ODM Solution Health Check 03/2024

EXAMPLE CLIENT -

DJ DES JARDINS

1 Intent of Health Check

The intent of the Health Check is to give you an idea of how well your solutions are doing in terms of overall Design and Implementation when compared to generally accepted Best Practices. It is also intended to point out areas of concern related to:

- Rules Manageability
- Solution Extensibility
- Governance issues
- Accuracy of results

(Not necessarily completeness of use case implementation unless noted)

Manageability and Extensibility is adversely affected by the current design and implementation of this solution and will be the major focus of **this HC.**

It is a tool for ODM Practitioners and Management to Improve the Solutions and to Mitigate Risk

Critical Issues

Critical issues are those that need to be addressed immediately and should be considered a "Material Defect" in the solution or a near guarantee to prevent your solution from meeting a required requirement or goal.

High Risk

High Risks are those that should be addressed immediately and considered putting your Solution at "High Risk" of not meeting one or more of your intended or requirements or goals.

Medium Risk

Medium Risks are those issues that should be addressed soon and considered potentially putting your Solution at "Risk" of not meeting one or more of your intended or requirements or goals.

Low Risk

Low Risks are those issues that should be reevaluated in terms of your ongoing design needs and requirements. The Solution may be at "Some Risk" of not meeting one of your intended or requirements or goals.

Informative

This is information that will help bring your Solution in line with Best Practices and prevent future issues.

1.2 Approach

The analysis does not take into consideration the size or scope of solution nor experience of the team doing the development. Therefor there is a likelihood that some of the issues that are identified exist due to where the development team is currently positioned on the maturity scale for ODM development.

That said, issues raised during a Health Check should not be taken negatively.

1.3 FOCUS of this current review

This review was an initial review with no specific identified focus since there have not been any specific areas of concern identified. It was a broad-ranging and detailed review.

2 New Items

Since this is an "initial" review **ALL** results are considered ***New***

2.1 XOM & BOM

The solution does not HAVE a XOM or BOM, all data is managed as Global String variables.

- Except a single virtual BOM object with a single method: isGreaterThan "{0} is greater than {1, min}"

2.1.1 FileNet Routing

Critical Issues

1. None Identified

High Risk

2. None Identified

Medium Risk

3. Not having an actual XOM/BOM severely limits the ability to enrich the vocabulary and to build "concepts" to simplify Rule Authoring and Testing
    a. List of Items that are limited or not possible to use without a XOM/BOM
        i. Virtual attribute &/or methods
        ii. The RETE algorithm
            1. https://www.ibm.com/docs/en/odm/8.11.1?topic=mode-reteplus-algorithm
    b. See Section 3.1 for additional details.

Low Risk

4. Strings are initialized to " " instead of ""
    a. Prevents utilization of "existence" vocabulary
    b. Cannot use "is empty" vocabulary on values only initialized and not changed.
    c. NOTE: initialized in the Variable Sets

Informative

5. The vocabulary is missing a "subject" and "article".
    a. Current vocabulary: "Doc Type"
    b. Suggested vocabulary:
        i. "the document type of the claim"
        ii. "the document type"
    c. See Section 3.2 for additional details.
6. The use of Domain values for defined lists of Strings would significantly simplify rule development and testing.
    a. Implementing Domains for Dropdowns in Rule Authoring and Decision Runner testing can still be accomplished even though there is no XOM/BOM for the solution.
    b. Implementation would only be slightly more complicated due to lack of XOM/BOM objects.
    c. **See Section 3.3 for additional details.**

2.2 Operational Framework

2.2.1 FileNet Routing

Critical Issues

1. Solution is still using the "Classic Engine".
    a. Deprecated since V8.9.0
        i. DE available for use since V8.5.1
        ii. Available on Z/OS even earlier
    b. Removed in V8.12.0
        i. Needs to be migrated to Decision Engine "Prior" to upgrading to 8.12.x
    c. Tagging as critical since choosing to NOT migrate to DE during the migration to 8.11 will require another "Migration Effort" when upgrading to later version of ODM instead of a simple platform upgrade.
        i. If the requirement is to get to the point where upgrades are not a major effort, then this is critical
        ii. If this is NOT a concern it can be dropped in severity
    d. Details:
        i. Migration: https://www.ibm.com/docs/en/odm/8.11.1?topic=engine-migrating-decision
        ii. API Differences: https://www.ibm.com/docs/en/odm/8.11.1?topic=rec-mapping-between-classic-rule-engine-decision-engine-api
    e. Assuming since the switch to Decision Engine (DE) from Classic Engine (CE) was not performed during the migration to 8.8.1 there are required changes to the Operational Framework to support the DE.
        i. This will require a deeper analysis that has not been completed as part of this HC.

High Risk

2. Solution is basically a rigid "Decision Tree" pattern ***(Figure 1).***
    a. Manual Rule Ordering, not using ANY of the available optimizations.
        i. There is a single rule artifact in EVERY task.
    b. Forces developer/rule author to change rule flows to add functionality beyond updating tables.
    c. Makes solution overly complicated:
        i. More difficult to implement Rule Change Management
        ii. Makes testing and debugging much more difficult.
    d. This is contrary to **MULTIPLE** Best Practices

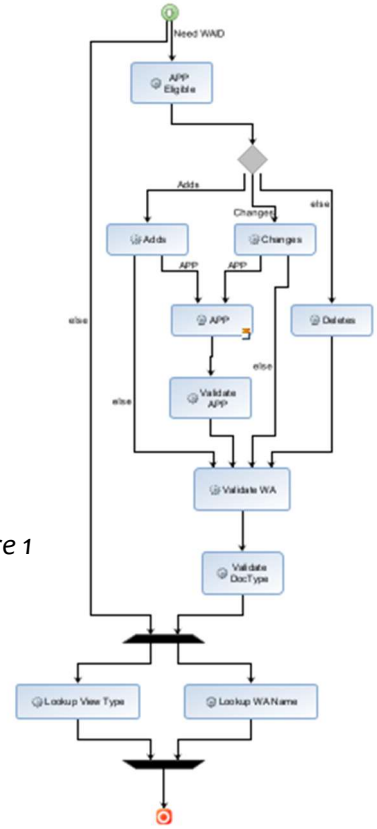


Figure 1

Medium Risk

3. Multiple Tasks have single rules selected instead of packages/directories.
    a. Best Practices is to select packages not individual rules.
    b. Individual rules per task is an acceptable pattern for "Technical" rules vs. "Business Rules".

4. Rule Flow is artificially more complex than required.
    a. There is no need for this complexity. A mix of these changes could consolidate this down to a single task ***(Figure 2)***.
        i. A simple "filter" in the Definition sections of three Decision Tables.
        ii. Use of an ***otherwise*** or changes to the existing validation check rules.
    b. This Rule Flow Section has two complexities that are unnecessary ***(Figure 3)***.
        i. There is no reason why these two tasks cannot currently be a single task.
        ii. They both implement the "winner" pattern but contain single Decision Tables whose rows are mutually exclusive and cannot use the pattern.

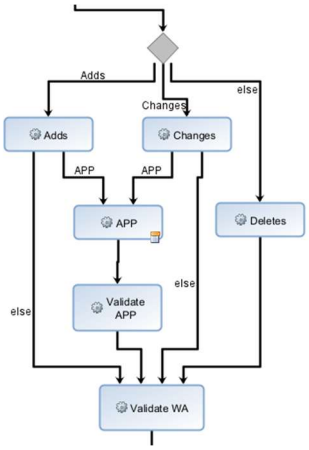


Figure 2

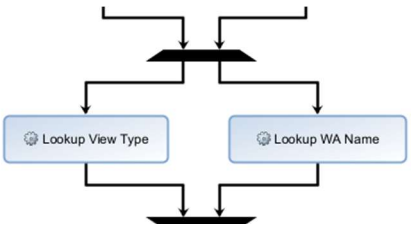


Figure 3

Low Risk

5. Actions in the Start and End nodes of the "Main" ruleflow.
    a. Hides functionality since there is no visual indicator that the node contains action code.
6. Using RETE algorithm when Fastpath algorithm, in each case here, is more appropriate and more performant.
    a. Cannot actually "use" RETE with this solution since there are no "objects" to reevaluate when its state is updated.

Informative

7. High-level rule flow Best Practices
    a. **See Section 3.4 for additional details.**

2.3 Business Rules

2.3.1 FileNet Routing

Critical Issues

1. There is a disconnect between HOW 3 key Decision Tables are ACTUALLY operating and how it is BELIEVED they are operating.
    a. This disconnect likely makes Changes and Testing more difficult.
    b. **See Section 3.5 for additional details.**

High Risk

2. There are 5 Decision Tables using the "Single row exit" or "Winner" pattern all with 100s of rows
    a. Extremely difficult to maintain and test.
        i. Related to [2.3.1 #1]
    b. Use of "Concepts" would significantly reduce complexity of this pattern.
        i. Possibly remove the need for the pattern to be applied to a list of "Possible" results after all possible "correct" answers identified.
    c. **See Section 3.5 for additional details.**

Medium Risk

3. A significant number of Condition Columns in most of the decision Tables are unused.
    a. A significant number of unused columns adds unnecessary complexity that makes reviewing, understanding, and updating the tables more difficult than it should be.
    b. This is a MEDIUM vs a LOW issue due to the size of the Decision Tables
    c. Some columns are not used at all, some are used in a single Row (Rule)
4. Using a single return attribute that designated a single rule execution: ruleFired
    a. It can only capture a single rule execution, the last one.
    b. Has to be manually added (a unique identifier) to each rule to set this value.
    c. This is a very outdated approach to track rule execution traces and to debug execution.
        i. **See Section 5.2 for recommended replacement.**

Low Risk

5. Using a response value as a "temporary" flag to control navigation through the Rule Flow
    a. `'Work Assigment ID' is "RULE4"`
    b. Forces a ridged Rule Sequence
    c. Makes testing and debugging more difficult.

Informative

6. ***Overly Complex implementation:***
    a. There are **918 rules** that determine what Work Assignment should be assigned.
        i. All these rules are implementing a "winner" pattern across 3 Decision Tables
        ii. They have to ALL be manually ordered
    b. There are ***262 possible Work Assignments*** that can be assigned.
        i. That is approximately 3.5 rules per Work Assignment
        ii. That seems high considering the way the Winning Pattern is being implemented

2.4 Testing Frameworks

Considering the Architecture implemented by the Operational Framework I have no useful feedback to provide on the current Testing Framework

If the Operational Framework is remediated, then the testing Framework can be enhanced to better support testing.

3 Detailed Issue Descriptions

The following provides more depth around an issue identified in section 2

3.1 Virtual methods and Attributes [2.1.1 #3]

3.1.1 General

Examples of when to use virtual methods.

- Boolean vocabulary enrichment.
- List management using rules vocabulary.
- Calculate information like an Age.
    - Data contains the context date: birth date, booking date, etc.
    - Today's date is known.
    - Calculate and return the "Age" at that point in time.

Examples of when to use virtual attributes.

- Model Flattening: Connection → ArrivalStation
    - Getter: `return this.inboundTrain.destination;`
    - Setter: `this.inboundTrain.destination = value;`
        - 'value' is a keyword that passes the input for the setter.
- Type Conversion. Have a String but need to use numeric in rules. NumericOfString
    - Getter: `return Integer.parseInt(this.stringValue);`
    - Setter: `this.stringValue = String.valueOf(value);`

3.1.2 Booleans

Boolean default vocabulary (except the default "true") uses true and false in the vocabulary. Most Business Users don't think in terms of true and false, they think in terms of "is" and "is not". Create virtual methods to implement vocabulary for the following functions:

- Determine if attribute is false.
    - Virtual Method: isNotCorp(), Return Type: Boolean
    - Vocabulary: {this} is not corporate
    - BOM to XOM: return !this.corporate;
- Set attribute to true.
    - Virtual Method: setCorp(), Return Type: void
    - Vocabulary: {this} is corporate
    - BOM to XOM: this.corporate = true;
- Set the attribute to false.
    - Virtual Method: setNotCorp(), Return Type: void
    - Vocabulary: {this} is not corporate
    - BOM to XOM: this.corporate = false;

Note: The vocabulary for the Action and Navigation vocabulary "**can**" be the same.

- The appropriate method is available in the Condition and Action section of the rules automatically, so the vocabulary can be the same for the Action and Navigation of an attribute.
- In the Vocabulary Outline Panel, it *only displays a single instance of the Vocabulary for both the "NOT" navigation and action.*

3.2 Vocabulary Overview [2.1.1 #5]

1. The "Rules Vocabulary" should closely match the language used by the ***Business*** to discuss the policies or requirements among themselves and with others in the organization.

    a. Sometimes groups have a language all their own that no one outside their bubble understands. This is **NOT** a good syntax to use for the rules.

    b. A COMMONLY accepted term understood by everyone within the organization is the better choice.

2. The Vocabulary should be designed to create a "complete syntax" in the chosen development language. A rule should be "Grammatically Correct" [within reason]

    a. Good

```
if
    'the itinerary' is corporate
    and there is at least one flown flight
            where the sales code of this flown flight is one of {RN, TV}
            and the flight number of this flown segment is at least 100,
then
    immediate reticketing is NOT possible;
```

    b. Bad

```
if
    'itin' is CORP
    and there is at least one flown flight
            where Sales Code of this flown segment is one of {RN, TV}
            and Flight Num of this flown segment is at least 100,
then
    set retickPoss to false;
```

3. Simplify physical and Virtual Methods, make them Business friendly

    a. Complex Vocabulary Implementation

```
then
   add SSR add event event to 'events of the current PNR'
   with code the code of 'the added SSR' ,
   type the fact type of 'the added SSR' ,
   origin the origin of the unflown segment ,
   destination the destination of the unflown segment ,
   flight number the operating flight number of the unflown segment ,
   departure date the gmt departure date time of the unflown segment
   and passenger the passenger name number of 'the current passenger' ;
```

    b. Enriched Vocabulary Implementation - Simplify complexity in the BOM2XOM Code

```
then
    add SSR add event to 'the events of the current PNR'
    using 'the added SSR',
    'the unflown segment',
    'the current passenger';
```

4. Use capitalization as appropriate within a sentence.

5. Enrich Vocabulary to avoid using "true" and "false".
    a. Use Enriched "is" and "is not" vocabulary in BAL rules.
    b. **Note:** Use "**true**" & "**false**" and not the enriched vocabulary in Decision Tables

3.3 Domain Usage [2.1.1 #6]

The focus on domains here is the implementation that provides a dropdown of values to enter for attributes during rule authoring.

3.3.1 When to use Dynamic Domains

- When the vocabulary to be used in the rule authoring is different than the actual Data Value used by the solution
    - Example: Coupon Status Type of "EXCH" using vocabulary "Exchanged"
- There is a "large" number of Domain values.
    - Management is easier in excel than the BOM editor in the Designer.
- The Domain values are volatile, and the desire is to have the Business manage the values.

3.3.2 Operational Characteristics of Dynamic Domains

- Domains are managed in the Decision Center, primarily by the Business.
- The Master, source, XLS for the domains is on the Decision Center

3.3.3 When to use Static Domains

- When the vocabulary and the actual data value are the same
- The number of elements is "small".
- The Domain values are static.

3.4 Ruleflows (High-Level Best Practices) [2.2.1 #7]

1. Avoid putting an Initial action code in the Start Node and Final action code in the End Node of a Ruleflow
    a. There is no visual indicator that there is code in the Start or End node, so it is possible that developers new to the project may miss that there is code located there.
2. Avoid assigning individual **Business Rules** to Rule Tasks
    a. Assign directories to Rule Tasks, that way when a new rule is added to a directory it is automatically included in the Ruleflow and "available" for execution. No additional steps necessary.
    b. Assigning individual **Operational Rules** to a Task may be appropriate and should be considered carefully before doing so.

3.5 Architectural Disconnect and Over Dependency on "Winner" pattern [2.3.1 #1, 2.3.1 #2]

3.5.1 Current Operational Pattern

This description is focused the operations within each Decision Table and NOT fully addressing the manual routing of the rule flow between the 3 Decision Tables.

If "CF Disposition" = "A"

1. Evaluate the CONDITION columns in Table "adds-table" starting on row 1
2. Continue to evaluate each row until all the conditions on that row evaluate to TRUE.
3. Execute the ACTIONS on that row:
    a. Set WS ID, Doc Type, and Rule Fired
4. Stop evaluating the remaining Rows.
5. IF the Work Assignment ID gets set to "RULE4" Evaluation the CONDITION columns in Table "Alpha Plan Prefix" starting on row 1
    a. Else, DONE
6. Continue to evaluate each row until all the conditions on that row evaluate to TRUE.
7. Execute the ACTIONS on that row:
    a. Set WS ID, Doc Type, and Rule Fired
8. If WA ID still set to "RULE4" set the WA ID to "BLU16"

3.5.2 Recommended Implementation of Winner Pattern

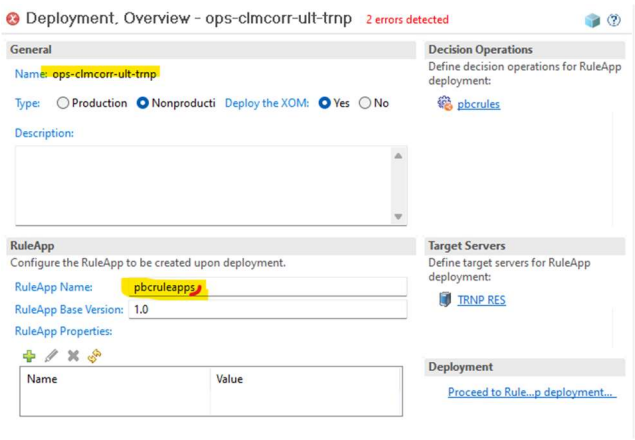
This is a Best Practices approach to using the Winner pattern, if pattern is appropriate.

1. Determine all the possible Work Assignments this Work Item is "Eligible" to be assigned to based on the criteria defined to assign a specific Work Item to that Assignment.
    a. Could be assigned to NONE or any number of Work Assignments
2. Implement a Winning Pattern to select the "Correct" (highest ranked) Work Assignment
3. Return the Winning result.

4 Questions

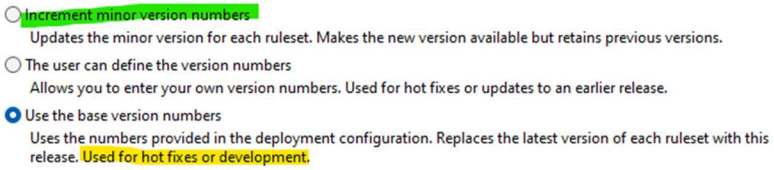
These are the questions I have. Based on the answers I may have recommendations. I'm asking these questions because what I see being done is questionable depending on "why" it is being done.

1. Why is there no XOM or BOM created for this solution?
    a. All data is sent in as individual String parameters and defined as Global Variables
    b. This "works" but significantly limits the usage of functionality/capabilities of ODM.
2. Why does this Deployment configuration use "pbcruleapps" and all others use "pbcruleapp"



a.

3. Why not using Automated Deployment versioning?



a.

4. There are currently 262 unique Work Assignment IDs implemented in the rules.
    a. Are they all still valid IDs?

5 Recommendations

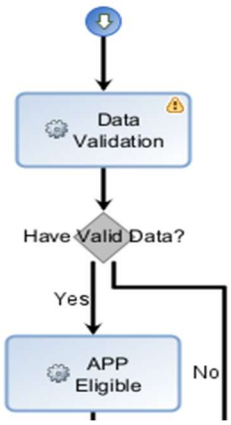
The following is a list of larger Architectural or Design Pattern recommendations for changes/enhancements, not in any particular order, that would improve the following:

- Rules Manageability
- Solution Extensibility
- Governance issues

5.1 Data Validation Pattern

5.1.1 Upfront Data Check

Perform data validation early to prevent/minimize execution of Rules with invalid or incomplete data.



5.1.2 Dedicated Error/Message Object

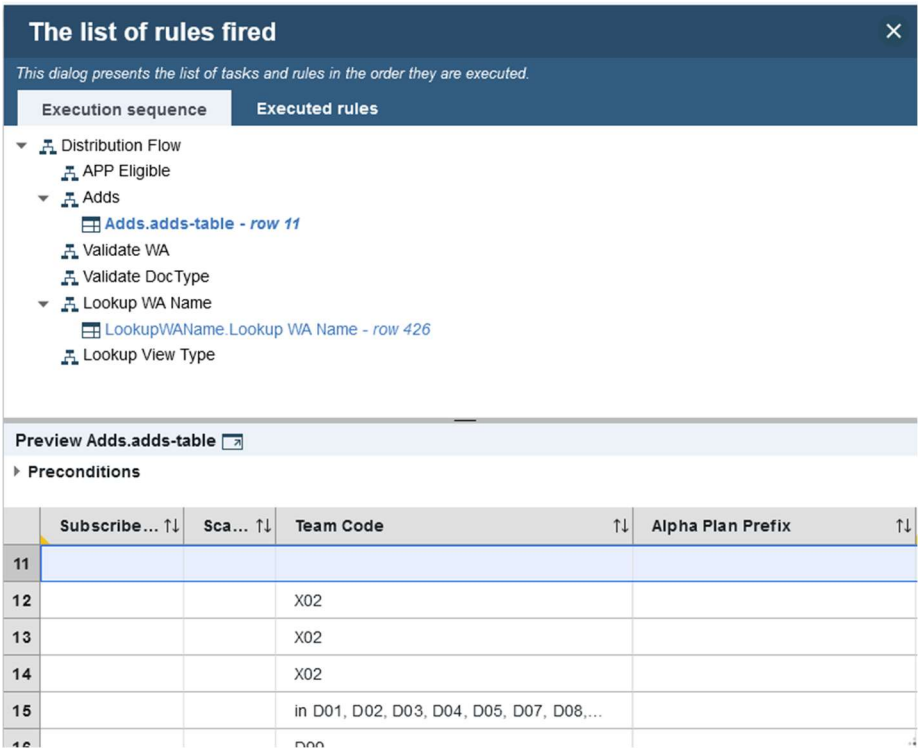
Use a dedicated Data Validation response object that collects all issues and contains a flag that can be checked independent of the standard response object.

Business Name	the data validation results		Returns error or other types of operational messages	
XOM/BOM Name	DataValidation			
Generate Auto Var and Insert				
Attributes				
Business/VOC	**XOM/IRL**	**DataType**	**Domain/Allowed Values**	**Notes**
{this} has failed	failed	boolean		default to FALSE
	messages	List<String>		free text
Methods				
Business/VOC	**XOM/IRL**	**Parameters**	**Return**	**Notes**
	Virtuals			
{this} has not failed	hasNotFailed		boolean (!this.failed)	
add message {0} to {this}	addMessage	String	void	adds a String to this.messages
{this} has failed	setHasFailed			set this.failed to TRUE
add error message {0} to {this}	addErrorMessage	String	void	adds a String to this.messages and set flag to fail

5.2 Use Built-in Rule Tracing Capabilities

5.2.1 Decision Runner Testing

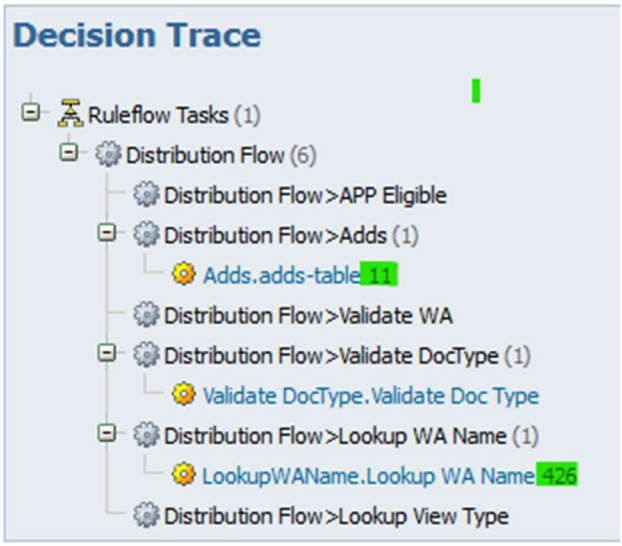
https://www.ibm.com/docs/en/odm/8.11.1?topic=rulesets-overview-testing-simulation



5.2.2 OOTB Decision Warehouse {Deprecated}

https://www.ibm.com/docs/en/odm/8.11.1?topic=warehouse-default-use-decision

For Tables row highlighted in green



5.2.3 Inline trace

https://www.ibm.com/docs/en/odm/8.11.1?topic=services-htds-decision-trace-filters

Trace Snippet. For Tables row highlighted in green


```
"rulesFired": {
  "ruleInformation": [
   {
    "name": "Adds.adds$45$table_10",
    "businessName": "Adds.adds-table 11",
            :
            :
   },
   {
    "name": "LookupWAName.Lookup_WA_Name_425",
    "businessName": "LookupWAName.Lookup WA Name 426",
            :
            :

   },
   {
    "name": "Validate_DocType.Validate_Doc_Type",
    "businessName": "Validate DocType.Validate Doc Type",
```

5.3 General List of Changes/Remediations

1. Create a Java Based XOM (with associated BOM) to support both the Operational & Testing Frameworks. Depending on integration requirements (current/future) any of these patterns would be viable improvements over the current approach.
    a. Simple "request" object(s) and Response Objects
        i. Could still have multiple Input if separate INPUT and IN/OUT parameters are required.
    b. Dynamic Model consisting of 1 or more Key-Value pair objects.
        i. Can be used for any of the parameters (IN, IN/OUT, OUT)
        ii. This is a good pattern if you want to add/remove input data without having to change the integration with the calling application.
    c. Combination of Both a & b above
2. Evaluate the Data to find patterns of date that can classify or identify a work item as something human understandable (a Concept)
    a. Rules can then be written to evaluate and act upon a "classified" work item instead of repeating the same data evaluation over and over again in multiple rules.
    b. This may or may not be feasible, I would need to discuss this in detail with the Subject Matter expert who Defines the routing rules to know if there are any to implement.
3. Understand/Document exactly what the business policies are that determine WHY a particular Work Assignment is selected.
    a. Focus on Work Assignment
        i. What criteria determines a "specific" work assignment should be selected?
    b. IF based on the selection criteria for a specific Work Assignment more than one Work Assignment is "eligible" to be selected, pick the best one.
        i. This is where the "Winner" pattern would come into play.
    c. NOTE: This will only work if there is quantifiable logic that says "this WA should be assigned over that WA"

6 Summary

6.1 Open Issues

***Critical Issues* – 2**

***High Risk* – 2**

***Medium Risk* – 5**

***Low Risk* – 4**

7 Appendix

7.1 Definitions

Decision Service – has multiple meanings depending on context:

1. ODM Solution Management construct as identified in Decision Center consisting of a Single "Main" project and 0-many "Standard" projects. [Project Structure]
2. A deployed ruleset or Endpoint that is exposed as an "executable" web service [Decision Operation]

Decision Solution – One or many related ***Decision Service Constructs*** [Decision Operations] within a single **Decision Service** [Project Structure] that may, or may not, share common rule artifacts.

DGF – IBM's ODM **Decision Governance Framework**, a ready-to-use prescriptive method for change management and governance.

Business Rules – Rules that directly implement and support Business policies. These are usually written and managed by the Business.

Operational Rules – Rules not directly responsible for implementing Business Policies (aka Business Requirements). Examples are rules that: build concepts, reset control flags, initialize values, etc. These are usually written and managed by IT.

Concepts – Physical or virtual Attributes or Objects created from the context of the input parameters, most often created with Rules. Intermediate data used to simplify, or make possible, the writing of Rules that implement a Business Policy

Change Activity – A quantifiable unit of work to be performed by 1 or more individuals resulting in rules being changed.

- **DGF**: A physical construct created within a Decision Service to isolate rules development and testing until it is complete and ultimately merged back into the *in-development* **Release**
- **Branch Based Governance:** Performed on one of the following depending on *Methodology* and *Tier*:
    - Directly on the in-development **Release Branch**
    - In a **sub-branch** of the *in-development* **Release Branch**
    - Directly on the **MAIN Branch**

Validation Activity – A conceptual unit of work to be performed by 1 or more individuals that tests, validates the rule changes.

- **DGF**: A physical construct created within a Decision Service that must be "Completed" in order for the associated **Release** to be "Completed"
- **Branch Based Governance**: Performance of Functional or Regression testing on appropriate depending on *Methodology* and *Tier*:

- Directly on the in-development **Release Branch**
- In a **sub-branch** of the *in-development* **Release Branch**
- Directly on the **MAIN Branch**

Release – A quantifiable designation of a set of requirements once completed will be placed into Production

- **DGF**: A physical construct created within a Decision Service that must be "Completed" for new rules to be deployed
- **Branch Based Governance (Release Branch):** A **Branch** of the Decision Service

Current Release:

- **DGF**: The *Completed* release that is currently deployed to and in (or planned to be in) Production
- **Branch Based Governance (Current Release Branch)** – Depending on whether **Cooldown Period** is to be used:
  - The **MAIN Branch** of the Decision Service
  - The **Candidate Release Branch** of the Decision Service

***In-Progress* Release –** A **Release** that is currently being changed. Actively under Change Management.

***Candidate* Release Branch –** The Tier 2 or Tier 3 Branch that is identified to be Deployed **next**

Operational Framework – The Decision Operation(s) and associated ruleflows, functions, XOM, BOM, variables, and Operational Rules that support the Business Rules and implement the Production Decision Operations. May also be used for UAT, Functional & Regression Testing if specialized Testing Frameworks are unwarranted.

Enhanced Testing Framework – The Decision Operation(s) and associated ruleflows, functions, XOM, BOM, variables, and operational rules that supports the full testing of single or multiple "internal" decisions using the Decision Runner capabilities. Used for Functional & Regression Testing

Concept Testing Framework – The Decision Operation(s) and associated ruleflows, functions, XOM, BOM, variables, and operational rules that supports the full testing of single or multiple Concepts using the Decision Runner capabilities.

Structural Changes – Changes to the Operational Framework that could include changes to: Ruleflows, Functions, Variable Sets, Parameters, etc.

Cooldown Period – The time between Deploying the **Candidate Release Branch** and merging the **Candidate Release Branch** with the **Main Branch** when using **Branch Based Governance**

7.2 Concepts In Detail

7.2.1 Broadest Definition

A "Concept" is a human understandable entity with the following characteristics:

- References a measurable entity.

- An order has 3 **Back Ordered Items**
- An order has 2 **Shipped Items**
- A segment's origin belongs to two **Regions.**
- A segment can have multiple **Profiles.**

- Has defining attributes.
  - A trip has an **Origin.**
  - An active flight has a **Departure date/time.**
  - A profile contains a **Value.**

7.2.2 Limited Definition (for this BP Document)

For this document a "Concept" will be defined as:

- Simple Concept – Example: Customer Profile
  - An attribute defined by Business Rules
  - A list of Strings
- Complex Concept – Example: Trips
  - An entity defined by Business Rules and/or other Rule Artifacts
  - All the pertinent information that makes up a Leg of a Domestic flight itinerary