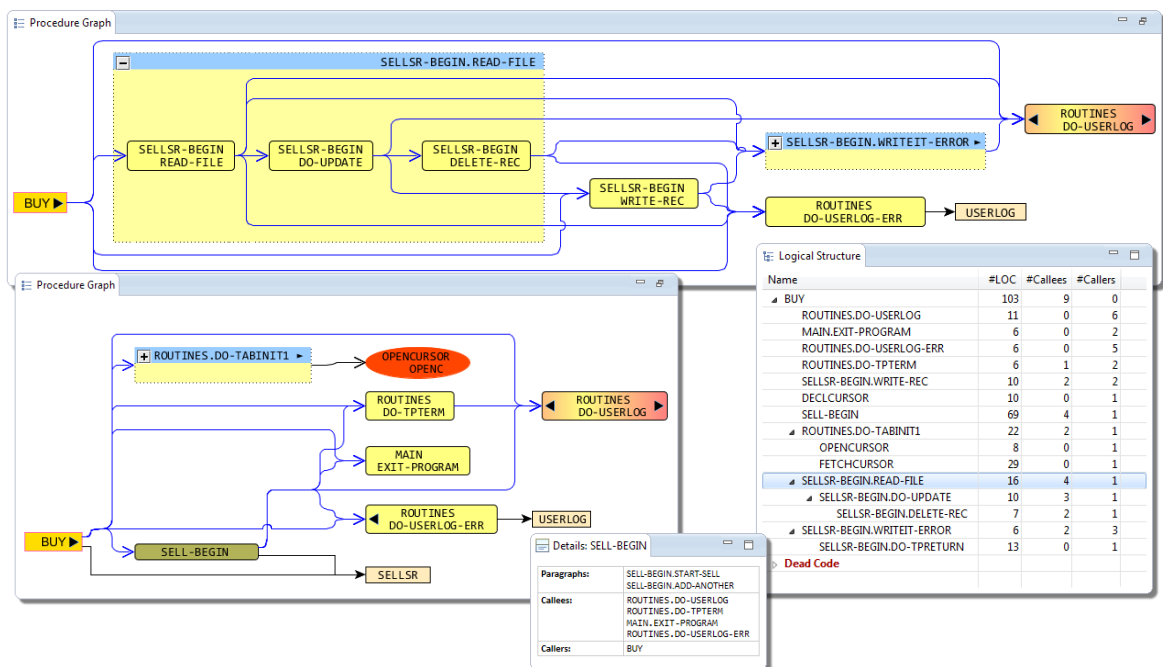


Prozeduren und Funktionen

Prozeduren und Funktionen gibt es in COBOL nicht, jedenfalls nicht explizit (in älteren COBOL-Versionen). Stattdessen gibt es in der Procedure Division eine Sammlung von Sections und darin enthaltene Paragraphen, die mal einzeln und mal gemeinsam ausgeführt werden und sich gegenseitig aufrufen. Variablen sind grundsätzlich global definiert, auch dann, wenn sie nur lokal verwendet werden. Diese Strukturierung ist prinzipbedingt, erschwert jedoch das Verstehen und die Modernisierung der Anwendung.

AMELIO Logic Discovery bestimmt Prozeduren und logische Gruppierungen, sowie deren Schnittstellen, und den Scope von Variablen anhand von Aufrufbeziehungen und Verwendungen. Die Ergebnisse werden übersichtlich darstellt. Auf diese Weise wird das Verstehen der Anwendung erleichtert und Vorschläge für ein Refactoring gemacht.



Prozeduren

In COBOL bilden Sections und Paragraphen implizit Prozeduren. AMELIO Logic Discovery ermittelt diese Prozeduren und macht sie sichtbar. Die Prozeduren werden jeweils mit ihrem Kontext angezeigt.

Composites

AMELIO Logic Discovery bestimmt, welche der Prozeduren logisch gruppiert werden können. Diese logischen Gruppierungen, die sog. Composites, werden im Gesamtkontext der Anwendung dargestellt.

Aufrufbeziehungen

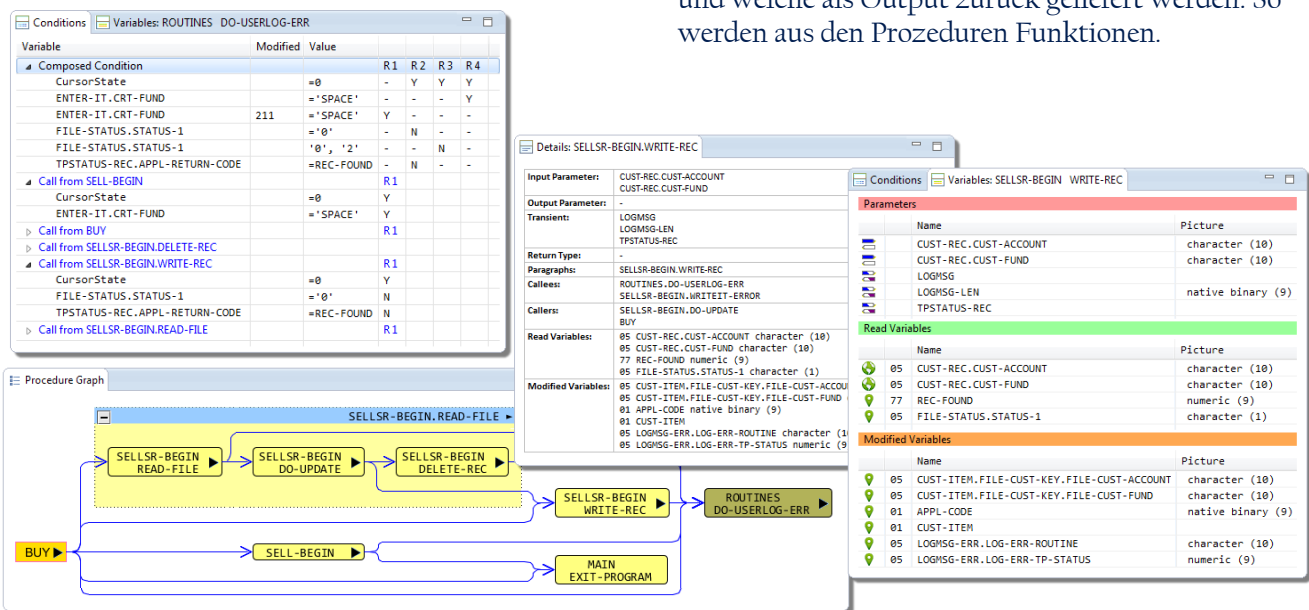
Für die gefundenen Prozeduren und Composites analysiert AMELIO Logic Discovery die Aufrufbeziehungen, die zwischen diesen Einheiten bestehen.

Außerdem wird ermittelt, welche Bedingungen erfüllt sein müssen, damit der Aufruf einer Prozedur oder eines Composites erfolgt. Die Analyse erfolgt sowohl ausgehend von der aufrufenden Prozedur als auch vom Programmstart.

Schnittstellen und lokale und globale Variablen

AMELIO Logic Discovery bestimmt für jede Variable deren Scope und ob die Variable tatsächlich global verwendet wird oder ob es sich eigentlich um eine lokale Variable handelt.

Mit dem Ergebnis der Scope-Analyse stellt AMELIO Logic Discovery für jede Prozedur und jedes Composite fest, welche Parameter als Input erwartet und welche als Output zurück geliefert werden. So werden aus den Prozeduren Funktionen.



The screenshot displays the AMELIO Logic Discovery interface with several panels:

- Variables: ROUTINES DO-USERLOG-ERR**: A table showing variable states across different routines.

Variable	Modified	Value	R1	R2	R3	R4
Composed Condition						
CursorState		=0	-	Y	Y	Y
ENTER-IT.CRT-FUND		= 'SPACE'	-	-	-	Y
ENTER-IT.CRT-FUND	211	= 'SPACE'	Y	-	-	-
FILE-STATUS.STATUS-1		= '0'	-	N	-	-
FILE-STATUS.STATUS-1		'0', '2'	-	-	N	-
TPSTATUS-REC.APPL-RETURN-CODE		=REC-FOUND	-	N	-	-
- Details: SELLSR-BEGIN.WRITE-REC**: A panel showing input/output parameters, return types, and callers for a specific routine.
- Conditions**: A panel showing parameters, read variables, and modified variables for the selected routine.
- Procedure Graph**: A flowchart showing the sequence of routines: BUY → SELL-BEGIN → SELLSR-BEGIN.READ-FILE → SELLSR-BEGIN.DO-UPDATE → SELLSR-BEGIN.DELETE-REC → SELLSR-BEGIN.WRITE-REC → ROUTINES.DO-USERLOG-ERR → MAIN.EXIT-PROGRAM.

Refactoring-Vorschläge

Die Composites stellen bei einem Refactoring Kandidaten für Unterprogramme dar. Da AMELIO Logic Discovery auch die Schnittstelle der Composites analysiert und ermittelt, welche globalen Variablen verwendet werden, stehen alle benötigten Informationen zur Verfügung.



Delta Software Technology GmbH
 Eichenweg 16, 57392 Schmallenberg
 phone +49 2972 9719-0
 e-mail info@delta-software.com
www.delta-software.com

AMELIO Logic Discovery

COBOL- und PL/I-Anwendungen verstehen:
 Kosten und Risiken für Wartung, Modernisierung und Neu-Implementierung senken.

www.delta-software.com/amld

