



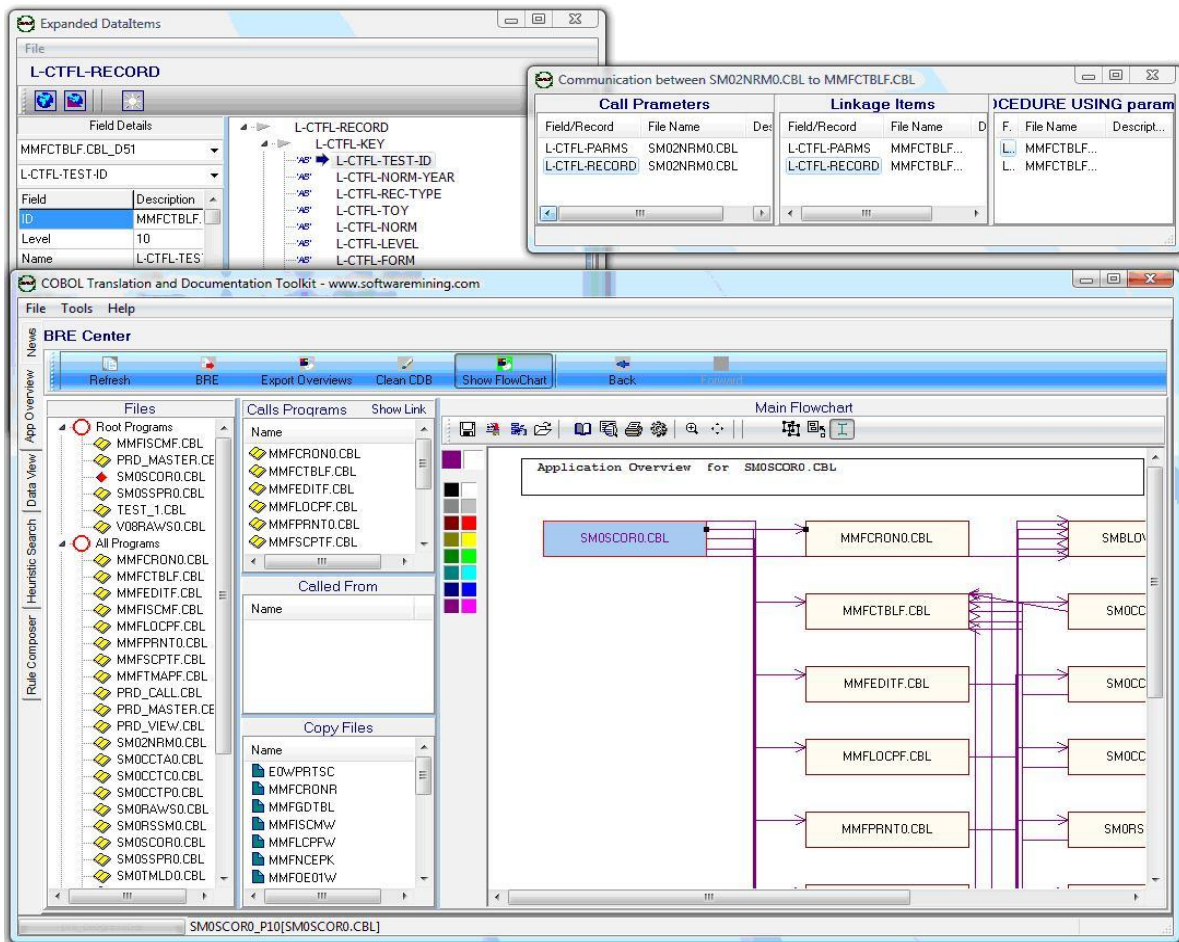
Business Rule Extraction

Business Rule Extraction is the process of isolating the code segments which are directly related to business processes. For example, how the system calculates a "discount" or "interest" on an account. In addition to Business-Related code.

The BRE Toolkit provides a set of utilities to:

- Application flowcharts and Code Inventory
- Program level flowcharts
- Heuristics assisted identification of important “rule” variables
- English Rules – Translation of COBOL rules from COBOL to Pseudo-English
- Java and C# Rules: translated program components can be used directly in modernized application.
- Data Dependencies - Identification and visual representation
- HTML Documentation - Business Rules, program level and application level documentation in HTML

BRE Toolkit uses sophisticated search algorithms to sieve through huge amount of code and isolate the *business related* code. It can hence significantly increase productivity, reduce the cost and increase accuracy of Knowledge Extraction from Legacy COBOL code.



Extracting Definition Rules

Definition rule can be defined as statements which provide a definition of a term (variable-of-interest) and a quantitative relationship between it and other variables. The terms in question are those responsible for achieving a particular business objective. e.g.

“order value is calculated by multiplying price of an item by quantity ordered”.

How does it work Given a variable-of-interest - (e.g. ORDER-VALUE), the BreToolkit can automatically find all the relevant statements and which goes into definition of this rule.

Take the following sample of code:

```
01      MOVE 123 TO ITEM-PRICE.
02      MOVE 10 TO I.
03      COMPUTE TEMP = 1 * J.
04      MOVE 40 TO J.
```



```
05      ACCEPT ORD-QTY.  
06      COMPUTE TEMP = ITEM-PRICE * ORD-QTY.  
07      MOVE TEMP TO ORD-VAL .
```

Lets say we are looking for Rules concerning calculation of Order Value. Searching for Definition-Rules on variable "ORD-VAL (representing Order Value) will result in identification of the following lines:

```
01      MOVE 123 TO ITEM-PRICE.  
02      MOVE 10 TO I.  
03      COMPUTE TEMP = 1 * J.  
04      MOVE 40 TO J.  
05      ACCEPT ORD-QTY.  
06      COMPUTE TEMP = ITEM-PRICE * ORD-QTY.  
07      MOVE TEMP TO ORD-VAL .
```

Considering typical COBOL programs containing thousands of lines of code, such automatic process can save significant amount of time on any business-rule extraction and re-documentation project.

The rule-definition facility traces application flow-logic backwards through all the possible routes, to identify all the most relevant operations on all variables involved in setting the value of Variable-of-Interest (ORD-VAL in above example).

Next, using the in-built Natural-Language translation facility, the above statements can be turned automatically into:

```
Item Price = 123  
read Order Quantity. (from screen)  
Temporary Variable = Item Price * Order Quantity.  
Order Value = Temporary Variable
```

Alternatively, the business rules can be saved as COBOL, Java or C# code segments, ready to use in re-factoring exiting applications, or re-development in new languages.

Where to start? How to find important Variables

The Business Rule Extraction provides a mean of identifying the operations and relationships associated with "Variables-Of-Interest" (VOI).

The identification of VOI is not a trivial task within large and complex legacy system. The original application developers usually can provide a good starting point, the subject matter expert will be also be invaluable. However, often such resources have long left the



organization, and other approaches will be required for identification of VOIs.

The "Heuristic Search" screen facilitates identification of VOIs, by providing "usage patterns" searches.

For example, a simple "usage pattern" is to identify variables used in mathematic operations (COMPUTE, ADD, SUBTRACT, ...) as the result of (financial) calculations are often important.

A slightly more complex pattern would be to identify variables involved in Mathematics, displayed on screen (ACCEPT / DISPLAY / CICS SEND / RECEIVE), and Read or Stored to files/database. Another pattern may include variables occurring in Mathematics and Reporting. The "Heuristic Search" screen facilitates such searches.

Program Name	Statement
V082NRM0.CBL	COMPUTE SAVE-AGE-MONTHS = (L-SAT10-CRON-TOT-MON / 3) + 1.
V082NRM0.CBL	COMPUTE WS-DOB-MONTHS = (L-SAT10-DOB-YR * 12) + L-SAT10-DOB-MON.
TEST_1.CBL	COMPUTE TEMP = ITEM-PRICE * ORD-QTY.
TEST_1.CBL	COMPUTE I = J * K.
TEST_1.CBL	COMPUTE TEMP = 1 * J.
PRD_CALL.CBL	COMPUTE WS-IDX = 1.
PRD_CALL.CBL	COMPUTE ORD-VAL = ORD-VAL * WS-DISCT-PRCNT
PRD_CALL.CBL	COMPUTE ORD-VAL = ORD-VAL + STD-LCL-DELV-CHRG
PRD_CALL.CBL	COMPUTE ORD-VAL = ORD-VAL + NXT-DAY-LCL-DELV-CHRG
PRD_CALL.CBL	COMPUTE ORD-VAL = ORD-VAL + DWRSEA-DELV-CHRG
PRD_CALL.CBL	COMPUTE CLNT-BALANCE = CLNT-BALANCE - ORD-VAL
PRD_CALL.CBL	COMPUTE WS-IDX = WS-IDX + 1.
PRD_CALL.CBL	COMPUTE ORD-VAL = ORD-VAL + (PROD-QTY * PROD-PRICE)

Number of datasets : 278