OpenESB
Intelligent Event Processing (IEP)
Topics

• Business Challenges
• Real-life use cases, RFID theft detection use case
• Key concepts
• Realtime event processing
  > Aggregation
  > Filtering
  > Correlation
  > Partition
• IEP as a JBI module
• IEP designer
Business Challenges
Event-aware Business Services

• Long-running business processes adjusting to ad-hoc external events
Business Activity Monitors

- Monitor production across multiple plants
  > Correlate and process events from multiple sources

Dashboard

- Event Processors
- EDA
- BPM
- SOA
- Back-end Systems
- Production Management System
- Clients, Trading Partners, Suppliers
- Inventory Management System
- Monitor Business Process Instances
Business Service Correlation

- Monitor business service consumptions
  - Detect spikes in service usage - “System Behavior”
  - Track events by user - “User Behavior”
- Automatic response
  - Alerts, Dashboard updates, service triggers, etc.
Real-life Use Cases of IEP

• How many times did Fred login as root for the last 24 hours?
• How many times is “purchase order” business process triggered for the last hour? And what is the average execution time?
• Is a credit card charged for gasoline twice within last one hour?
• Raise an alert when a stock price jump more than 10% relative to its 1 minute moving average price.
• Is the number of JMS messages in the broker increasing over time? What changed?
Use case: RFID Store
Theft Detection
Use case: RFID Store Theft Detection

- Z-Mart Store Layout
- Each product in the store has a RFID emitter that sends out its product ID $P_k$ every 2 seconds.
- 16 sensors: $S_0, \ldots, S_{15}$. One for each section. Sensor $S_i$ receives product ID $P_k$ and reports $(S_i, P_k)$ to the IEP system.
- One cannot remove a RFID emitter from its product without destroying it.
Use case; RFID Theft Detection

- Detect stolen product: A product is considered stolen if both of the following conditions are true
  > Its signal is received by S_15 (sensor at Exit)
  > It is not paid yet

- Detect RFID emitter: A RFID emitter (of a product) is considered dead if both of the following conditions are true
  > No sensor has received its signal for past 10 seconds
  > Its last reporting sensor is not S_15 (sensor at Exit)
Real time Event Processing:
Key Concepts
Table

- A finite collection of events that belongs to the same schema.
- The schema is also called the table’s schema.
- Example:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADBE</td>
<td>38.60</td>
</tr>
<tr>
<td>AMZN</td>
<td>40.12</td>
</tr>
<tr>
<td>AMGN</td>
<td>68.80</td>
</tr>
<tr>
<td>ADBE</td>
<td>38.62</td>
</tr>
</tbody>
</table>
Stream

- A collection of events that belongs to the same schema, and have *timestamps*.
- Example

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Price</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADBE</td>
<td>38.60</td>
<td>20070214T10:30:02.899-05:00</td>
</tr>
<tr>
<td>AMZN</td>
<td>40.12</td>
<td>20070214T10:31:01.674-05:00</td>
</tr>
<tr>
<td>AMGN</td>
<td>68.80</td>
<td>20070214T10:31:05.198-05:00</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Relation

• A collection of tables that have the same schema, and indexed by time.

• Example:

  > The relation represents the latest 2 stock transactions

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Price</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADBE</td>
<td>38.60</td>
<td>20070214T10:30:02.899-05:00</td>
</tr>
<tr>
<td>AMZN</td>
<td>40.12</td>
<td>20070214T10:31:01.674-05:00</td>
</tr>
<tr>
<td>AMGN</td>
<td>68.80</td>
<td>20070214T10:31:05.198-05:00</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Relation (Continued)

- Example

> \( R = \{\ldots, R(20070214T10:31:01.674-05:00), \)
> \( R(20070214T10:31:05.198-05:00), \ldots\} \)
> \( R(20070214T10:31:01.674-05:00) \) is

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADBE</td>
<td>38.60</td>
</tr>
<tr>
<td>AMZN</td>
<td>40.12</td>
</tr>
</tbody>
</table>

> \( R(20070214T10:31:05.198-05:00) \) is

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMZN</td>
<td>40.12</td>
</tr>
<tr>
<td>AMGN</td>
<td>68.80</td>
</tr>
</tbody>
</table>
Real time Event Processing: Aggregation
Types of Aggregators

- Time-based Aggregator
- Tuple-based Aggregator
- Window Aggregator
Time-based Aggregator

- Compute statistical summary **over a fixed time period**.
- Example: Given a stream of transactions of a stock, computes a new stream that holds hourly min, average, and max of the stock price.
Tuple-based Aggregator

• Compute statistical summary over a fixed number of events.
• Statistics: same as time-based aggregator.
• Example: Given a stream of transactions of a stock, computes a new stream that holds the min, average, and max of the stock price of every 10 transactions.
Window Aggregator

• Compute statistical summary over a sliding window
• Statistics: same as time-based aggregator.
• Example: Given the latest 2 hour window of transactions of a stock, computes the min, average, and max of the stock price of those transactions. As time passes by, the statistical summary will change too.
Real time Event Processing: Filtering
Filtering Example

• Given
  > a stream of transactions of a stock

• Compute
  > a new stream that holds only those transactions whose stock price is > $50.
Real time Event Processing:
Correlation
Types of Corelation

- Window Correlation
- Stream Serial Correlation
Window Corelation

• Given the following
  > latest 2 hour window of stock transactions
  > latest 2 hour window of news

• Compute
  > the latest 2 hour window of possible insider-trades by joining the trader’s name with the name mentioned in the news.
Stream Serial Corelation

- Example:
  - Given
    - a stream of stock transactions
  - Compute
    - a new stream in which each event is composed of 3 consecutive events from the original stream.

- This helps to build a trading model that predicts the next stock price using the latest 2 stock prices.
Real time Event Processing: Partition
Partition

- Given
  > a stream of stock transactions
- Compute
  > a window that holds the latest 10 transactions for each stock symbol.
IEP as a JBI Module
IEP is a Service Engine over JBI

- Access to external systems and services
- Receive and Send messages
- Currently supports Java DB & Oracle databases
IEP Designer (in NetBeans)
IEP Designer

Finished building build.xml (tests).
OpenESB
Intelligent Event Processing (IEP)