

Fast Flower Delivery Use Case

- implementation with ruleCore CEP Server

Description

This document describes how to solve (most of) the “Fast Flower Delivery” application using the ruleCore CEP Server.

The solution consists of a set of rules, grouped into three distinctive groups. Each group deals with a certain aspect of the application; initialization of the delivery process, supervising the process and maintaining driver rankings.

Event format

All events that are sent to or received from the ruleCore CEP Server share a common format, the ruleCore Event Format¹. In short, this event format consists of two sections; a header section and a body section. The header section contains meta information about the event such as security and causality information. The body section contains the actual event data or payload. The event type is the same as the local name of the root element. Throughout the document most event examples contain only the body section, this to make the examples more clear and concise.

Rule Sets

Delivery Process Initialization

This set of rules covers the first two phases of the specification; the Bid and the Assignment phase.

Rule: Create Bid Requests

This rule tracks all Drivers that are currently within a certain distance from the store, defined by a Zone with id=Store_<store_id>. When a DeliveryBid event is received for the store, one BidRequest event is generated for each of the Drivers.

Rule Definition

```
<Rule name="CreateBidRequest_store_id" limit="?" evalMode="once" level="1">
  <Description>This is rule CreateBidRequest_store_id</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="GPSLocation"]</base:XPath>
      </Event>
    </Assert>
  </Initialize>
</Rule>
```

1 http://rulecore.com/support/wiki/index.php?title=RuleCore_Event_Format

DRAFT – version 0.1 – DRAFT

```
</Assert>
</Initialize>
<Views>
  <ViewRef name="CreateBidRequest">
    <base:XPath>sim:ViewDef[@name="CreateBidRequest"]</base:XPath>
  </ViewRef>
</Views>
<Situations>
  <SituationRef name="CreateBidRequest">
    <base:XPath>sim:SituationDef[@name="CreateBidRequest"]</base:XPath>
  </SituationRef>
</Situations>
<Actions>
  <SituationDetected situationName="CreateBidRequest">
    <ActionRef name="CreateBidRequest" eventVisibility="external">
      <base:XPath>sim:ActionDef[@name="CreateBidRequest"]</base:XPath>
    </ActionRef>
  </SituationDetected>
</Actions>
</Rule>
```

View Definition

```
<View name="CreateBidRequest_store_id">
  <Description>This is view CreateBidRequest_store_id</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="GPSLocation"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryRequest"]</base:XPath>
      </Event>
    </Type>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="GPSLocation"]</base:XPath>
      </Event>
      <Expression>
        <Property name="Zone">
          <Value>Store_store_id</Value>
        </Property>
      </Expression>
    </Assert>
```

DRAFT – version 0.1 – DRAFT

```
<Assert>
  <Event>
    <base:XPath>sim:EventDef[@eventType="DeliveryRequest"]</base:XPath>
  </Event>
  <Expression>
    <Property name="Store">
      <Value>store_id</Value>
    </Property>
  </Expression>
</Assert>
<Match>
  <Zone/>
</Match>
<Match>
  <Property name="EntityId"/>
</Match>
</Properties>
</View>
```

Situation Definition

```
<SituationDef name="CreateBidRequest">
  <Detector>
    <EventPickup keep="last" see="new" name="P1">
      <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@event
Type="DeliveryRequest" and fn:get-property-value('MinimumRanking') <= fn:get-
property-value('Ranking', fn:get-matched-entity())]</base:XPath>
    </EventPickup>
  </Detector>
</SituationDef>
```

Action Definition

```
<ActionDef name="BidRequest">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="BidRequest" 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/child::text()" />
    </user:RequestId>
  </xsl:for-each>
</base:EventBody>
</xsl:template>
</xsl:stylesheet>
</Body>
</Event>
</ActionDef>
```

Zone Definition

```
<ZoneDef name="Store_store_id">
  <Description>Store_store_id</Description>
  <Circle center="center_coords" radius="pickup_distance"/>
</ZoneDef>
```

Rule: Create Assignments (Manual)

This rule triggers and generates a ManualAssignment event when a BidRequest event is seen and the Store id is present in the “ManualAssignmentStore” List.

Rule Definition

```
<Rule name="CreateManualAssignments" limit="?" evalMode="once" level="2">
  <Description>This is rule CreateManualAssignments</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="BidRequest"]</base:XPath>
      </Event>
      <Expression>
        <Property name="Store">
          <InList name="ManualAssignmentStore"/>
        </Property>
      </Expression>
    </Assert>
  </Initialize>
  <Views>
    <ViewRef name="CreateManualAssignments">
```

DRAFT – version 0.1 – DRAFT

```
<base:XPath>sim:ViewDef[@name="CreateManualAssignments"]</base:XPath>
</ViewRef>
</Views>
<Situations>
  <SituationRef name="CreateManualAssignments">
    <base:XPath>sim:SituationDef[@name="CreateManualAssignments"]</base:XPath>
  </SituationRef>
</Situations>
<Actions>
  <SituationDetected situationName="CreateManualAssignments">
    <ActionRef name="CreateManualAssignments" eventVisibility="external">
      <base:XPath>sim:ActionDef[@name="CreateManualAssignments"]</base:XPath>
    </ActionRef>
  </SituationDetected>
</Actions>
</Rule>
```

Situation Definition

```
<SituationDef name="CreateManualAssignments">
  <Detector>
    <After timeframe="00:02:00">
      <EventPickup keep="last" see="new" name="P1" evalMode="once">
        <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@eventType="DeliveryBid" ]</base:XPath>
      </EventPickup>
    </After>
  </Detector>
</SituationDef>
```

View Definition

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

```
</Match>
</Properties>
</View>
```

Action Definition

```
<ActionDef name="CreateManualAssignments">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="ManualAssignmentRequest" 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"
xmlns:fn="http://www.rulecore.com/2008/functions" 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/child::tex
t()" />
        </user:RequestId>
        <user:DeliveryBids>
          <xsl:for-each select="fn:sort-by-
property(user:Views/user:View[@default='true']/user:Events/user:Event[@eventType=
'DeliveryBid'])">
            <xsl:template match="@*|node()">
              <xsl:copy>
                <xsl:apply-templates select="@*|node()" />
              </xsl:copy>
            </xsl:template>
          </xsl:for-each>
        </user:DeliveryBids>
      </xsl:for-each>
    </base:EventBody>
  </xsl:template>
</xsl:stylesheet>
      </Body>
    </Event>
  </ActionDef>
```

Rule: Create Assignments (Automatic)

This rule triggers and generates an Assignment event when a BidRequest event is seen and the Store id is present in the “AutomaticAssignmentStore” List.

Rule Definition

```
<Rule name="CreateAutomaticAssignments" limit="?" evalMode="once" level="2">
  <Description>This is rule CreateAutomaticAssignments</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="BidRequest"]</base:XPath>
      </Event>
      <Expression>
        <Property name="Store">
          <InList name="AutomaticAssignmentStore"/>
        </Property>
      </Expression>
    </Assert>
  </Initialize>
  <Views>
    <ViewRef name="CreateAutomaticAssignments">
      <base:XPath>sim:ViewDef[@name="CreateAutomaticAssignments"]</base:XPath>
    </ViewRef>
  </Views>
  <Situations>
    <SituationRef name="CreateAutomaticAssignments">
      <base:XPath>sim:SituationDef[@name="CreateAutomaticAssignments"]</base:XPath>
    </SituationRef>
  </Situations>
  <Actions>
    <SituationDetected situationName="CreateAutomaticAssignments">
      <ActionRef name="CreateAutomaticAssignments" eventVisibility="external">
        <base:XPath>sim>ActionDef[@name="CreateAutomaticAssignments"]</base:XPath>
      </ActionRef>
    </SituationDetected>
  </Actions>
</Rule>
```

Situation Definition

```
<SituationDef name="CreateAutomaticAssignments">
```

DRAFT – version 0.1 – DRAFT

```
<Detector>
  <EventPickup keep="last" see="new" name="P1" evalMode="once">
    <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@event
Type="DeliveryBid" and position() = last() ]</base:XPath>
  </EventPickup>
</Detector>
</SituationDef>
```

View Definition

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

Action Definition

```
<ActionDef name="CreateAutomaticAssignments">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="Assignment" 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"
xmlns:fn="http://www.rulecore.com/2008/functions" version="1.0">
  <xsl:template match="child::*">
    <base:EventBody>
      <xsl:for-each
```


DRAFT – version 0.1 – DRAFT

```
select="user:Views/user:View[@default='true']/user:Properties">
  <user:RequestId>
    <xsl:value-of
select="descendant::user:MatchedProperty[@name='RequestId']/user:Value/child::text()" />
  </user:RequestId>
</xsl:for-each>
<xsl:for-each
select="user:Views/user:View[@default='true']/user:Events/user:Event[@eventType='
BidRequest']">
  <user:Store>
    <xsl:value-of select="fn:get-property-value('Store')"/>
  </user:Store>
  <user:Driver>
    <xsl:value-of select="fn:get-property-value('EntityId')"/>
  </user:Driver>
  <user:AddresseeLocation>
    <xsl:value-of select="fn:get-property-value('AddresseeLocation')"/>
  </user:AddresseeLocation>
  <user:RequiredPickupTime>
    <xsl:value-of select="fn:get-property-value('RequiredPickupTime')"/>
  </user:RequiredPickupTime>
  <user:RequiredDeliveryTime>
    <xsl:value-of select="fn:get-property-value('RequiredDeliveryTime')"/>
  </user:RequiredDeliveryTime>
</xsl:for-each>
</base:EventBody>
</xsl:template>
</xsl:stylesheet>
</Body>
</Event>
</ActionDef>
```

Delivery Process Supervision

This set of rules covers the third phase of the specification, the Delivery Process phase.

Rule: No Bidders

Rule Definition

```
<Rule name="NoBidders" limit="?" evalMode="once" level="2">
  <Description>This is rule NoBidders</Description>
  <Initialize>
  <Assert>
```

DRAFT – version 0.1 – DRAFT

```
<Event>
  <base:XPath>sim:EventDef[@eventType="DeliveryRequest"]</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="DeliveryRequest"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="BidRequest"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryRequestCancellation"]</base:
XPath>
      </Event>
    </Type>
    <Match>
      <Property name="RequestId">
    </Match>
```

```
</Properties>
</View>
```

Situation Definition

```
<SituationDef name="NoBidders">
  <Detector>
    <All>
      <After timeframe="00:02:00">
        <Not>
          <EventPickup keep="last" see="new" name="P1" evalMode="once">
            <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[
@eventType="DeliveryBid"]</base:XPath>
          </EventPickup>
        </Not>
      </After>
    </Not>
    <EventPickup keep="last" see="new" name="P1" evalMode="once">
      <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@e
ventType="DeliveryRequestCancellation"]</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
```

DRAFT – version 0.1 – DRAFT

```
select="descendant::user:MatchedProperty[@name='RequestId']/user:Value/child::text()"/>
    </user:RequestId>
  </xsl:for-each>
</base:EventBody>
</xsl:template>
</xsl:stylesheet>
</Body>
</Event>
</ActionDef>
```

Rule: No Manual Assignment

This rule triggers and generate a NoManualAssignment event when no ManualAssignment event is seen within 1 minute after its corresponding ManualAssignmentRequest has been seen.

Rule Definition

```
<Rule name="NoManualAssignment" limit="?" evalMode="once" level="3">
  <Description>This is rule NoBidders</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="ManualAssignmentRequest"]</base:XPath>
      </Event>
    </Assert>
  </Initialize>
  <Views>
    <ViewRef name="NoManualAssignment">
      <base:XPath>sim:ViewDef[@name="NoManualAssignment"]</base:XPath>
    </ViewRef>
  </Views>
  <Situations>
    <SituationRef name="NoManualAssignment">
      <base:XPath>sim:SituationDef[@name="NoManualAssignment"]</base:XPath>
    </SituationRef>
  </Situations>
  <Actions>
    <SituationDetected situationName="NoManualAssignment">
      <ActionRef name="NoManualAssignment" eventVisibility="external">
        <base:XPath>sim:ActionDef[@name="NoManualAssignment"]</base:XPath>
      </ActionRef>
    </SituationDetected>
  </Actions>
```

```
</Rule>
```

View Definition

```
<View name="NoManualAssignment">
  <Description>This is view NoManualAssignment</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="ManualAssignmentRequest"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="NoBiddersAlert"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="ManualAssignment"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryRequestCancellation"]</base:
    </Event>
  </Type>
  <Match>
    <Property name="RequestId">
    </Match>
  </Properties>
</View>
```

Situation Definition

```
<SituationDef name="NoManualAssignment">
  <Detector>
    <All>
      <NotWithin timeframe="00:01:00">
        <EventPickup keep="last" see="new" name="P1" evalMode="once">
          <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@e
            ventType="ManualAssignmentRequest"]</base:XPath>
        </EventPickup>
        <EventPickup keep="last" see="new" name="P1" evalMode="once">
          <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@e
            ventType="ManualAssignment"]</base:XPath>
        </EventPickup>
      </NotWithin>
    <Not>
      <EventPickup keep="last" see="new" name="P1" evalMode="once">
```

DRAFT – version 0.1 – DRAFT

```
        <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@event
Type="DeliveryRequestCancellation"]</base:XPath>
    </EventPickup>
    </Not>
</All>
</Detector>
</SituationDef>
```

Action Definition

```
<ActionDef name="NoManualAssignment">
    <Event>
        <EventDef>
            <base:XPath>sim:EventDef[@eventType="NoManualAssignmentAlert" 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/child::tex
t()" />
                    </user:RequestId>
                </xsl:for-each>
            </base:EventBody>
        </xsl:template>
    </xsl:stylesheet>
        </Body>
    </Event>
</ActionDef>
```

Rule: Too Late Pickup

This rule triggers and generates a PickupAlert event when a PickupConfirmation event is not seen before RequiredPickupTime.

Rule Definition

```
<Rule name="TooLatePickup" limit="?" evalMode="once" level="3">
```

DRAFT – version 0.1 – DRAFT

```
<Description>This is rule TooLatePickup</Description>
<Initialize>
  <Assert>
    <Event>
      <base:XPath>sim:EventDef[@eventType="Assignment"]</base:XPath>
    </Event>
  </Assert>
  <Assert>
    <Event>
      <base:XPath>sim:EventDef[@eventType="ManualAssignment"]</base:XPath>
    </Event>
  </Assert>
</Initialize>
<Views>
  <ViewRef name="TooLatePickup">
    <base:XPath>sim:ViewDef[@name="TooLatePickup"]</base:XPath>
  </ViewRef>
</Views>
<Situations>
  <SituationRef name="TooLatePickup">
    <base:XPath>sim:SituationDef[@name="TooLatePickup"]</base:XPath>
  </SituationRef>
</Situations>
<Actions>
  <SituationDetected situationName="TooLatePickup">
    <ActionRef name="TooLatePickup" eventVisibility="external">
      <base:XPath>sim:ActionDef[@name="TooLatePickup"]</base:XPath>
    </ActionRef>
  </SituationDetected>
</Actions>
</Rule>
```

View Definition

```
<View name="TooLatePickup">
  <Description>This is view TooLatePickup</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="Assignment"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="ManualAssignment"]</base:XPath>
      </Event>
    </Type>
  </Properties>
</View>
```

DRAFT – version 0.1 – DRAFT

```
<Event>
  <base:XPath>sim:EventDef[@eventType="PickupConfirmation"]</base:XPath>
</Event>
<Event>
  <base:XPath>sim:EventDef[@eventType="DeliveryRequestCancellation"]</base:
XPath>
</Event>
</Type>
<Match>
  <Property name="RequestId">
</Match>
</Properties>
</View>
```

Situation Definition

```
<SituationDef name="TooLatePickup">
  <Detector>
    <All>
      <After timeframe="00:05:00">
        <attr:basetime><base:xpath>fn:get-property-value('RequiredPickupTime',
fn:get-init-event())</base:xpath></attr:basetime>
        <Not>
          <EventPickup keep="last" see="new" name="P1" evalMode="once">
            <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[
@eventType="PickupConfirmation"]</base:XPath>
          </EventPickup>
        </Not>
      </After>
    <Not>
      <EventPickup keep="last" see="new" name="P1" evalMode="once">
        <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@e
ventType="DeliveryRequestCancellation"]</base:XPath>
      </EventPickup>
    </Not>
  </All>
</Detector>
</SituationDef>
```

Action Definition

```
<ActionDef name="TooLatePickup">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="PickupAlert" 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/child::text()" />
        </user:RequestId>
        <user:Driver>
          <xsl:value-of
select="descendant::user:MatchedProperty[@name='Driver']/user:Value/child::text()" />
        </user:Driver>
      </xsl:for-each>
    </base:EventBody>
  </xsl:template>
</xsl:stylesheet>
</Body>
</Event>
</ActionDef>
```

Rule: Too Late Delivery

This rule triggers and generates a DeliveryAlert event when a DeliveryConfirmation event is not seen before RequiredDeliveryTime.

Rule Definition

```
<Rule name="TooLateDelivery" limit="?" evalMode="once" level="3">
  <Description>This is rule TooLateDelivery</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="Assignment"]</base:XPath>
      </Event>
    </Assert>
    <Assert>
      <Event>
```

DRAFT – version 0.1 – DRAFT

```
        <base:XPath>sim:EventDef[@eventType="ManualAssignment"]</base:XPath>
      </Event>
    </Assert>
  </Initialize>
  <Views>
    <ViewRef name="TooLateDelivery">
      <base:XPath>sim:ViewDef[@name="TooLateDelivery"]</base:XPath>
    </ViewRef>
  </Views>
  <Situations>
    <SituationRef name="TooLateDelivery">
      <base:XPath>sim:SituationDef[@name="TooLateDelivery"]</base:XPath>
    </SituationRef>
  </Situations>
  <Actions>
    <SituationDetected situationName="TooLateDelivery">
      <ActionRef name="TooLateDelivery" eventVisibility="external">
        <base:XPath>sim:ActionDef[@name="TooLateDelivery"]</base:XPath>
      </ActionRef>
    </SituationDetected>
  </Actions>
</Rule>
```

View Definition

```
<View name="TooLateDelivery">
  <Description>This is view TooLateDelivery</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="PickupConfirmation"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryConfirmation"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryRequestCancellation"]</base:
XPath>
      </Event>
    </Type>
    <Match>
      <Property name="RequestId">
    </Match>
  </Properties>
```

```
</View>
```

Situation Definition

```
<SituationDef name="TooLateDelivery">
  <Detector>
    <All>
      <After timeframe="00:10:00">
        <attr:basetime><base:xpath>fn:get-property-value('RequiredDeliveryTime',
fn:get-init-event())</base:xpath></attr:basetime>
        <EventPickup keep="last" see="new" name="P1" evalMode="once">
          <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@e
ventType="DeliveryConfirmation"]</base:XPath>
        </EventPickup>
      </After>
      <Not>
        <EventPickup keep="last" see="new" name="P1" evalMode="once">
          <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@e
ventType="DeliveryRequestCancellation"]</base:XPath>
        </EventPickup>
      </Not>
    </All>
  </Detector>
</SituationDef>
```

Action Definition

```
<ActionDef name="TooLateDelivery">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="DeliveryAlert" 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/child::tex
```

```
t()"/>
    </user:RequestId>
    <user:Driver>
        <xsl:value-of
select="descendant::user:MatchedProperty[@name='Driver']/user:Value/child::text()"
"/>
    </user:Driver>
</xsl:for-each>
</base:EventBody>
</xsl:template>
</xsl:stylesheet>
</Body>
</Event>
</ActionDef>
```

Driver Ranking Maintenance

This set of rules covers the fourth phase of the specification; the Ranking Evaluation phase.

Rule: DeliveryThresholdReached

This rule triggers and generates a DeliveryThresholdReached event when a certain number (here 20) of DeliveryConfirmation events has been seen for Driver.

Rule Definition

```
<Rule name="DeliveryThresholdReached" limit="?" evalMode="once" level="4">
  <Description>This is rule DeliveryThresholdReached</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryConfirmation"]</base:XPath>
      </Event>
    </Assert>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryAlert"]</base:XPath>
      </Event>
    </Assert>
  </Initialize>
  <Views>
    <ViewRef name="DeliveryThresholdReached">
      <base:XPath>sim:ViewDef[@name="DeliveryThresholdReached"]</base:XPath>
    </ViewRef>
  </Views>
```

DRAFT – version 0.1 – DRAFT

```
<Situations>
  <SituationRef name="DeliveryThresholdReached">
    <base:XPath>sim:SituationDef[@name="DeliveryThresholdReached"]</base:XPath>
  </SituationRef>
</Situations>
<Actions>
  <SituationDetected situationName="DeliveryThresholdReached">
    <ActionRef name="DeliveryThresholdReached" eventVisibility="external">
      <base:XPath>sim:ActionDef[@name="DeliveryThresholdReached"]</base:XPath>
    </ActionRef>
  </SituationDetected>
</Actions>
</Rule>
```

View Definition

```
<View name="DeliveryThresholdReached">
  <Description>This is view IncreaseRanking</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryConfirmation"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryAlert"]</base:XPath>
      </Event>
    </Type>
    <Match>
      <Property name="RequestId">
      </Match>
    </Properties>
  </View>
```

Situation Definition

```
<SituationDef name="DeliveryThresholdReached">
  <Detector>
    <EventCount equal="20">
      <EventPickup keep="last" name="P1">
        <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@eventType="DeliveryConfirmation"]</base:XPath>
      </EventPickup>
    </EventCount>
  </Detector>
</SituationDef>
```

Action Definition

```
<ActionDef name="DeliveryThresholdReached">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="DeliveryThresholdReached" 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:Driver>
          <xsl:value-of
select="descendant::user:MatchedProperty[@name='Driver']/user:Value/child::text()
"/>
          </user:Driver>
        </xsl:for-each>
        <user:NumberOfAlerts>
          <xsl:value-of
select="count(user:Views/user:View[@default='true']/user:Events/user:Event[@event
Type='DeliveryAlert'])"/>
          </user:NumberOfAlerts>
        </base:EventBody>
      </xsl:template>
    </xsl:stylesheet>
  </Body>
</Event>
</ActionDef>
```

Rule: Increase Ranking

This rule triggers and generates a RankingDecrease event when a DeliveryThresholdReached event is seen where NumberOfAlerts = 0.

Rule Definition

```
<Rule name="IncreaseRanking" limit="?" evalMode="once" level="5">
  <Description>This is rule IncreaseRanking</Description>
  <Initialize>
  <Assert>
```

DRAFT – version 0.1 – DRAFT

```
<Event>
  <base:XPath>sim:EventDef[@eventType="DeliveryThresholdReached"]</base:XPath>
th>
</Event>
</Assert>
</Initialize>
<Views>
  <ViewRef name="IncreaseRanking">
    <base:XPath>sim:ViewDef[@name="IncreaseRanking"]</base:XPath>
  </ViewRef>
</Views>
<Situations>
  <SituationRef name="IncreaseRanking">
    <base:XPath>sim:SituationDef[@name="IncreaseRanking"]</base:XPath>
  </SituationRef>
</Situations>
<Actions>
  <SituationDetected situationName="IncreaseRanking">
    <ActionRef name="IncreaseRanking" eventVisibility="external">
      <base:XPath>sim:ActionDef[@name="IncreaseRanking"]</base:XPath>
    </ActionRef>
  </SituationDetected>
</Actions>
</Rule>
```

View Definition

```
<View name="IncreaseRanking">
  <Description>This is view IncreaseRanking</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryThresholdReached"]</base:XPath>
th>
      </Event>
    </Type>
    <Match>
      <Property name="Driver">
    </Match>
  </Properties>
</View>
```

Situation Definition

```
<SituationDef name="IncreaseRanking">
```

```
<Detector>
  <EventPickup keep="last" name="P1" evalMode="once">
    <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@event
Type="DeliveryThresholdReached" and number(fn:get-property-
value('NumberOfAlerts')) = 0]</base:XPath>
  </EventPickup>
</Detector>
</SituationDef>
```

Action Definition

```
<ActionDef name="IncreaseRanking">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="RankingIncrease" 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:Driver>
          <xsl:value-of
select="descendant::user:MatchedProperty[@name='Driver']/user:Value/child::text()
"/>
        </user:Driver>
      </xsl:for-each>
    </base:EventBody>
  </xsl:template>
</xsl:stylesheet>
      </Body>
    </Event>
  </ActionDef>
```

Rule: Decrease Ranking

This rule triggers and generates a RankingDecrease event when a DeliveryThresholdReached event is seen where NumberOfAlerts >= 5.

Rule Definition

```
<Rule name="DecreaseRanking" limit="?" evalMode="once" level="5">
  <Description>This is rule DecreaseRanking</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryThresholdReached"]</base:XPath>
      </Event>
    </Assert>
  </Initialize>
  <Views>
    <ViewRef name="DecreaseRanking">
      <base:XPath>sim:ViewDef[@name="DecreaseRanking"]</base:XPath>
    </ViewRef>
  </Views>
  <Situations>
    <SituationRef name="DecreaseRanking">
      <base:XPath>sim:SituationDef[@name="DecreaseRanking"]</base:XPath>
    </SituationRef>
  </Situations>
  <Actions>
    <SituationDetected situationName="DecreaseRanking">
      <ActionRef name="DecreaseRanking" eventVisibility="external">
        <base:XPath>sim:ActionDef[@name="DecreaseRanking"]</base:XPath>
      </ActionRef>
    </SituationDetected>
  </Actions>
</Rule>
```

View Definition

```
<View name="DecreaseRanking">
  <Description>This is view DecreaseRanking</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="DeliveryThresholdReached"]</base:XPath>
      </Event>
    </Type>
    <Match>
      <Property name="Driver">
    </Match>
```

```
</Properties>
</View>
```

Situation Definition

```
<SituationDef name="DecreaseRanking">
  <Detector>
    <EventPickup keep="last" name="P1" evalMode="once">
      <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@event
Type="DeliveryThresholdReached" and number(fn:get-property-
value('NumberOfAlerts')) >= 5]</base:XPath>
    </EventPickup>
  </Detector>
</SituationDef>
```

Action Definition

```
<ActionDef name="DecreaseRanking">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="RankingDecrease" 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:Driver>
          <xsl:value-of
select="descendant::user:MatchedProperty[@name='Driver']/user:Value/child::text()
"/>
        </user:Driver>
      </xsl:for-each>
    </base:EventBody>
  </xsl:template>
</xsl:stylesheet>
      </Body>
    </Event>
  </ActionDef>
```

Rule: Improvement Note

Rule Definition

```
<Rule name="ImprovementNote" limit="?" evalMode="once" level="1">
  <Description>This is rule ImprovementNote</Description>
  <Initialize>
    <Assert>
      <Event>
        <base:XPath>sim:EventDef[@eventType="RankingDecrease"]</base:XPath>
      </Event>
    </Assert>
  </Initialize>
  <Views>
    <ViewRef name="ImprovementNote">
      <base:XPath>sim:ViewDef[@name="ImprovementNote"]</base:XPath>
    </ViewRef>
  </Views>
  <Situations>
    <SituationRef name="ImprovementNote">
      <base:XPath>sim:SituationDef[@name="ImprovementNote"]</base:XPath>
    </SituationRef>
  </Situations>
  <Actions>
    <SituationDetected situationName="ImprovementNote">
      <ActionRef name="ImprovementNote" eventVisibility="external">
        <base:XPath>sim:ActionDef[@name="ImprovementNote"]</base:XPath>
      </ActionRef>
    </SituationDetected>
  </Actions>
</Rule>
```

View Definition

```
<View name="ImprovementNote">
  <Description>This is view ImprovementNote</Description>
  <Properties>
    <Type>
      <Event>
        <base:XPath>sim:EventDef[@eventType="RankingDecrease"]</base:XPath>
      </Event>
      <Event>
        <base:XPath>sim:EventDef[@eventType="RankingIncrease"]</base:XPath>
      </Event>
    </Type>
  </Properties>
</View>
```

```
</Type>
<Match>
  <Property name="EntityId">
</Match>
</Properties>
</View>
```

Situation Definition

```
<SituationDef name="ImprovementNote">
  <Detector>
    <SequenceMatch>
      <EventPickup keep="last" see="new" name="P1" evalMode="once">
        <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@event
ntType="RankingDecrease"]</base:XPath>
      </EventPickup>
      <EventPickup keep="last" see="new" name="P1" evalMode="once">
        <base:XPath>sim:Views/sim:View[@default='true']/sim:Events/sim:Event[@eve
ntType="RankingIncrease"]</base:XPath>
      </EventPickup>
    </SequenceMatch>
  </Detector>
</SituationDef>
```

Action Definition

```
<ActionDef name="ImprovementNote">
  <Event>
    <EventDef>
      <base:XPath>sim:EventDef[@eventType="ImprovementNote" 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:Driver>
          <xsl:value-of
select="descendant::user:MatchedProperty[@name='Driver']/user:Value/child::text()
"/>
        </user:Driver>
```

```
        </xsl:for-each>
      </base:EventBody>
    </xsl:template>
  </xsl:stylesheet>
</Body>
</Event>
</ActionDef>
```

Entities

This section contains the entity definitions for all entity types that are used in the solution.

Entity: Driver

```
<Entity name="Driver">
  <Description>Entity: Driver</Description>
  <Properties>
    <Position/>
    <Zone/>
    <Ranking/>
  </Properties>
</Entity>
```

Events

This section contains the event definitions for all event types that are used in the solution.

Event: GPSLocation

```
<EventDef eventType="GPSLocation">
  <Description>Event: GPSLocation</Description>
  <Properties>
    <Position>
      <PointWGS84>
        <Coordinates>
          <base:XPath>string (base:EventBody/user:Coordinates)</base:XPath>
        </Coordinates>
      </PointWGS84>
    </Position>
    <Property name="EntityId">
      <base:XPath>string (base:EventBody/user:EntityId)</base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Example event:

DRAFT – version 0.1 – DRAFT

```
<?xml version="1.0" encoding="iso-8859-1"?>
<GPSLocation xmlns="http://www.rulecore.com/2008/user"
  xmlns:base="http://www.rulecore.com/2008/base" >
  base:eventClass="user"
  base:eventId="5459e063-dfc3-42e1-82ba-64e3332d5908"
  base:eventTimestamp="2009-10-07T16:33:47.000044Z">
  <base:EventHeader>
    <base:SecurityInfo>
      <base:Credentials>12345678-1234-1204-1234-123456789abd</base:Credentials>
    </base:SecurityInfo>
  </base:EventHeader>
  <base:EventBody>
    <EntityId>60</EntityId>
    <Coordinates>57.597476,12.063016,0.000000</Coordinates>
  </base:EventBody>
</Location>
```

Event: DeliveryRequest

```
<EventDef eventType="DeliveryRequest">
  <Description>Event: DeliveryRequest</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
    <Property name="Store">
      <base:XPath>string (base:EventBody/user:Store) </base:XPath>
    </Property>
    <Property name="AddresseeLocation">
      <base:XPath>string (base:EventBody/user:AddresseeLocation) </base:XPath>
    </Property>
    <Property name="RequiredPickupTime">
      <base:XPath>string (base:EventBody/user:RequiredPickupTime) </base:XPath>
    </Property>
    <Property name="RequiredDeliveryTime">
      <base:XPath>string (base:EventBody/user:RequiredDeliveryTime) </base:XPath>
    </Property>
    <Property name="MinimumRanking">
      <base:XPath>string (base:EventBody/user:MinimumRanking) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: BidRequest

```
<EventDef eventType="BidRequest">
  <Description>Event: BidRequest</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
    <Property name="Store">
      <base:XPath>string (base:EventBody/user:Store) </base:XPath>
    </Property>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
    </Property>
    <Property name="AddresseeLocation">
      <base:XPath>string (base:EventBody/user:AddresseeLocation) </base:XPath>
    </Property>
    <Property name="RequiredPickupTime">
      <base:XPath>string (base:EventBody/user:RequiredPickupTime) </base:XPath>
    </Property>
    <Property name="RequiredDeliveryTime">
      <base:XPath>string (base:EventBody/user:RequiredDeliveryTime) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: DeliveryBid

```
<EventDef eventType="DeliveryBid">
  <Description>Event: DeliveryBid</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
    <Property name="Store">
      <base:XPath>string (base:EventBody/user:Store) </base:XPath>
    </Property>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
    </Property>
    <Property name="CommittedPickupTime">
      <base:XPath>string (base:EventBody/user:CommittedPickupTime) </base:XPath>
    </Property>
  </Properties>
```

```
</EventDef>
```

Event: NoBiddersAlert

```
<EventDef eventType="NoBiddersAlert">
  <Description>Event: NoBiddersAlert</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId)</base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: ManualAssignmentRequest

Property DeliveryBids contains five highest-ranked Delivery Bid events.

```
<EventDef eventType="ManualAssignmentRequest">
  <Description>Event: ManualAssignmentRequest</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId)</base:XPath>
    </Property>
    <Property name="DeliveryBids">
      <base:XPath>string (base:EventBody/user:DeliveryBids)</base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: ManualAssignment

```
<EventDef eventType="NoBiddersAlert">
  <Description>Event: NoBiddersAlert</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId)</base:XPath>
    </Property>
    <Property name="Store">
      <base:XPath>string (base:EventBody/user:Store)</base:XPath>
    </Property>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver)</base:XPath>
    </Property>
    <Property name="AddresseeLocation">
      <base:XPath>string (base:EventBody/user:AddresseeLocation)</base:XPath>
    </Property>
  </Properties>
```


DRAFT – version 0.1 – DRAFT

```
<Property name="RequiredPickupTime">
  <base:XPath>string (base:EventBody/user:RequiredPickupTime) </base:XPath>
</Property>
<Property name="RequiredDeliveryTime">
  <base:XPath>string (base:EventBody/user:RequiredDeliveryTime) </base:XPath>
</Property>
</Properties>
</EventDef>
```

Event: NoManualAssignmentAlert

```
<EventDef eventType="NoManualAssignmentAlert">
  <Description>Event: NoManualAssignmentAlert</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: Assignment

```
<EventDef eventType="NoManualAssignmentAlert">
  <Description>Event: NoManualAssignmentAlert</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
    <Property name="Store">
      <base:XPath>string (base:EventBody/user:Store) </base:XPath>
    </Property>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
    </Property>
    <Property name="AddresseeLocation">
      <base:XPath>string (base:EventBody/user:AddresseeLocation) </base:XPath>
    </Property>
    <Property name="RequiredPickupTime">
      <base:XPath>string (base:EventBody/user:RequiredPickupTime) </base:XPath>
    </Property>
    <Property name="RequiredDeliveryTime">
      <base:XPath>string (base:EventBody/user:RequiredDeliveryTime) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: PickupConfirmation

```
<EventDef eventType="PickupConfirmation">
  <Description>Event: PickupConfirmation</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
    <Property name="Store">
      <base:XPath>string (base:EventBody/user:Store) </base:XPath>
    </Property>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: PickupAlert

```
<EventDef eventType="PickupAlert">
  <Description>Event: PickupAlert</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: DeliveryConfirmation

```
<EventDef eventType="DeliveryConfirmation">
  <Description>Event: DeliveryConfirmation</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: DeliveryAlert

```
<EventDef eventType="DeliveryAlert">
  <Description>Event: DeliveryAlert</Description>
```

```
<Properties>
  <Property name="RequestId">
    <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
  </Property>
</Properties>
</EventDef>
```

Event: DeliveryThresholdReached

```
<EventDef eventType="DeliveryThresholdReached">
  <Description>Event: DeliveryThresholdReached</Description>
  <Properties>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
    </Property>
    <Property name="NumberOfAlerts">
      <base:XPath>string (base:EventBody/user:NumberOfAlerts) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: RankingIncrease

```
<EventDef eventType="RankingIncrease">
  <Description>Event: RankingIncrease</Description>
  <Properties>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: RankingDecrease

```
<EventDef eventType="RankingDecrease">
  <Description>Event: RankingDecrease</Description>
  <Properties>
    <Property name="Driver">
      <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```

Event: ImprovementNote

```
<EventDef eventType="ImprovementNote">
  <Description>Event: ImprovementNote</Description>
```

```
<Properties>
  <Property name="Driver">
    <base:XPath>string (base:EventBody/user:Driver) </base:XPath>
  </Property>
</Properties>
</EventDef>
```

Event: DeliveryRequestCancellation

```
<EventDef eventType="PickUpConfirmation">
  <Description>Event: PickUpConfirmation</Description>
  <Properties>
    <Property name="RequestId">
      <base:XPath>string (base:EventBody/user:RequestId) </base:XPath>
    </Property>
  </Properties>
</EventDef>
```