SOA Frameworks

Dave DiFranco
david.difranco@oracle.com
ddif@alum.mit.edu

What is SOA?

- “Service Oriented Architecture”
- It's a philosophy not a standard
- Composition of reusable, heterogeneous services
  - Multiple languages
  - Multiple protocols
  - Loosely coupled
  - Reuse legacy systems
- It's not WS-*,
  - often built on top of WS-* and XML
- Allow business users to assemble + customize applications

My SOA creds

- Developer at BEA/Oracle *2007-present*
  - Oracle Enterprise Repository: governance repository for SOA services + components
- OASIS tech committee for SCA *2007-2008*
- Developer at MetaMatrix *2005-2006*
  - (now JBoss Teiid)

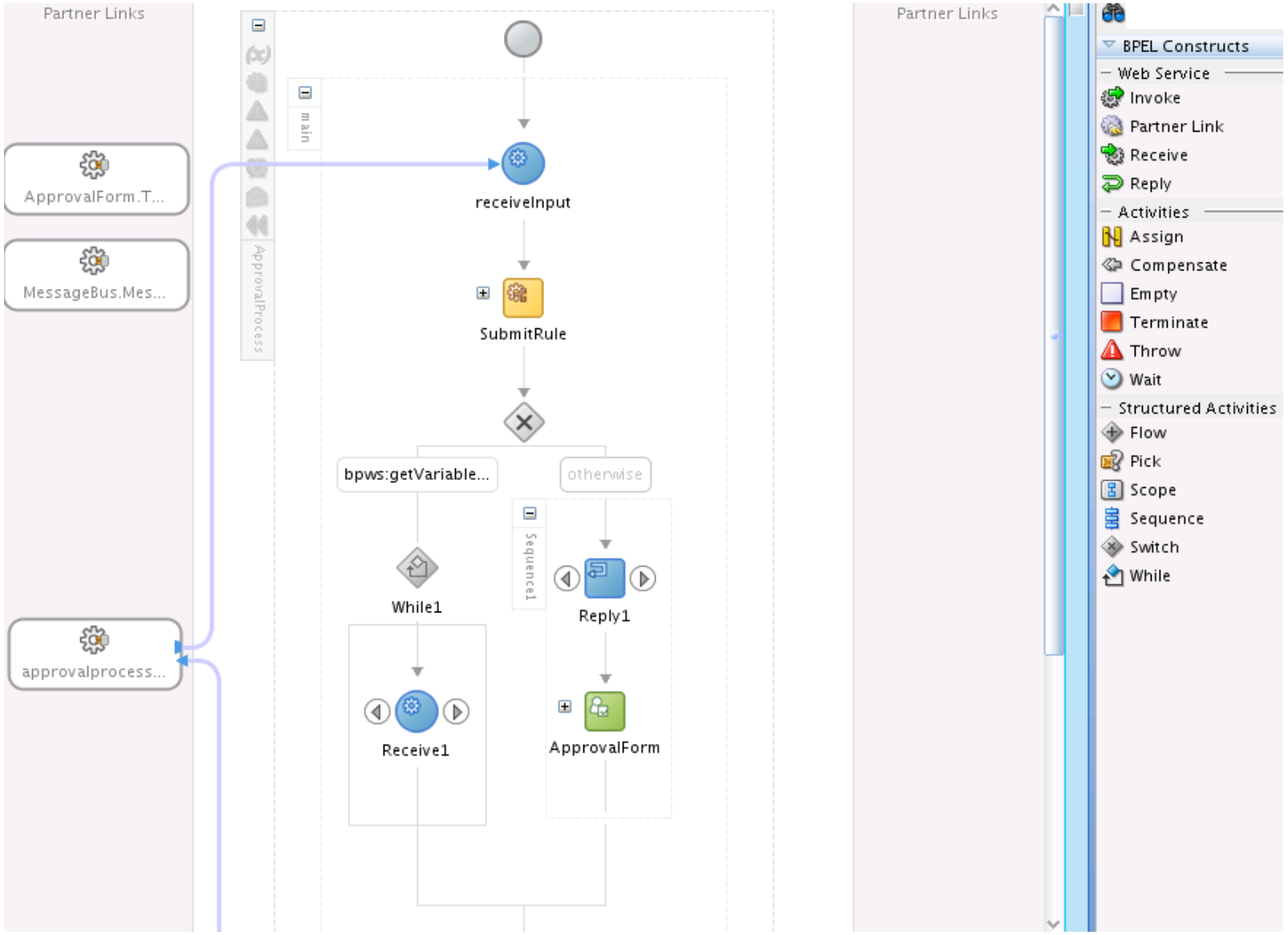
SOA Alphabet Soup

- **BPM** Business Process Modeling
- **BPEL** Business Process Execution Language
- **ESB** Enterprise Service Bus
- **CEP** Complex Event processing
- **BAM** Business Activity Monitoring
- **SCA** Service Component Architecture
- ...

BPM: Business Process Modeling

- two major standards
  - BPEL: Business Process Execution Language
    - OASIS Standard since 2003
  - BPMN: Business Process Modeling Notation
    - OMG Standard since 2004
- allow business analysts to assemble automated business processes
- Business processes are long-running: hours or days or more
- constructs for control flow, service invocation
  - proprietary tools have constructs for integration w/ other technologies

BPEL example



BPEL: standard activities

- Web Service
  - invoke
  - receive
  - reply
  - pick <!--selective event processing-->
- Variables
  - assign <!--assign variables: xpath, xsl, … variable datatypes are defined via XML schema-->
- Control flow
  - if
  - while
  - repeatUntil
  - forEach
  - wait
  - throw
  - rethrow
  - exit
  - sequence
  - flow <!--parallel processing -->
  - empty <!--no-op -->

BPEL example: some source

```
 <sequence xml:id="id7" name="main">
      <!-- Receive input from requestor. (Note: This maps to operation defined in ContractApprovalProcess.wsdl) -->
      <receive xml:id="id8" name="receiveInput" partnerLink="contractapprovalprocess_client"
portType="client:ContractApprovalProcess" operation="process" variable="inputVariable" createInstance="yes"/>


      <!--
        Asynchronous callback to the requester. (Note: the callback location and correlation id is transparently handled using W
addressing.)
      -->
      <switch xml:id="id35" name="ChooseBidder">
         <case xml:id="id36"
              condition="bpws:getVariableData('inputVariable','payload','/ns1:ContractBids/ns1:Bidder1/ns1:QuotePrice') &lt;
bpws:getVariableData('inputVariable','payload','/ns1:ContractBids/ns1:Bidder2/ns1:QuotePrice')">
            <assign xml:id="id39" name="AssignBidder1">
               <copy xml:id="id41">
                  <from xml:id="id42" variable="inputVariable"
                        part="payload"
                        query="/ns1:ContractBids/ns1:Bidder1"/>
                  <to xml:id="id43" variable="ChosenBidder"
                     query="/ns1:Bidder"/>
               </copy>
            </assign>
         </case>
         <otherwise xml:id="id37">
            <assign xml:id="id40" name="AssignBidder2">
               <copy xml:id="id44">
                  <from xml:id="id45" variable="inputVariable"
                        part="payload"
                        query="/ns1:ContractBids/ns1:Bidder2"/>
                  <to xml:id="id46" variable="ChosenBidder"
                     query="/ns1:Bidder"/>
                                 .....tApprovalTask_1"
```

BPEL: compensation

```
<invoke partnerLink="Seller"
   portType="SP:Purchasing"
   operation="Purchase"
   inputVariable="sendPO"
   outputVariable="getResponse">

   <compensationHandler>
      <invoke partnerLink="Seller"
         portType="SP:Purchasing"
         operation="CancelPurchase"
         inputVariable="getResponse"
         outputVariable="getConfirmation">
      </invoke>
   </compensationHandler>

</invoke>
```

BPEL example: some source

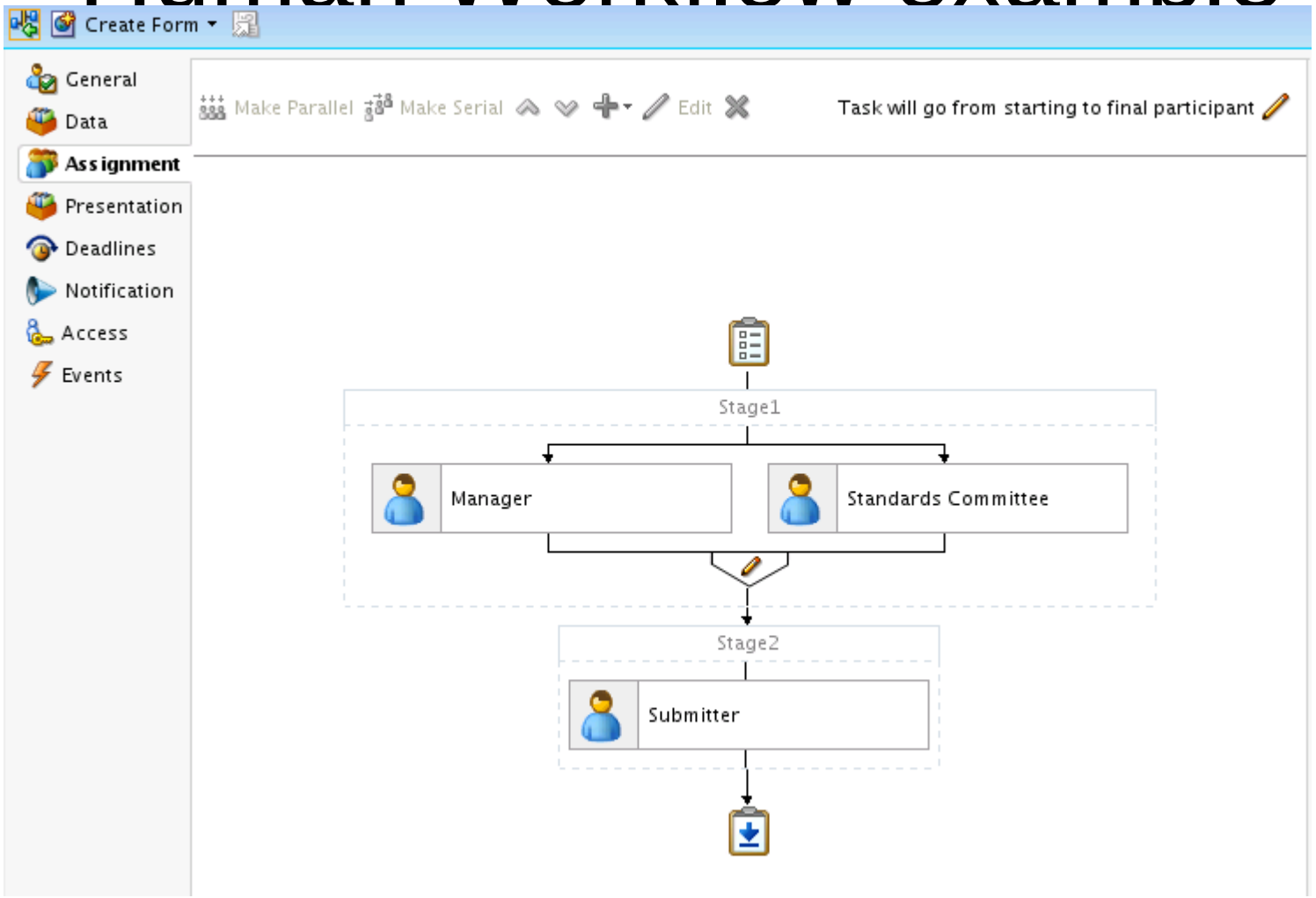
```
 <sequence xml:id="id7" name="main">
      <!-- Receive input from requestor. (Note: This maps to operation defined in ContractApprovalProcess.wsdl) -->
      <receive xml:id="id8" name="receiveInput" partnerLink="contractapprovalprocess_client"
portType="client:ContractApprovalProcess" operation="process" variable="inputVariable" createInstance="yes"/>


      <!--
        Asynchronous callback to the requester. (Note: the callback location and correlation id is transparently handled using W
addressing.)
      -->
      <switch xml:id="id35" name="ChooseBidder">
         <case xml:id="id36"
              condition="bpws:getVariableData('inputVariable','payload','/ns1:ContractBids/ns1:Bidder1/ns1:QuotePrice') &lt;
bpws:getVariableData('inputVariable','payload','/ns1:ContractBids/ns1:Bidder2/ns1:QuotePrice')">
            <assign xml:id="id39" name="AssignBidder1">
               <copy xml:id="id41">
                  <from xml:id="id42" variable="inputVariable"
                        part="payload"
                        query="/ns1:ContractBids/ns1:Bidder1"/>
                  <to xml:id="id43" variable="ChosenBidder"
                     query="/ns1:Bidder"/>
               </copy>
            </assign>
         </case>
         <otherwise xml:id="id37">
            <assign xml:id="id40" name="AssignBidder2">
               <copy xml:id="id44">
                  <from xml:id="id45" variable="inputVariable"
                        part="payload"
                        query="/ns1:ContractBids/ns1:Bidder2"/>
                  <to xml:id="id46" variable="ChosenBidder"
                     query="/ns1:Bidder"/>
                                 .....tApprovalTask_1"
```

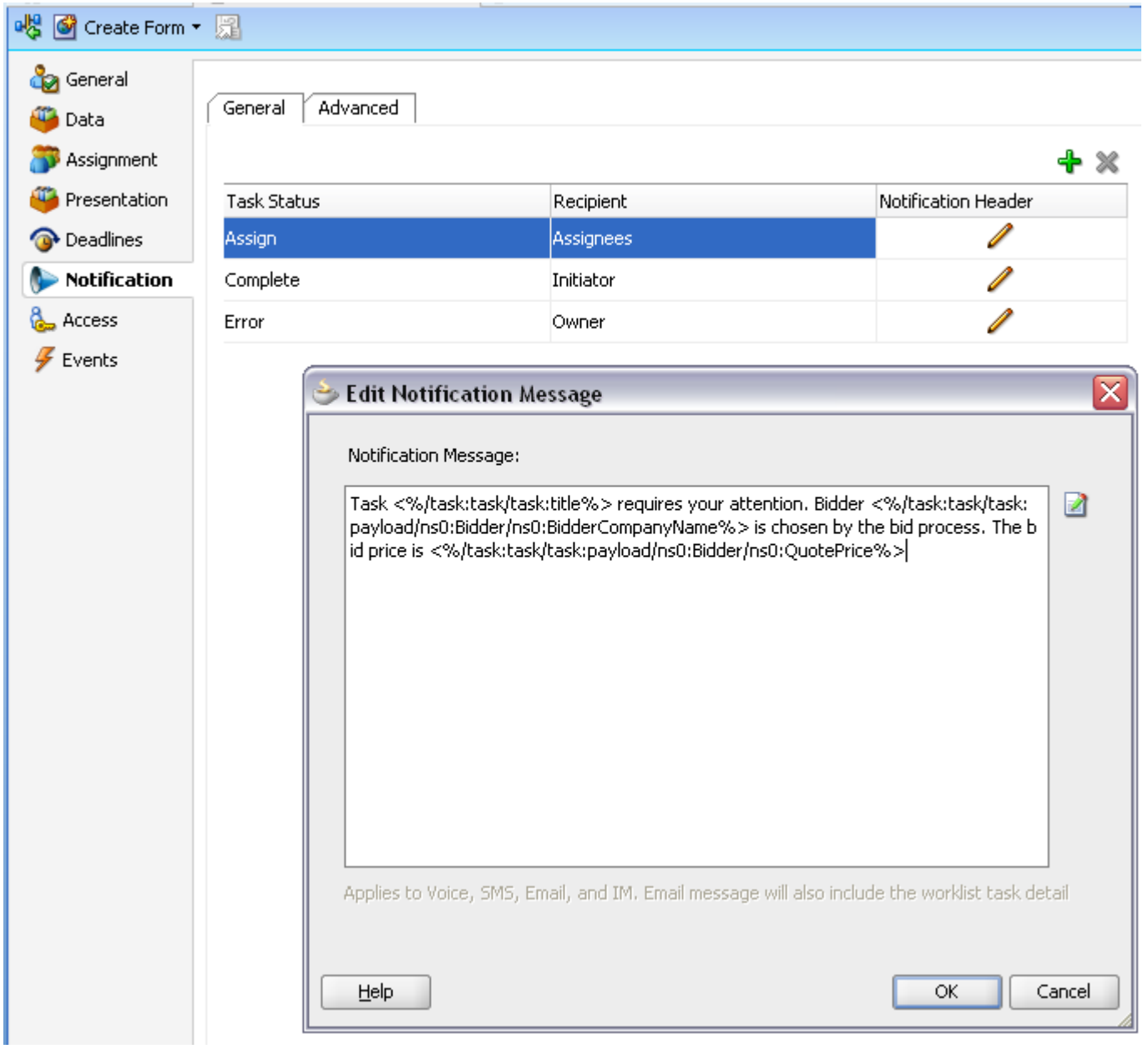
What BPEL forgot: Humans

- BPEL4People, WS-HumanTask specifications
  - OASIS Specification 2007
  - Users, Groups, Roles
  - Human Tasks
  - Notifications
  - Deadlines
  - Escalation
  - What's missing:
    - Actual notification mechanisms (email, SMS, voice, etc) – handled by proprietary implementations

Human Workflow example



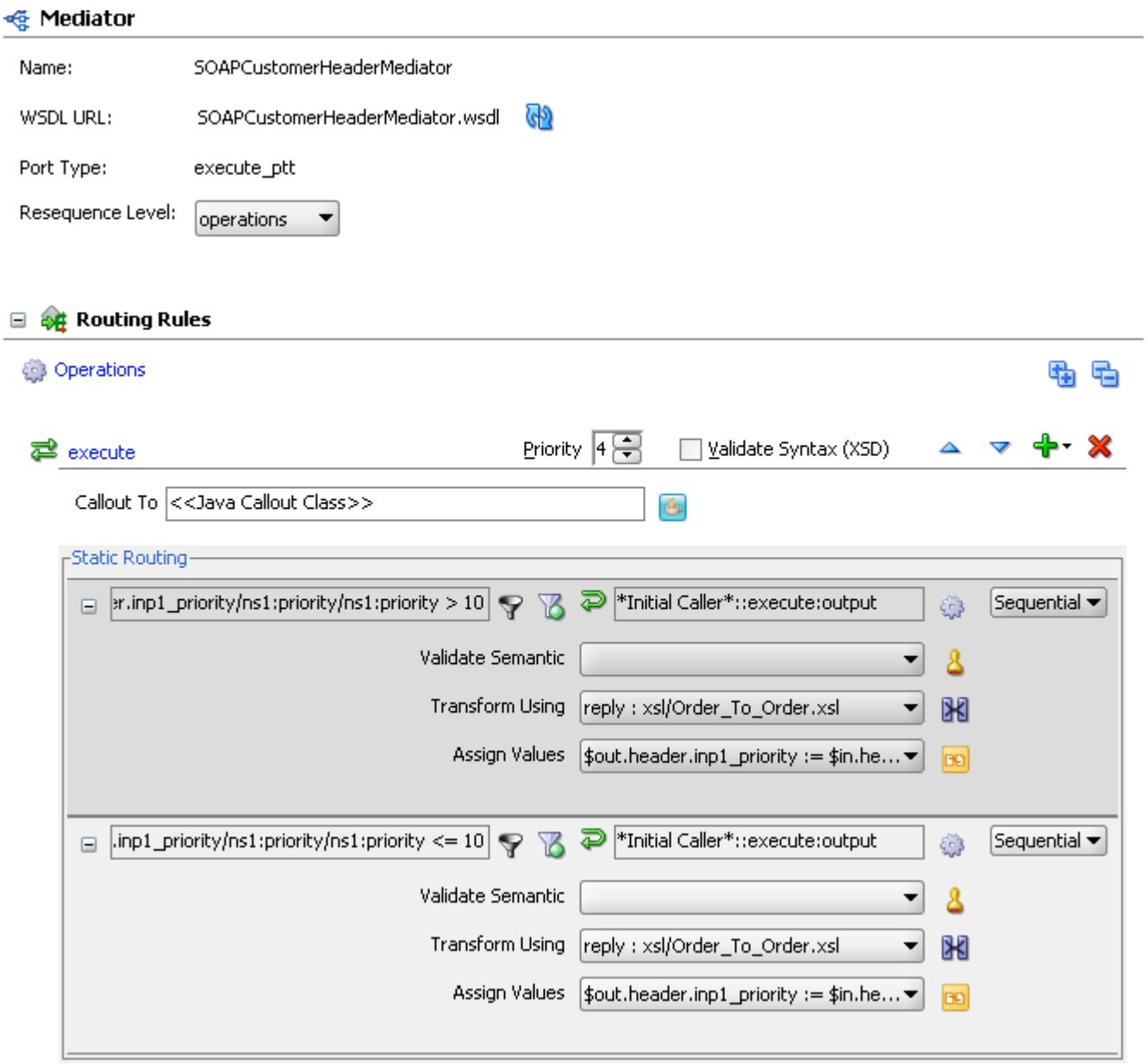
Human Workflow example: notification



ESB: Enterprise Service Bus

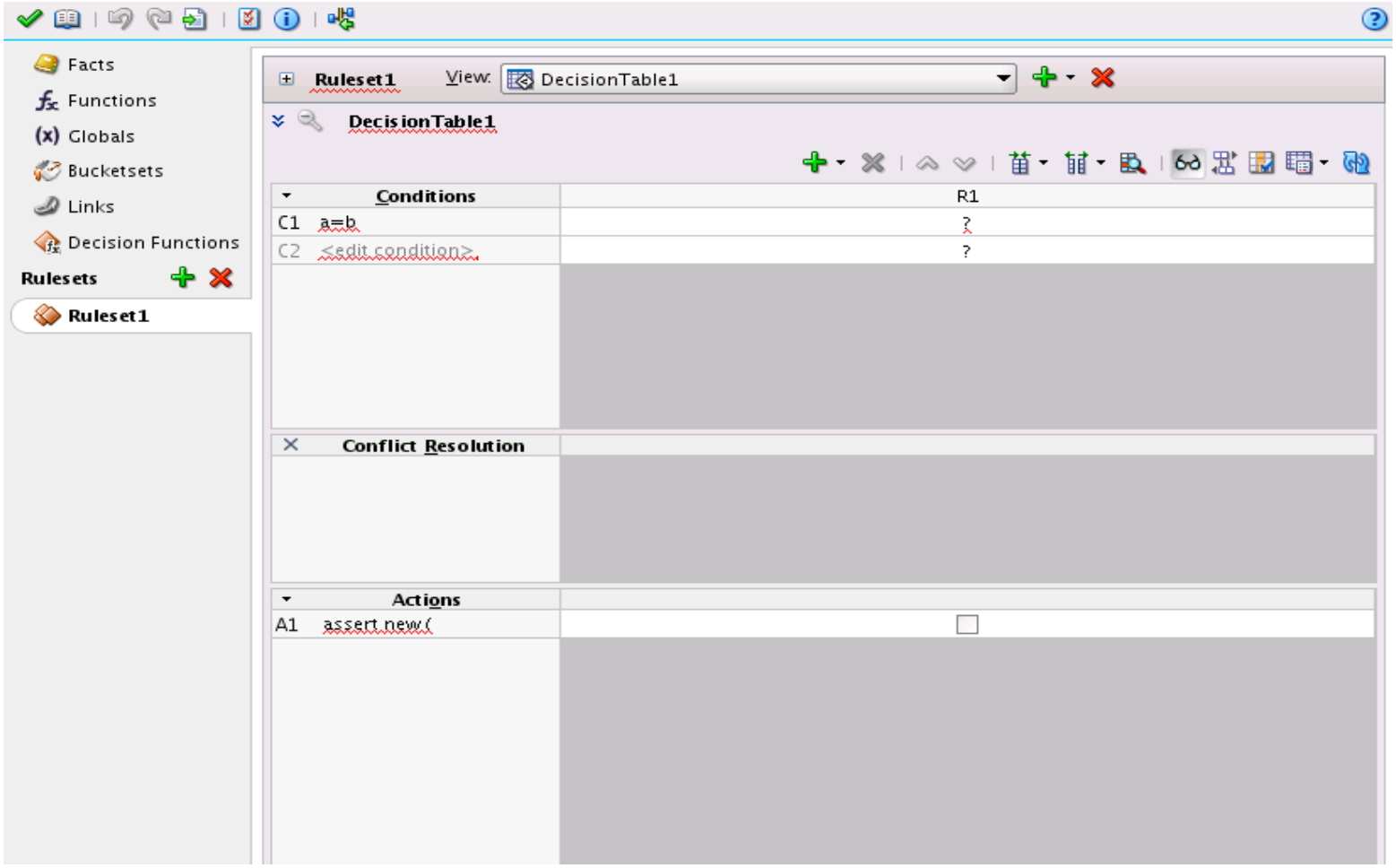
- Family of products, not a standard
- Route messages (service calls) across multiple technologies
- Routing
- Failover
- Transformation (e.g. XSLT / XQuery)
- Validation
- Lighter weight than BPM - “plumbing” not process

ESB example: route, transform



Business Rules

- Business users can customize rules + thresholds that drive application



More stuff

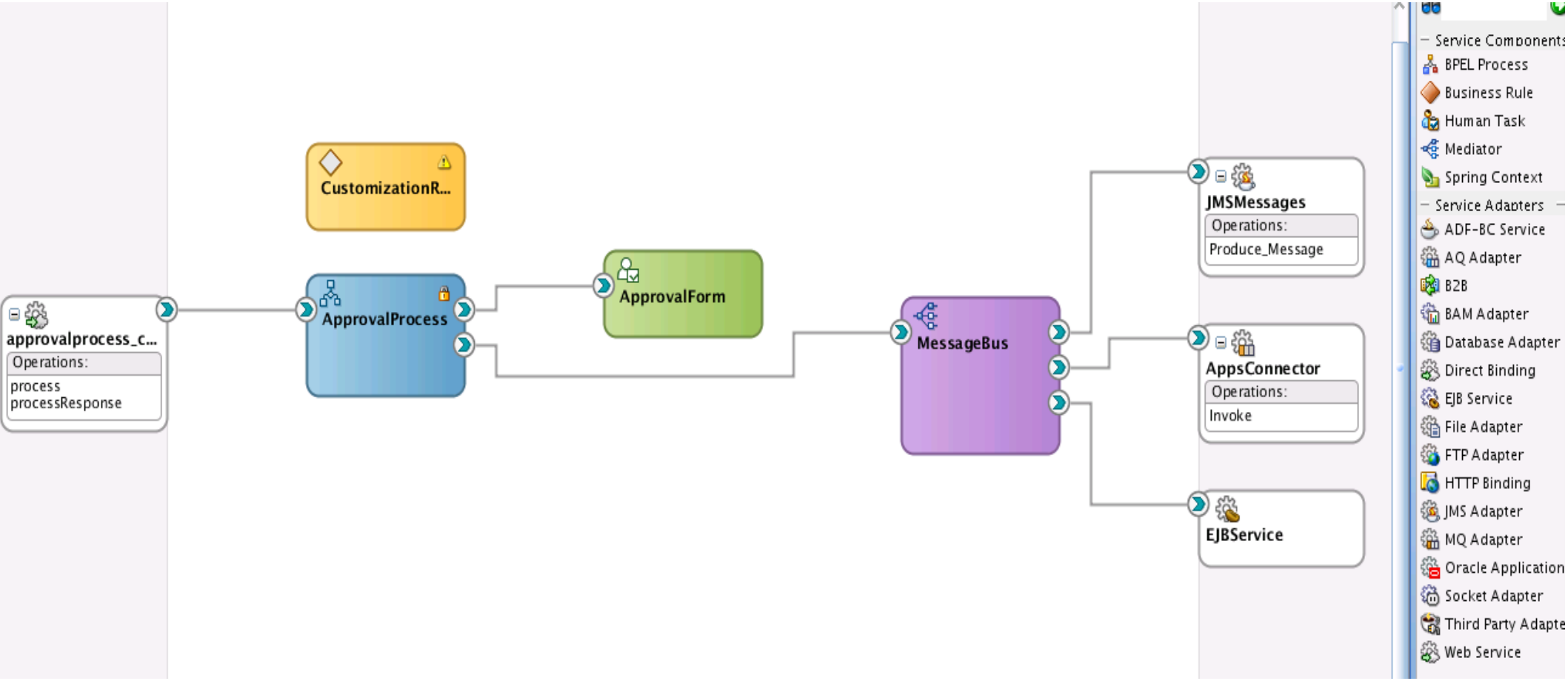
- **CEP** Complex Event processing
  - Orchestrate + act on complex sets of real-time events
- **BAM** Business Activity Monitoring
  - Monitor business activities, notify users of errors, thresholds
- Data Services / **EII** (Enterprise Information Integration)
  - Aggregate + transform data from multiple sources
  - Relational (database), XML, text etc

How do I assemble all this stuff?

SCA: Service Component Architecture

- OASIS Standard 2007
- Assemble + deploy heterogenous applications
- ‘deployment descriptor’ for SOA
- 'dependency injection' for SOA
- connect services to references
- set property / variable values
- set policies (WS-Policies) for security, reliability
- standard bindings for WS, Java, JMS, JCA, EJB, WS-Policy, Spring, BPEL, C, COBOL
  - proprietary tools have other bindings

SCA example: a composite application



SCA example: some source

```
<composite name="ContractApprovalComposite" ...>
  <service name="contractapprovalprocess_client_ep"
           ui:wsdlLocation="ContractApprovalProcess.wsdl">
    <interface.wsdl
interface="http://xmlns.oracle.com/ContractApprovalApp/ContractApprovalComposite/ContractApprovalProcess#wsdl.interface(Contract
pprovalProcess)"

callbackInterface="http://xmlns.oracle.com/ContractApprovalApp/ContractApprovalComposite/ContractApprovalProcess#wsdl.interface(C
ontractApprovalProcessCallback)"/>
    <binding.ws
port="http://xmlns.oracle.com/ContractApprovalApp/ContractApprovalComposite/ContractApprovalProcess#wsdl.endpoint(contractappro
valprocess_client_ep/ContractApprovalProcess_pt)"/>
    <callback>
      <binding.ws
port="http://xmlns.oracle.com/ContractApprovalApp/ContractApprovalComposite/ContractApprovalProcess#wsdl.endpoint(contractappro
valprocess_client_ep/ContractApprovalProcessCallback_pt)"/>
    </callback>
  </service>
  <component name="ContractApprovalProcess">
    <implementation.bpel src="ContractApprovalProcess.bpel"/>
  </component>
  <component name="ContractApprovalTask">
    <implementation.workflow src="ContractApprovalTask.task"/>
  </component>
  <wire>
    <source.uri>contractapprovalprocess_client_ep</source.uri>
    <target.uri>ContractApprovalProcess/contractapprovalprocess_client</target.uri>
  </wire>
  <wire>
    <source.uri>ContractApprovalProcess/ContractApprovalTask.TaskService_1</source.uri>
    <target.uri>ContractApprovalTask/TaskService</target.uri>
  </wire>
</composite>
```

SCA: java dependency injection

```
private HelloService helloService;

 @org.oasisopen.sca.annotation.Property(name="currency", required=true)
 private String currency;

 @org.oasisopen.sca.annotation.Reference(name="helloService", required=true)
 public setHelloService(HelloService service) {
        helloService = service;
 }

}
```

SCA: legacy system integration

- JCA Connector
  - Java Connector Architecture: since 2001
  - Specify integration with “legacy” information systems (EIS’s)
    - Siebel, JDEdwards, PeopleSoft, SAP, Tuxedo, Mainframe Apps ...
  - Connections, transactions, security

JCA Example

```
<adapter-config name="ImportOrderCP" adapter="Oracle Applications Adapter"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">

  <connection-factory location="eis/Apps/atg121d6" UIConnectionName="atg121d6"
UIOracleAppType="DBOBJECT" UIConcurrentPgmName="OEOIMP" adapterRef=""/>
  <endpoint-interaction portType="ImportOrderCP_ptt" operation="ImportOrderCP">
    <interaction-spec className="oracle.tip.adapter.apps.AppsStoredProcedureInteractionSpec">
      <property name="IRepInternalName" value="CONCURRENTPROGRAM:OEOIMP:Process"/>
      <property name="SchemaName" value="APPS"/>
      <property name="PackageName" value="XX_BPEL_FND_REQUEST_WRAPPER"/>
      <property name="ProcedureName" value="SUBMIT_REQUEST"/>
      <property name="Username" value="sysadmin"/>
      <property name="Responsibility" value="System Administrator"/>
    </interaction-spec>
    <output/>
  </endpoint-interaction>


</adapter-config>
```

I need some tools!

- SOA suites often have
  - developer workbenches / IDE plugins
  - web consoles for business analysts
    - Create / configure business processes etc
  - admin, deployment, monitoring, governance tools

When do I need SOA?

- Integrate legacy / heterogeneous projects
  - without code

- Need customization + orchestration
  - without (manually entered) code
  - by business users, not developers

SOA drawbacks (?)

- Which components + standards do I choose?
  - Many overlap
  - (software suites mitigate this)
- How do I connect all the components + standards?
  - (SCA mitigates this)
- Performance implications of SOAP
  - (But you’re not always limited to SOAP)
- XML-heavy
  - Implementation files are hard to read, refactor
  - (but you're not always limited to XML)
- Lagging support for bleeding-edge languages (?)

Who makes it?

- Oracle
- IBM
- HP
- SAP
- Microsoft
- Tibco
- SOA Software
- MuleSoft
- RedHat / JBoss
- Eclipse
- Apache
- Etc etc...