

Detecting Event Patterns

It's no exaggeration to say that event pattern detection is the jewel in the crown of event processing. Pattern detection lets us go beyond individual events to look for specific collections of events and the relationship between them. A pattern that is detected has a meaning that goes beyond the occurrence of any single event. Consider a personalized healthcare system where a patient is hooked up to multiple monitors; the individual events reported by the monitors might not in themselves be significant, but a combination of measurements occurring in a certain order might indicate a problem that cannot be detected by looking at any single monitor separately.

In this chapter we shall discuss:

- Pattern definition and its role in an EPA
- Pattern categories
- Specific patterns of various types
- Pattern oriented policies
- Patterns in the Fast Flower Delivery use case

Event pattern reference table

Category	Pattern	Pattern Assertion	Requires all relevant events?	Singleton Participant set?
Basic	All			
	Any			
Threshold	Count	Count threshold	X	
	Value max	Value max threshold	(X) ¹	

¹ (X) means – true for certain cases.

	Value min	Value min threshold	(X)	
	Value average	Value average threshold	X	
Relative	Relative min		X	
	Relative max		X	
Modal	Absence		X	
	Always	Always assertion	X	
	Sometimes	Sometimes assertion		
	Not selected		X	
Temporal	Sequence			
	Increasing	Evaluated attribute	X	X
	Decreasing	Evaluated attribute	X	X
	Stable	Evaluated attribute	X	X
	Non increasing	Evaluated attribute	X	X
	Non decreasing	Evaluated attribute	X	X
	Mixed	Evaluated attribute	X	X
Spatial	Min distance	Min distance threshold	(X)	
	Max distance	Max distance threshold	(X)	
	Average distance	Average distance threshold	X	
	Relative Min distance	Min distance threshold	(X)	
	Relative Max distance	Max distance threshold	(X)	
	Relative Average distance	Average distance threshold	X	
Spatio-temporal	Moving in consistent direction	Direction	X	X
	Moving in mixed direction		X	X
	Stationary		X	X
	Moving Towards		X	X

©Manning Publications Co. Please post comments or corrections to the Author Online forum:

<http://www.manning-sandbox.com/forum.jspa?forumID=XYZ>

Apama's example of the five highest bidders

```
386 // if manual, the system selects the 5 highest ranked bidders
387 // and send them to the store
388 if not isAuto then {
389     sequence<integer> keys :=bids.keys();
390     keys.sort();
391     keys.reverse();
392     integer rank, i:=0;
393     DeliveryBid bid;
394     for rank in keys {
395         if i=5 then {break;}
396         for bid in bids[rank] {
397             if i=5 then {break;}
398             route RankedDeliveryBid(bid.requestId, bid.store, bid.driver,
399                 bid.committedPickUpTime, dr.requiredDeliveryTime);
400             i:=i+1;
401         }
402     }
```

Rulecore example of the No Bidders alert

```

Rule Definition
<Rule name="NoBidders" limit="10000" evalMode="once" level="2">
  <Description>This is rule NoBidders</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="BidRequest"]</base:XPath>
      </Event>
    </Assert>
  </Initialize>
  <Views>
    <ViewRef name="NoBidders">
      <base:XPath>sim:ViewDef[@name="NoBidders"]</base:XPath>
    </ViewRef>
  </Views>
  <Situations>
    <SituationRef name="NoBidders">
      <base:XPath>sim:SituationDef[@name="NoBidders"]</base:XPath>
    </SituationRef>
  </Situations>
  <Actions>
    <SituationDetected situationName="NoBidders">
      <ActionRef name="NoBidders" eventVisibility="external">
        <base:XPath>sim:ActionDef[@name="NoBidders"]</base:XPath>
      </ActionRef>
    </SituationDetected>
  </Actions>
</Rule>

View Definition
<View name="NoBidRequests">
  <Description>This is view NoBidRequests</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="BidRequest"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryBid"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryRequestCancellation"]</base:XPath>
      </Event>
    </Type>
    <Match>
      <Property name="RequestId">
    </Match>
  </Properties>
</View>

```

©Manning Publications Co. Please post comments or corrections to the Author Online forum:

<http://www.manning-sandbox.com/forum.jspa?forumID=XYZ>

```

    </Properties>
  </View>

  Situation Definition
  <SituationDef name="NoBidders">
    <Detector>
      <All>
        <After timeframe="00:02:00">
          <Not>
            <EventPickup keep="last" see="new" name="bid" evalMode="once">
<base:XPath>sim:Views/sim:View/sim:Events/sim:Event [@eventType="DeliveryBid
"]</base:XPath>
            </EventPickup>
          </Not>
        </After>
      </Not>
      <EventPickup keep="last" see="new" name="cancel" evalMode="once">
<base:XPath>sim:Views/sim:View/sim:Events/sim:Event [@eventType="DeliveryReq
uestCancellation"]</base:XPath>
            </EventPickup>
          </Not>
        </All>
      </Detector>
    </SituationDef>

  Action Definition
  <ActionDef name="NoBidders">
    <Event>
      <EventDef>
        <base:XPath>sim:EventDef[@eventType="NoBiddersAlert" and
@eventClass="user"]</base:XPath>
      </EventDef>
      <Body>
        <XsltBuilder>
          <Stylesheet><![CDATA[ <?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:user="http://www.rulecore.com/2008/user"
xmlns:base="http://www.rulecore.com/2008/base" version="1.0">
  <xsl:template match="child:*">
    <base:EventBody>
      <xsl:for-each
select="user:Views/user:View[@default='true']/user:Properties">
        <user:RequestId>
          <xsl:value-of
select="descendant::user:MatchedProperty[@name='RequestId']/user:Value/chil
d::text()" />
        </user:RequestId>
      </xsl:for-each>
    </base:EventBody>
  </xsl:template>
</xsl:stylesheet>
        </Body>
      </Event>
    </ActionDef>

```

©Manning Publications Co. Please post comments or corrections to the Author Online forum:

<http://www.manning-sandbox.com/forum.jspa?forumID=XYZ>

Streambase example for automatic matching pattern

The screenshot shows the StreamBase Properties window for a 'SelectFirstBidder - Map operator'. The 'General' tab is active, showing configuration for input fields and additional expressions.

Input Fields:

Action	Field Name	Expression
Include	delivery_request_id	
Include	store_id	

Additional Expressions:

Action	Field Name	Expression
Add	driver_id	driver_ids_w_committime[0].driver_id -- first driver is earliest bid
Add	committed_time	driver_ids_w_committime[0].pickup_time_committed -- first driver's pickup time

The right pane shows the 'Streams' section with 'Input' and 'Output' fields. The 'Output' section lists the following fields:

- input1 (4 fields)
 - store_id int
 - delivery_request_id long
 - manual_process bool
 - driver_ids_w_committime list
- element type tuple
 - driver_id int
 - pickup_time_committed timesta

The bottom pane shows a visual flow diagram for the 'FlowerShop.sbapp' application. The diagram illustrates the data flow from input sources (DeliveryBids, OutstandingM...) through various operators (EnsureBids, ManualOrAutoAssign, SelectFirstBidder, ForEachDriver, AttachRankings, Top5RankedDrivers, Top5, RecordManualAssign) to output destinations (ManualAssignOut, AutoAssignOut). The diagram is highlighted in yellow.

©Manning Publications Co. Please post comments or corrections to the Author Online forum:

<http://www.manning-sandbox.com/forum.jspa?forumID=XYZ>

Bid Alert example in Esper

```
/**
 * No bid after 2 mins of a request
 */
insert into AlertW(requestId, message, driver)
select d.requestId, "no bidder", ""
from pattern[
every d=DeliveryRequest -> (timer:interval(120 sec) and not
DeliveryBid(requestId = d.requestId))
];
```