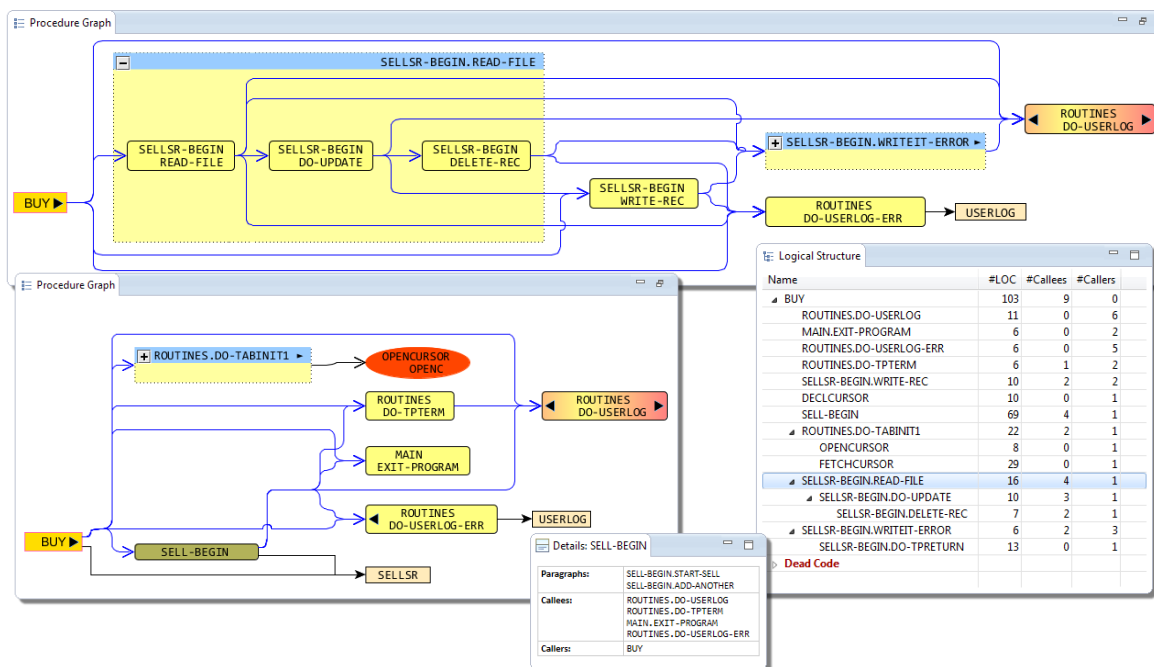


## Prozeduren und Funktionen

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

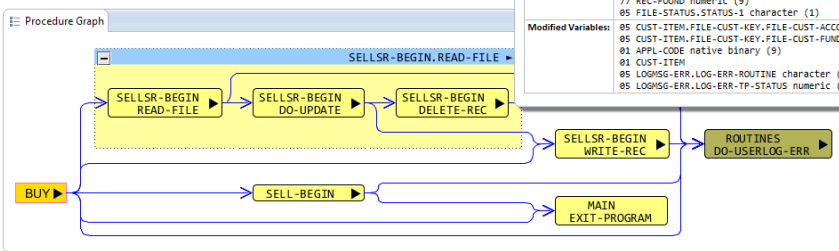
AMELIO Logic Discovery bestimmt, welche der Prozeduren logisch gruppiert werden können. Diese logischen Gruppierungen, die sog. Composites, werden im Gesamtkontext der Anwendung dargestellt.

### 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.

Variable	Modified	Value	R1	R2	R3	R4
<b>Composed Condition</b>						
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	-	-
<b>Call from SELL-BEGIN</b>						
CursorState		=0				R1
ENTER-IT.CRT-FUND		= 'SPACE'				Y
<b>Call from BUY</b>						
CursorState		=0				R1
ENTER-IT.CRT-FUND		= 'SPACE'				Y
<b>Call from SELLSR-BEGIN.DELETE-REC</b>						
CursorState		=0				R1
FILE-STATUS.STATUS-1		= '0'				N
TPSTATUS-REC.APPL-RETURN-CODE		=REC-FOUND				N
<b>Call from SELLSR-BEGIN.WRITE-REC</b>						
CursorState		=0				R1
FILE-STATUS.STATUS-1		= '0'				N
TPSTATUS-REC.APPL-RETURN-CODE		=REC-FOUND				N
<b>Call from SELLSR-BEGIN.READ-FILE</b>						
CursorState		=0				R1
ENTER-IT.CRT-FUND		= 'SPACE'				Y



Input Parameter:	Value
CUST-REC.CUST-ACCOUNT	CUST-REC.CUST-FUND
Output Parameter:	-
Transient:	LOGMSG LOGMSG-LEN TPSTATUS-REC
Return Type:	-
Paragraphs:	SELLSR-BEGIN.WRITE-REC
Callers:	ROUTINES.DO-USERLOG-ERR SELLSR-BEGIN.WRITE-REC
Callers:	SELLSR-BEGIN.DO-UPDATE BUY
Read Variables:	05 CUST-REC.CUST-ACCOUNT character (10) 05 CUST-REC.CUST-FUND character (10) 77 REC-FOUND numeric (9) 05 FILE-STATUS.STATUS-1 character (1)
Modified Variables:	05 CUST-ITEM.FILE-CUST-KEY.FILE-CUST-ACCOU character (10) 05 CUST-ITEM.FILE-CUST-KEY.FILE-CUST-FUND character (10) 01 APPL-CODE native binary (9) 01 CUST-ITEM 05 LOGMSG-ERR.LOG-ERR-ROUTINE character (1) 05 LOGMSG-ERR.LOG-ERR-TP-STATUS numeric (9)

Name	Picture
CUST-REC.CUST-ACCOUNT	character (10)
CUST-REC.CUST-FUND	character (10)
LOGMSG	character (10)
LOGMSG-LEN	numeric (9)
TPSTATUS-REC	native binary (9)
<b>Read Variables</b>	
Name	Picture
05 CUST-REC.CUST-ACCOUNT	character (10)
05 CUST-REC.CUST-FUND	character (10)
77 REC-FOUND	numeric (9)
05 FILE-STATUS.STATUS-1	character (1)
<b>Modified Variables</b>	
Name	Picture
05 CUST-ITEM.FILE-CUST-KEY.FILE-CUST-ACCOUNT	character (10)
05 CUST-ITEM.FILE-CUST-KEY.FILE-CUST-FUND	character (10)
01 APPL-CODE	native binary (9)
01 CUST-ITEM	character (1)
05 LOGMSG-ERR.LOG-ERR-ROUTINE	character (10)
05 LOGMSG-ERR.LOG-ERR-TP-STATUS	numeric (9)

### 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.

### 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](mailto:info@delta-software.com)  
[www.delta-software.com](http://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](http://www.delta-software.com/amld)

