Leveraging Event Manager for Key Data Changes Orchestration

Data Management Practice Team, Mastech InfoTrellis

Introduction

In a connected world, data should travel fast; in a real-time fashion, to serve the purpose – enabling the business to thrive. Data that is not available in time impacts business. Events that matter to business, that affects critical data changes, should orchestrate data changed sync up with connected systems that are the backbone of the business.

Statistics show that 200 businesses change addresses, 150 business telephone numbers will change, or be disconnected. Additionally, five suppliers or vendors were going through a rebranding, and retailers were losing USD 40 billion, or 3.5% of total sales due to product information inefficiencies. All of these taking place in one hour! Data that is not efficiently syndicated to the consuming systems by the System of Record hurts businesses.

We are illustrating IBM's Event Manager in action, in this whitepaper. We will see how it could seamlessly funnel event-driven data changes to connected systems just when they need them.

Contents

What is IBM InfoSphere® MDM Event Manager?

An Event Manager is one of the core components of IBM MDM solution, which detects events within InfoSphere MDM and performs relevant tasks like information proliferation asynchronously. Tasks are executed asynchronously to improve performance and throughput.

Some operations take a long time to complete. So, it makes sense to split those operations into two parts .i.e. a short synchronous interaction followed by a long asynchronous process. The end of the synchronous interaction responds to the user, and the remaining operation is performed asynchronously.

Real-Life Business Cases

Business Case 1:

A Global bank maintains Forex Exchange rates in one system. And, this rate is used by other downstream systems associated with currency exchange channels such as mobile, website, and branches. The exchange rate is fixed once in a day, mostly when the banking hour commences. The system that feeds the exchange rate sends daily updates at 9 AM to the MDM system. The MDM AE Event Manager triggers the update of the exchange rate to the downstream systems like CRM, ERP, and others.

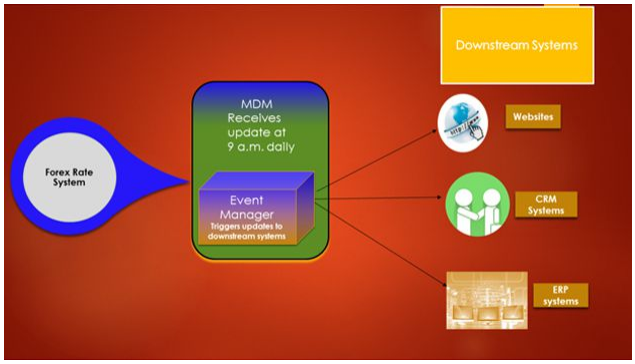


Figure 1: Forex Exchange Rate updates to downstream systems

Business Case 2

A vendor in the supply chain sends invoices to its suppliers daily. One of the suppliers changed the mailing address, and this was not updated in the CRM system. This is a perfect example of wreaking havoc when the supply chain is broken with wrongly addressed invoices.

Think of leveraging MDM AE Event Manager in this scenario. So, once the address changes in MDM, event is triggered, and all the other downstream systems will be updated with the latest address of the Supplier.

Business Case 3

A Global giant in tools manufacturing sells a specific product line at a 10% profit over the manufacturing cost incurred. In this case, the other subsystems such as e-Commerce Portal, ERP System, CRM System will also be maintaining the profit margin of 10 percent as seen in the picture below:

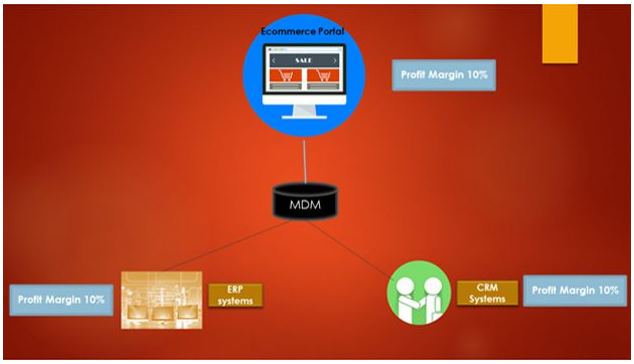


Figure 2: Unified Profit Margin maintenance -Source and Subsystems

To recover from dwindling sales, the client decided to jack up the profit margin from 10% to 12%. In this case, the Event Manager played the pivotal rule of orchestrating this change from the MDM system to other consuming systems. The event manager subsystem processes this change and sends a notification to other subsystems such as Ecommerce, ERP systems, and CRM systems. Thus, the profit margin at all associated subsystems is changed to 12 percent. The following diagram depicts the way IBM MDM's Event Manager handles this key data change.

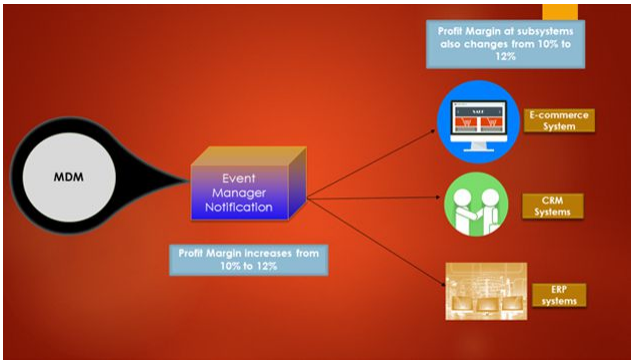
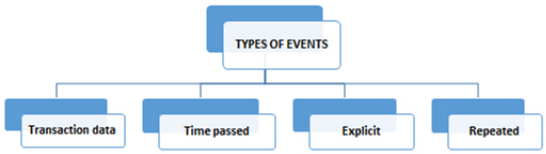


Figure 3: IBM Event Manager Data change Orchestration

Event Manager Capabilities

1. Detect business events based on the passage of time
2. Detect business events after business transactions have been executed
3. Record events that have occurred
4. Execute actions, such as trigger business transactions, as part of the business rules evaluating the occurrence of the event
5. Notify that an event has occurred, using the Notifications framework

Event Types



Transaction data event: This event occurs with a change in any key data change within MDM. A customer is updating his personal contact details, for example.

Time passed event: This event occurs over time. A promotion campaign expiring today should be informed to the e-Commerce systems, for example.

Explicit event: These events are typically user-defined. A category manager knows that a product category is being inactivated in two weeks. The category manager may want to persist in this type of event., for example.

Repeated event: This is used for repeated events. A banking user may enter a wrong username or password consecutively three times while accessing the banking portal. In such a scenario, the bank may want to lock the account for the day, and send a notification to the customer to avoid possible fraudulent transactions.

Event Manager Architecture

Now, let's dive deep into the inner workings. The InfoSphere® MDM Event Manager consists of five major subsystems

1. Services layer
2. Event detector message-driven bean (MDB) with the process controller
3. Event analyst or detector
4. Event persistence module
5. Notification module

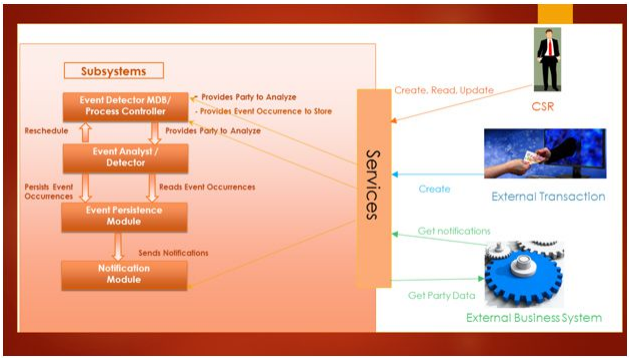


Figure 4: Event Manager Subsystems

Services layer: This layer provides a business interface to the Event Manager users. This interface consists of the ProcessController and EventServicesessionbeans. The business system calls the ProcessControllersessionbean to inform the Event Manager about a transaction performed against a particular business object. The ProcessControllerbean then sends this business object for processing to the event analyst or detector. This step is asynchronous, ensuring that the business transaction is not delayed by event processing.

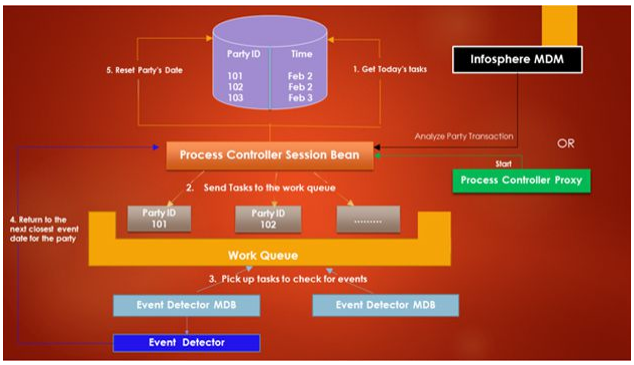


Figure 5: Event Manager Components

ProcessControllerbean with Event Detector MDB: This component ensures that business objects are sent to the event analyst or detector module for processing. The EventDetectionsScheduleController invokes the ProcessController bean according to the setting. The ProcessController bean checks for any business objects that are due for processing within a specific time and sends them to the event analyst or event detector.

Event analyst or detector: Executes event rules to

determine the list of the current event occurrences and the list of future potential event occurrences for a given business object.

Event persistence module: This module is responsible for persisting information about events that have taken place.

Notification Module: Sends notifications to other business systems notifying the changes effected by the event.

Event Manager Data

The Event Manager Data model consists of three operational tables:

1. PROCESSCONTROL
2. PROCESSACTION
3. EVENT

These tables hold data about business objects and events that are added by the Event Manager when it detects the related event. Click **here**. to know more about the data model from the official IBM documentation.

Building Business Solutions Using Event Manager Framework

Steps to implement IBM MDM Event Manager for any business case

Step1: Build Event Business Adapter Class

Business Adapter is called from the EventDetectorMDB. More specifically, called from the EventTaskObject by getDataObjects () method. Business Adapter has to implement the EventBusinessAdapter interface, and then carry out this method:

publicDataObjectCollectiongetDataObjects (SerializabletransObj, String busObjkey, String busEntity)

```
public class EventManagerBusinessAdapter implements IEventBusinessAdapter {

@Override
public DataObjectCollection getDataObjects(Serializable transObj, String busObjKey,
String busEntity) throws EMException {
TCRMPartyBObj partyBObj = partyComponent.getParty(partyIDPK, "2", control);
dObjColl = new DataObjectCollection();
asyncBObj = new MDMAsyncBObj(null, busObjKey, null, null, new DWLControl());

if(partyBObj!= null) {
      asyncBObj.setMdmBObj(partyBObj);
}
dObjColl.add(asyncBObj, busEntity, busObjKey);
```

For more details on Adapter Class refer the IBM Site **link**

Step 2: Configure Blueprint for Business Adapter

The snippet below provides a sample blueprint fragmentation to accommodate a custom Event Business Adapter.

```
<service id="EventManagerBusinessAdapter"
interface="com.ibm.mdm.common.servicefactory.api.CommonServiceFactory">
<service-properties>
      <entry key="common.service"
      value="EventBusinessAdapter.com.ibm.mdm.eventManager.adapater.
      EventManagerBusinessAdapter " />
  </service-properties>
      <bean class="com.ibm.mdm.common.servicefactory.CommonServiceFactoryImpl">
<argument type="java.lang.Class"
value="com.dwl.commoncomponents.eventmanager.IEventBusinessAdapter" />
          <argument type="java.lang.Class"
value="com.ibm.mdm.eventManager.adapater. EventManagerBusinessAdapter " />
      <argument ref="blueprintBundle" />
  </bean>
</service>
```

Step 3: Event Manager Rule class

Event Manager Rule is called from EventDetectorMDB's executeRules method. The EventDetectorMDB looks into the configuration and finds out the rule class and calls ***execute()*** method provided by BaseRule, which is a superclass of Event Manager Rules.

Step 4: Event Manager Custom Rule Class

```
public class EventManagerProcessRule extends BaseRule{

public void executeRules(EventTaskObject eventTaskObj) throws EMException {
          Serializable serializableData = eventTaskObj.getDataObj();

if (asyncBObj != null) {

      TCRMPartyBobj partyBObj = (TCRMPartyBobj) asyncBObj.getMdmBObj();

}
/* Here construct response and publish it to downstream System, and downstream system
will compose greeting message and send it to customer.*/
```

Author

Data Management Practice Team, Mastech InfoTrellis.

About

Mastech InfoTrellis partners with enterprises to help them achieve their business objectives by leveraging the power of data to derive deep, analytical insights about their business and its operations. We accelerate business velocity, minimize costs, and drastically improve corporate resiliency through personalized, process-oriented programs, consisting of strategy, data management (including master data management), business intelligence and reporting, data engineering, predictive analytics, and advanced analytics. Part of the NYSE-listed,$177.2M, digital transformation IT services company, Mastech Digital; we drive businesses forward around the world, with offices spread across the US, Canada, India, and Singapore.