



How Business Analysts Represent and Debug Complex Decision Services

Jacob Feldman, PhD
OpenRules Inc., CTO
www.OpenRules.com

June 30, 2022

I have a sneaking suspicion that this is how the world will end: *due to a small typo in one line of code*

For now, I'm more concerned about **debugging** than about superintelligence

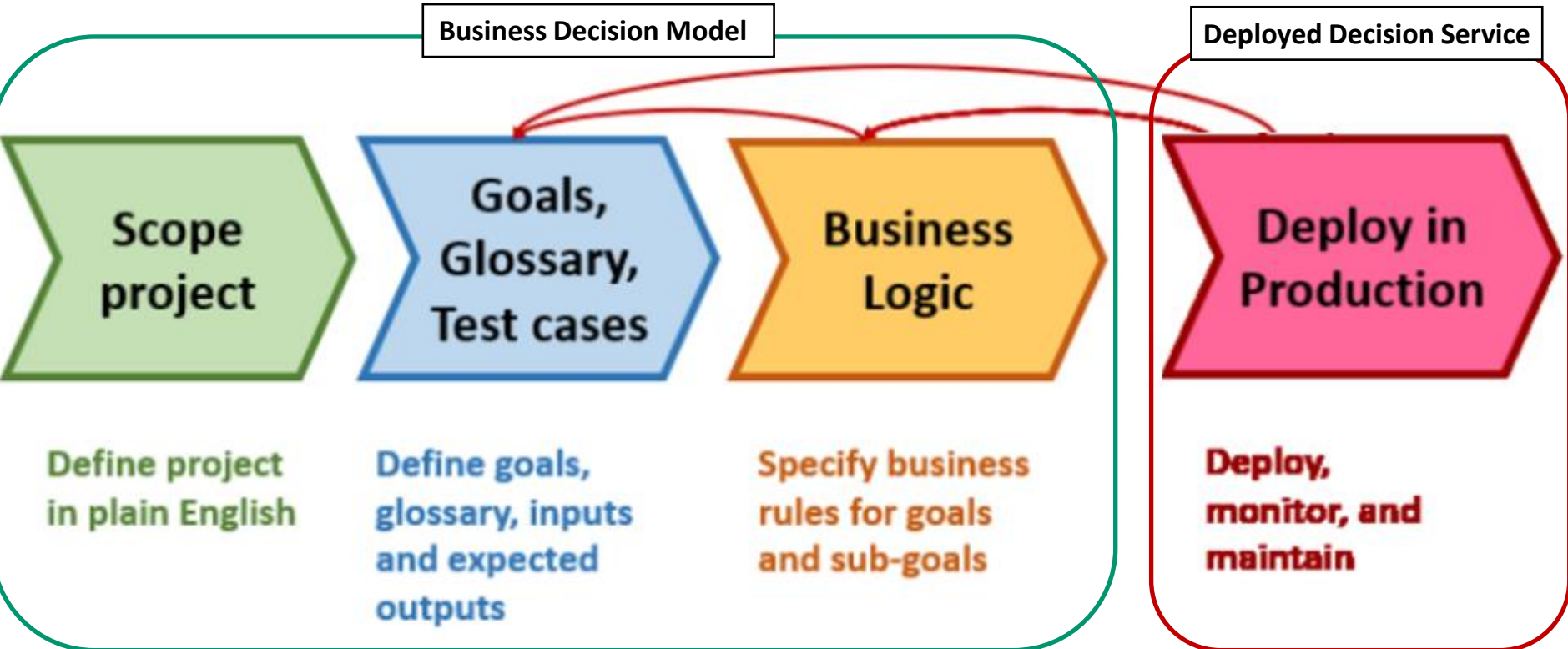
Sample: forgetting dot in “>***rm -rf ./***”

- /// Nowadays it is common for *business analysts to be in charge* of development, testing, and ongoing maintenance of complex business decision models
- /// What Business Analysts do:
 - /// **Represent Decision Logic** in a human-readable and machine-understandable ways (“Decision Model”)
 - /// **Create Test Cases** for the major business situations and test their decision models
 - /// **Debug their decision models** when they fail to produce expected results.
- /// In this presentation we will demonstrate how business analysts can do these 3 tasks without becoming de-facto programmers.

- /// Today you have a good choice of various tools for Business Decision Modeling –see DMCommunity.org
- /// Previously they were called “**Rule Engines**” but today the most products became “**Digital Decisioning Platform**” or even “**Decision Intelligence Platform**”
- /// Many of these tools follow the well-established international standard called **DMN** (Decision Model and Notation)
- /// Different tools may have different DMN compliance levels. However, the biggest practical achievements of the DMN is:

Looking at the de-facto standardized business rules, business users may understand the business logic

Decision Model Life Cycle



Let's Build a Simple Decision Model

Use-case:

- /// Patient therapy for Acute Sinusitis

Objective:

- /// Recommended medications and doses

Medication Rules:

If Patient is 18 years old or older, then a therapy choice is Amoxicillin.

If Patient is younger than 18, a therapy choice is Cefuroxime.

If Patient Penicillin allergic, the therapy of choice is Levofloxacin.

Dosing Rules:

For patients between 15 and 60, the dose is 500mg every 24 hours for 14 days.

If Patient's creatinine level (PCr) > 1.4, commence creatinine clearance (CCr) calculations according to the formula:

$$\text{CCr, in mL/min} = \frac{(140 - \text{age}) \times \text{lean body weight [kg]}}{\text{PCr [mg/dL]} \times 72}$$

If Patient's creatinine clearance < 50 ml/min, then the dose is 250 mg every 24 hours for 14 days.

Drug Interaction Rules:

Check if a patient on active medication. Coumadin and Levofloxacin can result in reduced effectiveness of Coumadin. Produce the proper warning.

Business Rules as Decision Tables

This is a complete Decision Model that can be tested and deployed

DecisionTableMultiHit DefineMedication				
Condition		Condition		Conclusion
Patient Age		Patient Allergies		Recommended Medication
>=	18			Is Amoxicillin
<	18			Is Cefuroxime
		Include	Penicillin	Is Levofloxacin

Medication Rules:
 If Patient is 18 years
 If Patient is younger than 18, a therapy choice is Cefuroxime.
 If Patient Penicillin allergic, the therapy of choice is Levofloxacin.

DecisionTable DefineDosing				
Condition		Condition		Conclusion
Patient Age		Patient Creatinine Level	Patient Creatinine Clearance	Recommended Dose
Within	[15..60]			Is 500mg every 24 hours for 14 days
		>	1.4	Is 250mg every 24 hours for 14 days
			<	Is More dosing rules is needed

Dosing Rules:
 For patient days.
 If Patient 24 hours
 If Patient 24 hours
 If Patient (CCr) calc

Glossary glossary			
Decision Variable	Business Concept	Attribute	Type
Encounter Diagnosis	DoctorVisit	encounterDiagnosis	String
Recommended Medication		recommendedMedication	String
Recommended Dose		recommendedDose	String
Drug Interaction Warning		warning	String
Patient Therapy	Patient	patientTherapy	String
Patient Name		name	String
Patient Age		age	int
Patient Weight		weight	double
Patient Allergies		allergies	String[]
Patient Creatinine Level			
Patient Creatinine Clearance			
Patient Active Medication			

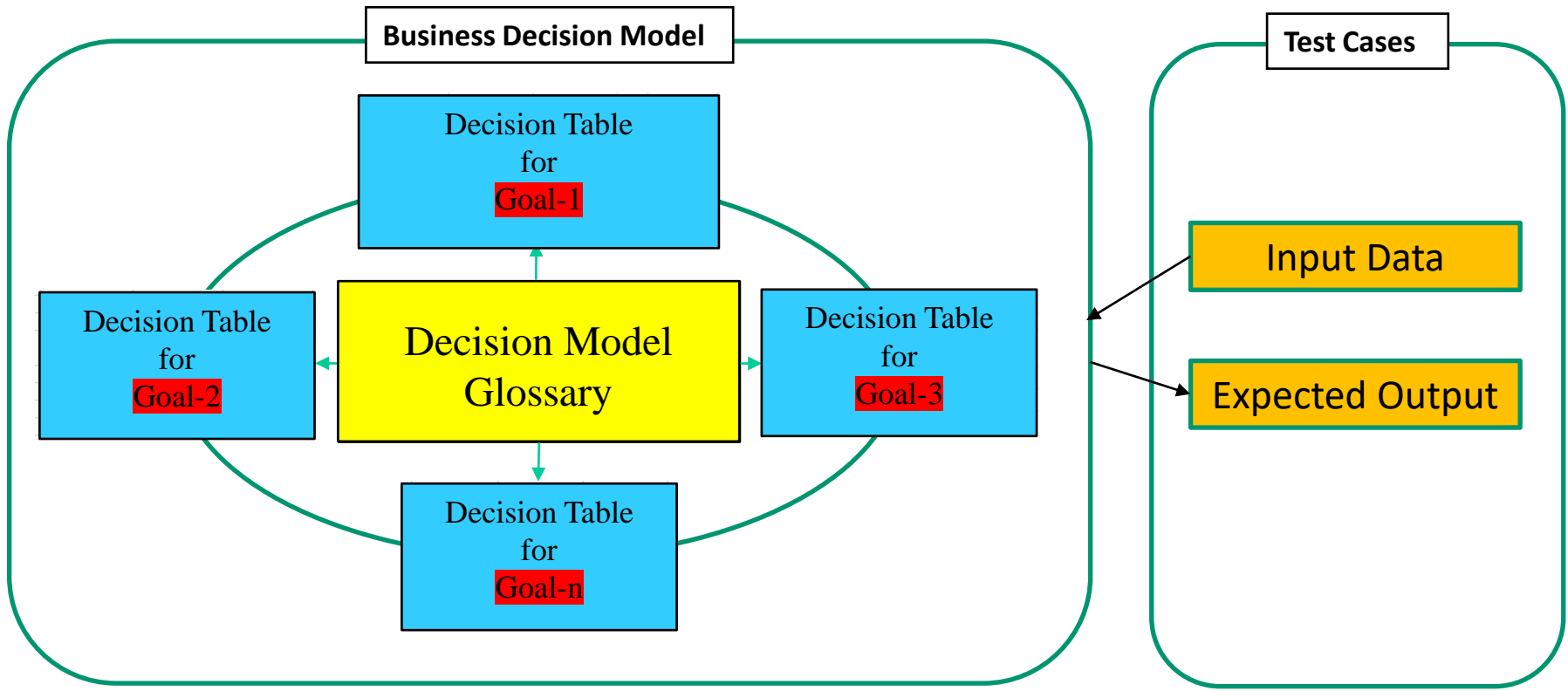
DecisionTable CalculateCreatinineClearance	
Action	
Patient Creatinine Clearance	
$(140 - \text{Patient Age}) * \text{Patient Weight} / (\text{Patient Creatinine Level} * 72)$	

DecisionTable WarnAboutDrugInteraction				
Condition		Condition		Action
Recommended Medication		Patient Active Medication		Drug Interaction Warning
Is	Levofloxacin	Is	Coumadin	Coumadin and Levofloxacin can result in reduced effectiveness of Coumadin

Drug Interaction Rules:
 Check if a patient on active medication result in reduced effectiveness of Coumadin

You don't need to manually define an execution order of different decision tables: Rule Engine can figure it out automatically!

Decision Model with a Glossary at the Heart

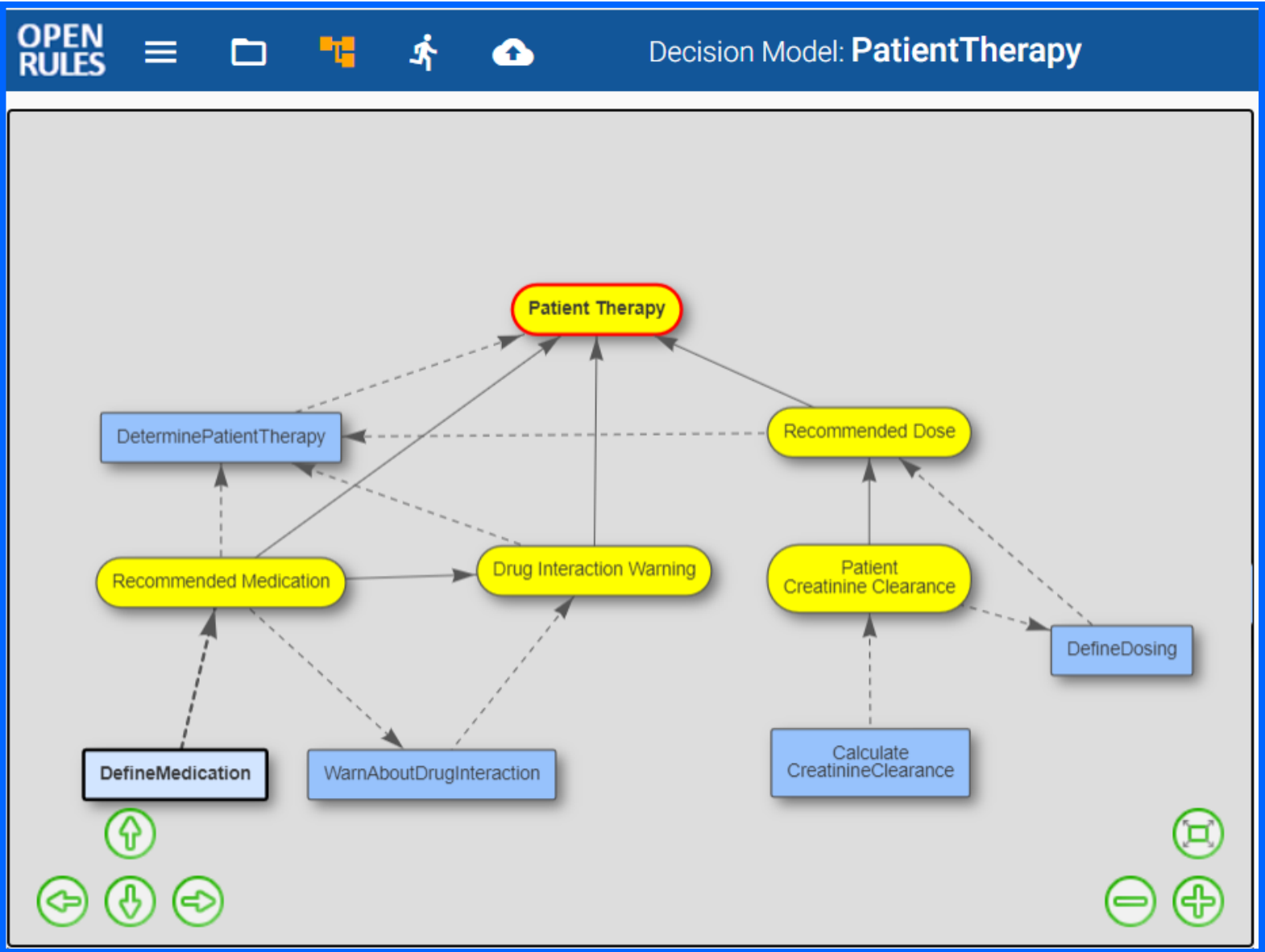


Testing Decision Model

Example of Test Cases

DecisionTest testCases										
#	ActionDefine	Action Define	Action Define	ActionDefine	ActionDefine	ActionDefine	ActionExpect	ActionExpect	ActionExpect	ActionExpect
Test ID	Encounter Diagnosis	Patient Age	Patient Weight	Patient Allergies	Patient Active Medication	Patient Creatinine Level	Recommended Medication	Recommended Dose	Drug Interaction Warning	Patient Creatinine Clearance
Test 1	Acute Sinusitis	58	78	Penicillin,Streptomycin	Coumadin	2.00	Levofloxacin	500mg every 24 hours for 14 days	Coumadin and Levofloxacin can result in reduced effectiveness of Coumadin	44.42
Test 2	Acute Sinusitis	65	83			1.80	Amoxicillin	250mg every 24 hours for 14 days	None	48.03
Test 3	Diabetes	27	110			1.88			None	

Live Demo with a Graphical Decision Manager



Execution protocol

```

Execute goal 'Patient Therapy'
  DefineMedication #2 (B6:G6)
  IF 'Patient Age' >= 18
  THEN 'Recommended Medication' Is Amoxicillin
  Variables:
    Patient Age: 27
    Recommended Medication: --> Amoxicillin

  CalculateCreatinineClearance #1 (B5:B5)
  THEN 'Patient Creatinine Clearance' = (140 - Patient Age) * P
  Creatinine Level * 72)
  Variables:
    Patient Age: 27
    Patient Creatinine Clearance: 0.0 --> 91.8291962174941
    Patient Creatinine Level: 1.88
    Patient Weight: 110.0

  DefineDosing #1 (B5:I5)
  IF 'Patient Age' Within [15..60]
  THEN 'Recommended Dose' Is 500mg every 24 hours for 14 days
  Variables:
    Patient Age: 27
    Recommended Dose: --> 500mg every 24 hours for 14 days

  WarnAboutDrugInteraction #2 (B7:F7)
  THEN 'Drug Interaction Warning' = None
  Variables:
    Drug Interaction Warning: --> None

  DeterminePatientTherapy #2 (B6:D6)
  IF 'Encounter Diagnosis' Is Not Acute Sinusitis
  THEN 'Patient Therapy' = Sorry, this decision model can handle
  Variables:
    Encounter Diagnosis: Diabetes
    Patient Therapy: --> Sorry, this decision model can handle

Test 'Test 3' completed OK. Elapsed time 45.03 ms
  
```

Decision Table: Rule# (Cells)	Executed Rule	Variables and Values
DefineMedication: 2 (B6:G6)	IF 'Patient Age' >= 18 THEN 'Recommended Medication' Is Amoxicillin	Patient Age=27 Recommended Medication={old:, new:Amoxicillin}
CalculateCreatinineClearance: 1 (B5:B5)	THEN 'Patient Creatinine Clearance' = (140 - Patient Age) * Patient Weight / (Patient Creatinine Level * 72)	Patient Creatinine Clearance={old:0.0, new:91.8291962174941} Patient Age=27 Patient Weight=110.0 Patient Creatinine Level=1.88
DefineDosing: 1 (B5:I5)	IF 'Patient Age' Within [15..60] THEN 'Recommended Dose' Is 500mg every 24 hours for 14 days	Patient Age=27 Recommended Dose={old:, new:500mg every 24 hours for 14 days}
WarnAboutDrugInteraction: 2 (B7:F7)	THEN 'Drug Interaction Warning' = None	Drug Interaction Warning={old:, new:None}
DeterminePatientTherapy: 2 (B6:D6)	IF 'Encounter Diagnosis' Is Not Acute Sinusitis THEN 'Patient Therapy' = Sorry, this decision model can handle only Acute Sinusitis	Encounter Diagnosis=Diabetes Patient Therapy={old:, new:Sorry, this decision model can handle only Acute Sinusitis}

- /// Whether you like it or not, your business decision models will contain bugs like regular software always does. They will fail to produce expected results under unforeseen circumstances.
- /// As a result, you may have troubles to understand why certain rules were not executed while others did.
- /// That's why a business analyst, the author of a decision model, needs a friendly graphical Rule Debugger

- /// A powerful while easy-to-use **Rule Debugger** should allow a business analyst to do the following:
 - /// Step through the execution of business decision models
 - /// Pause it at any rule or ruleset
 - /// Set Breakpoints
 - /// Inspect the state of the decision model in different breakpoint
 - /// Analyze the values of all used decision variables at any execution point
- /// Contrary to Program Debuggers, a **Rule Debugger** should be business friendly

Live Demo of a Business-Friendly Rule Debugger

OPEN RULES



Decision Model: PatientTherapy



▶ Next Rule

▶▶ Next Ruleset

▶| GoTo End/Breakpoint

↺ Restart

Executed Rules Only

Variables
 Breakpoints

Debugging testCases-Test 1

DefineMedication (DecisionTable)

Rule #1 (B5:G5)

IF 'Patient Allergies' Include Penicillin
THEN 'Recommended Medication' Is Levofloxacin

CalculateCreatinineClearance (DecisionTable)

Rule #1 (B5:B5)

THEN 'Patient Creatinine Clearance' (140 - Patient Age) * Patient Weight / (Patient Creatinine Level * 72)

DefineDosing (DecisionTable)

Rule #1 (B5:I5)

IF 'Patient Age' Within [15..60]
THEN 'Recommended Dose' Is 500mg every 24 hours for 14 days

WarnAboutDrugInteraction (DecisionTable)

Rule #1 (B6:F6)

IF 'Recommended Medication' Is Levofloxacin
AND 'Patient Active Medication' Is Coumadin

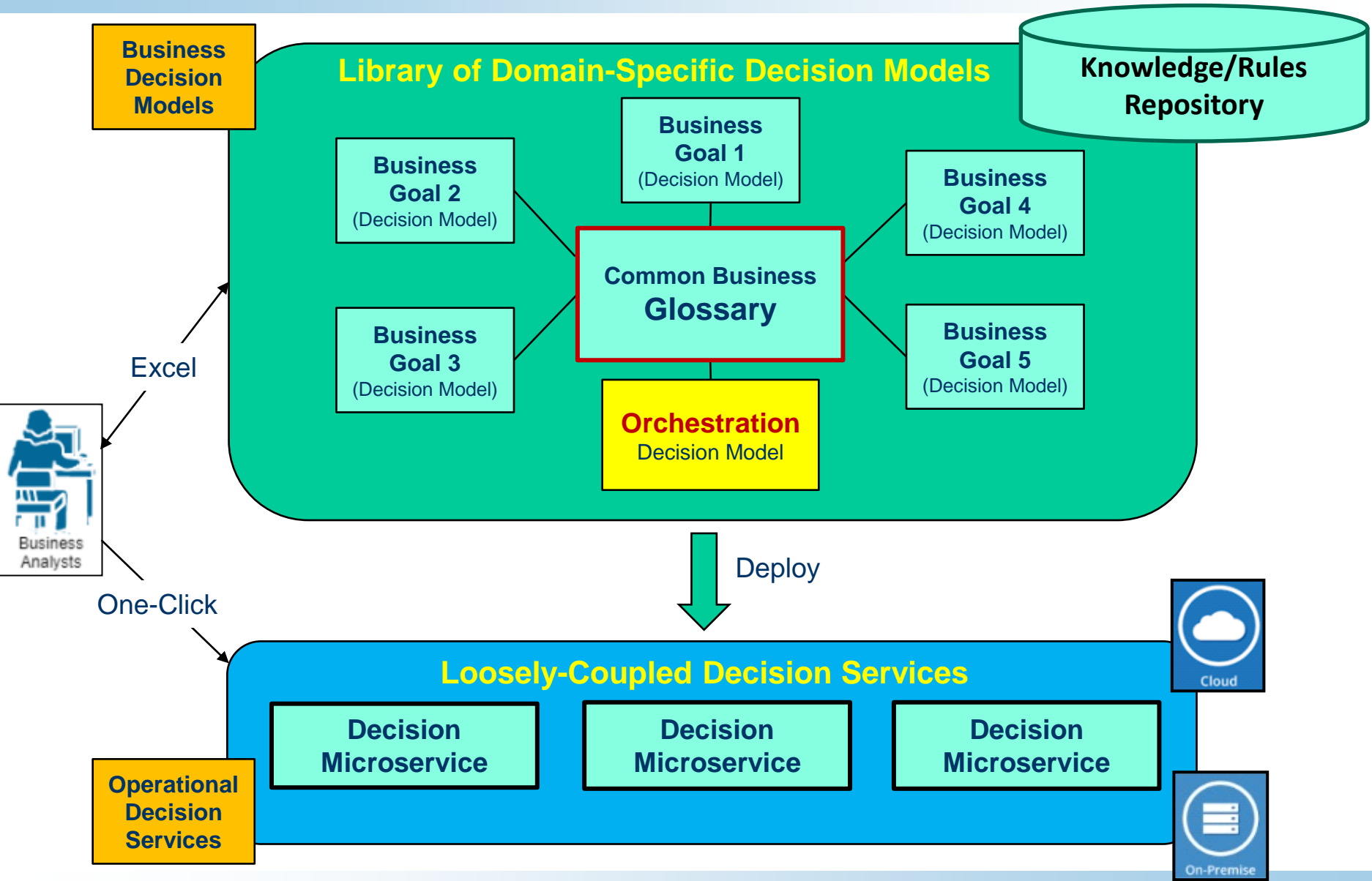
Related Variables Only

Variable	Value
Recommended Medication	"Levofloxacin"
Drug Interaction Warning	
Patient Active Medication	"Coumadin"

- ⌘ In real-world business analysts build and maintain not little pilots or demos but *production-level decision services!*
- ⌘ Their decision models are deployed as secured and highly efficient decision services.
- ⌘ However, they, business analysts, are still responsible for correctness and maintenance of their business logic!
- ⌘ The more complex decision model is the more difficult to maintain, expand, and debug it!

- ≡ That's why experienced people do not develop large (monolithic!) decision models
- ≡ Instead, they usually develop a set of smaller reusable decision models that implement 1-2 business function
- ≡ They test, debug, and deploy Decision Microservices separately
- ≡ Then they use them to compose large decision models (a LEGO approach)

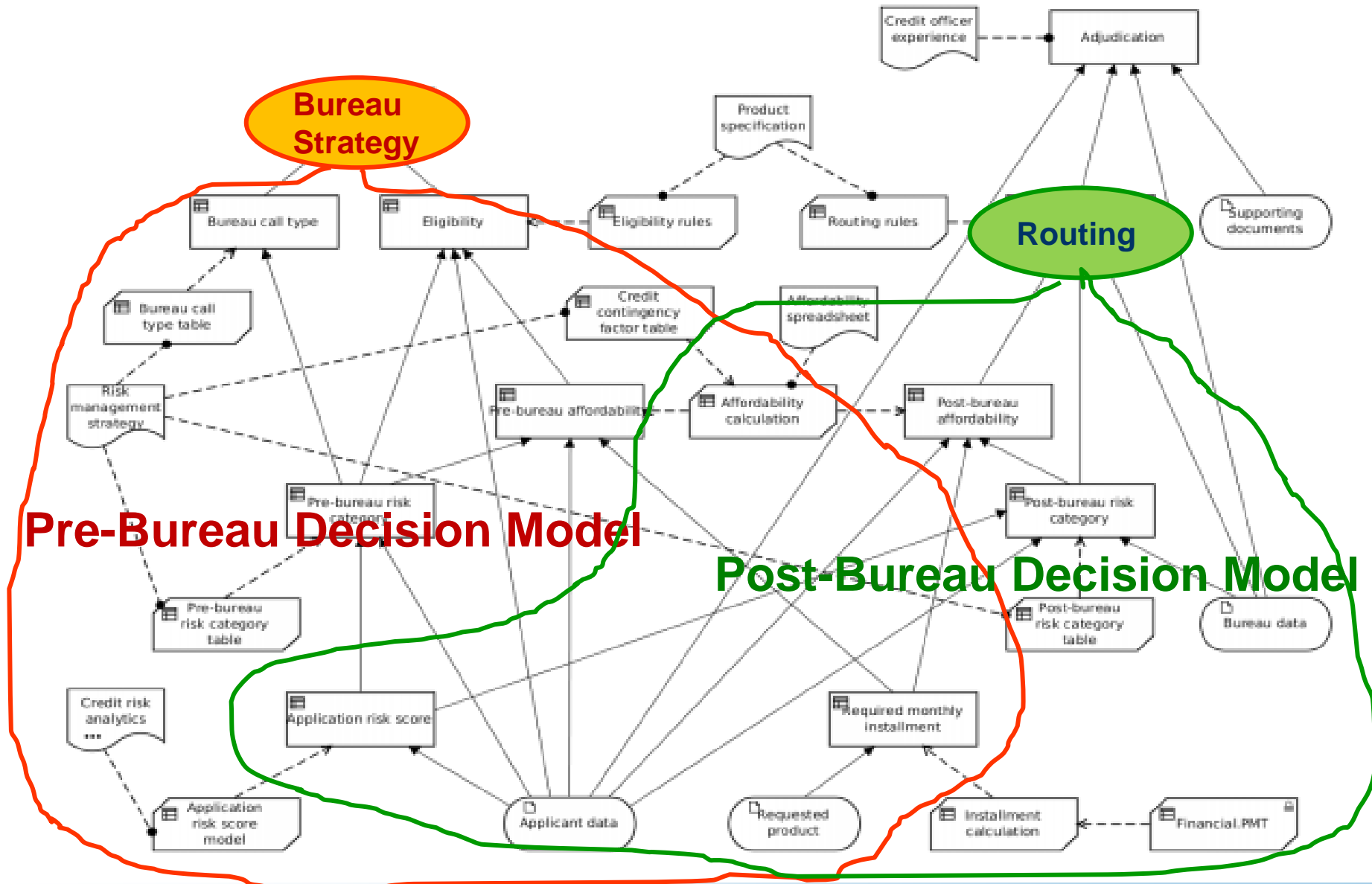
From Business Decision Models to Operational Decision Services



- ⌘ We will use the main sample from the DMN standard sample to build a library of Decision Models and Microservices for the **Loan Origination** domain

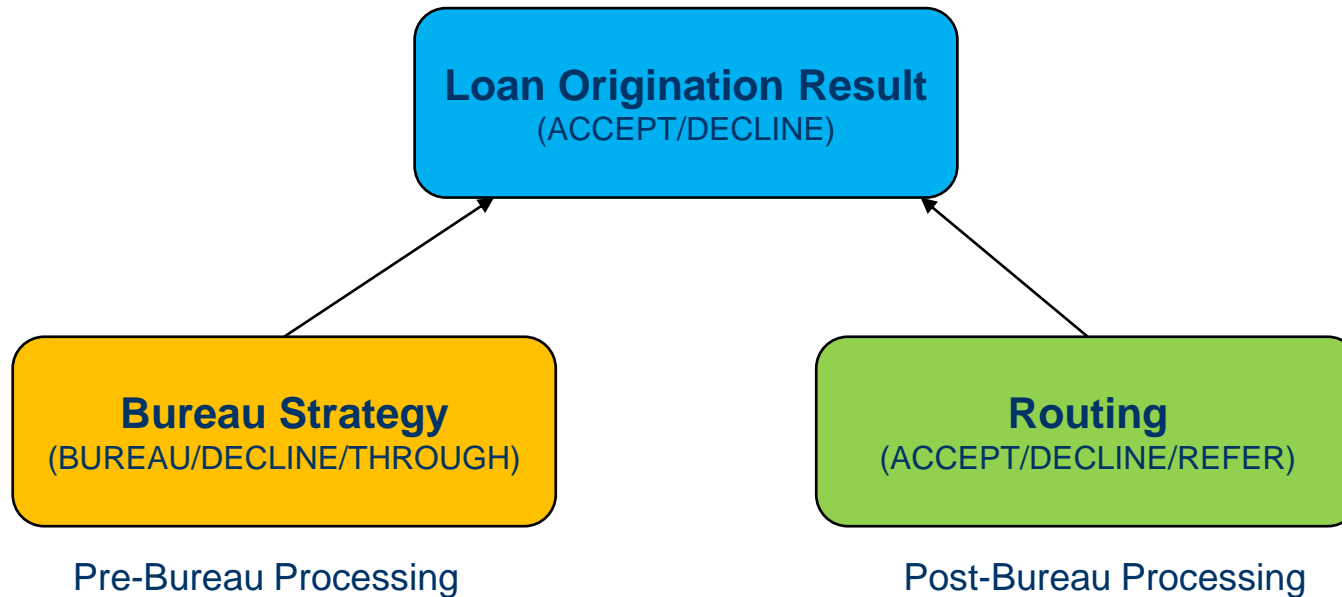
- ⌘ Here is our implementation plan:
 1. Build and test each model separately
 2. Use small models as “Lego Boxes” to create larger Decision Services and deploy them as cloud microservices
 3. Create a top-level Loan Origination Service by *orchestrating* smaller decision services

DMN-based Sample: Loan Origination Decision Model



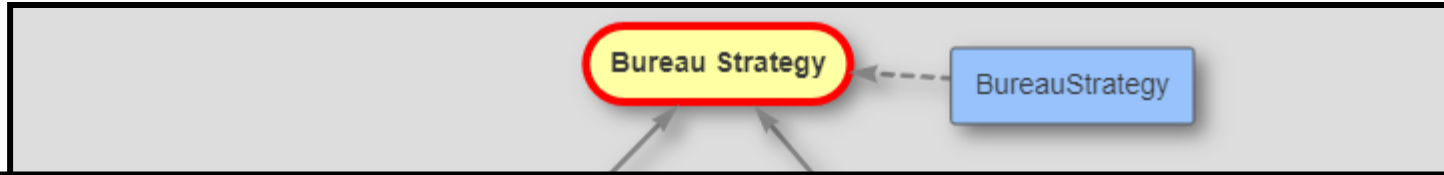
High-Level Goals for the Loan Origination Decision Model

- As business analysts, we may identify 3 major goals:



- These goals have common and specialized sub-goals like **Affordability** and **Risk Category** which could be implemented as stand-alone decision microservices

Live Demo: Debugging “Bureau Strategy”



OPEN RULES [Icons: Menu, Folder, Window, Person, Cloud] Decision Model: **BureauStrategy** [Help Icon]

▶ Next Rule ▶▶ Next Ruleset ▶| GoTo End/Breakpoint ↻ Restart

Executed Rules Only Variables Breakpoints

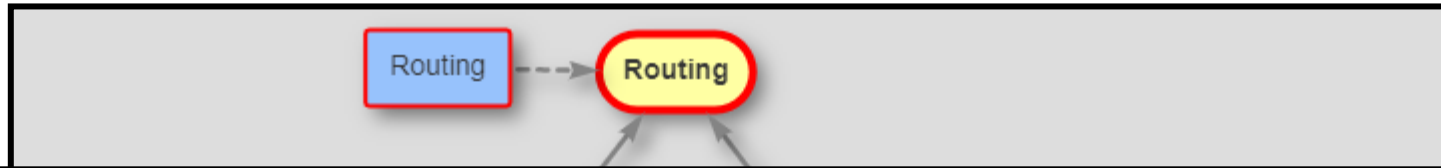
```

PMT (DecisionTable)
  Rule #1 (B27:B27)
    THEN 'PMT' ( Amount * Rate/12 ) / ( 1 - pow(1 +Rate/12,-Term) )
RequiredMonthlyInstallment (DecisionTable)
  Rule #1 (B5:D5)
    IF 'Product Type' Is SPECIAL LOAN
    THEN 'Required Monthly Installment' PMT + 25.00
  Rule #2 (B6:D6)
    THEN 'Required Monthly Installment' PMT + 20.00
DisposableIncome (DecisionTable)
  Rule #1 (B5:B5)
    THEN 'Disposable Income' Monthly Income - (Monthly Repayments +
    Monthly Expenses)
CreditContingencyFactor (DecisionTable)
  Rule #1 (B5:D5)
    IF 'Risk Category' Is One Of HIGH,DECLINE
    THEN 'Credit Contingency Factor' 0.6
  
```

Related Variables Only

Variable	Value
Required Monthly Installment	3153.636546143113
Disposable Income	4500
Credit Contingency Factor	0.8
Affordability	

Live Demo: Creating and Debugging "Routing"



OPEN RULES Decision Model: **Routing**

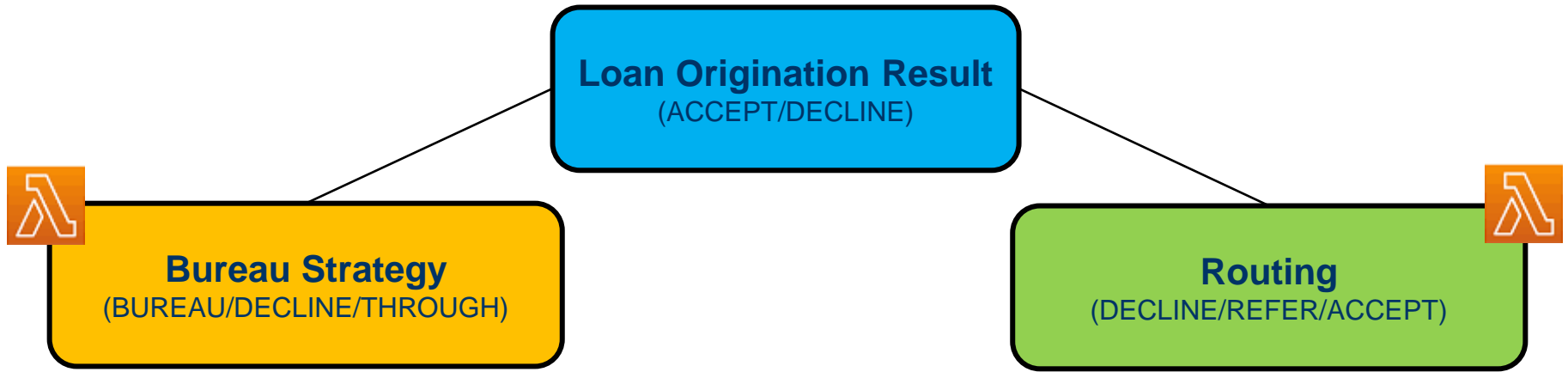
Next Rule Next Ruleset GoTo End/Breakpoint Restart Executed Rules Only Variables Breakpoints

```

  THEN 'Credit Contingency Factor' 0.7
  Rule #3 (B7:D7)
  IF 'Risk Category' Is One Of LOW, VERY LOW}
  THEN 'Credit Contingency Factor' 0.8
  Affordability (DecisionTable)
  Rule #1 (B5:C5)
  IF 'Required Monthly Installment' < Disposable Income * Credit
  Contingency Factor}
  THEN 'Affordability' true
  Routing (DecisionTable)
  Rule #1 (B5:F5)
  IF 'Affordability' false}
  THEN 'Routing' DECLINE
  Rule #2 (B6:F6)
  IF 'Affordability' true}
  AND 'Bankrupt' true}
  
```

Related Variables Only	
Age	51
Marital Status	"M"
Employment Status	"EMPLOYED"
Monthly Income	10000
Monthly Repayments	2500
Monthly Expenses	3000
Existing Customer	true
Id	"1"
Application Risk Score	138
Risk Category	"VERY LOW"

Deployment and Orchestration of Decision Services



- ⌘ Let's assumed that we deployed our decision models “Bureau Strategy” and “Routing” as AWS Lambda functions
- ⌘ Now we may orchestrate them to build the main decision model Loan Origination Result”

Orchestration Logic is Business Logic!

≡ We use a regular Excel table “Decision” to define “LoanOriginationResult”

Decision LoanOriginationResult					
Condition		Condition		ActionExecute	Action
Bureau Strategy		Routing		Execute	Loan Origination Result
				BureauStrategyService	
Is	DECLINE				DECLINE
Is Not	DECLINE			RoutingService	
Is Not		Is	DECLINE		DECLINE
Is Not		Is	REFER		REFER
Is Not		Is	ACCEPT		ACCEPT

ActionExecute can execute:

- internal decision tables or
- external decision services (!)

Decision Service decisionServices		
Service Name	Service Type	Service Endpoint
BureauStrategyService	REST	https://bfsu86u7u6.execute-api.us-east-1.amazonaws.com/test/bureau-strategy
RoutingService	REST	https://f7b53vrel.execute-api.us-east-1.amazonaws.com/test/routing

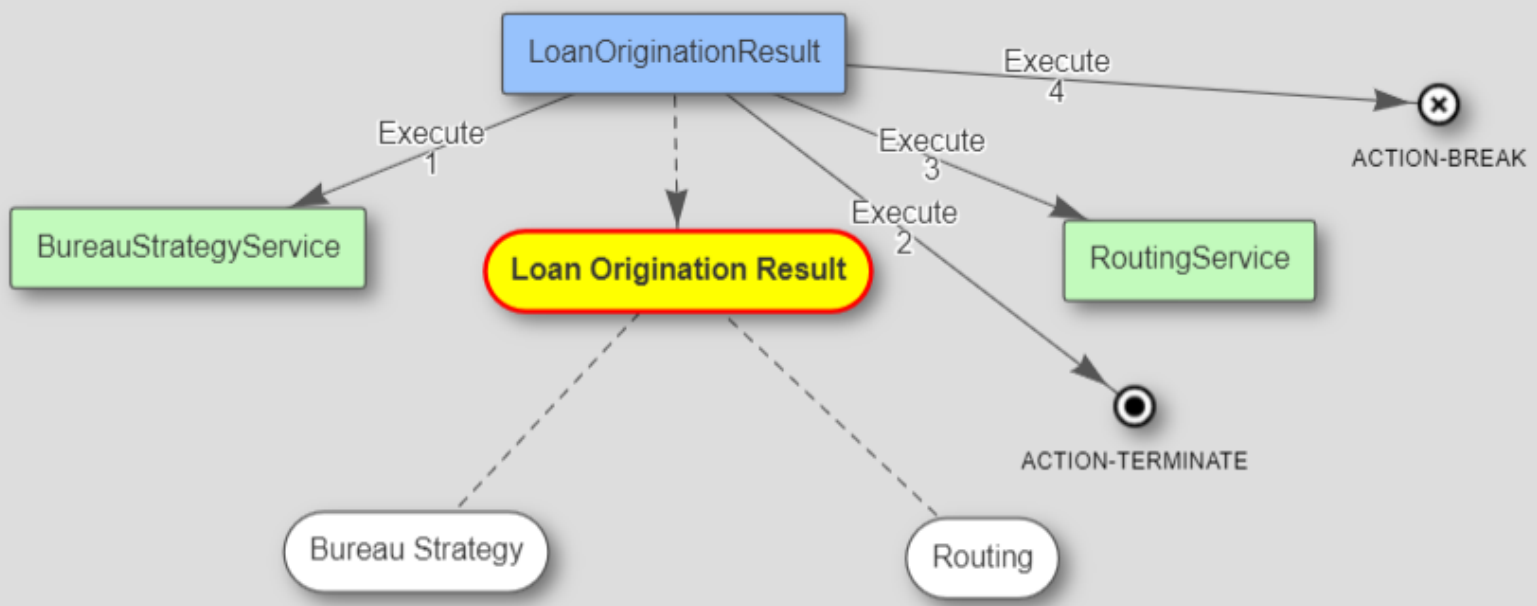
Test and Deploy the final decision model “Result”

≡ We can test decision model “Result” by a click on “*test.bat*”:

Decision Table: Rule# (Cells)	Executed Rule	Variables and Values
LoanOriginationResult: 1 (B17:G17)	THEN 'Execute' = BureauStrategyService	
LoanOriginationResult: 3 (B19:G19)	IF 'Bureau Strategy' Is Not DECLINE THEN 'Execute' = RoutingService	Bureau Strategy=THROUGH
LoanOriginationResult: 6 (B22:G22)	IF 'Bureau Strategy' Is Not DECLINE AND 'Routing' Is ACCEPT THEN 'Loan Origination Result' = ACCEPT	Bureau Strategy=THROUGH Routing=ACCEPT Loan Origination Result= {old:?, new: ACCEPT }

≡ Now we can also deploy and test the top-level decision model “Loan Origination Result” as an AWS Lambda function

Live Demo: Composed Decision Model "Result"



Live Demo: Debugging Composed Decision Model

“Result”

▶ Next Rule

▶▶ Next Ruleset

▶| GoTo End/Breakpoint

↻ Restart

Executed Rules Only

Variables

Breakpoints

LoanOriginationResult (Decision)

Rule #1 (B17:G17)

THEN 'Execute' = BureauStrategyService

BureauStrategyService (Decision Service)

Rule #2 (B18:G18)

IF 'Bureau Strategy' Is DECLINE

THEN 'Execute' = ACTION-TERMINATE

AND 'Loan Origination Result' DECLINE

Rule #3 (B19:G19)

IF 'Bureau Strategy' Is Not DECLINE

THEN 'Execute' = RoutingService

RoutingService (Decision Service)

Rule #4 (B20:G20)

IF 'Bureau Strategy' Is Not DECLINE

AND 'Routing' Is DECLINE

THEN 'Loan Origination Result' DECLINE

Rule #5 (B21:G21)

IF 'Bureau Strategy' Is Not DECLINE

AND 'Routing' Is REFER

THEN 'Loan Origination Result' REFER

Rule #6 (B22:G22)

IF 'Bureau Strategy' Is Not DECLINE

AND 'Routing' Is ACCEPT

THEN 'Execute' = ACTION-BREAK

AND 'Loan Origination Result' ACCEPT

Completed

Monthly Income	<input type="checkbox"/>	10000
Monthly Repayments	<input checked="" type="checkbox"/>	2800
Monthly Expenses	<input checked="" type="checkbox"/>	3000
Existing Customer	<input checked="" type="checkbox"/>	true
Id	<input checked="" type="checkbox"/>	"1"
Application Risk Score	<input checked="" type="checkbox"/>	138
Risk Category	<input checked="" type="checkbox"/>	"VERY LOW"
PMT	<input checked="" type="checkbox"/>	3133.636546143113
Required Monthly Installment	<input checked="" type="checkbox"/>	3153.636546143113
Disposable Income	<input checked="" type="checkbox"/>	4200
Credit Contingency Factor	<input checked="" type="checkbox"/>	0.8
Affordability	<input checked="" type="checkbox"/>	true
Eligibility	<input checked="" type="checkbox"/>	"ELIGIBLE"
Bureau Call Type	<input checked="" type="checkbox"/>	"NONE"
Service Stage	<input checked="" type="checkbox"/>	"Result"
Bureau Strategy	<input checked="" type="checkbox"/>	"THROUGH"
Routing	<input checked="" type="checkbox"/>	"ACCEPT"
Loan Origination Result	<input checked="" type="checkbox"/>	"ACCEPT"

- /// Nowadays we have solid rules-based Decisioning Tools that help Business Analysts create and maintain complex decision models
- /// Business Analysts can:
 - /// **Represent Decision Logic** in a human-readable and machine-understandable ways
 - /// **Create Test Cases** for the major business situations and test their decision models
 - /// **Debug their decision models** when it is difficult to interpret the produced results.

jacobfeldman@openrules.com

support@openrules.com

+1 732 993 3131

www.OpenRules.com

