WebSphere ESB Best Practices

WebSphere User Group, Edinburgh
17th September 2008

Andrew Ferrier, IBM Software Services for WebSphere
andrew.ferrier@uk.ibm.com

Contributions from:
Russell Butek (butek@us.ibm.com)
André Tost (andretost@us.ibm.com)

© 2008 IBM Corporation

Agenda

- Brief Refresher of WebSphere ESB
- ‘Large-scale’ Best Practices: **Design and architecture**
- ‘Small-scale’ Best Practices: **Hints and tips to make your life easier**
- References and Further Information

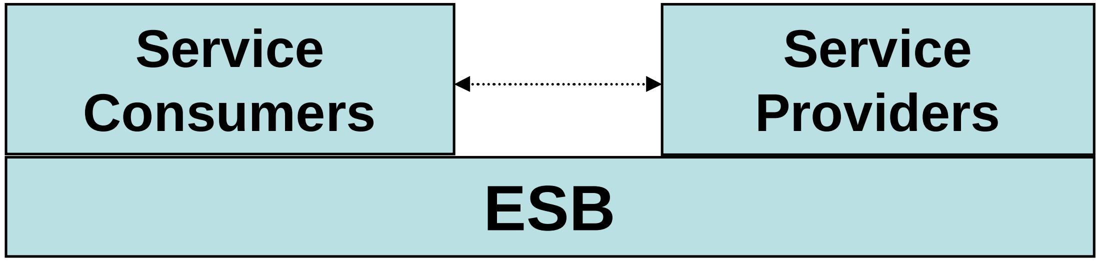
© 2008 IBM Corporation

Brief Refresher of WebSphere ESB

© 2008 IBM Corporation

What is WebSphere ESB?

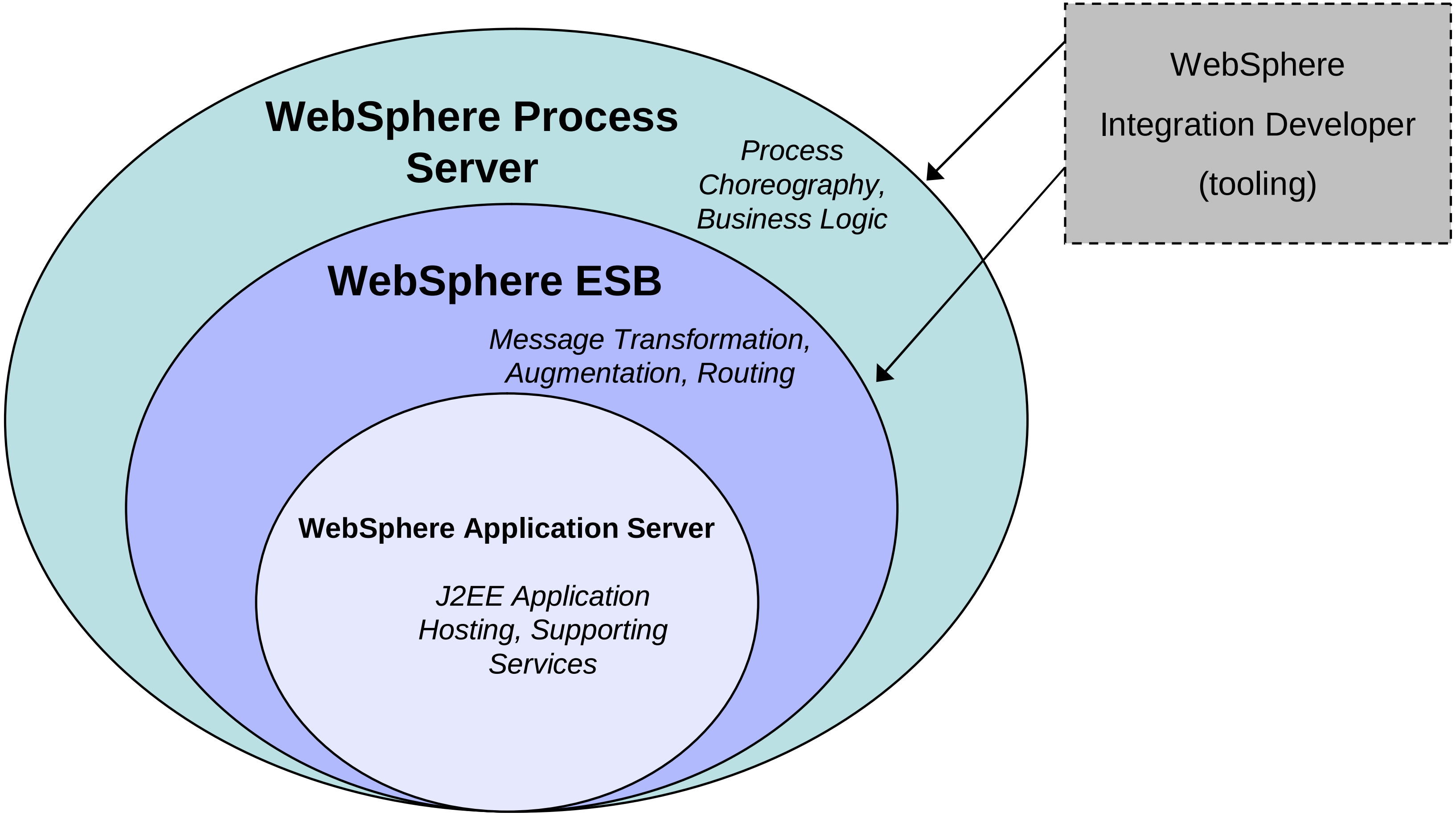
- An ESB (Enterprise Service Bus) is an architectural pattern that assists in creating an SOA environment.
- Enables routing, transformation, augmentation, aggregation, etc. of services by creating intermediary services.



- **WebSphere ESB** is one of IBM’s three ESB ‘products’.

© 2008 IBM Corporation

WebSphere SOA/Process Integration Stack



WebSphere Process Server

Process Choreography, Business Logic

WebSphere ESB

Message Transformation, Augmentation, Routing

WebSphere Application Server

J2EE Application Hosting, Supporting Services

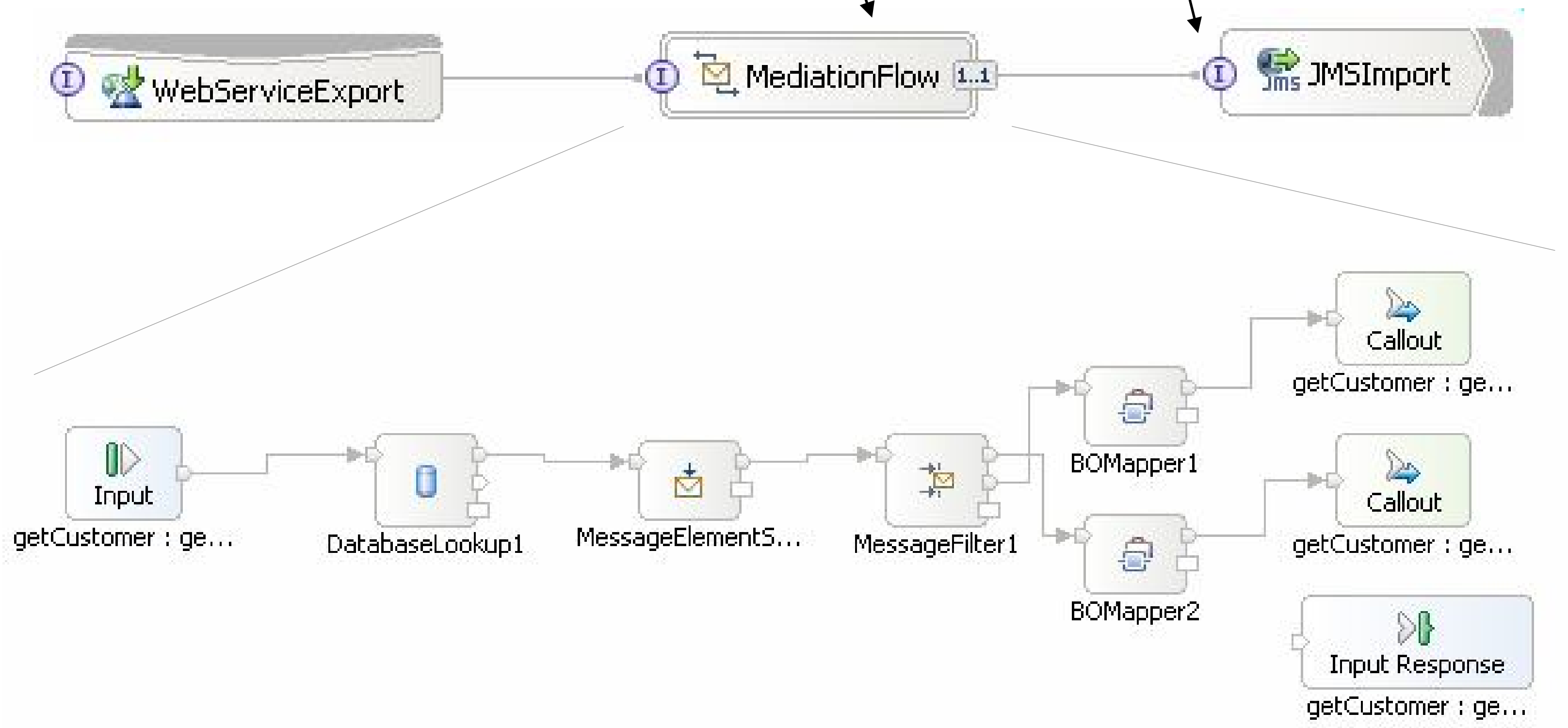
WebSphere Integration Developer (tooling)

© 2008 IBM Corporation

Mediation Module and Flow

SCA Component

(WSDL) Interface



- WPS adds Process (or Integration) modules.
- Mediation Modules can also contain Java Components.
- Mediation Modules can reference libraries that contain WSDLs, BOs, etc.

© 2008 IBM Corporation

Types of Import and Export (Bindings)

- Web Services (SOAP/HTTP and SOAP/JMS)
- Messaging:
  - WebSphere MQ and MQ / JMS
  - JMS (incl. Generic)
- HTTP
- JCA (WebSphere) Adapters
  - Application
  - Technology
- SCA ‘Default’ / Native
- Stateless Session Bean Binding (import only)
- Standalone Reference (export only)
- WebSphere Business Integration Adapters

© 2008 IBM Corporation

Service Data Object (SDO)

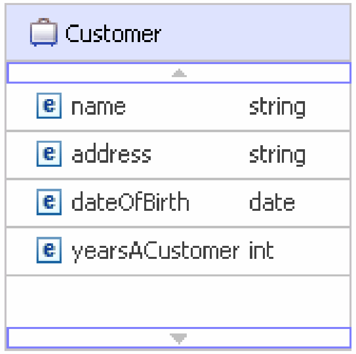
- Java API used for accessing (mostly) structured data.

```
DataObject customer = createCustomer();
customer.setString(“name”, “Fred”);
customer.setString(“address”, “123 Anytown”);
customer.setDate(“dateOfBirth”, new Date(1975, 2, 1));
customer.setYearsACustomer(0);
```

- Has serialized XML representation.

```
<customer>
  <name>Fred</name>
  <address>123 Anytown</address>
  <date>1975-02-01</date>
  <yearsACustomer>0</yearsACustomer>
</customer>
```

- **Business Object** is definition / type of SDO (underlying representation is XML Schema).



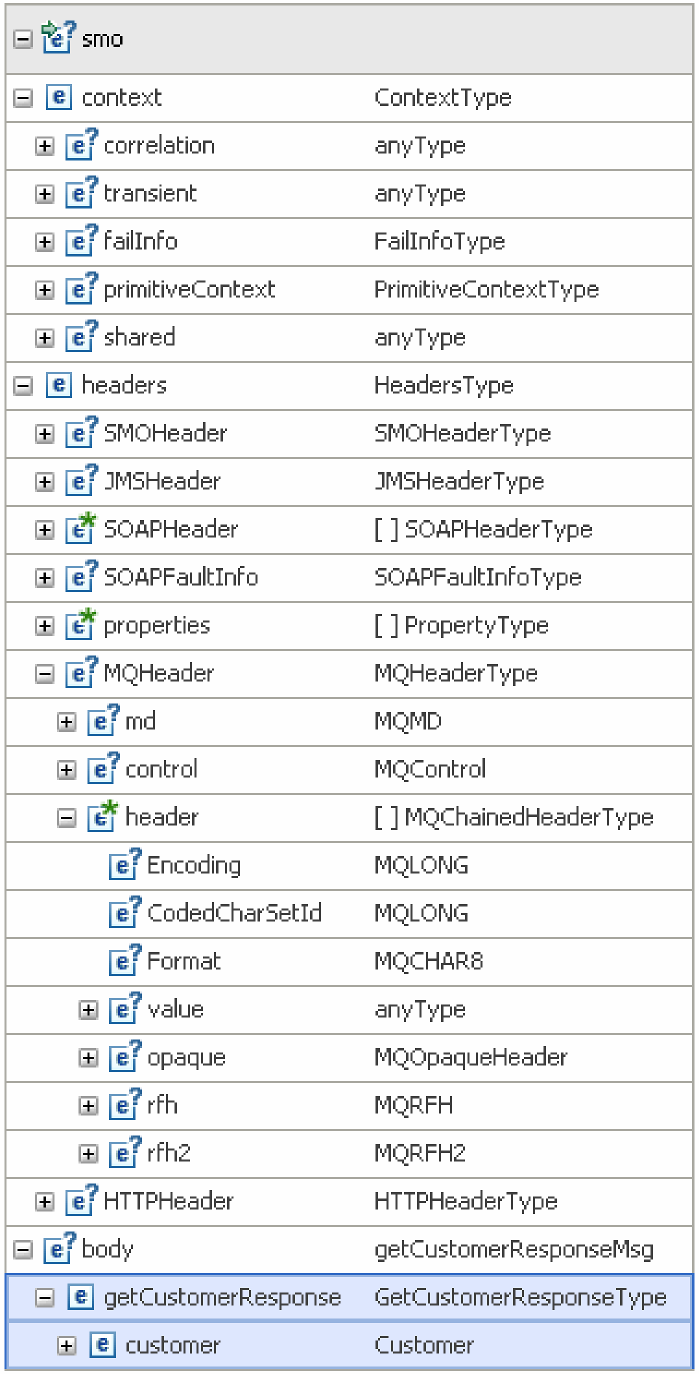
Customer

name	string
address	string
dateOfBirth	date
yearsACustomer	int

© 2008 IBM Corporation

Service Message Object

- Only used inside mediation flows
- Contains **context** (scratchpads) for various mediation functions
- Gives access to **headers** inaccessible in other SCA components
- As well as message **body** content



smo	
context	ContextType
correlation	anyType
transient	anyType
failInfo	FailInfoType
primitiveContext	PrimitiveContextType
shared	anyType
headers	HeadersType
SMOHeader	SMOHeaderType
JMSHeader	JMSHeaderType
SOAPHeader	[ ] SOAPHeaderType
SOAPFaultInfo	SOAPFaultInfoType
properties	[ ] PropertyType
MQHeader	MQHeaderType
md	MQMD
control	MQControl
header	[ ] MQChainedHeaderType
Encoding	MQLONG
CodedCharSetId	MQLONG
Format	MQCHAR8
value	anyType
opaque	MQOpaqueHeader
rfh	MQRFH
rfh2	MQRFH2
HTTPHeader	HTTPHeaderType
body	getCustomerResponseMsg
getCustomerResponse	GetCustomerResponseType
customer	Customer

© 2008 IBM Corporation

Large-scale best practices: **Design and architecture**

- Use the Right Type of Module
- Design your System Topology
- Spend Time on Interfaces and Business Objects
- Consider How you Split up Mediation Modules
- Select your Binding Types Carefully
- Document Modules and Components
- Consider your Custom Coding Strategy
- Consider your Logging Strategy
- Use Source Control & Do Automated Builds
- Do Unit Testing

© 2008 IBM Corporation

Use the Right Type of Module

- Think about mediation logic *vs.* process logic.
- Use **Mediation Modules** (WebSphere ESB & Process Server) for integration / mediation logic:
  - Short-running, minimal choreography.
  - Supports header manipulation.
- Use (Integration) **Modules** (WebSphere Process Server only) for business / process logic:
  - Can be long-running, powerful choreography and business logic.
- More Information:
  - http://www.ibm.com/developerworks/websphere/library/techarticles/0803_fasbinder2/0803_fasbinder2.html

© 2008 IBM Corporation

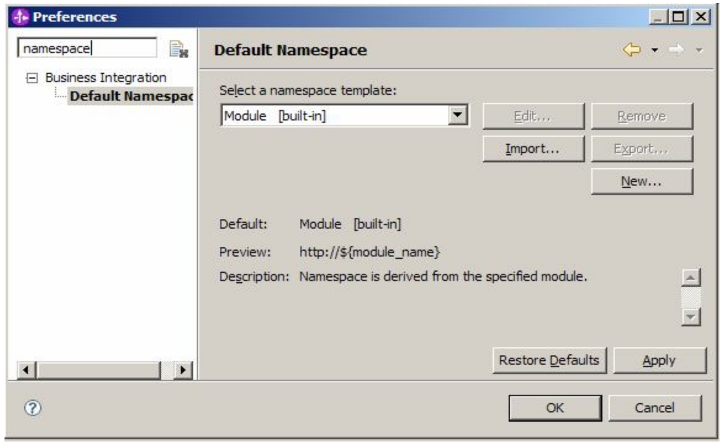
Design your System Topology

- Need more than one server?
- Using clustering? For scalability? For failover?
- Choice of topology – Bronze, Silver, Gold, …
- Mediation Modules on their own server?
- What databases do you need?
- Need a load balancer / HTTP server?
- What other systems are you connecting to and how will they ensure failover / scalability?
- DeveloperWorks article on clustering: http://www.ibm.com/developerworks/websphere/library/techarticles/0803_chilanti/0803_chilanti.html
- Redbook that discusses production topologies: http://www.redbooks.ibm.com/abstracts/sg247413.html?Open

© 2008 IBM Corporation

Spend Time on Interfaces and Business Objects

- Refactoring support is limited inside mediation flows, so good to get this right first time round.
- Adopt a naming convention.
- Add constraints?
- Add modelled faults?
- Think about namespaces.
- Configure default namespace policy before you start.



Select a namespace template: Module [built-in]

Default: Module [built-in]

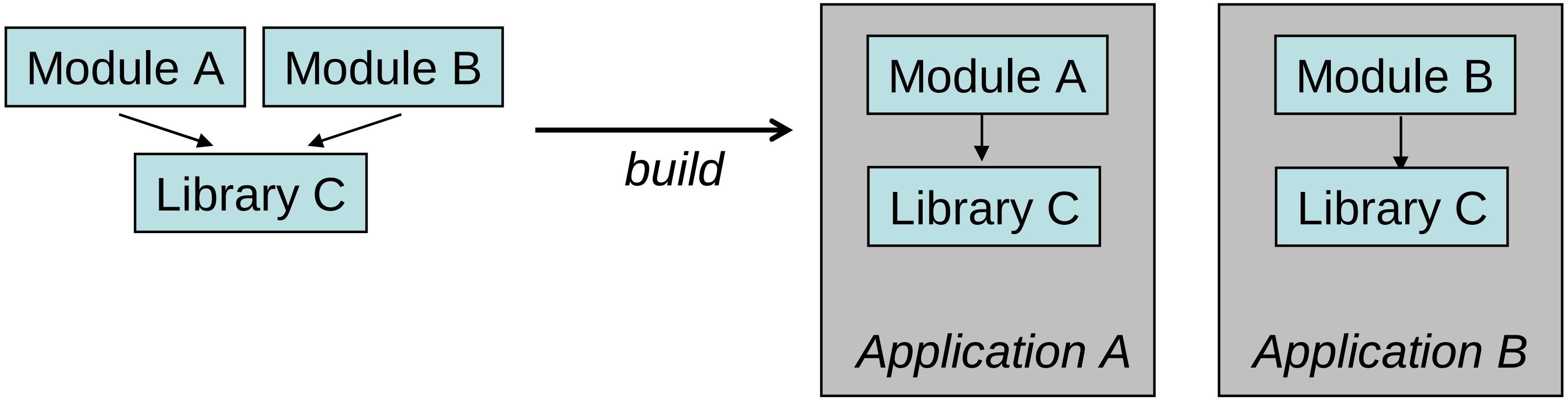
Preview: http://${module_name}

Description: Namespace is derived from the specified module.

© 2008 IBM Corporation

Consider How you Split up Mediation Modules

- How many mediation flows inside each mediation flow component?
  - **Large** number of modules impacts performance / deployment.



  - **Small** number impacts ease of development.
- Remove unused library content.

© 2008 IBM Corporation

Select your Binding Types Carefully

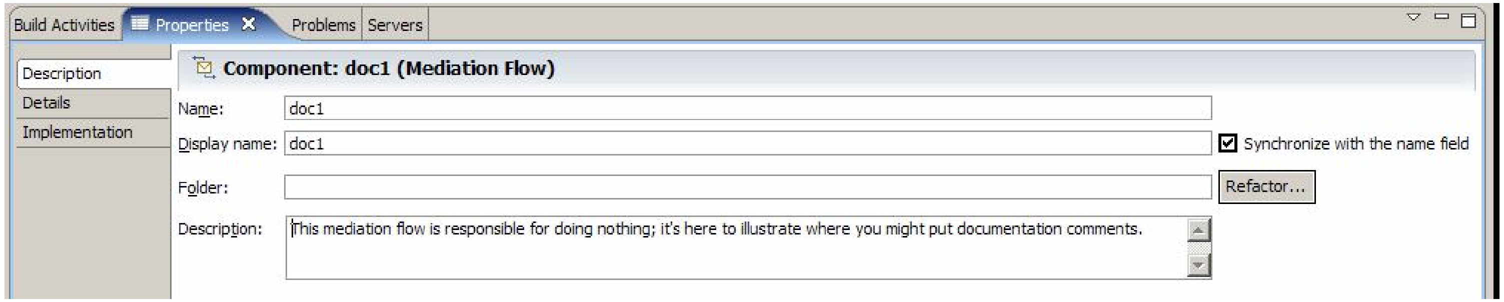
- Often binding type dictated by circumstance.
- But if you have the scope to decide:
  - Prefer **SCA default/native** for inter-ESB/WPS communications – fast, efficient, and simple
  - Prefer **Web Services** for synchronous service exposure – mature, integrates well into SDO model.
  - Prefer **JMS** for asynchronous service exposure – integrates well with WAS platform.
- Sometimes you have alternatives. For example:
  - Web Services binding – allows easy access to SOAP headers

    or
  - HTTP with SOAP data binding – allows access to HTTP headers but not SOAP headers

© 2008 IBM Corporation

Document Modules and Components (1/2)

- Specify description property to describe component in WID:



Component: doc1 (Mediation Flow)

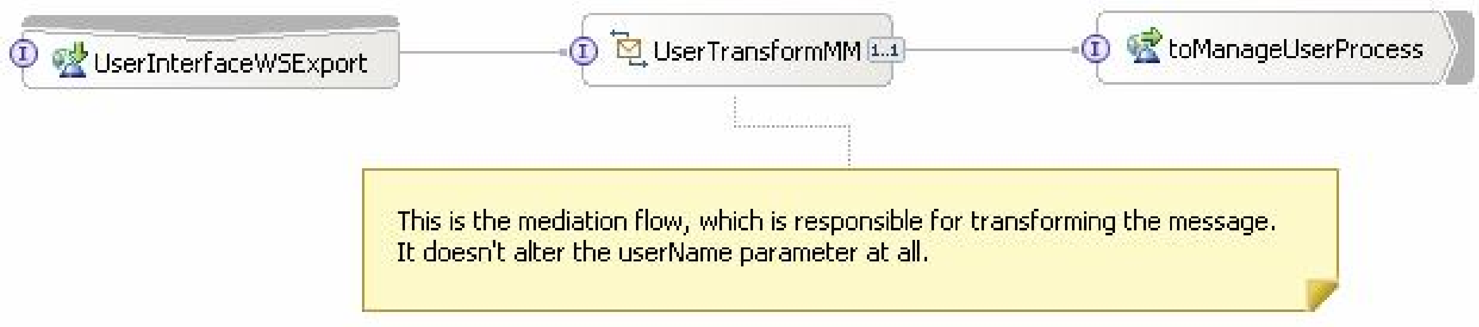
Name: doc1

Display name: doc1 ☑ Synchronize with the name field

Folder: Refactor...

Description: This mediation flow is responsible for doing nothing; it's here to illustrate where you might put documentation comments.

- As of WID 6.1.2, can also add notes:

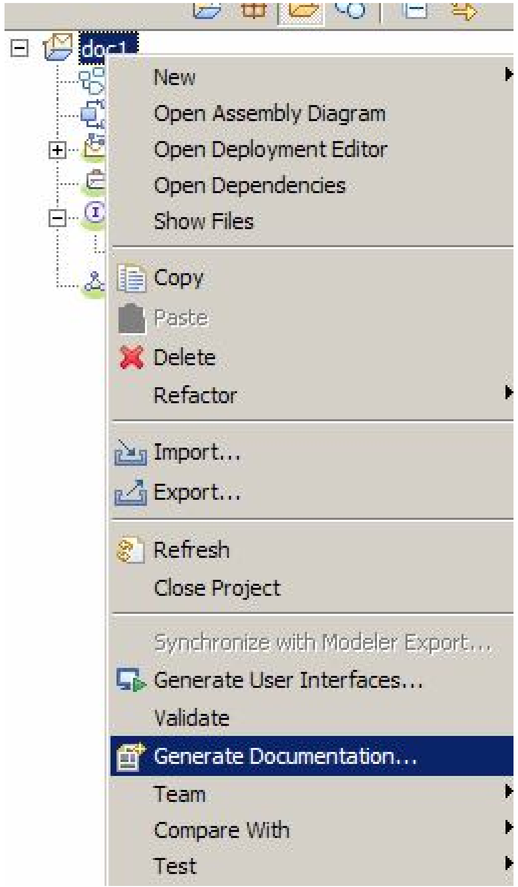


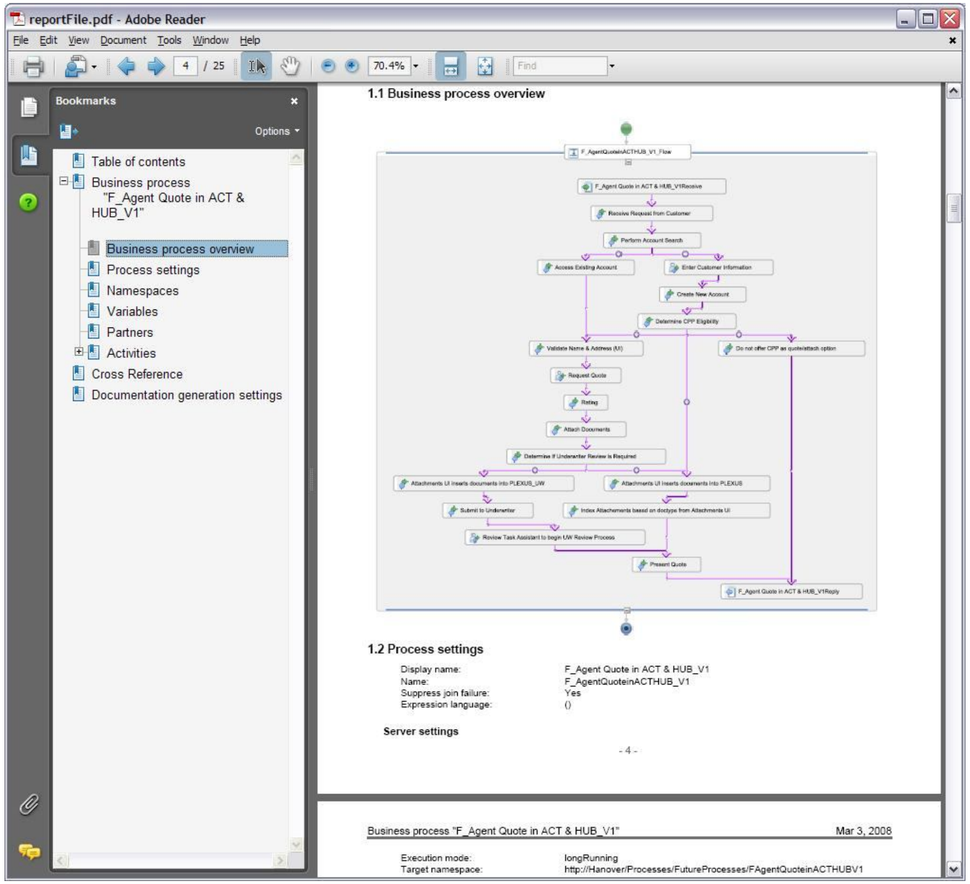
This is the mediation flow, which is responsible for transforming the message.
It doesn't alter the userName parameter at all.

© 2008 IBM Corporation

Document Modules and Components (2/2)

Can use ‘Generate Documentation’ to generate a full PDF describing your module:




reportFile.pdf - Adobe Reader
File Edit View Document Tools Window Help
4 / 25
70.4%
Find
Bookmarks
Options
Table of contents
Business process "F_Agent Quote in ACT & HUB_V1"
Business process overview
Process settings
Namespaces
Variables
Partners
Activities
Cross Reference
Documentation generation settings
F_AgentQuoteinACTHUB_V1_Flow
F_Agent Quote in ACT & HUB_V1Receive
Receive Request from Customer
Perform Account Search
Access Existing Account
Enter Customer Information
Create New Account
Determine CPP Eligibility
Validate Name & Address (UI)
Do not offer CPP as quote/attach option
Request Quote
Rating
Attach Documents
Determine if Underwriter Review is Required
Attachments UI inserts documents into PLEXUS_UW
Attachments UI inserts documents into PLEXUS
Submit to Underwriter
Index Attachements based on doctype from Attachments UI
Review Task Assistant to begin UW Review Process
Present Quote
F_Agent Quote in ACT & HUB_V1Reply

1.1 Business process overview

1.2 Process settings

Display name:	F_Agent Quote in ACT & HUB_V1
Name:	F_AgentQuoteinACTHUB_V1
Suppress join failure:	Yes
Expression language:	()

Server settings

- 4 -

Business process "F_Agent Quote in ACT & HUB_V1" Mar 3, 2008

Execution mode:	longRunning
Target namespace:	http://Hanover/Processes/FutureProcesses/FAgentQuoteinACTHUBV1

© 2008 IBM Corporation

Consider your Custom Coding Strategy

- Custom mediation:
  - Most useful for one-off coding.
  - Cannot be re-used between modules.
  - ‘Visual’ mode available which may be useful to those less comfortable with Java/SDO API.
- Custom primitive (also called roll-your-own primitive):
  - A first-class new primitive – same abilities as any other primitive type (XSLT, Endpoint Lookup…).
  - Can have customisable properties.
  - Appears in palette in WID.
  - More re-usable, but more work to create.

© 2008 IBM Corporation

Consider your Logging Strategy

- You will want one, consider it before you start developing.
- Options include:
  - Message Logger – limited functionality – logs only to a fixed schema database table.
  - JDBC or Flat File Adapter (in separate mediation module?)
  - Custom mediations – basic visual snippets for logging.
  - Custom primitives.

© 2008 IBM Corporation

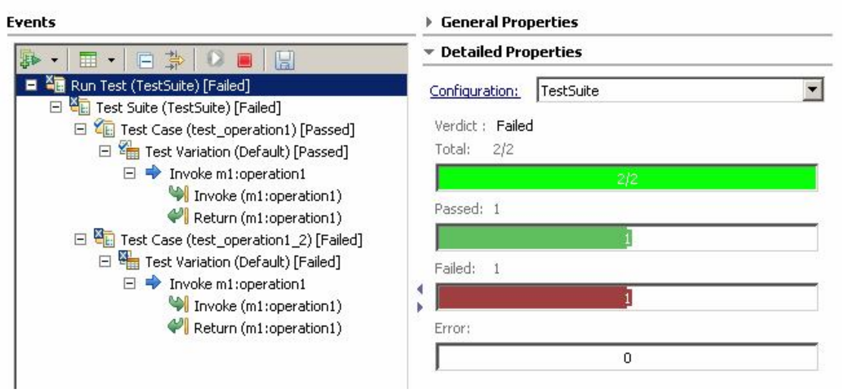
Use Source Control & Do Automated Builds

- Use source control – WID/Eclipse integrates with several.
- Only one developer per mediation module at once.
- Automated build direct from source control.
  - WebSphere ESB is supplied with the *serviceDeploy* tool for this purpose.
- Article gives a good example of this process, integrate with Rational ClearCase:
  - http://www.ibm.com/developerworks/websphere/library/techarticles/0711_manekar/0711_manekar.html

© 2008 IBM Corporation

Do Unit Testing

- As of version 6.1, WebSphere Integration Developer has support for unit testing.
- Use it before check-in.
- Can be run from command line as part of automation.
- Article with more information:
  - http://www.ibm.com/developerworks/websphere/library/techarticles/0806_gregory/0806_gregory.html



```
Run Test (TestSuite) [Failed]
  Test Suite (TestSuite) [Failed]
    Test Case (test_operation1) [Passed]
      Test Variation (Default) [Passed]
        Invoke m1:operation1
          Invoke (m1:operation1)
          Return (m1:operation1)
    Test Case (test_operation1_2) [Failed]
      Test Variation (Default) [Failed]
        Invoke m1:operation1
          Invoke (m1:operation1)
          Return (m1:operation1)
```

Configuration: TestSuite

Verdict : Failed

Total: 2/2

2/2

Passed: 1

1

Failed: 1

1

Error:

0

‘Small-scale’ Best Practices:
Hints and tips to make your life easier

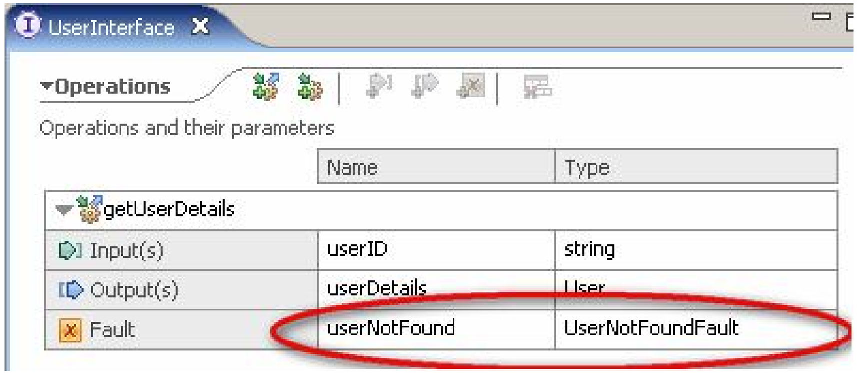
- Handle Modelled Faults
- Handle Unmodelled Faults where Appropriate
- Understand your Message Manipulation Choices
- Promote Properties where Relevant
- Use Visual Snippets in Custom Mediations
- Use Correct Message Context
- Understand the Synchronicity of Invocations
- Understand the Transactionality of Components
- Use Data Bindings (and Data Handlers) properly
- Use the Range of Debugging Tools Available

© 2008 IBM Corporation

Handle Modelled Faults

- a.k.a *business* or *checked*
- Don’t ignore them – consider them to be like declared faults in Java.
- Log it, then depending on nature of fault:
  - **Business level fault**: pass it on
    - Mediation does not include business logic
    - Maybe do transformation
  - **Infrastructure level fault**: Pass it on:
    - Mediate into a generic fault for business logic

    Or deal with it:
    - Retry?

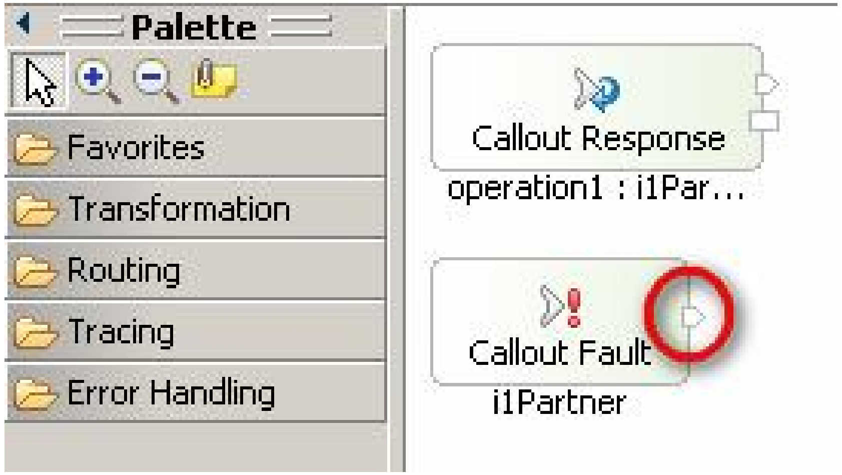


UserInterface

Operations

Operations and their parameters

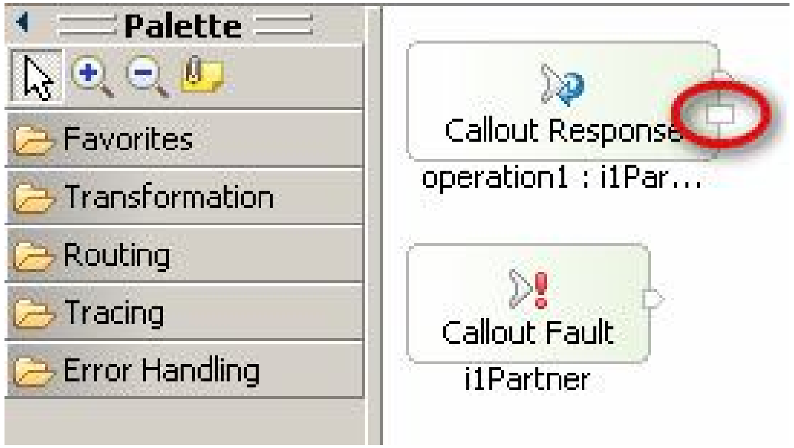
	Name	Type
getUserDetails		
Input(s)	userID	string
Output(s)	userDetails	User
Fault	userNotFound	UserNotFoundFault



© 2008 IBM Corporation

Handle Unmodelled Faults where Appropriate

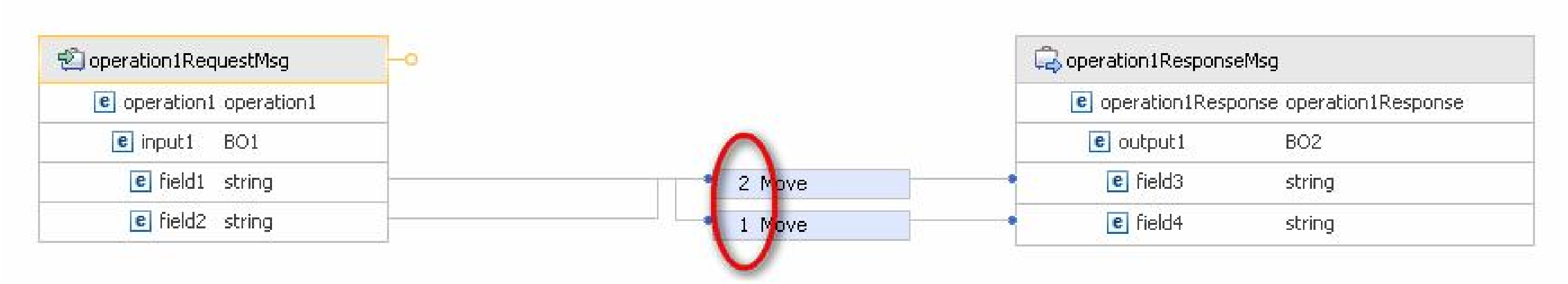
- Aren’t declared on an interface.
- a.k.a. *system*, *runtime* or *unchecked*.
- Appear at fail terminal of the callout node in the response flow
- Failure message found in SMO context.
- Useful where:
  - Interacting with a system that throws runtime faults that you want to capture.
  - Interacting with an interface that hasn’t declared faults, but should have. Façading approach: http://www.ibm.com/developerworks/websphere/library/techarticles/0802_lezajic/0802_lezajic.html



© 2008 IBM Corporation

Understand your Message Manipulation Choices

- *Message Element Setter* – simple, high performance. Cannot alter message type. Parts of element map are directly promotable.
- *XSLT* – prefer XSLT when you want to use XML functions or work with XSLT directly. Also more performant in some cases – such as when working with Web Services (but test performance!)
- *BO Map* – if want to share BO maps with WPS, or need/want ordering capabilities of the BO mapper:



operation1RequestMsg
operation1 operation1
input1 BO1
field1 string
field2 string

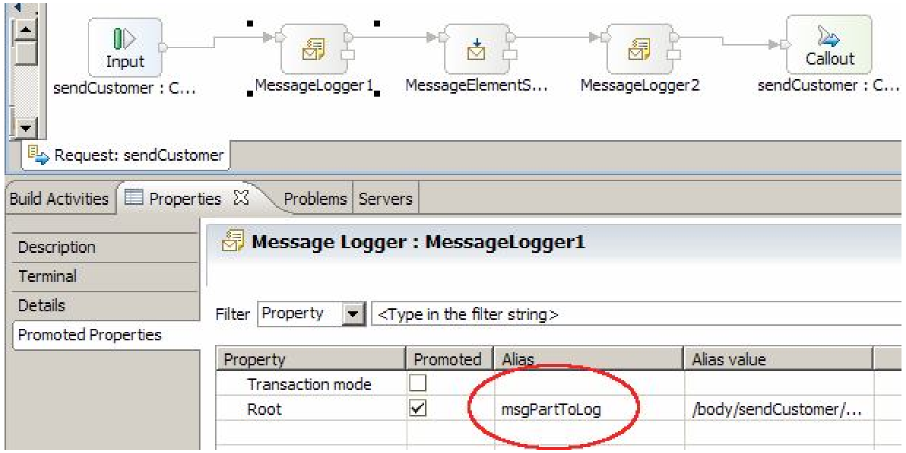
2 Move
1 Move

operation1ResponseMsg
operation1Response operation1Response
output1 BO2
field3 string
field4 string

© 2008 IBM Corporation

Promote Properties where Relevant

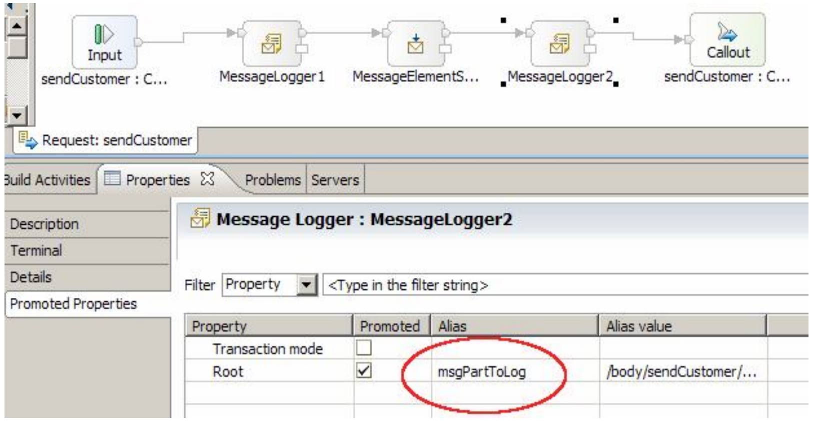
- Link them with the same name, where relevant, so that they can be changed together



Message Logger : MessageLogger1

Filter Property ▾ <Type in the filter string>

Property	Promoted	Alias	Alias value
Transaction mode	□		
Root	☑	msgPartToLog	/body/sendCustomer/...



Message Logger : MessageLogger2

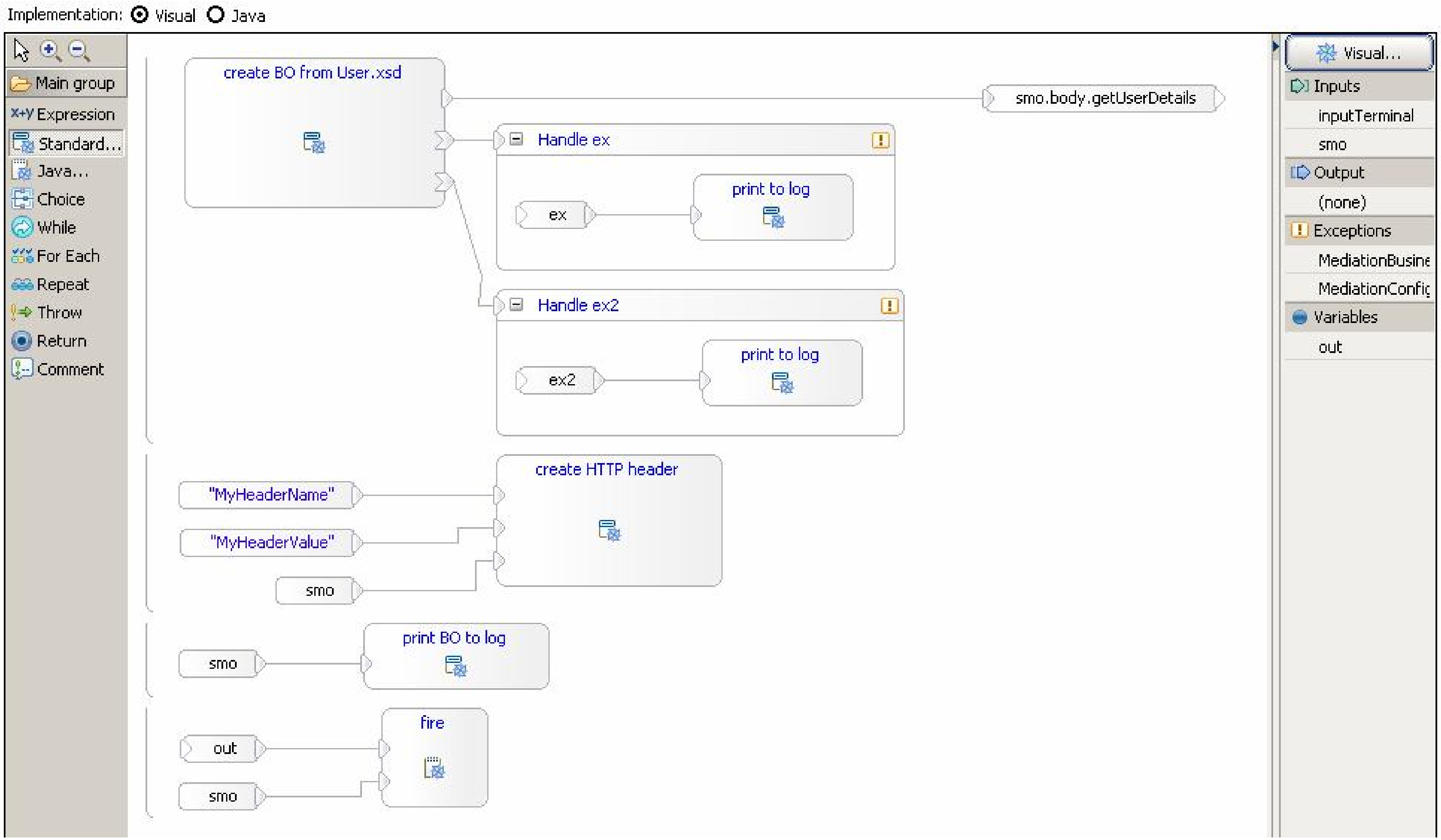
Filter Property ▾ <Type in the filter string>

Property	Promoted	Alias	Alias value
Transaction mode	□		
Root	☑	msgPartToLog	/body/sendCustomer/...

- Be aware that there is a minor performance penalty so don’t promote with abandon, particularly where performance is a concern.

© 2008 IBM Corporation

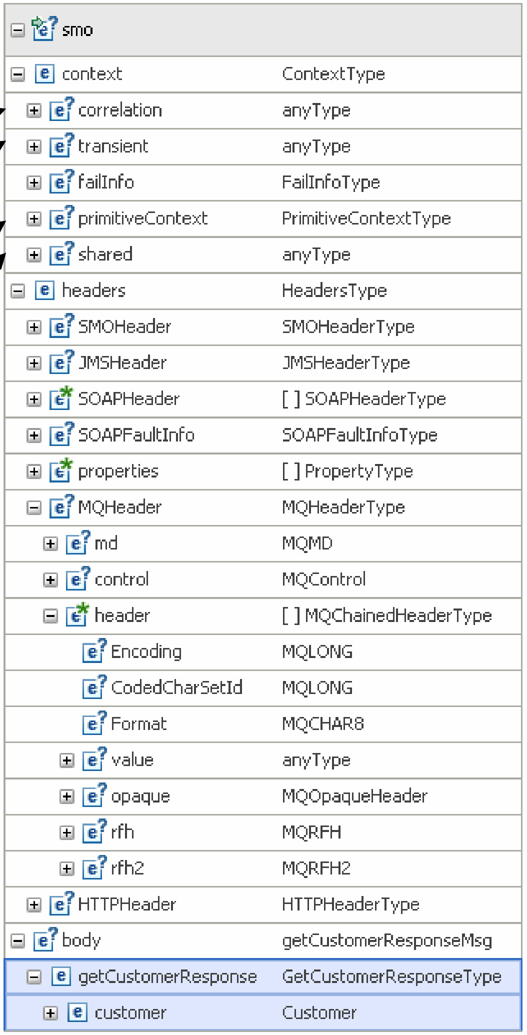
Use Visual Snippets in Custom Mediations



© 2008 IBM Corporation

Use Correct Message Context

- Use Message Context area appropriate for inter-primitive communication:
  - **`correlation`** – scratchpad for communicating between request and response flows.
  - **`transient`** – scratchpad within a flow.
  - **`primitiveContext/ FanOutContext`** – used when iterating using the Fan Out / In primitives.
  - **`shared`** – used to aggregate responses from Service Invokes during a Fan Out / In.
- More information:
  - http://www.ibm.com/developerworks/webservices/library/ws-websphereesb3/index.html?ca=drs-



smo	
context	ContextType
correlation	anyType
transient	anyType
failInfo	FailInfoType
primitiveContext	PrimitiveContextType
shared	anyType
headers	HeadersType
SMOHeader	SMOHeaderType
JMSHeader	JMSHeaderType
SOAPHeader	[ ] SOAPHeaderType
SOAPFaultInfo	SOAPFaultInfoType
properties	[ ] PropertyType
MQHeader	MQHeaderType
md	MQMD
control	MQControl
header	[ ] MQChainedHeaderType
Encoding	MQLONG
CodedCharSetId	MQLONG
Format	MQCHAR8
value	anyType
opaque	MQOpaqueHeader
rfh	MQRFH
rfh2	MQRFH2
HTTPHeader	HTTPHeaderType
body	getCustomerResponseMsg
getCustomerResponse	GetCustomerResponseType
customer	Customer

© 2008 IBM Corporation

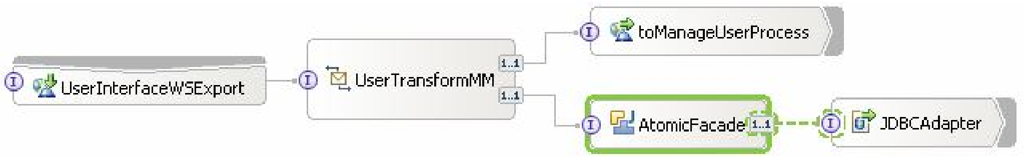
Understand the Synchronicity of Invocations

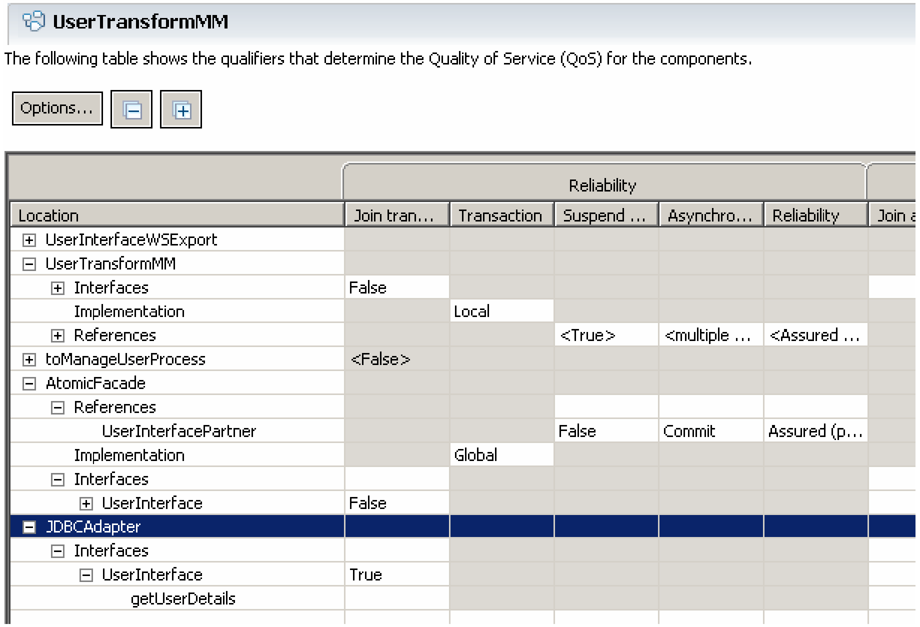
- Default ‘invocation style’ sometimes dictated by bindings.
- Often OK – but be aware of it.
- Async interactions (between components or modules) go via an SCA queue. Implies:
  - Breaking transactional scope
  - Runtime exception, after retry limit, roll onto exception destination. Handling method depends on product:
    - **WPS** has Failed Event Manager
    - **WESB** needs an app/human to read errors from system exception destination, or exception destination to be disabled
- Can be hard to predict when interactions will be async – subtleties in performance optimisation, etc. Assume async if in doubt, and use ‘preferredInteractionStyle’ liberally.
- More details here: http://www.ibm.com/developerworks/webservices/library/ws-sca-patterns/index.html?ca=drs-

© 2008 IBM Corporation

Understand the Transactionality of Components

- Affects what happens when errors occur.
- If you are interacting with JDBC or a messaging system – probably want transactionality.
- Not the default.
- Use the new Transaction Highlighting and Qualifiers editor in WID 6.1.2.
- More information: http://soatipsntricks.wordpress.com/2008/07/31/transactionality-in-sca-part-2-refactoring-interfaces/





UserTransformMM

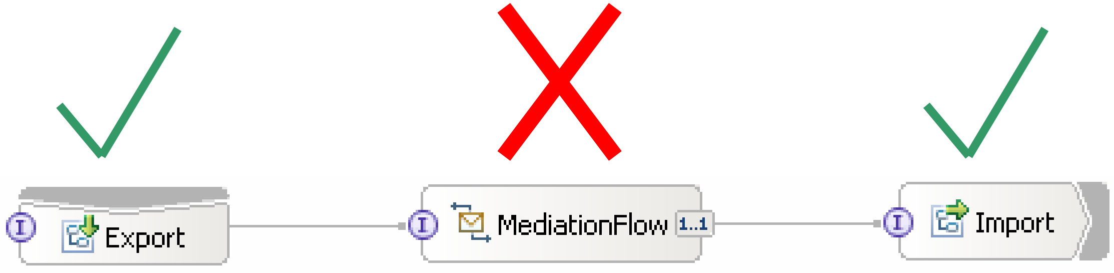
The following table shows the qualifiers that determine the Quality of Service (QoS) for the components.

	Reliability					
Location	Join tran...	Transaction	Suspend ...	Asynchro...	Reliability	Join a
⊞ UserInterfaceWSExport						
⊟ UserTransformMM						
⊞ Interfaces	False					
Implementation		Local				
⊞ References			<True>	<multiple ...	<Assured ...	
⊞ toManageUserProcess	<False>					
⊟ AtomicFacade						
⊟ References						
UserInterfacePartner			False	Commit	Assured (p...	
Implementation		Global				
⊟ Interfaces						
⊞ UserInterface	False					
⊟ JDBCAdapter						
⊟ Interfaces						
⊟ UserInterface	True					
getUserDetails						

© 2008 IBM Corporation

Use Data Bindings (and Data Handlers) properly

- Data Bindings (and Data Handlers) should be used on the boundaries of a set of modules to transform from and to proprietary formats.
- Don’t use the simple JMS and MQ bindings, then do the work of a data bindings in a mediation module:
  - Unnecessarily complicates the flow
  - Reduces opportunity for reuse with data handlers
- Only use the simple data bindings sparingly.



© 2008 IBM Corporation

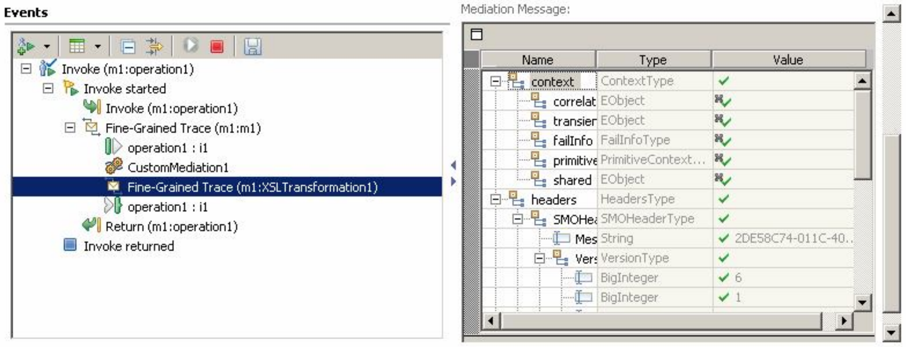
Use the Range of Debugging Tools Available (1/2)

- Re-run your unit tests so you know what’s failing.
- Review the content of application and server logs (such as *SystemOut.log*).
  - Read the whole stack trace.
  - Add more logging / increase logging levels.

© 2008 IBM Corporation

Use the Range of Debugging Tools Available (2/2)

- Use Component Test Client
- New fine-grained trace in WID 6.1.2



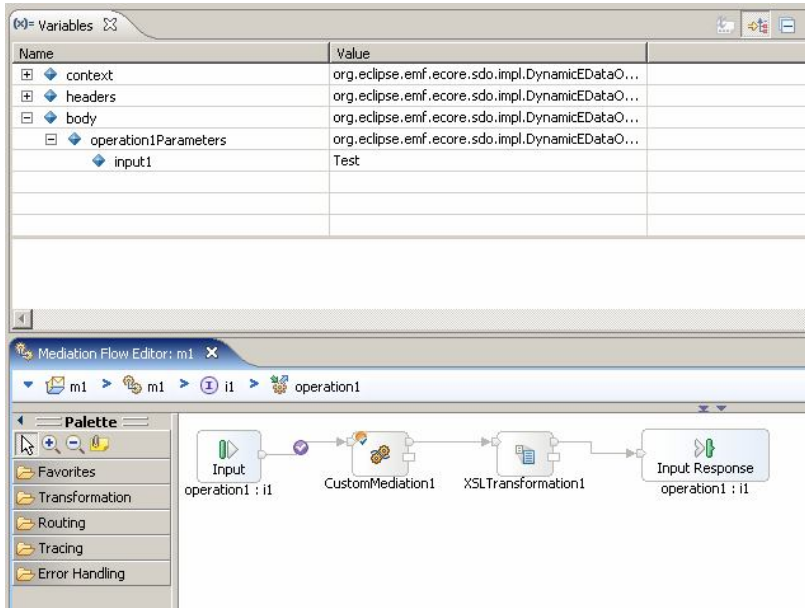
Events

```
⊟ Invoke (m1:operation1)
    ⊟ Invoke started
          Invoke (m1:operation1)
        ⊟ Fine-Grained Trace (m1:m1)
              operation1 : i1
              CustomMediation1
              Fine-Grained Trace (m1:XSLTransformation1)
              operation1 : i1
          Return (m1:operation1)
      Invoke returned
```

Mediation Message:

Name	Type	Value
⊟ context	ContextType	✓
correlat	EObject	✓
transier	EObject	✓
failInfo	FailInfoType	✓
primitive	PrimitiveContext…	✓
shared	EObject	✓
⊟ headers	HeadersType	✓
⊟ SMOHea	SMOHeaderType	✓
Mes	String	✓ 2DE58C74-011C-40…
⊟ Vers	VersionType	✓
	BigInteger	✓ 6
	BigInteger	✓ 1

- Use Debugger
- Supports Breakpoints, Step Over, Inspection of SMO, etc.



Variables

Name	Value
⊞ context	org.eclipse.emf.ecore.sdo.impl.DynamicEDataO…
⊞ headers	org.eclipse.emf.ecore.sdo.impl.DynamicEDataO…
⊟ body	org.eclipse.emf.ecore.sdo.impl.DynamicEDataO…
⊟ operation1Parameters	org.eclipse.emf.ecore.sdo.impl.DynamicEDataO…
input1	Test

Mediation Flow Editor: m1

Large-scale best practices: **Design and architecture**

- Use the Right Type of Module
- Design your System Topology
- Spend Time on Interfaces and Business Objects
- Consider How you Split up Mediation Modules
- Select your Binding Types Carefully
- Document Modules and Components
- Consider your Custom Coding Strategy
- Consider your Logging Strategy
- Use Source Control & Do Automated Builds
- Do Unit Testing

© 2008 IBM Corporation

‘Small-scale’ Best Practices:
Hints and tips to make your life easier

- Handle Modelled Faults
- Handle Unmodelled Faults where Appropriate
- Understand your Message Manipulation Choices
- Promote Properties where Relevant
- Use Visual Snippets in Custom Mediations
- Use Correct Message Context
- Understand the Synchronicity of Invocations
- Understand the Transactionality of Components
- Use Data Bindings (and Data Handlers) properly
- Use the Range of Debugging Tools Available

© 2008 IBM Corporation

References and Further Information

Andrew Ferrier

andrew.ferrier@uk.ibm.com

- InfoCenter: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r1mx/index.jsp
- WebSphere ESB Support Site: http://www-01.ibm.com/software/integration/wsesb/support/
- WebSphere ESB on DeveloperWorks: http://www.ibm.com/developerworks/websphere/zones/businessintegration/wesb.html
- SOA Tips 'n' Tricks Blog: http://soatipsntricks.wordpress.com/

© 2008 IBM Corporation