Composite Application: BookProcurement

- NeedApproval-to-Rejected
- Rejected-to-Modifying
- SubmitOrder
- ApproveOrder
- AddBook
- CreateOrder
- ShipOrder
- ShipmentSent
- CompleteOrder
- SendToApprover
- Requested-to-BackOrdered
- ShipBundle
- OrderShipped
- GetAllUnshippedShipments

- BookOrder
- Shipment

- Email
Business Artifact: BookOrder

Information Model

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>long</td>
</tr>
<tr>
<td>CurrentState</td>
<td>string</td>
</tr>
<tr>
<td>orderName</td>
<td>string</td>
</tr>
<tr>
<td>orderDescription</td>
<td>string</td>
</tr>
<tr>
<td>dateCreated</td>
<td>date</td>
</tr>
<tr>
<td>buyer</td>
<td>string</td>
</tr>
</tbody>
</table>

Business Artifact Lifecycle:
Edit lifecycle by copy/pasting states/transitions

Add Books Then Submit

Auto Approve When TotalCost < 50.00

Auto Order When Approved

Auto Complete When All Shipments Shipped

Receive signal from Shipments

Create Shipments

Ordering

Submitted

NeedsApproval

Approved

Rejected

Completed

Modify

Receive signal from Shipments

Receive signal from Shipments
Business Artifact: Shipment

Information Model

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>long</td>
</tr>
<tr>
<td>CurrentState</td>
<td>string</td>
</tr>
<tr>
<td>ShipmentName</td>
<td>string</td>
</tr>
<tr>
<td>dateCreated</td>
<td>date</td>
</tr>
<tr>
<td>buyer</td>
<td>string</td>
</tr>
<tr>
<td>bookOrderId</td>
<td>long</td>
</tr>
</tbody>
</table>

Business Artifact Lifecycle:

Edit lifecycle by copy/pasting states/transition

Signal Book Order when Shipment is Shipped

Analyze Shipment and determine either “Shipped” or “BackOrdered” exit condition

Requested → BackOrdered → Shipped
Service: BookOrder: CreateOrder

Universal Resource Identifier: /BookOrder/CreateOrder

Service Plan:

AssignCreationInfo

Map Input Parameters to Book Order

Service Input Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>long</td>
</tr>
<tr>
<td>OrderName</td>
<td>string</td>
</tr>
<tr>
<td>OrderDescription</td>
<td>string</td>
</tr>
<tr>
<td>BuyerEmail</td>
<td>string</td>
</tr>
</tbody>
</table>

Service Output Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Context:

- Modifying
- Submitted
- NeedsApproval
- Approved
- Rejected
- Ordered
- Completed
Service: BookOrder: AddBook

Universal Resource Identifier: /BookOrder/AddBook

Service Plan:

AdoBookToOrder → AddBookCostToOrder

Map Book Input Parameter to the Book Order Artifact

Increase the Total Cost

Context:

Submitted → NeedsApproval → Approved → Rejected

Completed → Ordered

Service Input Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>long</td>
</tr>
<tr>
<td>InBook</td>
<td>Book</td>
</tr>
</tbody>
</table>

Service Output Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>long</td>
</tr>
</tbody>
</table>
Service: BookOrder: SubmitOrder

Universal Resource Identifier: /BookOrder/SubmitOrder

Service Plan: NoOP

Context:

Service Input Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>

Service Output Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Modifying</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submitted</td>
</tr>
<tr>
<td>NeedsApproval</td>
</tr>
<tr>
<td>Approved</td>
</tr>
<tr>
<td>Rejected</td>
</tr>
<tr>
<td>Ordered</td>
</tr>
<tr>
<td>Completed</td>
</tr>
</tbody>
</table>
Service: BookOrder: SendToApprover

Universal Resource Identifier: /BookOrder/SendToApprover

Service Plan:

Notify the Approver that he needs to look at a Book Order and either Approve or Reject it

Service Input Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>

Service Output Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>

Context:

Modifying

Submitted → NeedsApproval

Approved → Rejected

Ordered → Completed
Service: BookOrder: ApproveOrder

Universal Resource Identifier: /BookOrder/ApproveOrder

Service Plan:

- NotifyBuyer

Take the input parameter call InBundle and assign it to the newly created Shipment Artifact

Service Input Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>long</td>
</tr>
</tbody>
</table>

Service Output Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>long</td>
</tr>
</tbody>
</table>

Context:

- Submitted
- NeedsApproval
- Approved
- Rejected
- Ordered
- Completed
Service: BookOrder: ShipOrder

Universal Resource Identifier: /BookOrder/ShipOrder

Service Plan:

- ShipBundle1
- ShipBundle2

Context:

- Modifying
- Submitted
- NeedsApproval
- Approved
- Rejected
- Ordered
- Completed

Service Input Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>

Service Output Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>
Service: BookOrder: ShipmentSent

Universal Resource Identifier: /BookOrder/ShipmentSent

Service Plan:

- GetAllUnshipped
- SetAllShipped
  - Set AllShipped Attribute based on result set
  - Test Exit Condition

Context:

- Modifying
  - Submitted
    - NeedsApproval
      - Approved
      - Rejected
  - Ordered
  - Completed

Service Input Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>

Service Output Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>
**Service: BookOrder: CompleteOrder**

**Universal Resource Identifier:** /BookOrder/CompleteOrder

**Service Plan:**

- NotifyBuyerOfCompletedOrder

**Context:**

- Data Type
- Attribute
- Service Input Message
- Data Type
- Attribute
- Service Output Message

- NeedsApproval
- Submitted
- Approved
- Rejected
- Ordered
- Completed

**Notify Buyer that his Book Order is complete**
Service: Shipment: ShipBundle

Universal Resource Identifier: /Shipment/ShipBundle

Service Plan:

AssignBundleToShipment

Setup Shipment Data

AddBooks

Add Books to Bundle

Requested

BackOrdered

Shipped

Service Input Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>long</td>
</tr>
<tr>
<td>ShipmentName</td>
<td>string</td>
</tr>
<tr>
<td>BookOrderID</td>
<td>long</td>
</tr>
<tr>
<td>Buyer</td>
<td>string</td>
</tr>
</tbody>
</table>

Service Output Message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Data Type</th>
</tr>
</thead>
</table>
Service: Shipment: OrderShipped

Universal Resource Identifier: /Shipment/OrderShipped

Service Plan:

- SendShipmentNotification
- SignalBookOrder

Notify Buyer a Shipment has been sent

Communicate to the Book Order that a shipment has been sent

Requested
BackOrdered
Shipped

Context:

<table>
<thead>
<tr>
<th>Service Input Message</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Service Output Message</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td></td>
</tr>
</tbody>
</table>

© 2006 IBM Corporation
Data Services: BookOrder: DetermineBundling

Universal Resource Identifier: /BookOrder/DetermineBundling

SQL Query:
Data Services: Shipment: GetAllUnshippedShipments

**Universal Resource Identifier:** /Shipment/GetAllUnshippedShipments

**SQL Query:**

```
SELECT Shipment.ID, Shipment.shipmentName, Shipment.bookOrderId FROM Shipment where Shipment.CurrentState!='Shipped' and Shipment.bookOrderId=?
```
External: REST: Email

**Universal Resource Identifier:**  http://localhost:8080/SienaUtilWeb/Email

**Type:**  GET

*Type is one of GET / POST / PUT / DELETE. Please consult the REST service documentation*

An externally available REST Service may be registered for use in the composite application using this template. It can then be used via invoke-declaration in any service-plan.

The REST service has:
- **URI:** Universal Resource Identifier (typical an HTTP URL)
- **Type:** GET/POST/PUT/DELETE
- **Input Message:** Input
- **Output Message:** Output

Right Click anywhere on the slide to view/update service messages