators are available or obscured.

<table>
<thead>
<tr>
<th>Nature</th>
<th>Regulatory</th>
<th>Competitor</th>
<th>Supply Chain Partner</th>
<th>Self</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seemly Random but patterns emerge</td>
<td>Due Process</td>
<td>Little visibility until in open market</td>
<td>Visibility to a point</td>
<td>Own Process</td>
</tr>
<tr>
<td>Impact small to momentous/catastrophic</td>
<td>Rules by Law &amp; Procedure</td>
<td>Actively hiding info</td>
<td>Vested interest in mutual success</td>
<td>Easy instrumentation</td>
</tr>
<tr>
<td></td>
<td>Impact significant</td>
<td>Share Market Space</td>
<td>Other Customers</td>
<td>Have Understanding</td>
</tr>
</tbody>
</table>

Processes hidden And opaque indicators Processes understood And available indicators
Data POV vs. Process POV

When processes and indicators are understood and available, we take a Process POV approach. When they are hidden and opaque, we take a data oriented approach.

**Data POV**
- Nature
- Regulatory
- Competitor
- Supply Chain Partner
- Own

**Processes hidden And opaque indicates**

**Processes Understood And Available indicators**

**Instrumentation**
- Sensors, Market Intelligence Collectors, Industry

**Tooling**
- Streaming, Big Data, Master data management

**Process, Object self-publish, transactions, feeds**

**Processing / Transaction flow**

---

6/29/2013
The Awareness Engineer job is to figure out how we become aware. Below are starting points for the awareness engineer based on the Point of View.

Starting Point

<table>
<thead>
<tr>
<th>Data POV</th>
<th>Process POV</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Identifying situations, streams / datasets available</td>
<td>1. Identifying process triggers in terms of situation</td>
</tr>
<tr>
<td>2. Determine patterns that indicate situation</td>
<td>2. Identify significant nouns, its states and properties</td>
</tr>
<tr>
<td>3. Iterate over intermediate patterns until you get to raw data/events</td>
<td>3. Map situation to noun-state change</td>
</tr>
<tr>
<td></td>
<td>4. Instrument in processes code significant state changes (detect)</td>
</tr>
<tr>
<td></td>
<td>5. For all situations that does not directly map, iterate top-down or bottoms-up until you connect (derive)</td>
</tr>
</tbody>
</table>
Awareness of events – what are the main reasons?

EP Solution Segments – Business Value

- Getting the right information in the right granularity to the right person at the right time
- Diagnose problems based on symptoms and resolve them
- Quick observation into exceptional business behavior and notification to the appropriate people
- Mitigate or eliminate predicted events

Reactions to events are done as part of business transactions – achieving low latency decisions, and quick reaction to threats and opportunities
Reaction types

Decision first: A decision is needed in order to make reaction; the decision can be simple, or complex (requiring OR methods)

Actuators: Automatic activation of actuator

Notification in various ways:

Activation of process/workflow/task – manual or automatic
By 2015, 80% of all available data will be uncertain

- By 2015 the number of networked devices will be double the entire global population. All sensor data has uncertainty.

- The total number of social media accounts exceeds the entire global population. This data is highly uncertain in both its expression and content.

Data quality solutions exist for enterprise data like customer, product, and address data, but this is only a fraction of the total enterprise data.
Representative sources of uncertainty

- Thermometer
- Human error
- Fake tweet
- Sensor disrupter
- Wrong hourly sales summary

Source

Malfunction

Malicious Source

Projection of temporal anomalies

Uncertain input data/Events

Propagation of uncertainty

Sampling or approximation

Source Inaccuracy

Visual data
Rumor

Wrong trend

Inference based on uncertain value
Types of uncertainty in event processing

- incomplete event streams
- insufficient event dictionary
- inconsistent event annotation
- imprecise event patterns
- erroneous event recognition

Uncertainty in the event input, in the composite event pattern, in both

6/30/2013
Uncertainty handling

Traditional event processing needs to be enhanced to account for uncertain events.

Two main handling methods:

Uncertainty propagation
The uncertainty of input events is propagated to the derived events.

Uncertainty flattening
Uncertain values are replaced with deterministic equivalents; events may be ignored.
Pattern matching: Sequence (1/3)

Suspicious Observation

| Certainty | 0.8          |
| Occurrence time | Uni(9:45AM,10:05AM) |
| Id       | ‘John Doe’   |

Crime report

| Certainty | 0.9          |
| Occurrence time | 10:02AM |
| Id       | NA           

Crime report matching

Pattern: Sequence [Suspicious observation, Crime report]
Context: Location, Crime type
Pattern matching: Sequence (2/3)

The ‘uncertainty propagation’ approach

<table>
<thead>
<tr>
<th>Suspicious Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certainty</td>
</tr>
<tr>
<td>Occurrence time</td>
</tr>
<tr>
<td>Id</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Crime report</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certainty</td>
</tr>
<tr>
<td>Occurrence time</td>
</tr>
<tr>
<td>Id</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Crime report matching
Pattern: Sequence
Context: Location, Crime type

\[
\text{Certainty} \cdot \text{crime.certainty} \cdot \text{Prob}\{\text{obs.time}<\text{crime.time}\}
\]

Matched crime

- Certainty: 0.612
- Occurrence time: Uni(9:45AM,10:02AM)
- Id: ‘John Doe’

\[
\text{obs.time} \mid \text{obs.time}<\text{crime.time}
\]

The ‘uncertainty propagation’ approach

Pattern matching: Sequence (2/3)
Pattern matching: Sequence (3/3)

The ‘uncertainty flattening’ approach

Suspicious Observation

| Certainty | 0.8 |
| Occurrence time | Uni(9:45AM,10:05AM) |
| Id | ‘John Doe’ |

......

......

Crime report

| Certainty | 0.9 |
| Occurrence time | 10:02AM |
| Id | NA |

......

......

Matched crime

| Certainty | 0.72 |
| Occurrence time | 9:55AM |
| Id | ‘John Doe’ |

......

......

**Occurrence time → percentile(occurrence time, 0.5)**
Summary of topic III:

The ontology of event-based systems is based on awareness of events – either directly or by derivation from other events.

The awareness is enabler for reactions.

Next - we describe the anatomy of event-based systems.
Topic IV – Anatomy of reactive systems
Two separate but connected goals: Awareness and Reaction

Awareness

Did Something Happen?

Detect

Derive

Event

It Happened

Reaction

What should we do about it?

Decide

Do
Reactions are Events too

Becoming aware of an event and **then** doing something about it.

- **Single Event**
  - May need multiple iterations
  - Detect Mechanism
  - May require addition reference / state information

- **Multiple Events**
  - Derive Mechanisms
  - Derived Event

- **Ancillary Info**
- **Event of Interest**
  - Trigger
  - Decide Mechanism
  - Order
  - Feedback Events

- **Something we want to react to**
  - A Situation

- **Should be indicating**
  - Decisions
  - Entity State Change

6/30/2013
Detect

The act of bringing into a system’s sphere of understanding knowledge about an event.

A person recognizes the change and enters it into some system. This is the classic case! This is the most flexible because humans are ingenious.
Detect

The act of bringing into a system’s sphere of understanding knowledge about an event.

A sensor senses the indicators and creates the corresponding event in the system.
Detect

The act of bringing into a system’s sphere of understanding knowledge about an event.

A data feed or systemic interface allows events to be published into the system.
Detect

The act of bringing into a system’s sphere of understanding knowledge about an event.

When the processes under the system’s control makes changes to nouns it should published these changes as events.
Detect

The act of bringing into a system’s sphere of understanding knowledge about an event.

As connected things are becoming more self-aware of their inner-workings, they can publish their own state changes.
Derive

The act of becoming aware of events that are not directly detectable by bringing together events with other events, data, patterns and publishing the observation as a derived event.

A Person recognizes the pattern and enters the derived event or just reacts to it directly. Shown a lot of time by dashboards and analysis.
Derive

The act of becoming aware of events that are not directly detectable by bringing together events with other events, data, patterns and publishing the observation as a derived event.

A Neural network processes the various inputs and determines a new situation expressed by a derived event.
The act of becoming aware of events that are not directly detectable by bringing together events, data, patterns and publishing the observation as a derived event.

A software applies pattern matching over multiple events and data to find derived events.
Derive

The act of becoming aware of events that are not directly detectable by bringing together events, data, patterns and publishing the observation as a derived event.

The most common place is hidden inside of every day system’s code.
Decide

The act of determining the course of action to do in response to the situation. This includes the background information needed to be collected to make the decision.

Pass through: Sometimes there is no decision. There is only one course of action.
Decide

The act of determining the course of action to do in response to the situation. This includes the background information needed to be collected to make the decision.

No Decision

Manual Decision: Many times the ecosystem asks a person to decide the course of take.
Decide

The act of determining the course of action to do in response to the situation. This includes the background information needed to be collected to make the decision.

Automated Decision: Algorithmic decision via a decision management system.
Decide

The act of determining the course of action to do in response to the situation. This includes the background information needed to be collected to make the decision.

Automated Goal Oriented: Algorithmic decision via a decision management system that seeks a optimizing quantitative goals.
Do

The act of performing the course of action that was decided upon.

Notification: Sending a signal of sort to either a person or system. This would include calling a web-service or subscription to alerts.
Do

The act of performing the course of action that was decided upon.

Manual Action: This is an order for a human to go do an action.
Do

The act of performing the course of action that was decided upon.

Applying Actuator: cause a action or setting change on an actuator.
Do

The act of performing the course of action that was decided upon.

Trigger process: Execute a process or potential a single action.
Richard Hackathorn’s Response Time Latency

Value in terms of Competitiveness decreases

Source: Richard Hackathorn – Active data warehouse, from nice to necessary, Teradata Magazine, 2006
4D Version of Response Time Latency

Converting Richard Hackathorn’s flow to the 4D Prospective.
Detect Latency

The time it takes for a ecosystem to detect either the business event or indicators that can be used derive the event.
Derive Latency

The time necessary to combine the indicators, events, ancillary data or human analysis to derive situations that can’t be directly detected.
Decide Queue Latency

The time waiting for someone (manual decisions) or something (automated decisions) to make a decision.
Decide Act Latency

The time it takes to decide the course of action in reaction to the situation.
Do Queue Latency

The time waiting for someone (human actor) or something (machine actor) to start executing or orchestrating the course of action.
The time it takes to execute the course of action.
Until the course of action is completed, everything is a value investment. After the course of action is completed, value can be realized.
Business velocity – a key competitive edge

Business velocity is constantly increasing as the world moves to on-line business with multiple alternatives.

“There’s no patience on the Internet”

6/30/2013
Summary of topic IV:

An event-driven system consists of the 4D architecture

Latency reduction is the key to business velocity

Next we’ll present basic ideas about the business oriented approach
Topic V – pragmatics – a computational independent model for event-based systems
Approaching Event Processing in an Enterprise

- Business Goals
- Application characteristics

Current state in the maturity model
Calculating the benefits of event processing

- Business Predictability
- Business Agility
- Business Velocity
- Business Optimization
- operational support
- Compliance with Regulation

Quantify the impact of:

- proactively seeking and adapting to patterns that might indicate an emerging event (threats and opportunities)
- Enable the support of event-driven adaptive business processes
- Increase business velocity by faster response to opportunities and early detection of threats
- Make faster and better (manual or autonomic) decisions based on timely multi-source information;
- Increasing level of automation and thus increase productivity (e.g. of back office) and reduce cost
- Continuous audit ensures timely handling of violations.
Situation awareness

- Enables the business logic to be context sensitive

Context sensitive

- Taking advantage of information whose value decreases in time.

Real-time dissemination

- Enables fast deployment of new versions when the business logic dynamically changes

Fast change

- Complexity stems from one or more of: event rate, quantity of event sources, state and context handling, event order sensitivity.
EPMM: Event Processing Maturity Model (IT View)

0: unused
No event awareness

1: manual
Subscription to some events, manual handling

2: implicit
Events are stored in databases and are processed as part of process oriented

3: islands
Explicit event processing for some applications as islands; instrumentation and actions are hard coded and sporadic

4: integrated
Event processing is integrated with main business processes; instrumentation and actuators are well established

5: strategic
Strategic view of event processing cross the enterprise
Next Step: Develop Multi-Step Method to Select Entry Points

Select Goals

Improve Business Agility
Improve Business Optimization

Prioritize Goals

1. Improve Business Agility
2. Improve Business Optimization

Select applications that best fit these goals

Liquidity Management
Anti Money Laundering
...

Analyze the possible entry point Applications based on their characteristics

By entry point, map both satisfaction of business goals and fitness characteristics for best ROI.

Produce plan of action
Based on the maturity model

115
Vision:
Shift Governance from Programmer to Knowledge Worker

TODAY

Governance occurs through development and maintenance of program code.

TOMORROW

Governance occurs through development and maintenance of event models.
Process Oriented View
Next step – event processing language (ESPER)

// Big cash deposit
insert into BigCashDeposit
select * from Transaction where amount > 100,000 and
transaction_cash_deposit_indicator = 'Y'

// Frequent (At least three) big cash deposits
create context AccountID partition by accountId on Transaction;
Context AccountID
Insert into FrequentBigCashDeposits select count(*) from
BigCashDeposit having count(*)>3;

// Transfer abroad
insert into TransferAbroad select * from Transaction where
transferabroad_indicator = 'Y'

// Frequent cash deposits followed by transfer abroad
Context AccountID
insert into SuspiciousAccount select * from pattern [ every f=FrequentCashDeposit -> t=TransferAbroad where timer.within(15 days)]
Modeling according to the concept computing principles

The application logic should be expressed by a semantically declarative, directly executable, implementation independent, and rigorously structured knowledge model.

Knowledge Model

Automatic translation to code in regular or specific engine language

Free of implementation assumptions

Rigorous verifiable structure with all connections

Represented as a collection of tables

The term was coined by Mills Davis in 2012
## TEM Logic Specification

### Frequent big cash deposits

<table>
<thead>
<tr>
<th>Row #</th>
<th>Context</th>
<th>Event Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When</td>
<td>Partition By</td>
</tr>
<tr>
<td></td>
<td>Expression</td>
<td>Start</td>
</tr>
<tr>
<td>1</td>
<td>last 15 days</td>
<td>same</td>
</tr>
</tbody>
</table>

### Lack of account activity

<table>
<thead>
<tr>
<th>Row #</th>
<th>Context</th>
<th>Event Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>When</td>
<td>Partition By</td>
</tr>
<tr>
<td></td>
<td>Expression</td>
<td>Start</td>
</tr>
<tr>
<td>1</td>
<td>last 10 days</td>
<td>same</td>
</tr>
</tbody>
</table>

- **Expression**: Count(Big cash deposit)
- **Regular Conditions**: is
- **Multiple events Conditions**: Absent
- **Conclusion**: Derived
Next step – model view

Derive Suspicious account

- Frequent big cash deposits
- Frequent deposits followed by transfers abroad
- Lack of account activity

Always

Suspicious account

Frequent big cash deposits
Frequent deposits followed by transfers abroad
Lack of account activity

Account ID

Frequent big cash deposits

Big cash deposit

- Last 10 days
- Account ID

Always

Customer ID

Frequent deposits followed by transfers abroad

Deposit followed by transfer abroad

- Last 30 days
- Account ID

Account ID

Deposit followed by transfer abroad

Big cash deposit

- deposit occurrence
- Account ID

Lack of account activity

Transaction is absent

- Last 10 days
- Account ID

Lack of account activity

Transaction

Transfer abroad

Bank transaction system

Suspicious account

Transaction

Derive Suspicious account

Transaction is absent

Account ID

Transaction

Transfer abroad

Bank transaction system

Derive Suspicious account
Concepts of

Glossary
- Actors
- Facts
- States
- IT elements

Logic
- Event Derivation Logic
- Goals
- Transitions
- Computation Logic
Summary of topic V:

Systematic approach in an enterprise based on ROI and maturity models - the business world is not interested in technology but in business outcomes

Business user oriented modeling as a key solution point

Next we’ll present basic ideas about computing independent model
Topic VI – Summary
What is the main take away from this tutorial – 1/3?

The exploitation of events is a game changer in the universe

In the business level – it is not well understood how to think in events and how to utilize events

In the IT level – most of what can be functionally thought as event processing is implemented using the traditional thinking within application code
What is the main take away from this tutorial – 2/3?

The ontology of event-based systems is based on awareness to events – either directly or by derivation from another events

An event-driven system consists of the 4D architecture

Latency reduction is the key to business velocity
What is the main take away from this tutorial – 3/3?

- Systematic approach in an enterprise based on ROI and maturity models – the business world is not interested in technology but in business outcomes.
- Business user oriented modeling as a key solution point.

Next step: help the organizations help themselves with realizing the benefits of events.
One more thing

OUR DRIVING FORCE IS TO HELP EVERYBODY REALIZE THE POWER OF EVENTS TO CREATE A BETTER WORLD

A BETTER WORLD
Thanks for listening