Achieving Business Agility in your SOA Initiatives using IBM BRMS

Jerome Boyer
STSM ISSW BRMS-BPM
Agenda

- Challenges
  - Reference Models
  - IBM BRMS
  - Decision Service
  - ABRD Methodology
  - Demonstration
  - Conclusion
## Current Business Drivers

### Efficiency
- **Eliminate Manual Data Entry**
  - Reduction in time from 9 hours to 10 minutes.
- **Reduce Process Cycle Time**
  - Cut processing timing for compensation approval from 33 days to 7 days.
- **Reduce Manual Work**
  - Elimination of 80% of the manual routing of disputed invoices.
- **Support Business Growth**
  - 250% productivity improvement by existing staff.

### Effectiveness
- **Handle Exceptions Better**
  - Rescuing distressed shipments yielded $2M per quarter in saved revenues.
- **Make Better Decisions**
  - Better process controls saved $3M in incorrect write-offs.
- **Consistent Execution**
  - Consistent, proactive communication raised customer satisfaction to 92%.

### Agility
- **Faster Regulatory Compliance**
  - Compliance with new customs requirements within 90 days.
- **Support New Business Models**
  - Ability to change shipping partners in core process to eliminate vendor lock-in.
Business and IT Alignment

- **Pure SOA Play**
  - Lack of disciplined approach to process definition and optimization
  - Lack of context for business operational excellence and for managing business operational risk
  - Lack of explicit metrics for the business value of service reuse
  - Lack of capability to adapt the service behavior to support changing decisions

- **For pure BPM Play**
  - Lack of a disciplined approach to creating and managing an agile library of well-architected and reusable building blocks (including all of services, processes and information assets)
  - Lack of governance and lack of explicit contracts between business and IT participants in an end-to-end process
  - Lack of context for optimizing investment across business and IT
  - Lack of clear separation of concern between process and rules
Agenda

- Challenges

Reference Models

- IBM BRMS
- Decision Service
- ABRD Methodology
- Demonstration
- Conclusion
SOA Reference Architecture

**Business Innovation & Optimization Services**
Facilitates better decision-making with real-time business information

**Interaction Services**
Enables collaboration between people, process & information

**Process Services**
Orchestrates and automates business processes

**Information Services**
Manages diverse data and content in a unified manner

**Partner Services**
Connect with trading partners

**Business App Services**
Build on a robust, scalable, and secure services environment

**Access Services**
Facilitates interactions with existing information and application assets

**Infrastructure Services**
Optimizes throughput, availability and performance

Integrated environment for design and creation of solution assets

Monitor, manage and secure services, application assets & resources
Enablers in the Language of Business Design

**Business Outcomes**
- Improved customer sat & retention
- Reduced operational costs
- Better and faster opportunity id
- Maximized productivity
- Improved cross-business unit coord
- More successful promotions
- Expanded market share
- Improved customer acquisition
- ... 

**Design Discipline**

**Business Agility**

**Lifecycle Services**
- SOA Foundation

**Enterprise Strategy and Planning**
- Banking
- Insurance
- Health Care
- Telecom
- Retail
- Government
- Others

**Model and Development Services**
- Interaction and Collaboration Services
- Process and Decision Services
- Information and Analytic Services

**Partner Services**
- Business Application Services
- Access Services
- Asset and Registry Services

**Enterprise Service Bus**

**Monitoring and Management Services**
Agenda

- Challenges
- Reference Models
- IBM BRMS
- Decision Service
- ABRD Methodology
- Demonstration
- Conclusion
When to think BRMS

ILOG BRMS is the IBM technology for creating, maintaining and implementing decision services…

- IT Executives are under pressure to respond more quickly to their internal customers
- LOB wants to respond more quickly to changes in the external environment
- Automating or standardizing business decisions across the enterprise or LOB
- Business users need to be more involved as part of the change management process
- Transforming or modernizing legacy applications
Traditional Approach for Managing Change

- Decisions used in operations = “business rules”
  - Derived from organizational policies, procedures and internal/external regulations

- For operational systems, business rules are usually contained within application code, although they can also be in documentation or the knowledge of employees

Where Business Rules Typically Exist

- Applications
- Processes
- People
- Documents

Issues

- Rules are hidden in code or isolated within the organization
- Changes are hard to track and maintain over time
- Rules used by systems have to be programmed and require IT resources
- Duplication and multiple versions of the same rules
- Lack of auditability, traceability
- Decision changes cannot be easily tested or simulated
WebSphere ILOG BRMS Introduction

- **Development**
  - JRules BRMS
    - Rule Studio (Eclipse-based)
  - Manage
    - Decision Validation Services
  - Share
    - Rule Repository
  - Deploy
    - Rule Execution Server for Java
    - Rules for COBOL
    - SOA
  - Line Of Business
  - Enterprise-wide
  - Production

- **Common Usage**
  - Rule Team Server
  - Rule Solutions for Office
  - Rule Repository

- **Rules for .NET BRMS**
  - Rule Studio (Visual Studio-based)
  - Rule Execution Server for .NET
Concept of Operation

Development

Rule Studio (Eclipse)

Rule Repository

Maintenance

Rule Team Server - Decision Validation Services

Operation

Deploy

Code Generation

Rule Execution Server J2EE/J2SE
Rule Execution Server (Web Service)
COBOL
Rule Execution Server .NET

IT Architect
IT Developer
Business Analyst
Process Owner
Business Leader
Business User

Rule Solutions for Office
Agenda

- Challenges
- Reference Models
- IBM BRMS
- Decision Service
- ABRD Methodology
- Demonstration
- Conclusion
BPM 1997 - Workflow

Claim Entry → Coverage Verification → Manage Claim Issues → Adjudication → Payment

Enter Claim
Coverage Policies
List Issues
New Issue
Update Claim
Close Issue

Data

Claim Entry
Coverage Guideline
Enterprise Application

Data
BPM 2005+ -> Decision Service Integration
SOA Reference Architecture - BRMS Component

Business Innovation & Optimization Services

Interaction Services

Process Services

Information Services

Partner Services

Business App Services

Access Services

Rule Team Server

ESB: Facilitates communication between services

Decision Service

Rule Studio

Development Services

IT Service Management

Infrastructure Services

Rule Execution Server

Facilitates communication between services
Decision Services = Business Services

- Part of the business application services
  - owned and defined by Line Of Business
  - Supports business task (Examples: OpenBankAccount(bankAccount), processLoanApplication(application), ...)
  - Works on business entities, and their life cycle.

- Designed using SOA service definition best practices:
  - Loosely coupled,
  - Coarse grained,
  - Minimum payload.
Decision Service = Business Services

- A Decision Service is a service operation within a business service
  - ClaimManagerService
    - `verifyCoverage`
    - `adjudicateClaim`
  - LoanManagerService
    - `assessLoanApplication`
    - `rateApplicant`

- Part of a component:
  - Service operations are defined within larger components (service end-point) responsible to manage the main business entities:
    - Claim Manager component
      - Manages Claim entity
      - Defines operations `loadClaim`, `saveClaim`, `verifyCoverage`, `adjudicateClaim`
    - Loan Application Manager component
      - Manages LoanApplication entity
      - Defines operations `RateApplicant`, `AssessLoanApplication`
Service Specification - Semantic

- Service interface usually exposes more than one operation:
  - Having one interface per business function may lead to a huge number of service interfaces
  - Service proliferation, in turn, results in service governance problems, making it harder to pursue effective code reuse

- Business transactions related to the same data should be mapped to operations within the same interface.

- The definition of an operation can follow different approaches, usually reflected by its signature:
  - Synchronous or asynchronous,
  - Stateless or stateful,
  - Header based or carrying payload,
  - Using faults or not.
Identify Decision Services

- Decision points are points of variability and decision in a business process
  - Process tasks
  - Process branching

- DP helps to drive the rule analysis
- DP helps business analyst team to focus on where the business rule are enforced
Different Models For Different Purposes
Service Characteristic: Loosely Coupled

- Interfaces should be developed with minimal assumptions about both the provider and the consumer.

- Change in one application/module should not force a change in another application/module.

- Coupling factors:
  - **Time:**
    - Availability of one system does not affect the other.
    - Supported by the use of an ESB.
  
  - **Data:**
    - Differences in data models do not prevent integration.
    - Supported by the use of an ETL process or a mediation flow.
  
  - **Implementation:**
    - The service implementation is hidden from the caller
Service Characteristic: Coarse Grained

- The service should provide significant business process usefulness
- It usually includes a composition of finer grain services, in particular data services
- The business rules render significant business decisions that requires complex processing on a large amount of data elements.
  - `VerifyCoverage`, which needs Claim and Policy information
  - `AdjudicateClaim`, needs all the medical invoices, and other documents related to the claim
  - `AssessLoanApplication`, which needs Credit History and Property information
Service Characteristic: Minimum Payload

- Information passed into a service should be reduced to the minimum like key data:
  - Primary key information
  - Data Value Object

- Driving factors:
  - Performance
  - Dependencies between components
  - Reusability
  - Lighter change management

- The responsibility of acquiring the data should be on the service. The service should not ask the consumer to provide this data.
  - This is especially true for any reference data needed by the service.
  - Service consumers should not change because of provider’s requirements
METHODOLOGY
Agenda

- Challenges
- Reference Models
- IBM BRMS
- Decision Service
- ABRD Methodology
- Demonstration
- Conclusion
BP and BR have different cycles
Agile development for Decision Service

Agile Business Rule Development

*Discovery*
- Harvest rules using short workshop sessions

*Analysis*
- Understand and prepare the rule for implementation

*Design*
- Define rule set, BOM and project structure
- Prototype rules

*Authoring*
- Develop unit test and rules

*Validation*
- Functional test involving SME for feedback

*Deployment*
- Use rule execution server staging platform
### Agile Business Rules Development (ABRD)

- An Open-Source methodology to develop Business Rules application
  - Development Process, Roles, Tasks, Architecture, Best Practices
  - Governance

<table>
<thead>
<tr>
<th>Breakdown Element</th>
<th>Steps</th>
<th>Index</th>
<th>Predecessors</th>
<th>Model Info</th>
<th>Type</th>
<th>Planned</th>
<th>Repeatable</th>
<th>Multiple Occurrences</th>
<th>Ongoing</th>
<th>Event Driven</th>
<th>Optional</th>
<th>Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discovery Activities</td>
<td>2</td>
<td></td>
<td></td>
<td>Capability Pattern</td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Review Decision Point Table</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Define roadmap</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Organise workshop</td>
<td>5</td>
<td>4</td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Execute Rule Discovery Workshop</td>
<td>6</td>
<td>5,4</td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis Activities</td>
<td>7</td>
<td></td>
<td></td>
<td>Capability Pattern</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Transform Rules</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Define Rule Schema</td>
<td>9</td>
<td>8</td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Determine Rule implementation</td>
<td>10</td>
<td>9</td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Build rule project structure</td>
<td>11</td>
<td>10</td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Build Object Models</td>
<td>12</td>
<td>11</td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Prototype_rules</td>
<td>13</td>
<td>12</td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Synchronize Data Model</td>
<td>14</td>
<td>11</td>
<td></td>
<td></td>
<td>Task Descriptor</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use tools as soon as possible

Incremental implementation of the rule set

Start by simple decision service: data validation

Time boxed iteration
A good analysis leads to a good implementation

Use refactoring feature to improve object models, rules, and unit tests

Start by tests helps speed up the development – even business rules

Build the rule set by iterations, and integrate it at each build with the core business application

Each iteration brings business value

Ruleset decomposition helps to drive iteration content
  - Rule package
  - Rule flow content
**Goal**: gather business rules for a given decision point

Cut the day in two: discovery workshop – Off line analysis

Just enough documentation

Short analysis focusing on the conceptual data model

Come back to the SME with issues

Stop when you are able to write some rules and organize rule projects

**Rules:**
- The **Claim** should be initiated within 30 working days after the **accident**.
- The **date of loss** should be before the **expiration date** of the **policy** and after the **effective date**
- Do not reimburse **medical expenses incurred abroad** if the claim is presented more than 1 year after the expenses has been incurred or if the claimant spent more than 182 days abroad within the past year
Conceptual Model

- **Claim**
  - claimNumber : String
  - status : String
  - dateOfLoss : Date

- **InsurancePolicy**
  - policyNumber : String
  - effectiveDate : Date
  - expirationDate : Date

- **MedicalInvoice**

- **Location**

- **Claimant**

- **Property**

- **InsuredPerson**

- **MedicalLineItem**

- **MedicalProvider**

- **Treatment**
Different Models For Different Purposes

Managed Server
- Process Engine
  - Process Variables
    - BPD | BPEL
  - Message XSD
  - Canonical Model

DataService
- Beans/SDO
- XOM

DB Server
- BizEntity DB
- RefDataDB

Managed Server
- Rule Execution Server
- Service Contract Centric
- Decision Centric Model

Process – coaches centric
Agenda

- Challenges
- Reference Models
- IBM BRMS
- Decision Service
- ABRD Methodology
- Demonstration
- Conclusion
• Decision Service is embodied by a Rule Project
• Input & Output correspond to Ruleset Parameters
Modeling in JRules

- The goal is to use the product as soon as possible in Rule Discovery process
  - To capture the conceptual model
  - Iterate/refine using refactoring capabilities

- Rules are written using Business Terms
  - Need to capture the vocabulary in the Product
  - 2 ways:
    - Bottom-up: Use a Java Model or XSD => import in Rule Studio
    - Create a BOM using BOM Editor in Rule Studio
      disconnected from any implementation
Capturing Business Vocabulary

Business Rules are written against a Language-specific Business Vocabulary

- Business Vocabulary
- Business Object Model (BOM)
- BOM to XOM mapping
- eXecution Object Model (XOM)

Verbalize
Import / Update
Top-down
Bottom-up
Map
Defining Ruleset input & output

- Signature of the ruleset is different than Decision Service
- Root to access data necessary to the decision
- Output to store decision result
Maintain Business Vocabulary & Rules

- **Browse and learn Vocabulary**
  - Vocabulary tools in editors
  - Find usage

- **Enrich Vocabulary to improve Rules**
  - Good vocabulary makes good rules

- **Maintain Vocabulary**
  - Refactor rules
  - Deprecate terms and phrases
Agenda

- Challenges
- Reference Models
- IBM BRMS
- Decision Service
- ABRD Methodology
- Demonstration

Conclusion
CONCLUSION
### Nice Wedding

<table>
<thead>
<tr>
<th>Business Constraints</th>
<th>SOA (web service)</th>
<th>BPM- BPEL</th>
<th>BRMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agility</td>
<td>Design by service simplify reuse XSD for message definition WSDL for common interface</td>
<td>orchestrate and control the execution Externalize application logic</td>
<td>quickly develop and deploy business rules that reflect new business policies</td>
</tr>
<tr>
<td>Transparency</td>
<td>Business function map Business service map Technical service map</td>
<td>Business process Technical process</td>
<td>Business Rule Execution reporting</td>
</tr>
<tr>
<td>Cost Reduction</td>
<td>Reduce integration</td>
<td>Cost application development</td>
<td>Cost of maintenance</td>
</tr>
<tr>
<td>Reduce IT dependence</td>
<td>Reduce risk Reduce time</td>
<td>Better Communication Process Visibility</td>
<td>Empower user on business logic maintenance</td>
</tr>
<tr>
<td>Flexible Deployment</td>
<td>Autonomous services</td>
<td>Executable business process- BPEL engine</td>
<td>Hot deployment Rule Set BRE engine</td>
</tr>
</tbody>
</table>
Conclusion

- Agile methodology enforces working software / ruleset over documentation.
- Decision service is a business service
- Get values by using the product as early as possible:
  - Better data model design
  - Centered to execution
  - Understand the business logic and rules
  - Ask the good questions
  - Get subject matter experts’ feedbacks earlier and often
- The product support iterative development
Q&A
## Business Agility Factors

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptability</td>
<td>Measure the ability to change the business logic easily. The motivation may be due to short deadline constraint, frequent small changes or important change that may occur every day, week, month or quarter</td>
</tr>
<tr>
<td>Transparency</td>
<td>Represents the need to clearly implement the business logic as what was agreed upon between the business unit and the IT teams</td>
</tr>
<tr>
<td>Auditability</td>
<td>Understand what the logic behind a decision is</td>
</tr>
<tr>
<td>Reusability</td>
<td>Need to share business logic across processes or applications and stay consistent across applications/transactions</td>
</tr>
<tr>
<td>Manageability</td>
<td>Addresses the life cycle management of the business logic. Business users control process and decision logic</td>
</tr>
</tbody>
</table>