**Step 5: Automation (Development) with IBM ODM**

**Introduction**

Automation (Development) is a critical phase in the Business Rules Management System (BRMS) lifecycle, as outlined in the USAA 9-step BRMS overview. This step, labeled as "#5: Automation (Development)" in the diagram, focuses on developing decision logic within IBM Operational Decision Manager (ODM) using Rule Designer, followed by moving the decision to Decision Center for governance, testing, and simulations, with rule execution managed by Decision Server. ODM, an enterprise-grade BRMS, enables organizations to create scalable, maintainable, and efficient automated decision-making systems, ensuring flexibility for future updates through its comprehensive tools.  
  
This phase builds on the foundation laid in previous steps:

* **Step 1**: Opportunity Evaluation
* **Step 2**: Rule Harvesting
* **Step 3**: Decision Modeling
* **Step 4**: Verification & Validation

The automation phase ensures that the decision logic in ODM is ready for production implementation (Step 7) and testing (Step 6).

**Objectives of Automation (Development) with IBM ODM**

1. **Develop Rule-Based Decision Logic in IBM ODM’s Rule Designer**
   * Encode business rules using Implementation Rule Language (IRL) or Business Action Language (BAL).
   * Example: A loan approval rule: *If a customer’s credit score exceeds 700 and their income is above $50,000, approve the loan.*
2. **Ensure Modular, Reusable, and Efficient Rule Implementation**
   * Structure rules as independent, reusable components within rule projects.
   * Optimize performance in Decision Server’s rule engine.
3. **Maintain Separation Between Business Rules and Application Code**
   * Isolate decision logic in rule artifacts (e.g., decision tables, rule flows).
   * Allow business users to modify rules via Decision Center without developer intervention.
4. **Support Easy Updates Without Extensive Coding Changes**
   * Adjust rules using Decision Center’s user-friendly interface.
   * Deploy changes seamlessly via Decision Server.

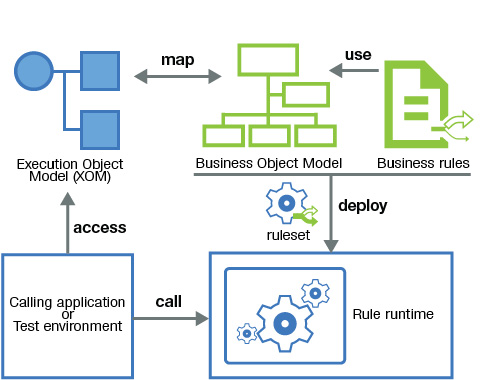
**Steps for Developing a Decision in IBM ODM**

1. **Rule Design and Development in ODM Rule Designer**

Develop the decision logic in ODM’s Rule Designer, creating rule artifacts such as decision tables, rule flows, and rule sets, which will later be moved to Decision Center for testing and simulations. This process involves defining the Execution Object Model (XOM) and Business Object Model (BOM) to structure data for rule development.

* **Step 1: Create a New Rule Project in Rule Designer**
  + Open IBM ODM Rule Designer and create a new rule project (e.g., "LoanApprovalRules"). Define the project structure, including the **Execution Object Model (XOM)** and **Business Object Model (BOM)**, which are critical for structuring data used in rule logic.
  + **Execution Object Model (XOM)**:
* The XOM represents the technical, runtime data model used by the application integrating with ODM. It defines the actual Java or other programming language objects (e.g., POJOs, EJBs) that ODM interacts with during rule execution via Decision Server. For example, in a loan approval system, the XOM might include Java classes like Customer (with fields creditScore, income) and Loan (with fields status, interestRate).
* XOM is typically provided by the application developers and imported into Rule Designer to ensure ODM rules can access and manipulate the application’s data structures during runtime.
* **Practical Insight**: Ensure the XOM aligns with the application’s data model, using Rule Designer’s XOM editor to import or define Java classes, ensuring compatibility with Decision Server execution.
  + **Business Object Model (BOM)**:
* The BOM is a business-friendly abstraction of the XOM, created in Rule Designer to simplify rule authoring for business users and developers. It maps XOM objects to business-oriented concepts, such as transforming Customer (XOM) into Customer (BOM) with attributes like creditScore and income, but presented in a way that hides technical details (e.g., Java class names).
* Use Rule Designer’s BOM editor to create the BOM by mapping XOM objects, defining virtual attributes, and ensuring the model is intuitive for rule logic in decision tables and flows. For example, the BOM for a loan approval might include Customer with creditScore (integer) and income (double), linked to the XOM’s Java Customer class.
* **Practical Insight**: Keep the BOM simple and business-focused, avoiding technical jargon, to facilitate collaboration in Decision Center and rule execution in Decision Server.

Define the project structure, ensuring the XOM and BOM are aligned for rule development, with the BOM serving as the primary interface for rule authoring and the XOM ensuring runtime compatibility with the application via Decision Server.



**Step 2: Design Decision Tables**

Create a decision table in Rule Designer to represent the decision logic, using the BOM for conditions and actions. For example for loan validation  
  
A screenshot of a computer

AI-generated content may be incorrect.

**Step 3: Design Rule Flows**

* Create a rule flow in Rule Designer to define the sequence of rule execution, using BOM objects and linking to decision tables or rule sets, with XOM ensuring runtime compatibility. For example, a loan approval flow:
  1. **Check Customer Eligibility**: Use a decision table or rule set to verify Customer.creditScore and Customer.income from the BOM, mapping to XOM Customer objects.
  2. **Assess Credit Risk**: Evaluate Customer.creditScore and debt-to-income ratio with another rule set, using BOM and XOM mappings.
  3. **Determine Loan Terms**: Set Loan.interestRate and Loan.duration based on risk and eligibility, executed via XOM in Decision Server.
* Use Rule Designer’s flow editor to build this flow, connecting decision tables and rule sets with arrows to indicate sequence, ending with a decision node (e.g., "Approved/Rejected?").  
    
  A diagram of a process flow

  AI-generated content may be incorrect.

**Step 4: Implement Rules in IRL or BAL**

* Code the rules in Rule Designer using IRL or BAL, linked to decision tables and flows, using BOM for authoring and XOM for runtime execution. For example:

A close-up of a computer screen

AI-generated content may be incorrect.

A white background with black text

AI-generated content may be incorrect.  
  
Ensure rules reference BOM attributes (e.g., customer.creditScore), which map to XOM objects (e.g., Customer.creditScore) for execution in Decision Server.

**Step 5: Test Locally in Rule Designer**

* Use Rule Designer’s built-in testing features and Decision Validation Services (DVS) to run unit tests on decision tables, rule flows, and rule sets, ensuring they produce expected outputs (e.g., input {creditScore: 750, income: 60000} → Approved). Test using BOM data, validating XOM mappings for runtime.
* **Tools**: Leverage Rule Designer’s test scenarios and DVS for initial validation, ensuring compatibility with Decision Server execution.

**6. Move the Decision to ODM Decision Center for Testing and Simulations**

After developing the decision in Rule Designer, migrate it to Decision Center for business user collaboration, testing, and simulations, preparing it for execution via Decision Server, ensuring BOM and XOM alignment.

* **Step 1: Deploy Rule Project to Decision Center**
  + From Rule Designer, package the rule project (e.g., "LoanApprovalRules") and deploy it to Decision Center via ODM’s deployment wizard, including XOM and BOM definitions. Ensure the project includes decision tables, rule flows, and mappings.
  + **Tools**: Use Rule Designer’s deployment options to synchronize with Decision Center, creating a branch or baseline for governance.
* **Step 2: Review and Refine in Decision Center**
  + Business users access Decision Center’s web interface to review and refine the decision logic, including decision tables and rule flows, ensuring BOM clarity and XOM compatibility. For example, adjust the loan approval threshold from $50,000 to $60,000 in the decision table, verifying BOM-to-XOM mappings.