CHAPTER 6

ORCHESTRATING SUPPLY CHAIN ACTIVITIES

In each business, the supply chain management encounters the extraordinary difficulty to respond and progress. The businesses are wrestling with supply chains to tackle the multifaceted operations impacted by the most disruptive market pressures these days. There exist a lot of pressures such as the worldwide financial crisis, worldwide recognition of sustainability, weather change, insufficiency of assets and fuel cost creates universal trading complications and necessitates the fulfillment, delays and mislaid sales issues. All of these pressures are producing great impacts and grind down the capability of the supply chain to draw off that the majority fragile of enterprise juggling activities achieving most favorable balance in cost & customer satisfaction. The main aim of every industry is to perform the operations at the lowest cost consistent with the target quality and customer service.

The huge shifts in working environments are forcing supply chain managers to modify the approach of addressing the market, their suppliers and their distributors. These factors formulate the managers to alter the way of organizing, operating and managing the interactions with their customers. Hence there is the perfect storm of market pressures affecting the performance parameters of the supply chain management system. So achieving the maximized customer satisfaction through maintaining lowest product production cost in the supply chain, it is essential to bring specializations to all the components of the chain.

6.1 SERVICE ORCHESTRATION

The business processes plays extremely significant in administrating the competitive challenges in any company. The service oriented architecture (SOA) may be considered as an approach to budding enterprise systems by loosely coupling interoperable services. Instead of writing the program again, the SOA provides the mechanism and a collection
of various services to design new business services through reusing the codes of previous designed programs for the enterprise. The critical feature of service oriented architecture is service orchestration. The SOA based enterprise systems and integration projects are directly based on thriving service orchestration and superior service orchestration capabilities. The service orchestration supports the synchronization between numerous services uncovered in the solitary cumulative service. The enterprise application developers exploit the service orchestration mechanism to sustain the automation of business processes through utilizing the loosely coupling services across dissimilar applications and enterprises. It is the grouping of service connections to produce higher-level business services.

The service orchestration shares the information through message exchange in the domain layer of enterprise applications. The messages exchange takes place regarding the predetermined business logic and execution order to implement the compound service to fulfill the customers’ needs. Let discuss the example to discuss the usage of the service orchestration. A retailer requests supply chain manager to provide the product stock request on fulfilling customer needs and uses an automated stock request module. The retailer accesses the stock request module to initialize stock request and then it is sent to the SCM orchestrator which operates as the central messaging engine. The SCM orchestrator calls and invokes other services to process that request. Other services include a service to obtain current manufactures capabilities to retrieve a list of suppliers to avail the raw materials from manufacture department and provide the availability of manufactured products. With the help of this information, the orchestrated services comprise the sock request service, which then returns a list of quotes from impending retailers and response original request. This example illustrates the service orchestration as a more essential facet for effectively implementing SOA.

6.1.1 Challenges of service orchestration and SOA

The service orchestration and SOA are being applied as reasonably unsophisticated concepts of fulfilling the business requirements of the customers. The service orchestration and service oriented architecture can be tricky to attain devoid of the precise set of tools. The existing service orchestration scheme requires in elevated
economic expenditure to orchestrate services programmatically. It minimizes the pattern of tradition coding through reusability of the existing programs code.

Next challenge regarding the service orchestration is to choose the right platform. Having selection of the right platform, the SOA generates the programming environment to support business processes with IT systems and diminish the costs involved in designing and implementation phase required to grip the changes in customer demands and the integration of new applications. During the composition of new business services, the processes orchestrate one service with others successfully.

6.2 SUPPLY CHAIN ORCHESTRATION

Due to more challenges faced in the markets the SCM necessitate innovative approach to achieve the enhanced performance. As discussed in the last sections, these pressures make the SCM manager to think about the supply chain orchestration for improving the performances of the SCM. There are numerous anxieties to satisfy before selecting the SCM approaches as given below:

- Current business model is still the precise business model.
- Best possible solutions to address all requirements of supply chain.
- The SCM is capable of managing the conflicting demands.

The SCM challenges are being faced from the customer demand processing and ending with order management, fulfillment and delivery to the end customers. There exists following three key characteristics that help in improving the performance of the SCM as given below:

- **Orchestration**
  This factor helps in managing the combination of complex business process. This task is required to attain best possible solutions for end-to-end processes in the supply chain. This task generates the step change in operating performance of the SCM system.

- **Collaboration**
  Next factor may be helpful in enhancing the SCM performance is the implementation of collaborative supply chain delivery models. It benefits of
working in collaboration with competitors, suppliers, distributors and transport service operators.

- **Independence**
  The independence factors are required to finish the supply chain process of the product deliverance of mutual services by a third party to convince the global business competition. This factor provides the shortage of fulfilling the customer constraint to convey the most favorable arrangement of third parties in the resolution for the customer requirements.

![Figure 6.1 Supply Chain levels](image)

6.2.1 **Benefits of supply chain orchestration**

The supply chain orchestration involves the integration of various components likes the labors, activities, and technology expertise with business accelerators to accomplish most favorable solutions for end-to-end processes in the supply chain. The supply chain orchestration can convey the lowly sustainable price of operations reliable on the objective excellence and buyer services. The supply chain orchestration can offer benefits in the areas of process excellence, IT, visibility, and sustainability.
In the process excellence field, the supply chain orchestration consists the exploitation of comprehensive progression replicas for consistency and accelerative apparatus. It can convey noteworthy improvements in customer accomplishment and client services through supplementary precise, proficient, standardized and steady processes. The IT technologies include the cloud computing, SaaS, contemporary workflow tools, business process management and business intelligence systems. On the spot of visibility, the supply chain orchestration offers the faultless sight of third-party systems from various components likes the logistics providers, freight forwarders, roads, rails, and other operators. And in the area of sustainability, the supply chain orchestration guarantee that supply chain activities follows the innovative key performance indicators (KPIs) & comprises built-in sustainability management tools and features. The supply chain orchestration provides the following benefits as given below:

- The supply chain orchestration with the help of the collaboration features combines the supply chain actions of similar companies and competitors. It can deliver cost reductions of more than 20% in the regions of transport costs.
- With the help of the independence features, it handles the third party issues with reliable product delivery with improved customer satisfactions.
- It is capable of managing the global business competitions successfully.
- It can lessen lead times and supply chain cost. The reduced cost improves the supply chain performance.
- It reduces supply chain timelines to avoid the dead time for maintaining the customer satisfactions.
- It is capable of decreasing manual process management. The mechanism of managing the business process manually is the untrustworthy and unsafe
- It supports the enhancement in the supply chain visibility. The supply chain visibility resolves the problems faced with the supply chain entities.
- It assimilates every essential enterprise business partner. It compliments the every supply chain partners necessitated to complete the supply chain cycle.

These facts show the advantages of the supply chain orchestrations in running the supply chain activities. All entities involved in the supply chain orchestration are being shown in the figure 6.2 as given below:
As shown in the figure 5.2 the SCM contains following process as given below:

- Order Management
- Planning
- Distribution
- Supplier Management
- Sourcing
- Inventory Management
- Freight Management
- Master Data Management

These processes are the core of the SCM process initializing from customer requirement with ending to the reliable delivery of the product to end users.

### 6.3 ORCHESTRATION ENGINE

The supply chain management system involves various entities and process taken place during the operation of fulfilling the customers’ requirement. For successfully completing the SCM operations, the demand forecasting should be performed in the right way to manipulate the customer actual product demands. The SCM involves the flow of
materials, cash and the information. There are following problems faced in running the supply chain management activities as given below:

- The business process changes naturally transform with respect to the changing in roles and responsibilities of various the SCM entities, the information supplied to various entities and their flow between them. For example, for fulfilling the requirement of the customer product, the retailers send the request to the distributors. However, generally it doesn’t require the product stock status from the manufacturers.

- The SCM entities within the whole operations are ignorant of the worldwide business competitions. At time point when their pre-requisites are fulfilled, these entities don’t distinguish about past happened, not know about future plans to gain more profits such as customer satisfactions.

- At every instance of the time, the SCM process changes with customer requirement. It is not possible to modify business processes quickly.

- The SCM entities are interested in a range of operational aspects of their trade activities. For example, the suppliers want to know about the customer demand for upcoming month for the purpose of designing the production scheduling.

All these issues faced in the SCM may be resolved through coordination aspect which is successfully furnished by the orchestration engine. It implements the business process and deals exclusively with the coordination facet. With efficient coordination between the SCM entities, the efficient decisions are being taken by SCM manager. The orchestration engine provides following advantages in handling the business process as given below:

- **Superior modifiability**
  The orchestration engine is capable of handling the changing the process characterization. It defines the operational functionalities of the business process involved in the supply chain process. It improves the SCM application’s suppleness.

- **Simpler application code**
  The orchestration engine makes the way of designing the application simpler. There is no need of writing the glue code by the developers for the purpose of code compatibility. The glue code is not included in the application logic.
• **Enhanced reusability**
  It supports the feature of reusing the existing program code without writing the code again for process control flow. The developer can utilize the reusable code again and again as needed.

• **Domain independence**
  The orchestration engine deals only with control & data flow. It enables the developers to design the applications in various domains. It is dependent of the particular domain. The developers orchestrate business processes from any application domain such as financial, insurance, health care, telecommunications, etc.

These facts show the benefits of the OE in the field of designing the business applications used for financial, insurance, health care, telecommunications, etc.

### 6.3.1 Orchestration language

The orchestration engine provides the mechanisms for coordinating the supply chain entities such as suppliers, manufacturers, distributors, retailers and customers etc. participating in the business process. Unfortunately these entities have no inherent awareness about whole SCM behavior. The orchestration engine specifies following factors for maintaining the coordination between various entities as given below:

- Order by transferring the control to each entity.
- Specifying the data passed into and retrieved from each entity.
- Exceptions handling.

These tasks are being handled by the orchestration engine through the help of the specified language known as the orchestration language. The orchestration language defines the steps for maintaining the coordination between various entities.

The Orchestration Language is different from general-purpose programming languages as it focuses on the abstractions necessary to illustrate the coordination facet. The orchestration engine transfers control to and data between the business process entities. The orchestration languages entail dissimilar abstractions than the ones provided by conventional programming languages such as Java or Ruby. However, having access to
constructs such as variables, objects and messages from orchestrations increases their self-expression. The orchestration languages contain following elements as given below:

- The orchestration languages consist the diminutive set of uncomplicated control structures such as sequence, repetition, conditional, fork, and join etc. These structures can symbolize a wide range of coordinated behaviors that helps to represent some complex control flow constructs particularly involving concurrency.

- The activity plays very important role in managing the orchestrations. The activity execution represents the spot which indicates the control flow crosses the margins between the orchestration and application dominions. The entities receive the data to bring information into the orchestration by the orchestration engine.

- The control flow is the major mechanism for defining the coordination between various business processes. It may be in various forms during the executions of business process. The sequence implements a linear flow of control. The orchestration language shatters this linearity with the help of the conditional activity. The conditional activity provides another way of composing activities. It holds two different children activities and control flow is being passed to only single activity depending on the condition.

- Another more important element of orchestration languages is the repetition. The While activity endows with the method of repeating one or numerous step. Like the Conditional, the While has the circumstance that establishes the flow of control. The orchestration engine estimates the circumstance and executes the While’s body only if the condition is true. The engine repeats this process until the condition is no longer satisfied.

The Fork and Join control add support for concurrency and synchronization features for managing the execution of multiple business processes. The Fork control structure enables the developers to confirm the execution of the business process by different entities at the same time. The Join control provides a synchronization point through OR, XOR and AND logic operators. After combing these logic operators these Join control statement becomes OR-Join, an XOR-Join and AND-Joins. These control statements synchronize the external concurrent orchestrations in different ways. The OR-Join
continues as soon as the control flow from any one of the incoming orchestrations reaches it. It allows any control flow of any single incoming orchestrations. In distinction, an AND-Join waits for all the synchronized orchestrations to reach it before it continues execution. It allows when both control flow of incoming orchestrations. The XOR-Join continues as soon as the control flow from only one of the incoming orchestrations reaches it. It does not allow for both control flow of incoming orchestrations.

All these control structures represent a basic set of abstractions that allows the developers to define the coordination aspect. Various control structures are transposable. The orchestration language is comprised of activities, then the grammar would look like the following:

\[
\text{activity := Primitive|Sequence|While|Fork|Join}
\]

\[
\text{Sequence := Primitive*}
\]

\[
\text{While := Condition activity activity}
\]

\[
\text{Fork := activity *}
\]

\[
\text{Join := activity *}
\]

\[
\text{Condition := expression evaluating to a Boolean}
\]

The Orchestration Language must also endow with a means of composing Primitives, thus allowing its users to define more realistic business processes. The Sequencing is the simplest form of composition. The sequence defines a sequential ordering between two or supplementary communications with the entities. We can say that it enables users to identify orchestrations between the various process entities. The Business Process Execution Language (BPEL) is one of the orchestration languages. It is the standard executable language for defining the actions within business processes with web services. The process in BPEL export and import information by using web service interfaces completely.

The Orchestration Context facilitates the OE to bypass the outputs from every activity to the subsequently and the data flow. It provides the resolution to the following situations as given below:

- Several activities only necessitate data provided by the supplicant. Other activities require information formed by the process.
• Various outputs are worn by other activities; other outputs encompass the efficiency of the progression.
• Some situations involve knowledge concerning the downstream activities in categorize to abandon.

The Orchestration Engine involuntarily generates the orchestration context for each orchestration, and maintains it during the orchestration’s existence. The orchestration context carries orchestration significant information about the execution of the business processes. The OE utilizes this context to bring information among various entities and stock up supplementary orchestration significant data.

6.4 BPEL

The BPEL stands the Business Process Execution Language which contains the mechanisms for composing numerous synchronous and asynchronous services into shared & transactional process flows. BPEL provides the following features:
• Web services/WSDL used as a component model
• XML used as data model providing data loose-coupling
• Synchronous and asynchronous message exchange patterns
• Deterministic and nondeterministic flow coordination
• Hierarchical exception management
• Long-running unit of work/compensation

The BPEL specification was presented by the Organization for the Advancement of Structured Information Standards (OASIS) in March 2003. The features of BPEL are being supported by other languages like Java and Ruby for industry applications. This language provides multiple advantages in designing the enterprise applications implementing related business processes in a customary and transferable manner. It prevents vendor-specific rules to the extent not before probable.

6.4.1 Elements of BPEL

All BPEL elements are written in XML language which enables to use XML queries for expressions. The XML provides the multiple advantages over other approaches. XML provides the mechanism that makes the user to read and understand the information easily. It consists no predetermined collection of tags. There exists the simple way to
create new tags as the user requirements. XML tags illustrate significance not appearance. The look and feel of an XML manuscript can be proscribed by XSL style sheets that allow the look of a document to be distorted without modifying the contents of the document. Numerous sights of the similar content are easily rendered. XML documents can consist of nested elements that are scattered over numerous distant servers. XML is presently the majority complicated configuration for distributed data. Let’s discuss the various elements as given below:

6.4.1.1 Process

The BPEL process forever starts with the `bpel:process` element. This element specifies the name of the process along with a target namespace. This element should contain least one activity. The activity can be a basic activity or a structured activity.

```xml
<bpel:process name="pcname">
  activity
</bpel:process>
```

In `bpel:process` element there are numerous attributes which help in adding to supervise dissimilar options of the process. It also provides the alternative to stifle join failures.

6.4.1.2 Partners

The BPEL provides the explanation of the business process interaction among partners communicating with Web Services. With the help of the partners elements that relationships between the business processes may be précised among their associates. Every partner maintains the WSDL description file that explains the ways of communicating with this partner for the purpose of the interaction of the BPEL process. BPEL modeled the business process as a `partnerLink`. Every `partnerLink` is characterized by a `partnerLinkType` which is defined in the WSDL definition. A `partnerLinkType` specifies the role and the type of a partner.

```xml
<bpel:partnerLinks>
  <bpel:partnerLink name="plname" partnerLinkType="pltype"
    myrole="plname" partnerRole="plrole">
    </bpel:partnerLink>
  </bpel:partnerLinks>
```
Every *partnerLink* specifies the unique name for identifying the *partnerLink*. The *myRole* specifies the role & responsibilities of the partner link. & the attribute *partnerRole* specifies the role of the partner. The *partnerLinkType* attribute is used with only one role.

### 6.4.1.3 Variable

As other programming languages, the BPEL use variables for storing the values & helps in arithmetic calculations. The variable is always associated with a message from a WSDL. In the WSDL file, the message is being structured as given below:

```
<message name="supplychain">
    <part name="supplier" type="xsd:string"/>
    <part name="manufacture" type="xsd:string"/>
    <part name="retailer" type="xsd:string"/>
</message>
```

Variables suggest the prospect to accumulate the messages that clutch the condition of the method. The messages pass between multiple partners. In BPEL, The variables are capable of storing the data which are derived from the basis of the states of the business process and never shared between the partners. There are three types of variables:

- WSDL message type
- XML Schema simple type
- XML Schema element.

The syntax of a variable declaration is shown in the code example below.

```
<variables>
    <variable name="varname"
        messageType="messname"
        type="messtype"
        element="messele"/>
</variables>
```

The variable name should be unique. The *messageType* attributes can be used to identify the type of the variable. Every variable has to be declared in a certain scope and only belongs to this scope and underlying scopes. The variable may be declared globally to make it the visible through the whole process.
Mostly the variables are used to drive and collect information, but BPEL also gives the possibility the read the content of a variable in the process. This can be done with an expansion defined through the BPEL namespace. The function getVariableProperty can examine a value from a variable as given below.

```
getVariableProperty ('variableName', 'propertyName')
```

The assignment operation on the variable is being performed. It involves the mechanism of copying the data from one variable to the other in a business process. The assign activity can perform the task of copying new data into a variable.

```
<assign>
  <copy>
    <from variable="source" part="srcname"/>
    <to variable="destination" part="dstname"/>
  </copy>
</assign>
```

The `to` and `from` attributes can also be changed to copy from other places. For example, you could copy from a partnerLink to a certain variable.

### 6.4.1.4 Basic activities

Every basic activity has numerous customary attributes and elements which can be used to indicate certain properties. There are following basic activities as given below:

- **Invoke**

  The invoke activity allows the process to call other Web Service that acted as a partner. The invoke activity take place either asynchronous or synchronous manner. In the asynchronous mode, the invoke only requires to identify an input variable which has no direct reaction having no output variable. In synchronous mode, then invoke requires both an input and an output variable. The synchronous invoke can return a WSDL error message. This will result in a BPEL error which could be handled locally.

  ```
  <invoke partnerLink="plinkename"
    portType="ptype"
    operation="opname"
    inputVariable="invarname"
  ```
Receive
The business process provides the services to its partners to help of receiving and matching reply activities. The receive activity specifies three attributes: partnerLink, portType and operation. The receive activity plays an important role in the life cycle of a business process. There are predefined ways to begin the process by initializing the attribute createInstance to yes value in the receive activity. Its default value of this attribute is no. The process shouldn't enclose two receive activities having same partnerLink, portType, operation and correlationSet.

\[
\text{receive partnerLink="parlinkname"}
\text{portType="ptypename"}
\text{operation="opname"}
\text{variable="varname"}
\text{createInstance="yes|no"}\
\]

Reply
The reply activity sends the response after calling the receive activity. The reply is only useful in the synchronous interaction. The reply activity transfers data by specifying a variable. It can only be placed after receive activity.

\[
\text{reply partnerLink="plname"}
\text{portType="ptname"}
\text{operation="opname"}
\text{variable="varname"}
\text{faultName="fname"}\
\]

6.4.1.5 Signaling faults
The process is capable of reporting an internal fault though throw activity. Every fault must assign the globally unique name. The fault activity specifies the value of
the faultVariable. This variable contains supplementary information about the fault. During handling the fault, this information is being utilized to analyze the generated fault.

```xml
<throw faultName="fname" faultVariable="fvarname" />
```

The *wait* activity makes the business process to erect it for a certain time to sleep or wait till a specified deadline has accepted.

```xml
<wait (for="duration-expr" | until="deadline-expr") />
```

To indicate that process is doing nothing in fault handling, the empty activity is used.

```xml
<empty />
```

### 6.4.1.6 Structured activities

Structured activities enable the developer to structure a BPEL process with a description of the flow of a process by structuring basic activities. In this structure, it manages the data flow, fault handling and coordination of messages between the processes. The structured activities of BPEL are:

- Basic sequence control between various activities is presented by the sequence, switch and while activities.
- Synchronization and concurrency of activities is handled by the flow activity.
- The pick activity manages the choice based on information from the outside.

Structured activities can be used recursive and it's important to see that structured activities can be used in any way to create process flow.

#### Sequence

The sequence activity manages the execution of one or more activities sequential in the order of placing within a sequence element. The sequence activity stops when all activities within it are executed.

```xml
<sequence>
  activities
</sequence>
```

#### Switch

The switch activity is capable to execute the multiple activities on the basis of specified conditional behavior. It maintains the ordered list of conditional branches. Every branch is defined by a case element followed by one optional
otherwise element. The case element in the switch defines the sequence of process placed. The activities specified in the case are executed depending on the condition. When the condition of the case is true then particular activity is being executed otherwise another element are executed. The switch activity is done when all the activities of one of the branches are completed.

\[
\begin{align*}
\text{<switch>}
\text{<case condition="bool-expr">}
\text{activity}
\text{</case>}
\text{<otherwise>}
\text{activity}
\text{</otherwise>}
\text{</switch>}
\end{align*}
\]

- **While**

  The while activity executes the specific activities in an iterative way. The activities in the while activity are executed until the Boolean expression in the condition attribute is true.

\[
\begin{align*}
\text{<while condition="bool-expr">}
\text{activity}
\text{</while>}
\end{align*}
\]

- **Flow**

  The flow activity supports the execution of numerous activities in parallelism. The flow activity is considered as done after execution of all activities. The flow activity supports the synchronization in various activities within the flow.

\[
\begin{align*}
\text{<flow>}
\text{activities}
\text{</flow>}
\end{align*}
\]

### 6.4.1.7 Compensation handler

The BPEL consist the compensation handler component which is capable of defining the set of activities to resolve the problem faced during the execution of the process. This
component may be initialized from the process itself to disengage definite steps which have already been completed. The compensation handler can also classify substitution steps which require performing them in case of occurrence of the certain events.

6.4.1.8 Fault handler

The elective fault handlers identify the set of fault handling activities called catch activities. The catch activity catches the fault by its faultName and faultVariable. In case of occurring fault without a faultName, then the fault will be associated with every faultHandler having the same type of faultVariable. To catch the activities, there is one catchAll activity that catches the entire fault which are not handled by the defined catch activities.

```xml
<faultHandlers>
  <catch faultName="fname" faultVariable="fvarname">
    activity
  </catch>
  <catchAll>
    activity
  </catchAll>
</faultHandlers>
```

For having no catch or catchAll activities, the fault will be given to the parent scope. In case of fault which isn't caught by the global scope, then the process will stop with a terminal activity.

6.4.1.9 Event handler

The event handler in the BPEL is used to specify the events to handle the happening the particular events. It stands loose from the rest of the activities in the scope so when there are activities in the scope active, it's still possible to receive an event in the event handler.

```xml
<eventHandlers>
  <onMessage partnerLink="plname"
    portType="ptname"
    operation="opname"
    variable="varname">
    activity
```

These elements are core components of the BPEL. They perform the various tasks when they are being embedded in the BPEL code. The simplified form of the process in BPEL would look something like the code below.

```
<bpel:process name="SupplyChain"

targetNamespace="http://supplychain"

suppressJoinFailure="yes"

xmlns:tns="http://supplychain"

xmlns:bpel="http://docs.oasis-open.org/wsbpel/2.0/process/executable"
>}

<!-- Import the client WSDL -->

<bpel:import location="SupplyChainArtifacts.wsdl"

namespace="http://supplychain"

importType="http://schemas.xmlsoap.org/wsdl/" />

<!-- PARTNERLINKS -->

<!-- List of services participating in this BPEL process -->

<!-- The 'client' role represents the requester of this service. -->

<bpel:partnerLink name="client"
partnerLinkType="tns:SupplyChain"
myRole="SupplyChainProvider"
/
</bpel:partnerLinks>

<!--
=========================================================================
== -->

<!-- VARIABLES -->
<!-- List of messages and XML documents used within this BPEL process -->
<!--
=========================================================================
== -->

<bpel:variables>
<!-- Reference to the message passed as input during initiation -->
<bpel:variable name="input"
messageType="tns:SupplyChainRequestMessage"/>
<!--
Reference to the message that will be returned to the requester
-->
<bpel:variable name="output"
messageType="tns:SupplyChainResponseMessage"/>
</bpel:variables>

<!--
=========================================================================
== -->

<!-- ORCHESTRATION LOGIC -->
<!-- Set of activities coordinating the flow of messages across the
services integrated within this business process -->
<!--
=========================================================================
== -->
This example represents the model of the supply chain orchestration engine which takes the supply chain parameters as the input and generates the output.

6.5 SUPPLY CHAIN ORCHESTRATOR

The supply chain orchestration involves the coordination between entire major activities occurred in whole supply chain operation. It occupies all of the processes related to the material, cash and information from the processing of the sales order, demand and delivery of product, warehousing and luggage compartment, picking and dispatch, through to the delivery of goods and consequent cash compilation. The key intention is to maximize the return on resources and assets while optimizing customer order throughput and speed of delivery and minimizing wastage and the cost of error.

Typically the supply chain management involves improving and integrating major processes into a seamless flow of accurate information, allowing household tasks to be superior enabled, processes to be monitored and change controls effortlessly executed transversely the following procedures:

- Demand forecasting
- Product Order flow
- Effectiveness of resource planning
- Storehouse throughput
- Inventory control
- Transport planning
- Delivery effectiveness
- Cash collection cycle
- Fulfillment with regulation
- Faithfulness to business authority necessities

The supply chain orchestrator is the supply chain entity whom aimed to improve the performance of every process involved in the SCM activities. It has sufficient authority to manipulate all entities of the supply chain such as suppliers, manufacturers, distributors, retailers and the shopkeepers. The orchestration is typically an managerial lead squad duty which is being championed by an entity coordinating the group drawn from all levels of the functions. Orchestrators not only rely on information drawn from their own experience but also from creative and innovative technology suppliers, who draw from their experience of implementing change across a variety (hundreds) of deployments. The Orchestrator is responsible for managing the functions across the supply chain. In some situations, the OEM or product proprietor may fulfill this requirement. The orchestrator is capable to make the decisions for preparing the coordinated and cohesive way to support the goals and specific needs of the brand, the specific product, and ultimately of the end customer. And in terms of the chain of responsibility, they literally carry liability insurance to address warranty, accidents, damage and injury that may occur across the chain. The OEM and/or Freight Forward may compliment these financial liabilities. In other cases, the brand owner may hire a supply chain Orchestrator like Li & Fung to perform these functions.

In this case, the Orchestrator interfaces deeply and frequently with the brand owner to understand and fulfill the end customer requirements and the orchestrator possesses deep domain expertise, such as understanding the quirks of local politics, regulations, and culture, as well as a global knowledge of logistics, supply chains, and industry-specific processes and players. The Figure 5.3 shows the supply chain orchestrator’s sphere of the influence.
6.5.1 Role of supply chain orchestrator

The Orchestrators act as the key actors in the business network, they organize and govern for combining of a variety of potentials for explicit consequences expected at the consumer. Its responsibilities are being defined in wide range. It is responsible for organizing the all activities initializing from the upstream activities finishing with completion of the downstream activities. It acts as the interface between the upstream & downstream activities in the supply chain. In traditional supply chain system all activities took place at the single office. But in the age of the globalization, the supply chain activities are being handled from various offices located at different locations. So the task
of organizing the supply chain activities becomes a very complex task. This nature of the SCM activities requires the distributed information flow for various SCM entities. Hence the supply chain orchestrator builds the network between the supply chain entities with the following targets as given below:

1. Efficient process control
2. Mastering the linking of capabilities in a network of different actors.
3. Manage, end-to-end, discrete processes distributed in a network of business partners.

The Orchestrator are being aimed to manage the IT platform, choose the suppliers, make the decisions on locations, finalize the transportation providers, logistics, and other service providers, and mandates the performance requirements for all of the SCM entities. In other words, the orchestrator is responsible for coordinating, guiding, and aligning the supply chain on many levels. The real challenge in the SCM is still to come: process management that follows a business-oriented architecture i.e. that gives the full flexibility to design and execute business processes irrespective of the operational environment. The roles of the supply chain orchestrator are given as below:

- **Strategic alignment**
  The SCM entities coordinate their market strategies, product/solution strategies, technology roadmaps, infrastructure investments etc. Hence it provides the alignment between the strategies of different entities.

- **Financial alignment**
  Each entity has motivation to optimize the accomplishment of the customer’s requirements and needs. For this purpose it provides the alignment for the finance required for the operations occurred in SCM.

- **Executional alignment**
  In SCM multiple activities are being taken place in SCM. It provides the synchronization of operations across the chain.

- **Metrics alignment**
  It is aimed to provide the alignment between the success metrics. The success is measured in a consistent manner, using enterprise scorecards to align with the needs of the customer.
These roles are basic facts of showing the responsibilities of the SCO. It chooses the IT technologies which provide the following features as given below:

- **Sustainable**
  The technique should be capable of providing high quality products from the supplier side over time.

- **Highly functional**
  It measures the degree of fulfilling the current and potential future requirements of the products without deploying add-ons or customization.

- **Upgradeable**
  The technique should be capable to enable the supplier is to fix bugs.

- **Future-Proof**
  These technologies should provide the alignment of the system with new technologies over time.

On the basis of these factors the supply chain orchestrators select the technology for designing the supply chain. On wrong selection of the technology it degrades the overall performance of the supply chain system.

### 6.6 ORCHESTRATING ACTIVITIES IN MAS BASED SCM

Traditional technologies are not sufficient in implementing the automation of the business processes in the supply chain. The multi-agent system is being found more efficient in managing and automating the business processes during the operation of the supply chain process. Hence MAS based SCM system is designed for reducing the bullwhip effect faced during whole system operation. This system is capable of handling the causes of bullwhip effect in supply chain management system. This factor prompts the performance of the whole system.

#### 6.6.1 Challenges faced in MAS based SCM

There are a lot of challenges which are being faced in working of MAS based SCM. A main major issue of this system is to manage the process of coordinating asynchronous communication between SCM activities. All SCM activities manipulates its input parameters and pass this final and intermediate values to other related activities occurring
in this system. These final and intermediate values are required for decision making, initializing other activities and controlling these activities in whole system working.

Next challenge faced in this system is to correlate messages exchanges between various SCM entities like supplier, manufacturer, distributor and retailer etc. In whole working of the MAS based SCM system, a bulk of the messages is being transmitted by all entities involved in this system. The particular information is being passed between these SCM entities. Now it is major issues of processing this information shared or passed by other entities is to manipulate the information either the information is associated with particular event taking place at that particular time period.

Another challenge of implementing simultaneously processing of SCM activities is being faced in this system. There occurs a numerous of the processes in this system at the same time. The main problems encounter during phenomena of management and implementation of these simultaneous processes having affects on the results of other running processes. It is a very complex task to implement the parallel processes with the aim of avoiding the conflicts in the results produced by these processes. To manipulate data between SCM entities interactions is one of major challenge faced in the MAS based SCM system. All SCM entities are directed to achieve their goals. In during the process of achieving their goals, these entities interact with each others. These SCM entities share the data for collaboration of these entities. To achieve the coordination and collaboration between these entities, the shared data should be analyzed and manipulate with great concern with supporting extensive consecutively industry transactions and activities. These activities should be able to hold the data during long time to wait till the completion of the whole system operation. The individual processes manipulate some data during in the time of its execution. But it should be remained till the completion of execution of whole system operation.

The last major issue faced in this system is aim of providing consistent exception handling. The exception may be defined as the situation which prevent the designed system to achieve the predetermined system goal and may be interrupt the system working in its execution time. Hence it is one of the major concern to avoid the exception occurred in working of the whole system. It requires efficient exception handling mechanism. These exceptions may cause numerous worst effects which make a great
impact on the whole system performance. These issues are major challenges faced in working of MAS based SCM system.

### 6.6.2 Resolving challenges faced

To resolve all these challenges faced in MAS based SCM, all SCM activities are being orchestrated. The programming language is not capable of performing the integration and automation of the business processes. There is the most suitable solution is the SOA which provides efficient implementation of the business process automation. For resolving the challenges faced in MAS based SCM system, the Business Process Execution Language is being applied to smoothly running on this system. The BPEL is top down approach to the service-oriented architecture. The BPEL is used to achieve the orchestration in the supply chain activities. The BPEL process is made of various steps called as an activity. It contains two types of the activities- primitive and structure activities. The primitive activities perform the common tasks such as initializing other services and manipulate data variables and structure activities perform the complex operations and may be combinations of the primitive activities.

There are following activities exists that are used to orchestrate the supply chain activities in MAS based SCM system as given below:

- To appeal to other business activities services, using `<invoke>`
- To compose the client to wait for invoking the business process through sending a message, using `<receive>`
- To produce a reply for synchronous operations, using `<reply>`
- To operate the data variables, using `<assign>`
- To specify the faults and exceptions, using `<throw>`
- To remain ideal for a quantity of time, using `<wait>`
- To lapse the whole process, using `<terminate>`.
- To define a set of activities for invoking in an ordered sequence using `<sequence>`.
- To define a set of activities that will be invoked in parallel, using `<flow>`.
- To implement the branches, using `<switch>`.
- To define the loops, using `<while>`.
- To choose one of a number of substitute paths, using `<pick>`.
The BPEL supports the different type of the process running in the synchronous or asynchronous mode. To resolve the challenges of implementing simultaneously processing of SCM activities, the BPEL process blocks the SCM activities which are using the process until the process finishes and returns a result of the activities in the synchronous mode. For maintaining the data manipulation result for the longer lasting period, the BPEL supports the concept of the asynchronous processes and synchronous for processes that return a result in a relatively short time. The BPEL has the feature of message correlation feature for associating the message exchange in different business process. The BPEL enables the process to maintain a correlation set which helps to find the message whether the coming message is precise to receive by the concerned SCM entities among multiple messages sent by different SCM entities. For handling the exceptions, the BPEL consist the fault handling component which is capable of managing the errors takes place during the execution of business processes. The BPEL contains some standard faults occurring in business process execution like conflictingRequest, invalidReply, selectionFailure and bindingFault etc. Using this standard faults, the BPEL manages the all types of the errors taken place in the execution of the business process handling the working of the different agents involved in MAS based SCM system.

In this way, all challenges faced in the MAS based SCM system are being resolved successfully. The BPEL enhances the working and performance of the supply chain activities. After resolving the challenges faced in this system, the bullwhip effect can be reduced by orchestrating the supply chain activities in the MAS based SCM system. After orchestrating the supply chain activities, the particular intelligent agent’s activities are being controlled and exceptions are being resolved in order to achieve the system goals in the MAS based SCM system.

6.7 ORCHESTRATION FOR REDUCING BULLWHIP EFFECT

In MAS based SCM system, there exist various multiple intelligent agents designed for performing the supply chain activities efficiently with the aim of managing all causes of bullwhip effect. These intelligent agents manage various operations such as supplying, manufacturing, inventory and distributing etc. The main functions of these intelligent agents include:

- To supervise all other agents during their operations.
• To take delivery of the orders from the designated customers.
• To check the inventory for fulfillment the customer requirements.
• To issue the order of raw materials from suppliers.
• To manage and schedule the production
• To perform the financial activities.
• To store the information of stock, components and material.

In the MAS based SCM system, the orchestrator agent is designed with aim of managing and coordinating the supply chain activities. The orchestrator agent works on the concept of the orchestration. This concept is very useful in the business process automation. The orchestrator agent in MAS based SCM can play very important role in controlling the bullwhip effect. The goals, perceptions and action of this agent are directly oriented on objective of the coordination between the supply chain activities. There are some parameters involved in whole supply chain activities during whole system operations. These parameters are being used in the intelligent agents’ operation. These parameters help the particular intelligent agent to take the decisions during their operations. These entities are being listed as given below:
• Product volume
• Product cost
• Order quantity
• Customer demand
• Price updating
• Pending order
• Stock status
• Customer responses
• Delivery time
• Consumer RFQs
• Supplier RFQs
• Inventory levels
• Stock threshold
• Market conditions
These parameters may play very important role in tackling the bullwhip effect in the supply chain system. All causes of the bullwhip effect can be minimized with proper handling of these parameters. The supply chain activities are directly oriented on the aims of achieving the organizational goals and customer satisfactions. The bullwhip effect makes the worst consequences on the performance of whole supply chain management system. With the aim of controlling the bullwhip effect in the supply chain management system, the events such as demand forecasting, price frustration, order batching, and rationing and shortages games should be orchestrated for resolving the bullwhip effect occurred in the supply chain system.