



IBM Software Group

Enterprise – Business Process Management/ Business Performance Management: Architecture, Technology, Standards

Donald F. Ferguson

IBM Fellow, SWG Chief Architect,
dff@us.ibm.com





Agenda

- Introduction
 - ▶ Disclaimer
 - ▶ A sample business problem
 - ▶ Context and major trends
- Business Process/Performance Management
 - ▶ Model, and why more than “process.”
 - ▶ Assemble and Customize; EDA and SOA
 - ▶ Deploy
 - ▶ Manage/Monitor
- Summary, Discussion and Challenges,
and **“A Grand Challenge!”**

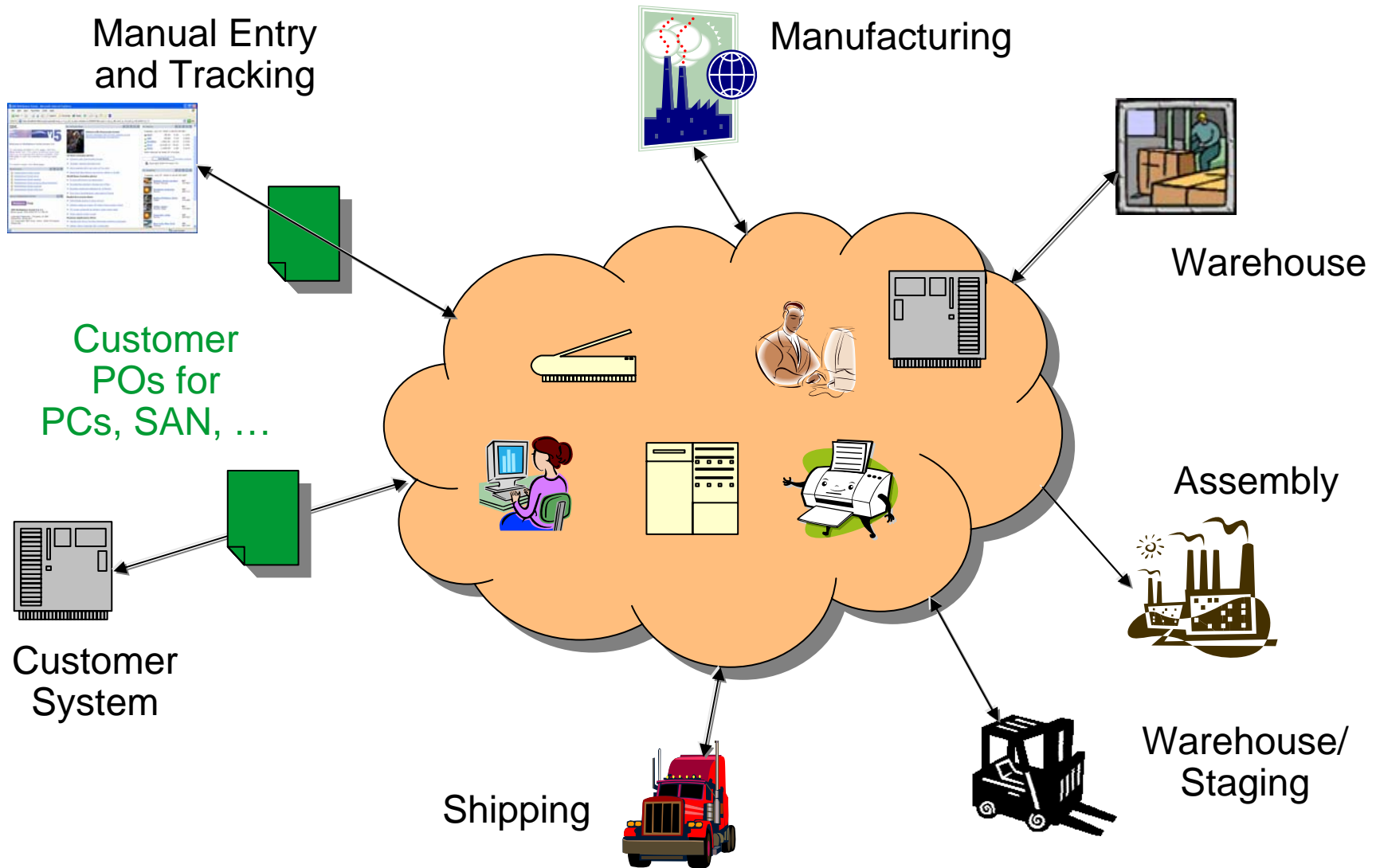


Agenda

- Introduction
 - ▶ Disclaimer
 - ▶ A sample business problem
 - ▶ Context and major trends
- Business Process/Performance Management
 - ▶ Model, and why more than “process.”
 - ▶ Assemble and Customize; EDA and SOA
 - ▶ Deploy
 - ▶ Manage/Monitor
- Summary, Discussion and Challenges,
and **“A Grand Challenge!”**



A Sample Business Problem – HW PO/Supply Chain



5



Some Challenges and Trends

■ Challenges

- ▶ Seams between
 - “people activities,” “information integrated” and “automated SOA activities”
 - Choice may change in a solution over time
- ▶ Reusable solution templates with customization/configuration.
- ▶ Policy and rules
- ▶ Information –
 - Processes are a mix of “documents,” “people” and choreography.
 - Rich information model – PO, Customer,
- ▶ Legacy integration
- ▶ Federated/decentralized control and goals

■ Some trends

- ▶ SOA and Web services, obviously.
- ▶ Improved but fragmented formal modeling standards
- ▶ Coming together of IT processes and business processes (MUWS)
- ▶ Coherent models for EDA and {BPM, EAI}
- ▶ Domain standards
- ▶ Governance.

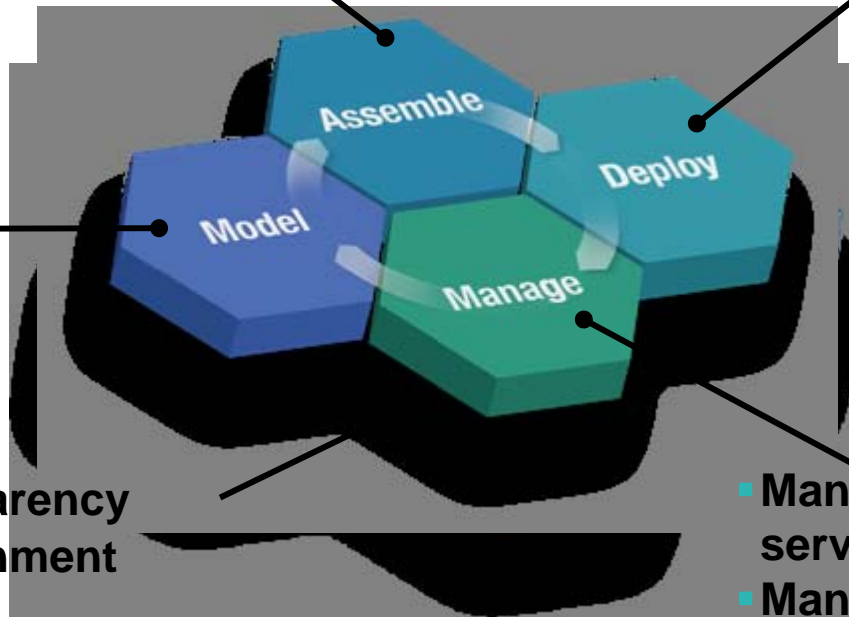


Business Process and Performance Management: Model – Assemble – Deploy – Manage/Monitor

- Discover
- Construct & Test
- Compose

- Integrate people
- Integrate processes
- Manage and integrate information

- Gather requirements
- Model & Simulate
- Design
- A “better napkin”



- Financial transparency
- Business/IT alignment
- Process control
- Charge back
- Who saw what and did what for whom?

- Manage applications & services
- Manage identity & compliance
- Monitor business metrics



Agenda

- Introduction
 - ▶ Disclaimer
 - ▶ A sample business problem
 - ▶ Context and major trends
- Business Process/Performance Management
 - ▶ Model, and why more than “process.”
 - ▶ Assemble and Customize; EDA and SOA
 - ▶ Deploy
 - ▶ Manage/Monitor
- Summary, Discussion and Challenges,
and **“A Grand Challenge!”**

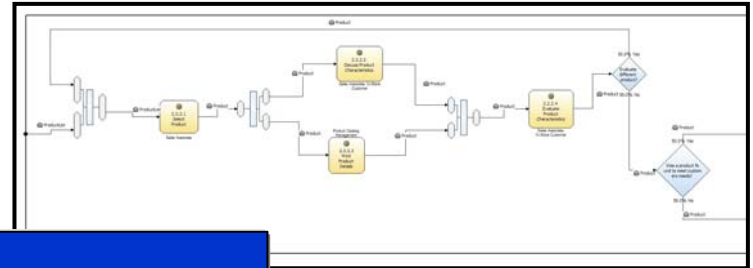


Model

■ BPM is more than process

- ▶ Business Objects, Content Model (e.g. scans)
- ▶ Business and object state
- ▶ Organization (people)
- ▶ Interaction
- ▶ Policy
- ▶ Key Performance Indicators
- ▶ Observation, Events
- ▶ **Business Vocabulary**

Model in "business user" terms and concepts, not XSD or com.payroll. ...



■ Standards matter

- ▶ Runtime interoperability
- ▶ **Federated tools**
- ▶ **Portability**
- ▶ Monitoring and reporting
- ▶ Evolution, Substitutability

Standards are more than runtime protocols

Design Time

- What does company A's tools give to B's?
- How does the caller know invocation seqs?
- What CA's does the service support?

Portability

- Flexible placement over disparate product choices
- Decouple BPM from infrastructure evolution

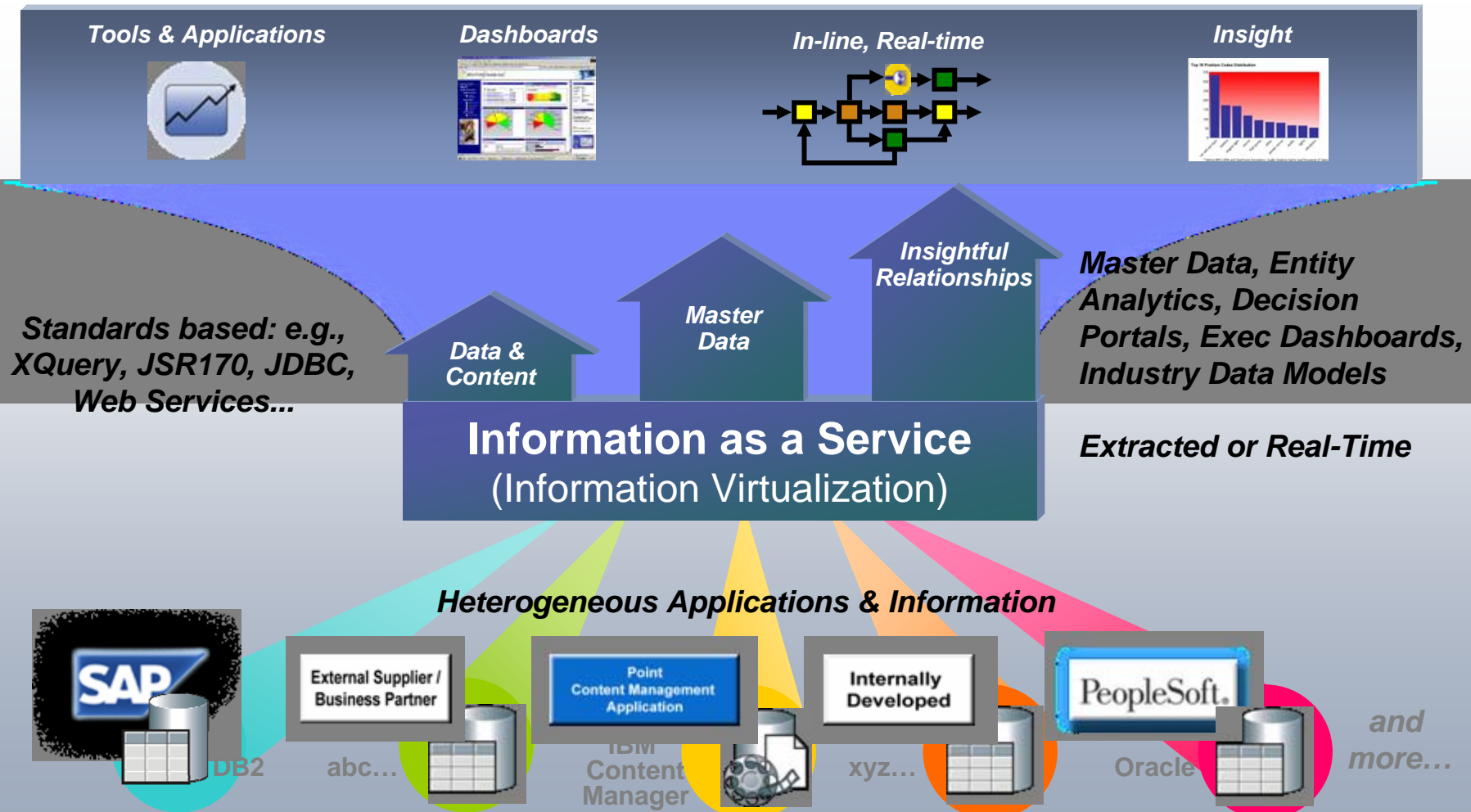
■ Currently a bit of a mess

- ▶ UML
- ▶ E-R
- ▶ WSDL, BPEL, ...
- ▶ BPMN
- ▶ SBVR



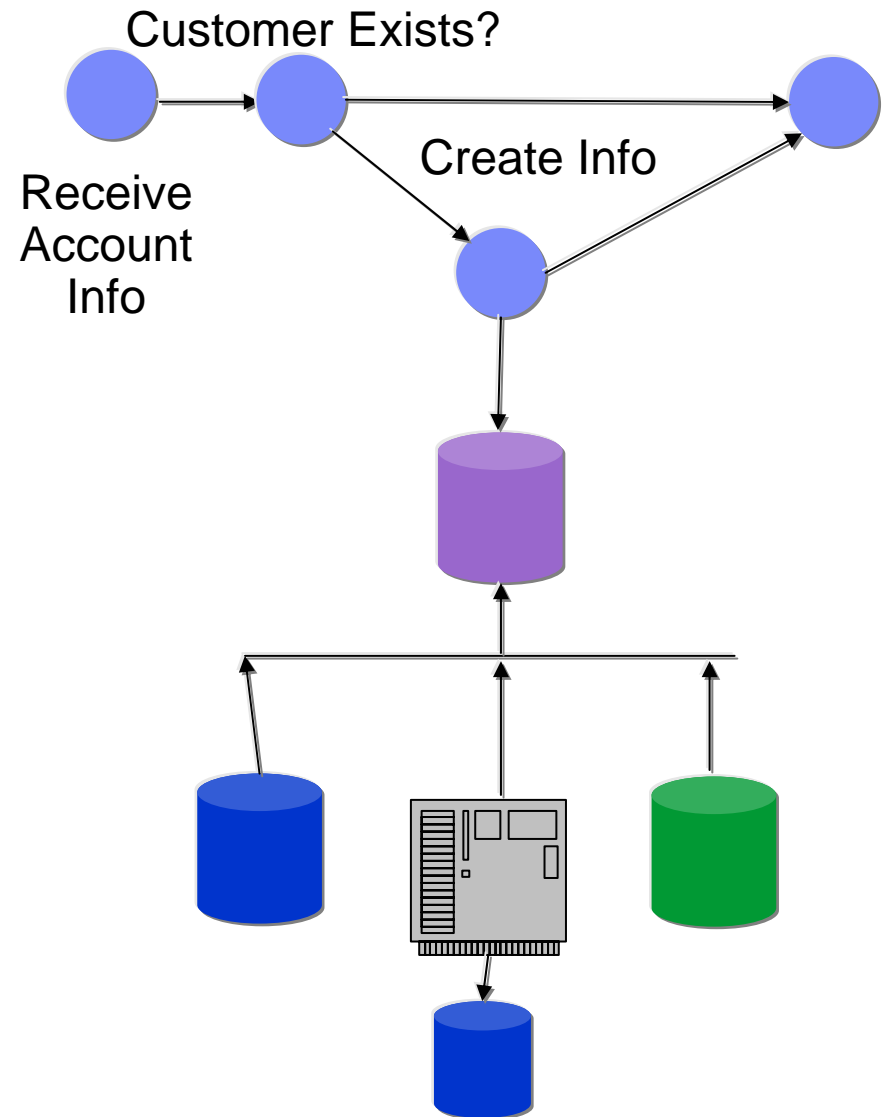
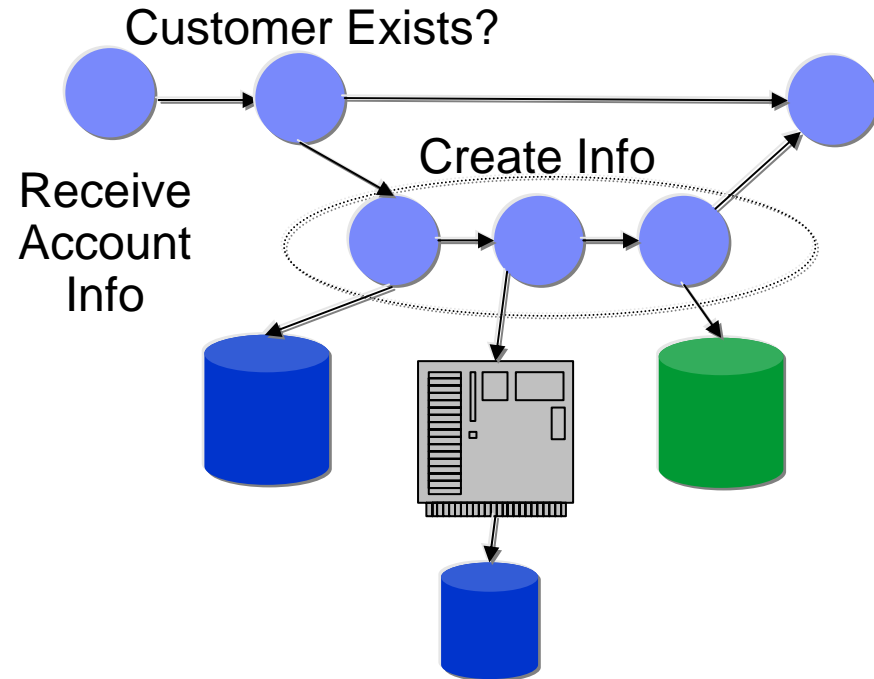


Information as a Service – An Example of Going Beyond Process





Why Use Information Services in SOA?



Why use Information Server/Services?

- Efficiency
- Sophisticated clean/link/... .. functions
- Simplifies process design
- Reuse in other processes, as well as applications, portal,

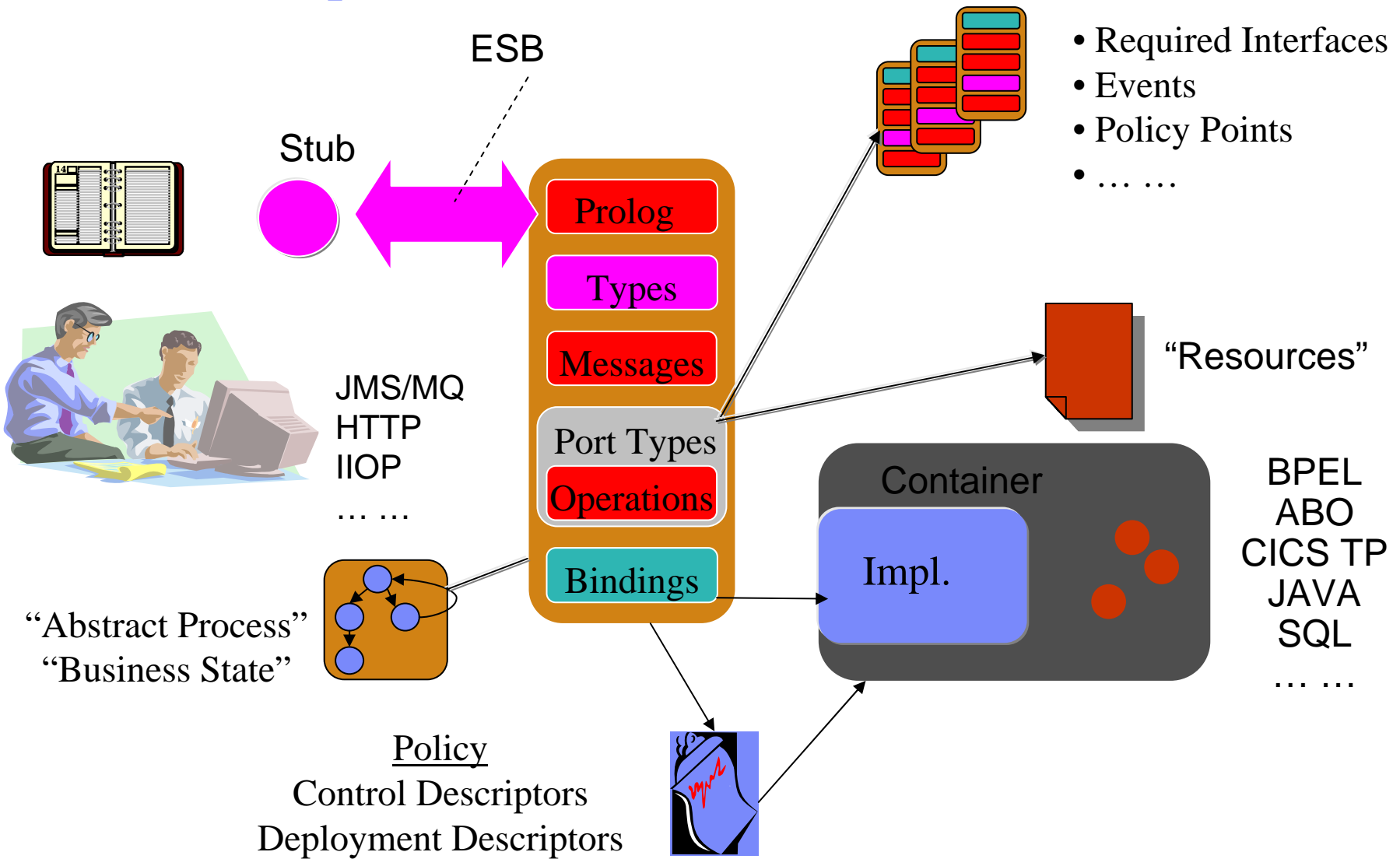


Agenda

- Introduction
 - ▶ Disclaimer
 - ▶ A sample business problem
 - ▶ Context and major trends
- Business Process/Performance Management
 - ▶ Model, and why more than “process.”
 - ▶ Assemble and Customize; EDA and SOA
 - ▶ Deploy
 - ▶ Manage/Monitor
- Summary, Discussion and Challenges,
and **“A Grand Challenge!”**



Service Components





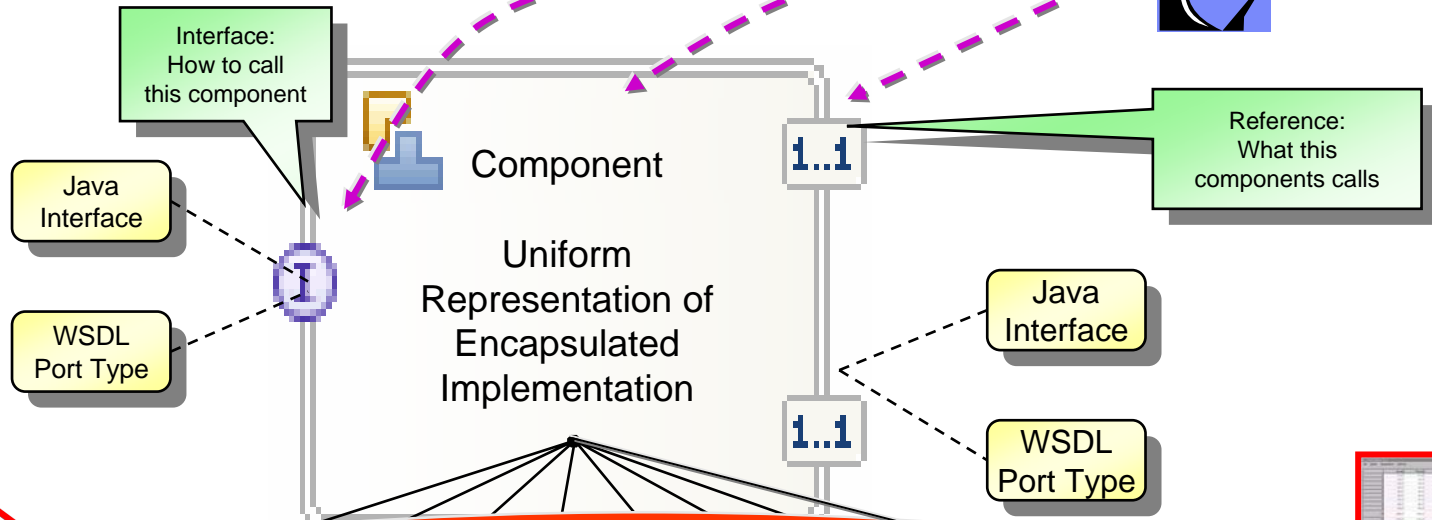
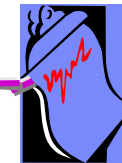
Service Components

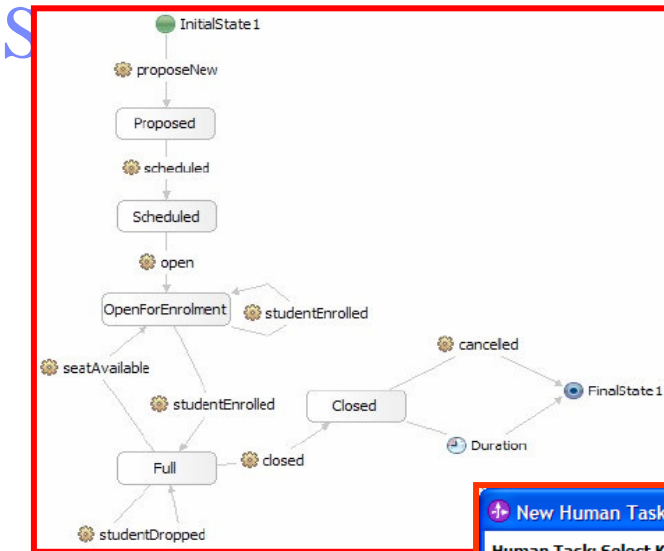
“Web services describes the outsides. How do you implement a service?”

How do you compose services?”

- Encapsulate Components for Reuse; All look the same from outside
- Components may be wired together and aggregated via flow
- Business Objects are the data flowing on wires between Components
- *Enable type, role and skills specific tools.*

Policy





sides

use;

ther a

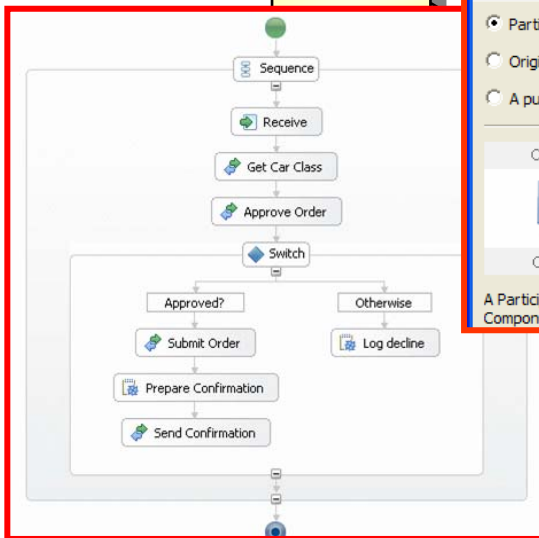
owing

specific

Table Widget Test

	January	February	March	April	May	June	
1	r.1 c.1	r.1 c.2	r.1 c.3	r.1 c.4	r.1 c.5	r.1 c.6	1
2	r.2 c.1	r.2 c.2	r.2 c.3	r.2 c.4	r.2 c.5	r.2 c.6	2
3	r.3 c.1	r.3 c.2	r.3 c.3	r.3 c.4	r.3 c.5	r.3 c.6	3
4	r.4 c.1	r.4 c.2	r.4 c.3	r.4 c.4	r.4 c.5	r.4 c.6	4
5	r.5 c.1	r.5 c.2	r.5 c.3	r.5 c.4	Spanning Item		5
6	r.6 c.1	r.6 c.2	r.6 c.3	r.6 c.4	r.6 c.5	r.6 c.6	6
7	r.7 c.1	r.7 c.2	r.7 c.3	r.7 c.4	r.7 c.5	r.7 c.6	7
8	r.8 c.1	r.8 c.2	r.8 c.3	r.8 c.4	r.8 c.5	r.8 c.6	8
9	r.9 c.1	r.9 c.2	r.9 c.3	r.9 c.4	r.9 c.5	r.9 c.6	9
10	r.10 c.1	r.10 c.2	r.10 c.3	r.10 c.4	r.10 c.5	r.10 c.6	10
11	r.11 c.1	r.11 c.2	r.11 c.3	r.11 c.4	r.11 c.5	r.11 c.6	11
12	r.12 c.1	r.12 c.2	r.12 c.3	r.12 c.4	r.12 c.5	r.12 c.6	12
13	r.13 c.1	r.13 c.2	r.13 c.3	r.13 c.4	r.13 c.5	r.13 c.6	13
14	r.14 c.1	r.14 c.2	r.14 c.3	r.14 c.4	r.14 c.5	r.14 c.6	14
15	r.15 c.1	r.15 c.2	r.15 c.3	r.15 c.4	r.15 c.5	r.15 c.6	15

Java



New Human Task

Human Task: Select Kind

Human Task exist in various kinds. Select the appropriate kind for your task template.

☒ Participating Task
☐ Originating Task
☐ A pure Human Task

Originator Receiver

Computer Interface Human

A Participating Task is a Human Interaction that is invoked from another Service Component, i.e. a BPEL Business Process.

Rules

Rule2

Template: Template 1

Presentation: If the assets are greater than [2000000] and the liabilities are less than [500000], then the rating is [A]

Rule1

Template: Template 1

Presentation: If the assets are greater than [2000000] and the liabilities are less than [1000000], then the rating is [B]

Templates

Template 1

Presentation: If the assets are greater than [0] and the liabilities are less than [1], then the rating is [2].

Index	Name	Type	Constraint
{0}	var1	double	None
{1}	var2	double	None
{2}	var3	string	None

If: all of the following are true

- info.assets > var1
- info.liabilities < var2

Then: rating = var3

Business Rule

Human Task

Interface Map

Decision Table

info.assets	> 500000	<= 500000	
info.liabilities	< 10000	>= 10000	< 10000 >= 10000
rating	"A"	"B"	"C" "D"



A Simple Example and Some Concepts

Something a DB dude recognizes

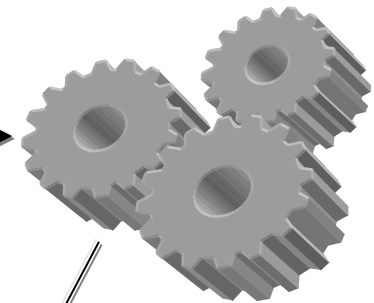


Author Tools
or
Text Editor

```

/*
Pragma This;
Pragma That;
*/
SELECT      ticker,value, activity
FROM        StockQuotes
WHERE       ticker == QuoteRequest.ticker
INTO        quoteResponse.ticker
            quoteResponse.value,
            quoteResponse.sharesTraded;
  
```

Deployment Tools



Deployment Package

SDOs

```

public class
String
Date
}
  
```

```

Public class
String
float
float
}
  
```

```

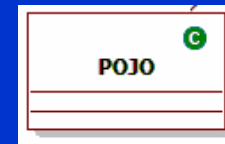
QuoteRequest {
  ticker;
  when;
}
  
```

```

quoteResponse {
  ticker;
  value;
  sharesTraded;
}
  
```



Interface(s)



Data Svc.

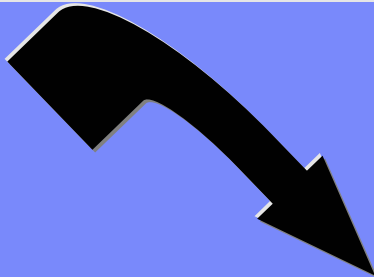


Generated Code
Interpreted Metadata



Assembling Services – Modules and Subsystems

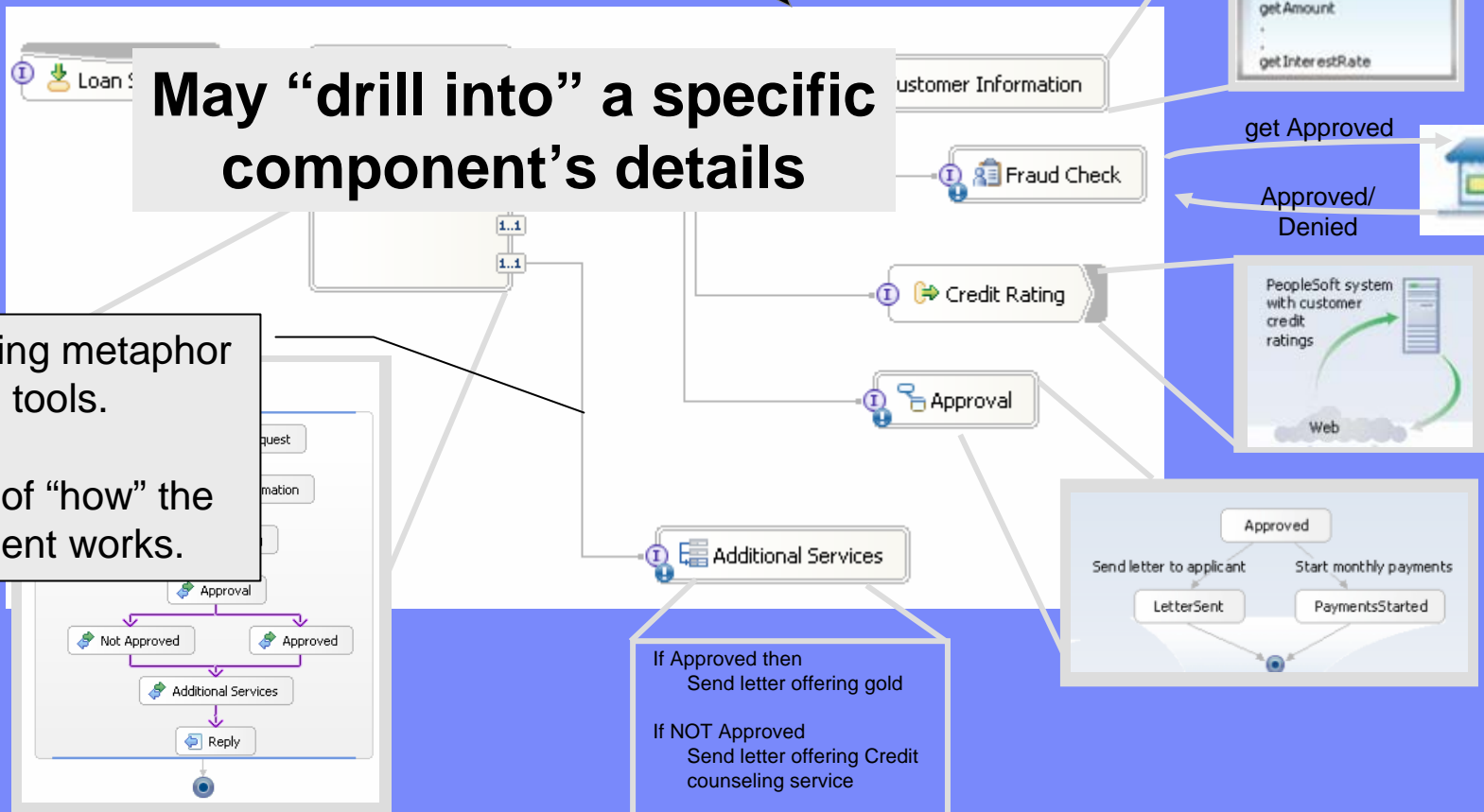
UDDI, Service Registry
Project, SCCS
ad hoc exchange,
... ..



May “drill into” a specific component’s details

Simple wiring metaphor
and tools.

Unaware of “how” the
component works.

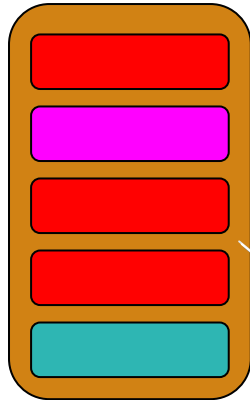




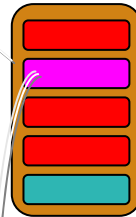
Services Oriented Architecture and Mediations

Credit Card Service

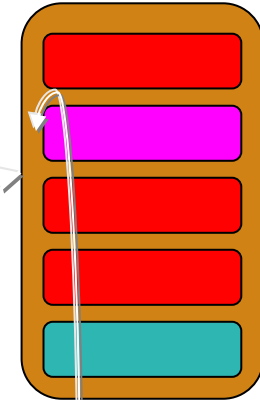
Commerce Service



Needs a CC
Service



Found a CC Service



But it's not quite right. Darn

Transform
Route

Augment

Side Effect

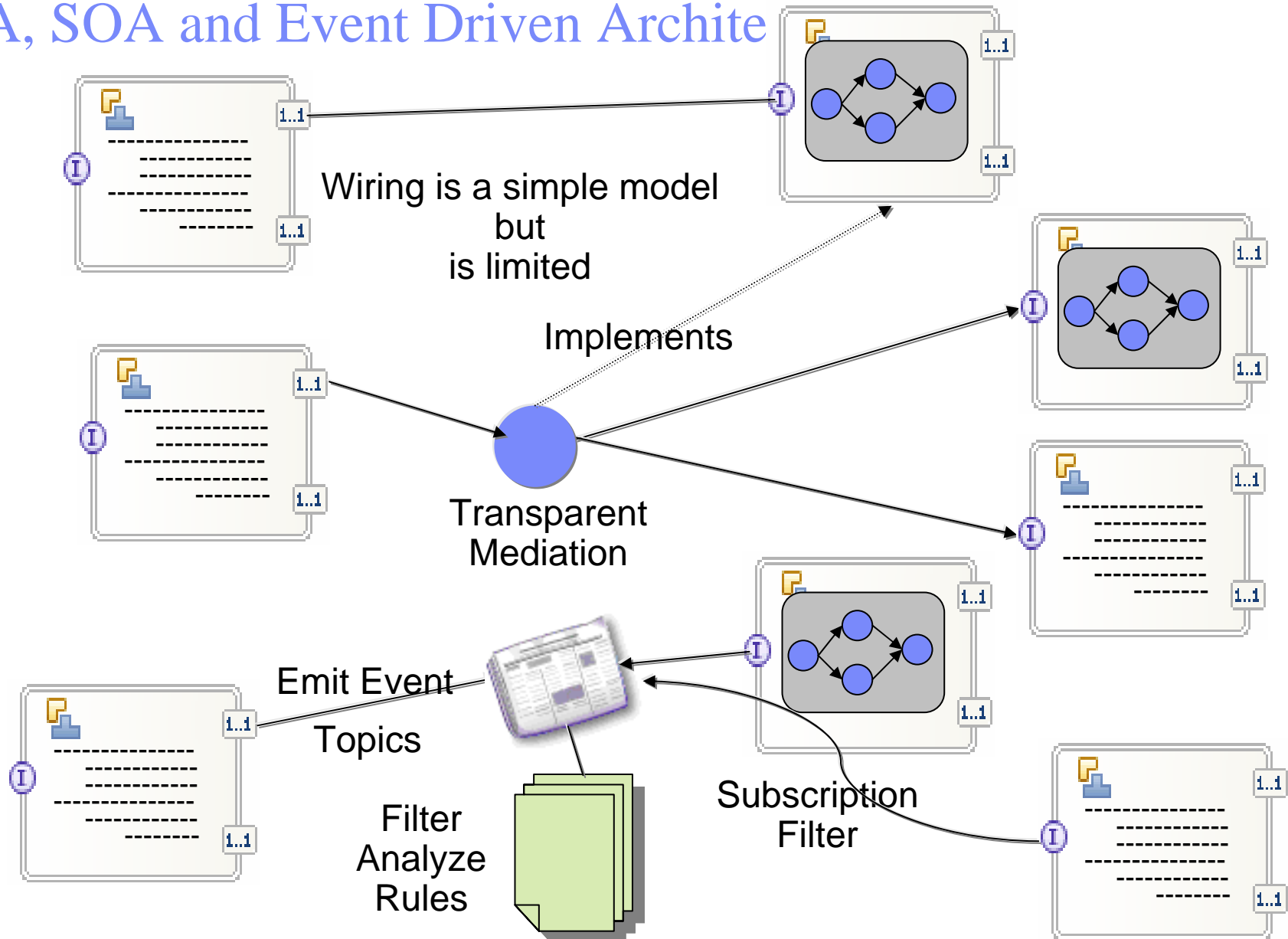
Mediation

Policy Selection
Matching



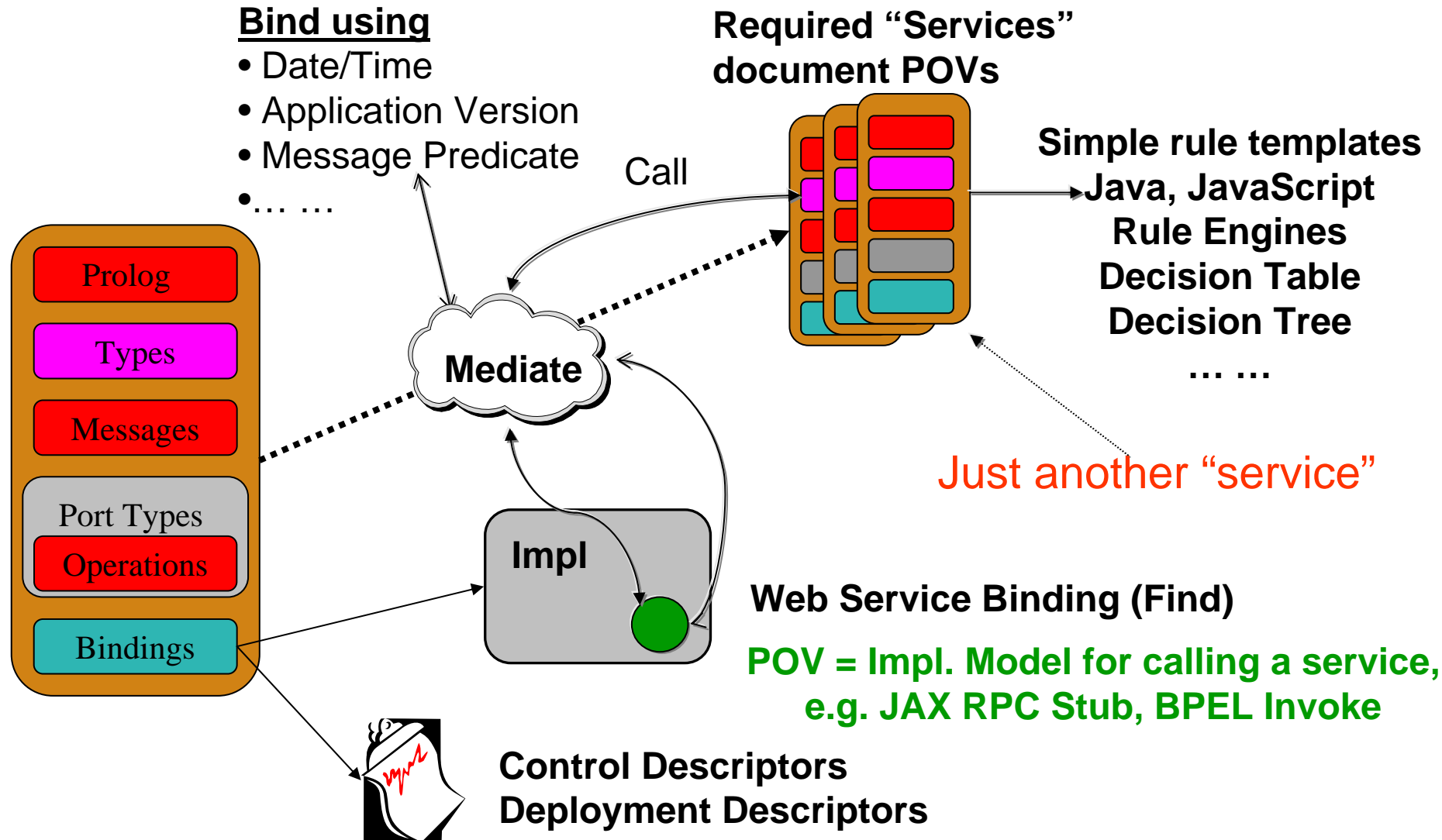


SCA, SOA and Event Driven Archite





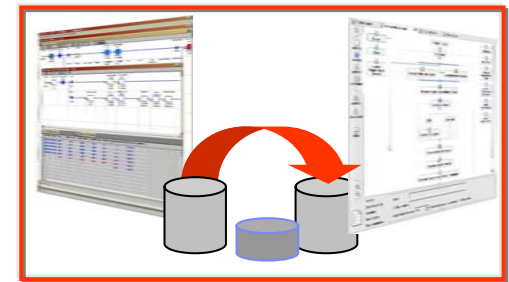
Customizing Services – A “Design Pattern”





Services and Components

- There is a set of standards (emerging) for formally representing a service component's behavior
 - ▶ WSDL, XSD
 - ▶ BPEL, UML
 - ▶ WS-Policy
 - ▶ WS-ResourceFramework
- There is an emerging, extensible set of service component kinds that provide a natural mapping for model elements
 - ▶ Process, Business State Machine, Selector
 - ▶ *Eliminates the “miracle happens here” model*
 - ▶ High level, portable implementations emerging
- Support for very dynamic
 - ▶ Structural composition
 - ▶ Behavioral composition
 - ▶ Configuration/customization
- Bridge to existing skill sets.



This “BPM” solution is a set of documents enabling a “wiki like” BPM evolution.



Some Perspective

- Haven't we heard this before? OO, RPC, MDP,?
 - ▶ This time we really mean it.
 - ▶ Can't you take a joke?
- There are some differences
 - ▶ XML is language neutral;
previous approaches *implied* a language model.
 - ▶ WSDL and XML are more forgiving of changes
 - ▶ Supports RPC and message/document approaches
from beginning
 - ▶ Common type model for applications/servers, message systems
and DBs
 - ▶ Builds on Internet protocols already deployed for
“Web browsing.”
 - ▶ Uniform model for events/pub-sub, message routing and RPC
 - ▶ More focus on logical behavior – WSDL, Policy, etc.
 - ▶ ***SOA component model derives from business modeling, making MDD simpler and eliminating “spooky transformations!”***



Agenda

- Introduction
 - ▶ Disclaimer
 - ▶ A sample business problem
 - ▶ Context and major trends
- Business Process/Performance Management
 - ▶ Model, and why more than “process.”
 - ▶ Assemble and Customize; EDA and SOA
 - ▶ Deploy
 - ▶ Manage/Monitor
- Summary, Discussion and Challenges,
and **“A Grand Challenge!”**



A Patch Management – Implementing the Process

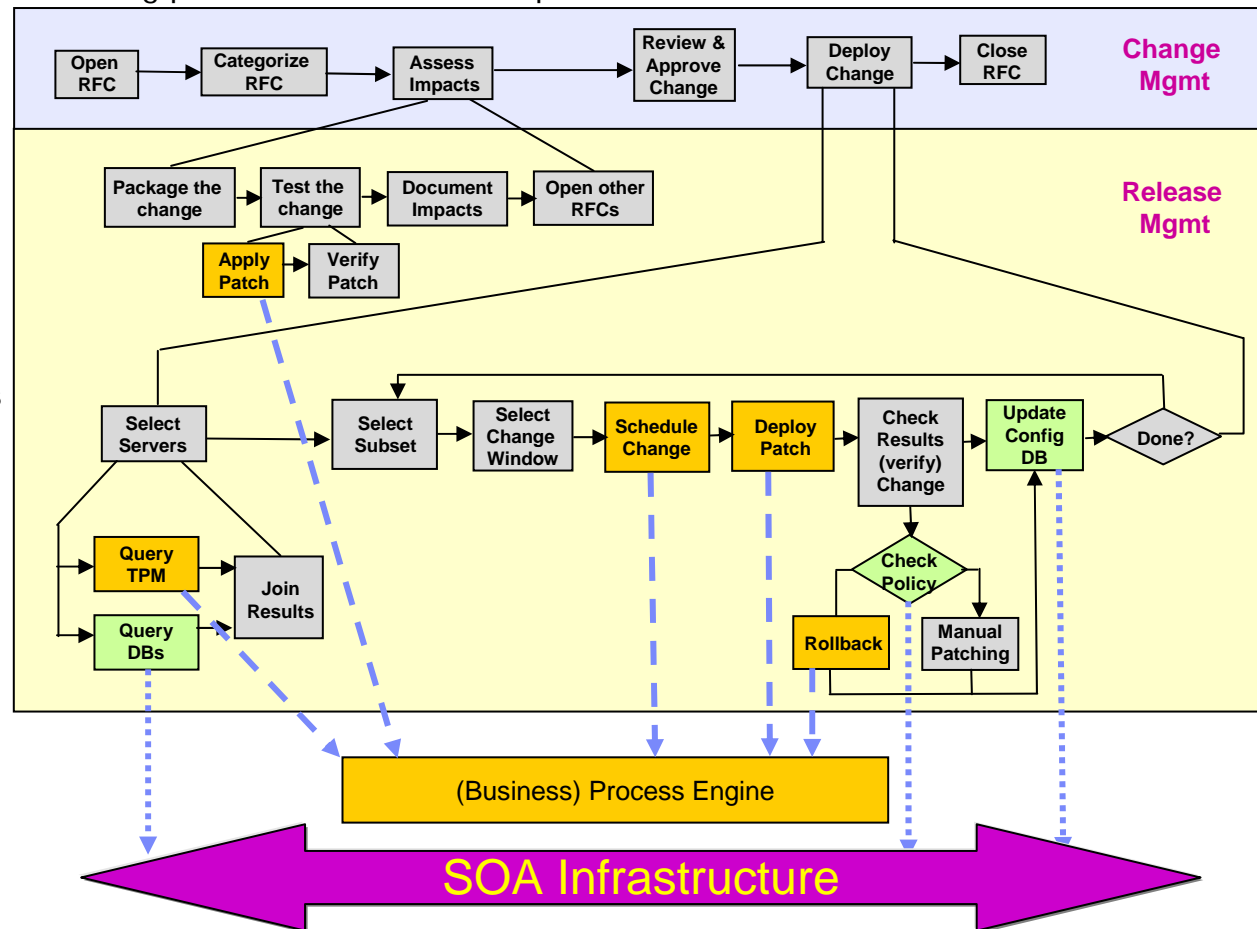
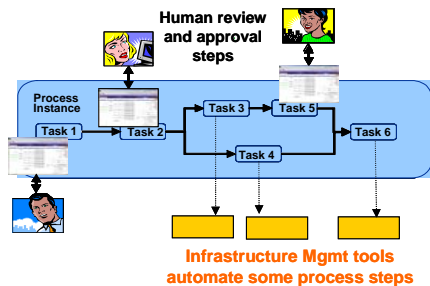
Questions:

- How best to implement the process and make it operational?
- What tools should be used for sequencing work between people?
- What tools for automating particular activities in the process?

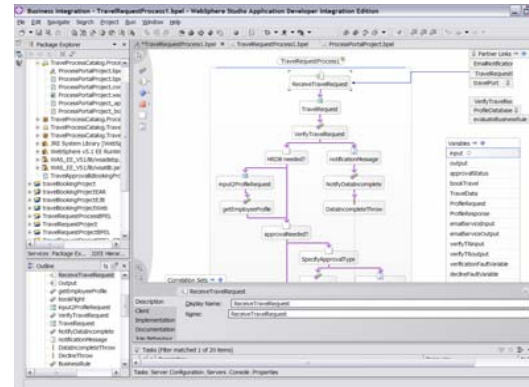
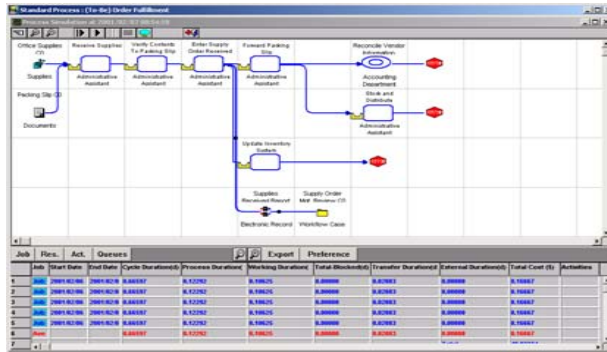


Need to have:

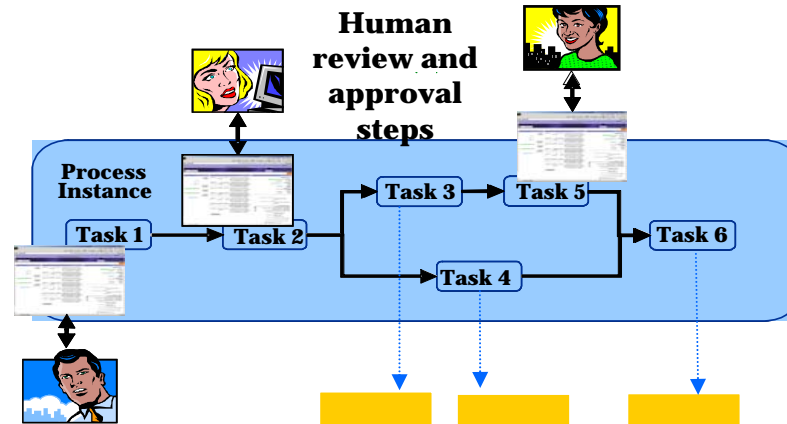
- Logical Process implemented in executable workflows
- User Interfaces for process steps
- Adapters to Management Apps
- Documentation to customize steps



- Model
- Assemble



- Deploy

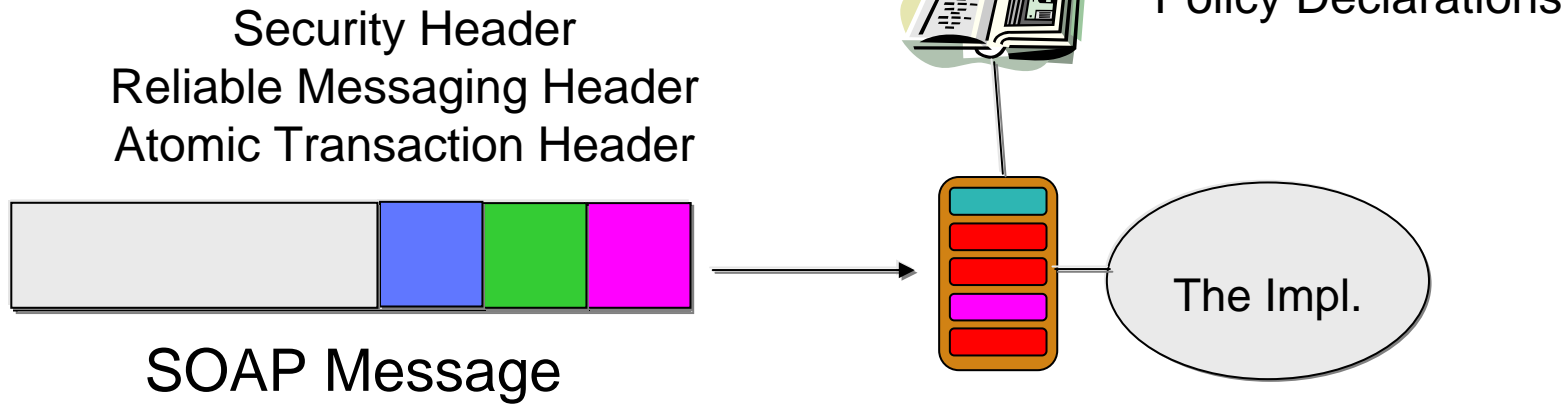


Infrastructure Mgmt tools automate process steps

- Provide portal console and run process using Process Choreographer (WPC)



The Role of the Container

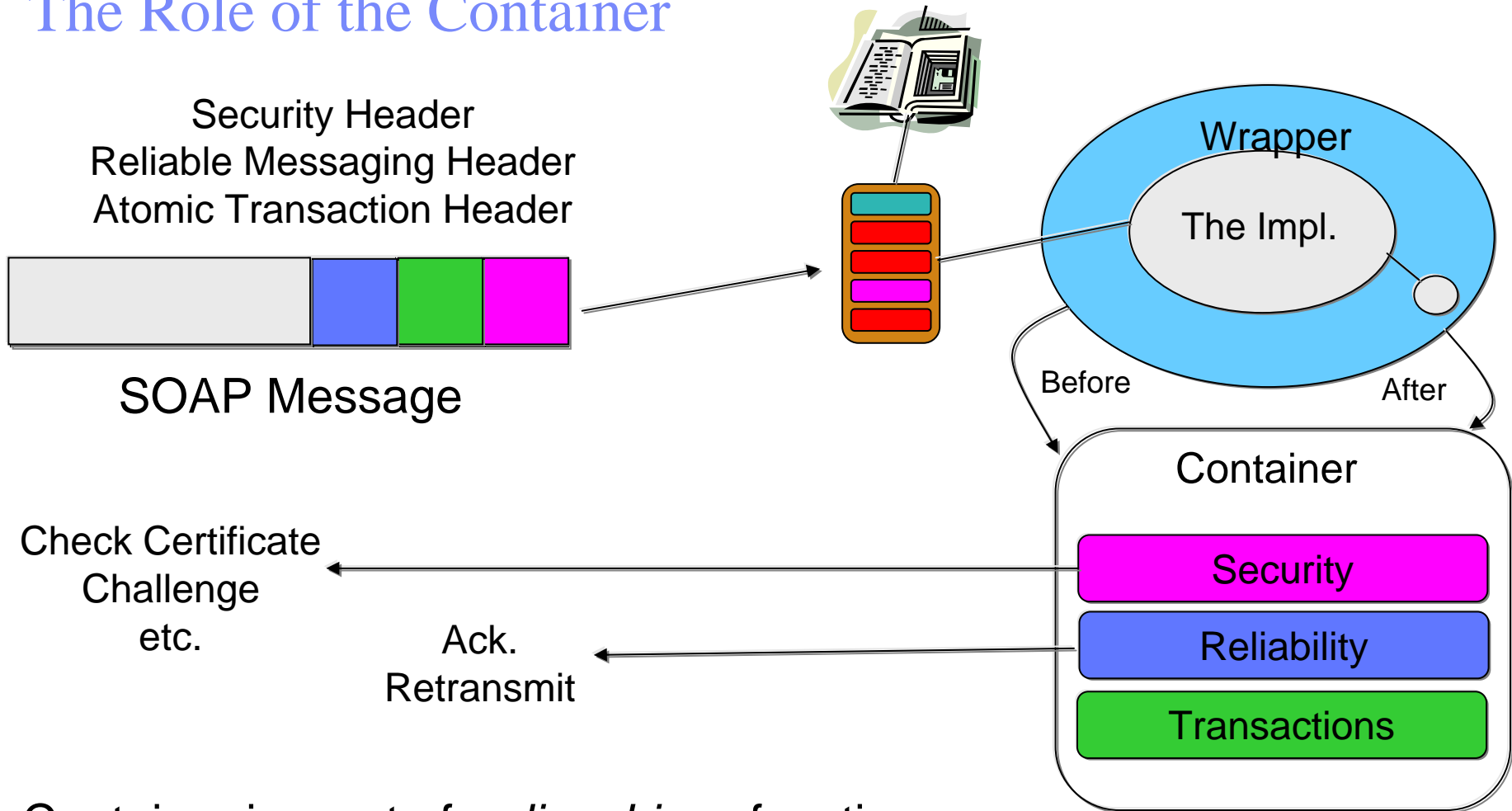


```
double    deposit(Message m) {  
    checkForDuplicate(m.seqNo);  
    registerForTransaction(m.context);  
    isCAValid(m);  
    checkSignature(m);  
    updatePerformanceInfo();  
  
    balance += m.amount;  
  
    // ... ..  
    updatePerformanceInfo();  
}
```

This is fragile,
changes over time,
complex for business programmers,
error prone,
etc.



The Role of the Container

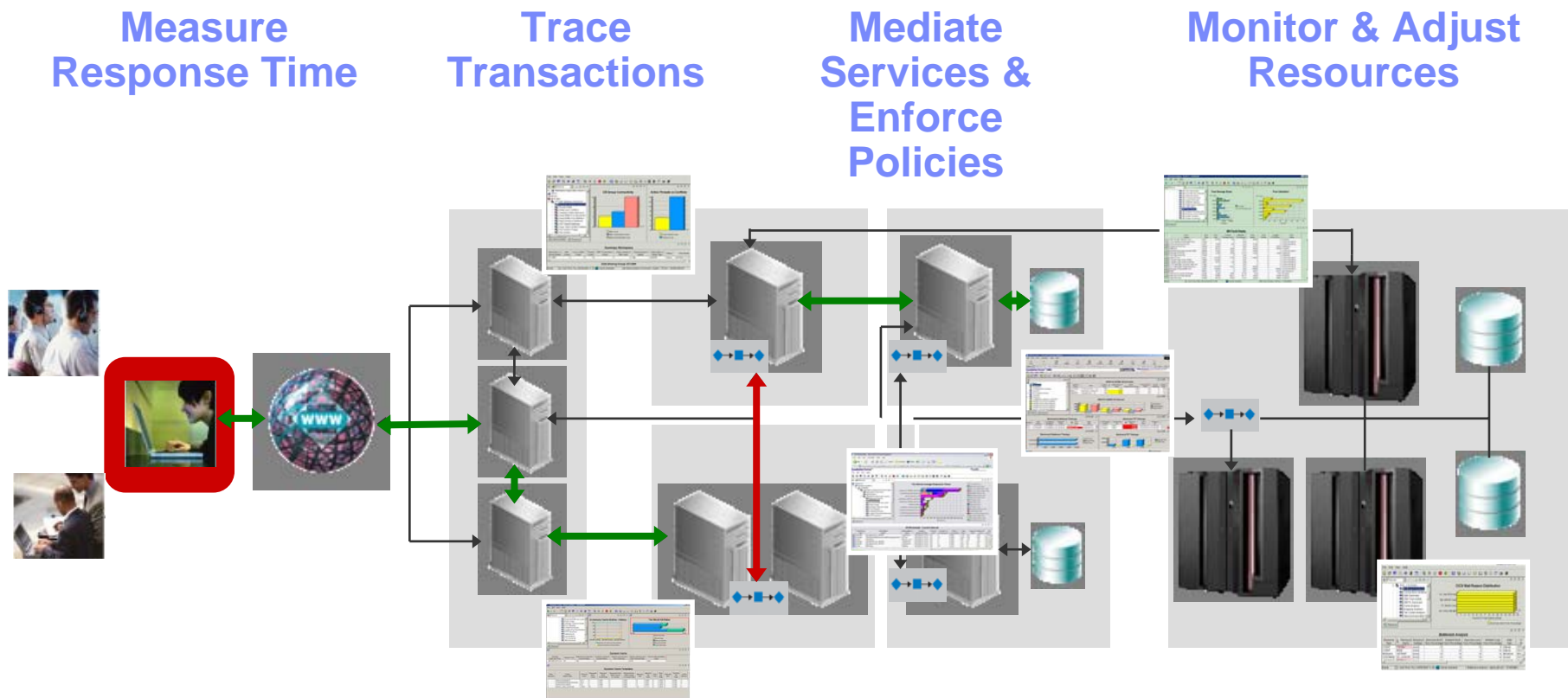


Container is a set of *policy driven* functions.
Interceptor pattern for business logic and "stubs."
Before and After factoring of code.



4 Principles Of Application Management Confidence

A repeatable approach to sense and respond to performance problems within the composite application infrastructure.





A Typical Management/Monitor Portal

Web Services Profiling - Eclipse Platform

File Edit Navigate Search Project Run Window Help

Service Topology Navigator

Topology View:
aggregate interactions among services.

Statistics View:
A table view of the raw data collected by the monitoring agent at each interception point

Sequence Diagram:
Shows exact sequence of messages over time

Content View:
Shows content of a SOAP message

Transaction Flows Flow Patterns

Message Content

Message Name	Requester Service	Requester Machine	Provider Service	Provider Machine	Send Time
lookupCustomerInNewDB2...	Customer	LOCALSERVER	NewDB2	LOCALSERVER	07:41:08.882 C...
lookupCustomerInNewDB2...	Customer	LOCALSERVER	NewDB2	LOCALSERVER	07:41:08.890 C...
lookupCustomerInLegacyS...	Customer	LOCALSERVER	Legacy	LOCALSERVER	07:41:08.896 C...
lookupCustomerInLegacyS...	Customer	LOCALSERVER	Legacy	LOCALSERVER	07:41:08.904 C...
getEnterpriseCustomerReq...	Customer	LOCALSERVER	EnterpriseCustomer	CENTRALSERVER	07:41:08.910 C...
getEnterpriseCustomerRes...	Customer	LOCALSERVER	EnterpriseCustomer	CENTRALSERVER	07:41:09.038 C...
getFromEnterpriseCustome...	EnterpriseCustomer	CENTRALSERVER	EnterpriseCustomerDB	CENTRALSERVER	07:41:09.020 C...
getFromEnterpriseCustome...	EnterpriseCustomer	CENTRALSERVER	EnterpriseCustomerDB	CENTRALSERVER	07:41:09.029 C...
lookupCustomerRequest	Client	LOCALCLIENT	Customer	LOCALSERVER	07:41:09.111 C...
lookupCustomerResponse	Client	LOCALCLIENT	Customer	LOCALSERVER	07:41:09.825 C...
lookupCustomerInNewDB2...	Customer	LOCALSERVER	NewDB2	LOCALSERVER	07:41:09.789 C...
lookupCustomerInNewDB2...	Customer	LOCALSERVER	NewDB2	LOCALSERVER	07:41:09.798 C...
lookupCustomerInLegacyS...	Customer	LOCALSERVER	Legacy	LOCALSERVER	07:41:09.806 C...
lookupCustomerInLegacyS...	Customer	LOCALSERVER	Legacy	LOCALSERVER	07:41:09.814 C...
lookupCustomerRequest	Client	LOCALCLIENT	Customer	LOCALSERVER	07:41:09.877 C...
lookupCustomerResponse	Client	LOCALCLIENT	Customer	LOCALSERVER	07:41:10.003 C...
lookupCustomerInNewDB2...	Customer	LOCALSERVER	NewDB2	LOCALSERVER	07:41:09.984 C...

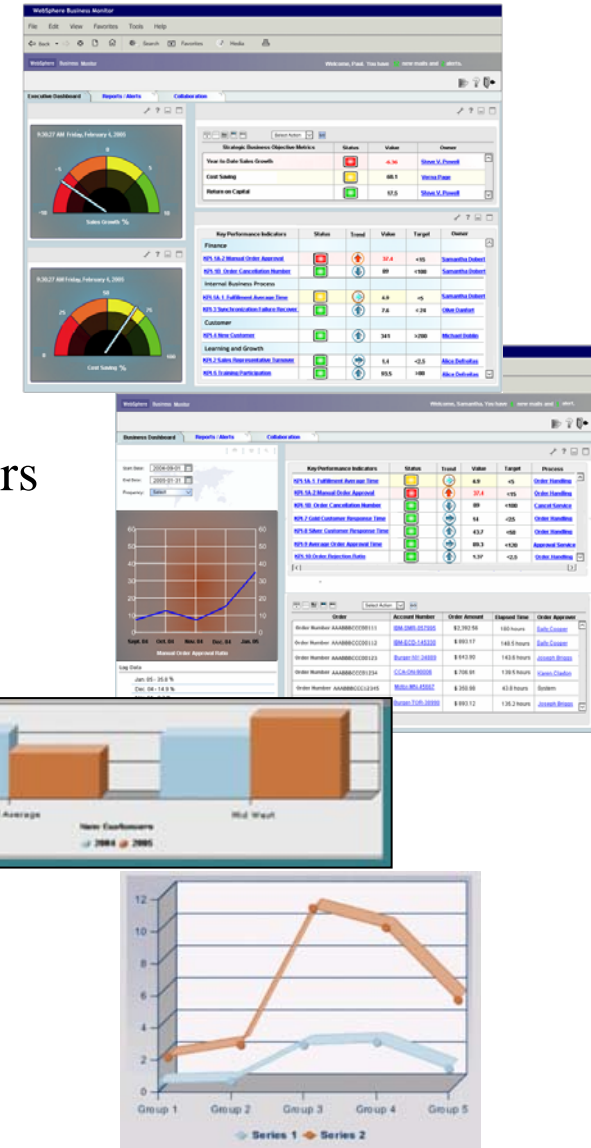
Message Content

Structure	Value
xmlns:soapenc	http://schemas.xmlsoap.org/soap/encoding/
xmlns:soapenv	http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsd	http://www.w3.org/2001/XMLSchema
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
soapenv:Body	
getEnterpriseCustomer...	
getEnterpriseCusto...	http://ec.retail.samples.wsm.ibm.com
address	
xmlns	http://valueobjects.retail.samples.wsm.ibm.com
city	Johnson City
country	USA
postalCode	11111
state	NY
streetName	Johns Street
streetNumber	111
unit	Apt
unitDescription	1A
dialable	1



Business Process/Performance Manage/Monitor

- SOA infrastructure monitoring
 - ▶ Dynamic and policy driven
 - ▶ “Transparent” to application logic
 - ▶ Sense – Respond – Act via MUWS
- Observation/Performance Modeling
 - ▶ Observation points are part of modeling
 - ▶ Define scorecard view of Key Performance Indicators
- SOA event infrastructure and event database
- Enables
 - ▶ Dashboard, monitoring, management.
 - ▶ Ability to intervene in deployed processes
 - Set situational triggers and notifications
 - Dynamically respond to these alerts
 - ▶ Supporting continuous process improvement
 - Monitor in-flight business processes
 - Make process modifications based upon real-time data sent back to the Modeler for simulations





Deploy, Manage and Monitor

- System and application management is *a business process*
 - ▶ Complex, multi-step process with compensation for change management
 - Patches, OS/middleware upgrades, application enhancements
 - Automated steps, manual approval
 - Compensation, recovery, retry
 - ▶ Common Management (Information) Database
 - ▶ Events, Sense/Respond, Monitor, Act, History, reports
- Using “standard” BPM technology
 - ▶ Enables existing skills (e.g. no arcane SM language and APIs)
 - ▶ First class tool support
- **The line between systems/application management and BPM is an illusion**
 - ▶ Why are PO submissions failing? IT error or process design?
 - ▶ Impact analysis
 - ▶ Think about “customer on-boarding.”
 - Some updates to CRM, account system, etc.
 - Also calls to security for UID, ACLs,





Agenda

- Introduction
 - ▶ Disclaimer
 - ▶ A sample business problem
 - ▶ Context and major trends
- Business Process/Performance Management
 - ▶ Model, and why more than “process.”
 - ▶ Assemble and Customize; EDA and SOA
 - ▶ Deploy
 - ▶ Manage/Monitor
- Summary, Discussion and Challenges,
and **“A Grand Challenge!”**



Summary and Directions

- Business Process/Performance Modeling/Management
 - ▶ Is a broad space with many “sub-models.”
 - ▶ We are only beginning to
 - Bring them into a coherent whole
 - Without overwhelming business professionals and IT professionals
 - ▶ We must drive standards and convergence. Standards are broader than runtime formats and protocols.
 - Tool federation
 - Flexible function placement
 - Evolution
- There are many open areas
 - ▶ Governance
 - ▶ Web 2.0
 - ▶ Explosion of casual programmers in the workplace
 - ▶ Modeling support for *Recipes, Patterns and Templates*
 - ▶ Service/component identification, factoring, good “size,”



A Grand Challenge

- There is good and improving models for Process, Information and State,
 - ▶ Events
 - ▶ Components
 - ▶ Use cases
 - ▶ Component collaboration
 - ▶
- Policy is where we are the most “broken”
 - ▶ Examples
 - “All POs over \$10,000 must be approved by regional sales manager.”
 - “An employee cannot close a customer complaint that he created or marked complete.”
 - “Business class is authorized for flight more than 8 hours or overnight.”
 - ▶ We typically write these down in text.
 - ▶ Programmers read the text and write code.
- Can we do better?
 - ▶ Many domains have nascent business vocabularies, e.g. law, dentistry
 - ▶ We write specs with nascent “grammars,” e.g. MUST, CANNOT, RECOMMEND
 - ▶ Documents are often simple combinations of policy
 - Nested lists (Decision Trees)
 - Tables/Forms (Decision Tables)
 - Hyperlinks (Decision Flows)
 - ▶ Can we improve hand-offs and traceability for rules and codes through structured language, business vocabularies





Some Clarification

- An *Asset* is, well an Asset. Can be anything
 - ▶ Word document, Powerpoint Presentation
 - ▶ Handy code that I keep lying around
 - ▶ Excel spreadsheet for costing a project
 - ▶
- A *Pattern* is a recurring solution for solving a problem
 - ▶ Patterns for eBusiness (<http://www-106.ibm.com/developerworks/patterns/>)
 - ▶ Enterprise Integration Patterns (<http://www.eaipatterns.com/>)
 - ▶ J2EE Patterns (<http://community.jboss.org/en/j2eepatterns>)
 - ▶

Read the book and start typing
- A *Template* is a Pattern (or subset of a Pattern) that
 - ▶ Has associated metadata
 - ▶ Comes with a design time control
 - ▶ Uses code generation or “data driven” to convert to an instance.
- A *Recipe* is an directed graph of Templates
 - ▶ Which arcs to follow
 - ▶ Metadata flows through the graph
 - ▶ Subsets, augments, modifies the metadata
- A *Solution Template* is
 - ▶ A complete solution, with install images
 - ▶ Well-defined POVs for tailoring the elements and wizards

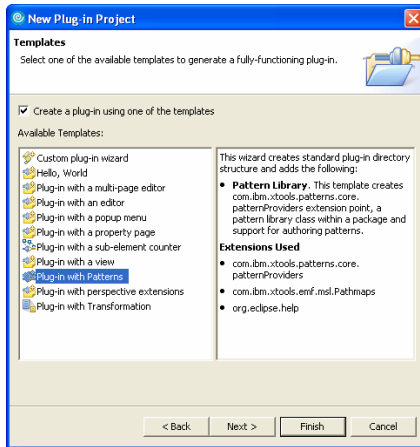
We should be more helpful than this

This is my terminology.
We are trying to come to a simple, common terminology.

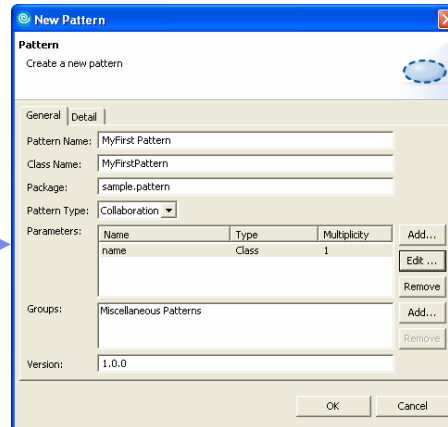


Pattern Authoring in Rational Software Architect

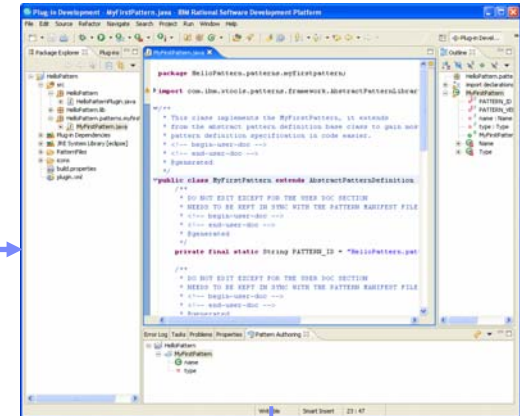
1. Create Plugin Project with Patterns Templates



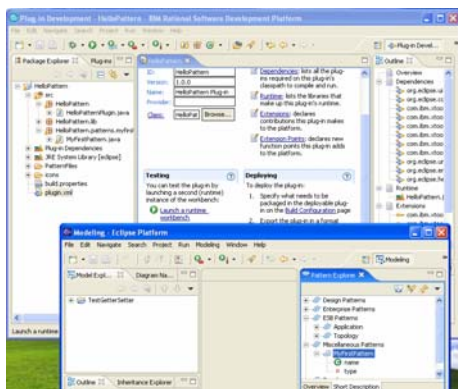
2. Create the skeleton of the pattern. Define parameters for the pattern and constraints on the parameters



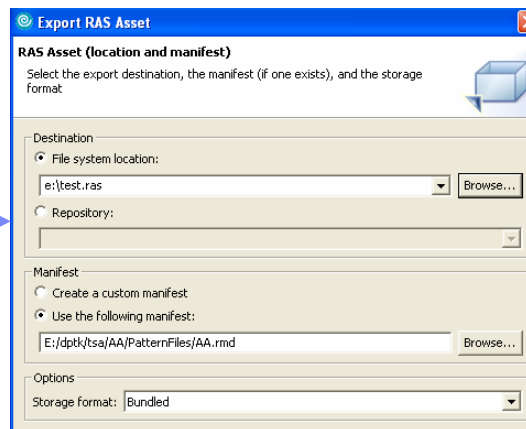
3. Add your custom code using the RSA Pattern Framework. It provides many extension points



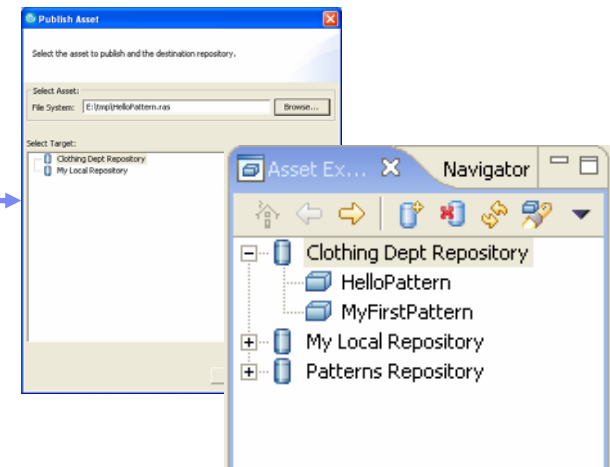
4. Test pattern by launching a new runtime workbench



5. Export the pattern as a RAS asset. Classify your pattern to facilitate searching



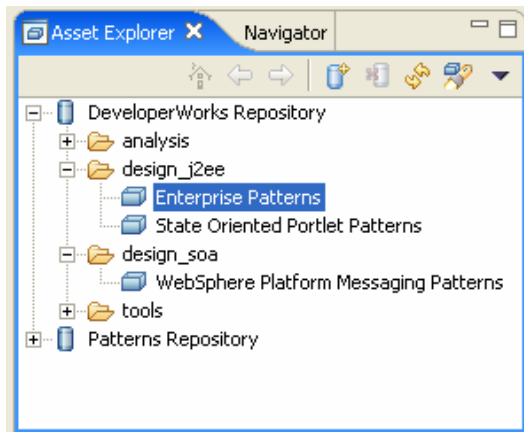
6. Publish Asset to a RAS repository



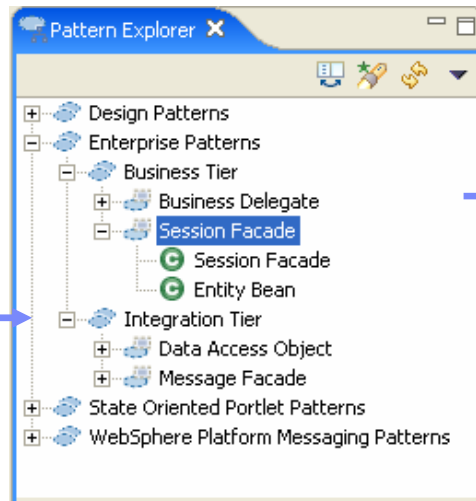


Example of applying a Pattern

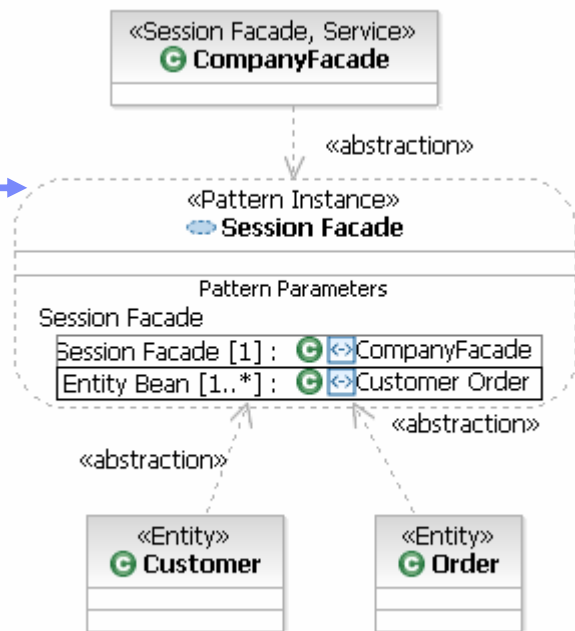
1. Locate and Import the Asset from a RAS repository



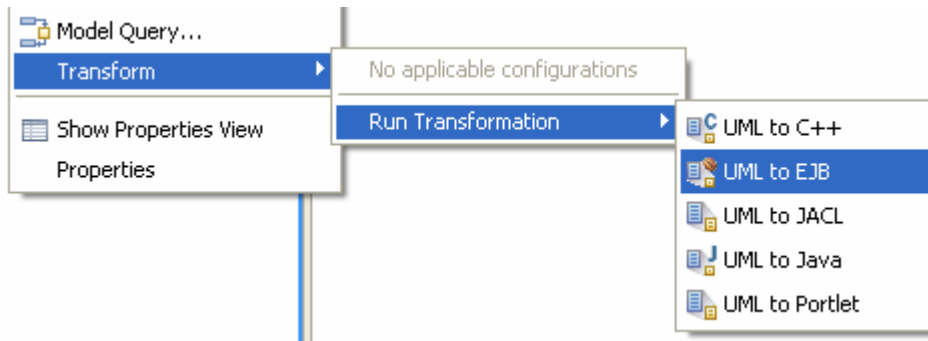
2. Select the Pattern from Pattern Explorer



3. Specify Pattern Parameters



4. Run a Transformation to convert models into Artifacts (code, scripts, docs)





Some Principles

- Minimize concepts and rely on *patterns*
 - ▶ Pub-sub is a SOA pattern
 - ▶ Mediation is a *style* of service
 - ▶ Business rules are *a way* to implement services
- There are very few *new things*
 - ▶ Focus less on code reuse and more on pattern reuse
 - ▶ There is no difference between pattern, code, developer tool
 - ▶ Basic building blocks (e.g. SCA) enable flexible patterns
- Benefits
 - ▶ Productivity
 - ▶ ***Reduce risk, more predictable projects, technical community, ...***
- There are two tiers of programmer
 - ▶ Architecture/Pattern/Template provider
 - ▶ Template/Pattern user
 - Instantiate
 - Compose
 - Configure
- Code for customization





To Do

- Governance
- ITSM
- Haven't we heard this before?
- Web 2.0/ESRI
- Service Identification and Factoring
- Component Business Model and abstract models
- Everyone is a “casual programmer”
- Patterns