**BRE TOOL Solution and Requirements**

**Business Rules Extraction**

**What is business rules extraction?**

Business rules extraction refers to the process of harvesting, and in some instances reverse-engineering, legacy code to understand underlying business methods. It helps catalog embedded business logic and dependencies to improve the accuracy of IT systems, and documents technical rules so they can be validated and preserved.

**What are the business benefits of business rules extraction?**

With business rules extraction, businesses can:

* Improve understanding of legacy applications
* Document business logic in an easy-to-understand manner
* Reduce human errors
* Help automate modernization efforts.

**BRE Tool Significant:**

* It is a process of getting requirement specifications and functions of a product from an analysis of its code.
* It builds a program database and generates information from this.
* The purpose of BRE Tool is to facilitate the maintenance work by improving the understandability of a system and to produce the necessary documents for a legacy system.
* Business rules are operational rules that business organizations follow to perform various activities. a tailored solution approach to the business rule extraction problem, which combines variable classifications, program slicing, and hierarchical abstraction among other maintenance techniques.

**Requirements:**

**Requirements:**

* PyCharm – version 9.0 (or above) for both Python and PySpark Programming.
* Python – version 3.7

For PySpark Analytical code – Log-Enabler.py, Modified-logenabler.py

* We must add the spark library to the Project Structure

File -> Settings -> Project Structure -> Add Content Root -> path for Spark\python\lib

File -> Settings -> Project Structure -> Add Content Root -> path for Spark\python\pyspark

* Install required packages like pandas, matplotlib, xlrd (excel reader), xlwt (excel write), seaborn.

**Expectations from the code:**

* Rigth now the code can give the output of variable for
* Arthmetic operations like ADD, SUB, MUL, DIV and COMPUTE.
* File operations like READ, WRITE,OPEN, CLOSE, UPDATE, REWRITE and DELETE keywords.
* It can give the data types of the variables that are declared in any section like File section (FD), Working-Storage Section (WS), Declare Section (DB2). Keywords like
* PIC, COMP, VALUES
* For DB2 Code it will gives the Exec Commands between EXEC SQL and END-EXEC.
* Core Logic
* Conditional Verbs

1. <IF-ELSE –ENDIF logic>> ++ Nested Conditional
2. EVALUATE

* A mechanism to identify core paragraphs << Set out a bypass logic for default paragraphs>>
* Reserved Words
* SEARCH ALL, INDEXED BY, INSPECT

There are several types of reserved words in COBOL.

* Keywords: they appear in uppercase. ADD, READ, etc.
* Optional Words: they improve readability. GIVING, AFTER, etc.
* Figurative constants: they refer to constant values. ZEROES, SPACES, etc.

**Identification Division** – Full features

* PROGRAM-ID ✓
* AUTHOR ✓
* INSTALLATION ✓
* DATE WRITTEN ✓
* DATE COMPILED ✓
* SECURITY ✓

**Environment Division** – Full features

* File Name ✓
* Organization ✓
* Access mode ✓
* Key – Alternate Key (KSDS) ✓
* File – status ✓

**Data Division** – Full features

* Variable Name ✓
* Data type ✓
* Identifying COMP ✓
* Key word like VALUE, INDEX, OCCURS

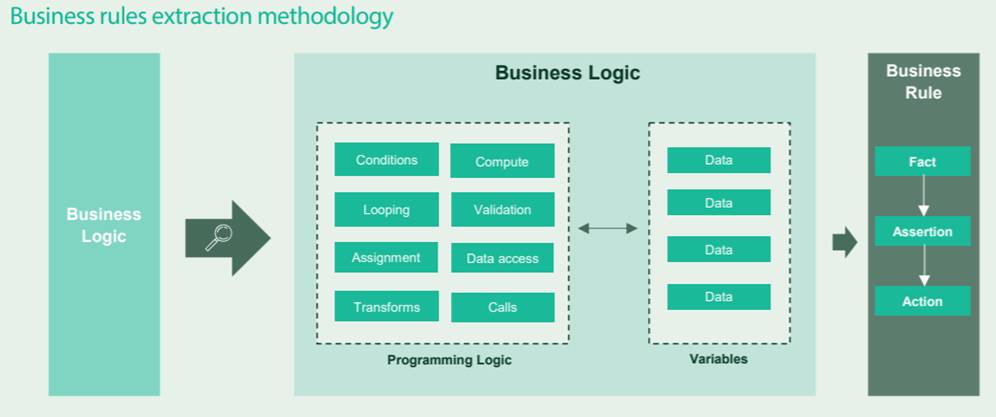
**Procedure Division** – Full features

* File operations ✓
* Arithmetic operations ✓
* Control blocks (if-else, evaluate)
* Perform Paragraph

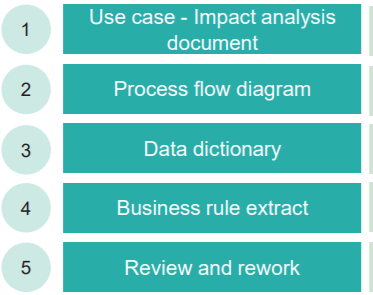
**EXEC commands in DB2** – Full features

* Categorizing for both working storage and procedure division ✓
* Include and Declare commands for Working storage ✓
* SQL quires for procedure division ✓

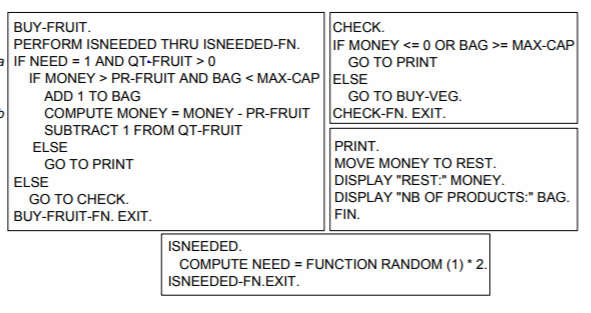
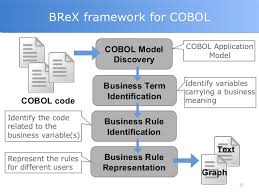
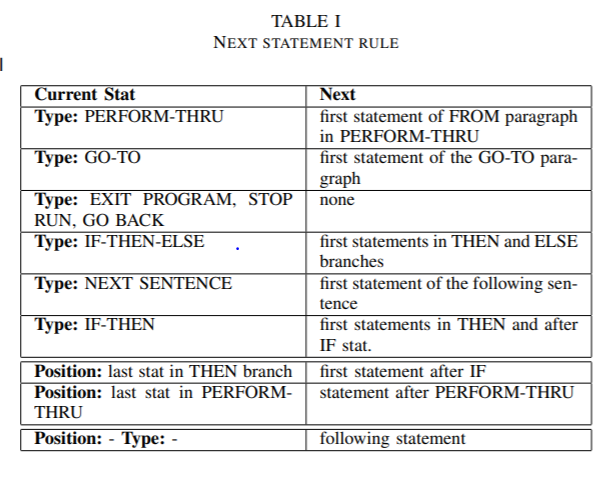
**Flow diagram of Solution**



**Project Stages**



**Advanced Features & Tokens**

* Control blocks like (If-else, Evaluate) and Perform paragraphs for core solution in procedure division.
* Keywords like INDEXED BY, OCCURS, VALUE in data division.
* From organization, access mode and remaining features to get what kind file we are using from environment division

**References:**

* <https://modeling-languages.com/wp-content/uploads/2013/09/paper-cameraready.pdf>
* <https://www.researchgate.net/publication/334946627_From_COBOL_to_Business_Rules-Extracting_Business_Rules_from_Legacy_Code>
* <https://www.infosys.com/modernization/documents/business-rules-extraction.pdf>