

**AMELIO Logic Discovery** ist ein innovatives Analyse-Werkzeug, das hilft, PL/I-Anwendungen besser zu verstehen und die relevante Funktionalität zu extrahieren. Es analysiert auch große und komplexe Anwendungen und leitet aus Implementierungsdetails Erkenntnisse über die Funktionen der Anwendung ab. Die Anwendungslogik wird von der technischen Infrastruktur getrennt. Analyseergebnisse werden modellorientiert dargestellt, so dass sie als sprachneutrale Dokumentation zur Verfügung stehen. Mit dem so gewonnenen Wissen sind eine effiziente Neuentwicklung, Modernisierung, Weiterentwicklung oder Wartung der Anwendung mit minimalisiertem Risiko möglich.

The screenshot displays the AMELIO Logic Discovery interface with several key components:

- Procedure Graph:** A central flowchart showing the execution flow from 'main' through 'input', 'output', and 'varsscopes' to various SQL operations (SELECT, DELETE, INSERT) and control structures (then, if).
- Control Flow varsscopes:** A detailed flowchart showing decision points like 'if (var1 = 10)' and 'if (var1 = 1 to 10)', leading to different execution paths.
- Coverage Report of Include IM010:** A table showing the usage of primary sources across different lines of code.
- Dead/Unused Data Definitions:** A table listing variables and their usage status.
- Database Access:** A table showing SQL operations performed by different routines.
- Source Code:** A window displaying the original PL/I code being analyzed.
- Conditions and Variables:** Panels showing the state of variables like 'BISHOP' and 'varsscopes' during execution.

## Analyse der Anwendungslogik

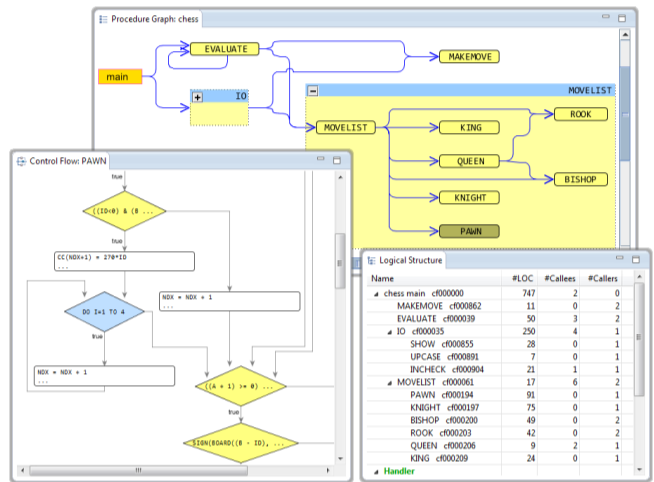
Zum Verständnis der Anwendungslogik von PL/I-Anwendungen bietet **AMELIO Logic Discovery** Analysen mit unterschiedlichen Zielsetzungen und Granularitäten.

### Anwendungsarchitektur

- Analysiert, aus welchen Programmen und Unterprogrammen sich eine Anwendung zusammensetzt und ermittelt ihre Aufrufbeziehungen, inklusive Aufrufbedingungen und Schnittstelleninformationen.
- Die Analyseergebnisse werden graphisch dargestellt. Zusätzliche Details, wie Aufrufbedingungen und Schnittstellendetails, werden textuell eingeblendet.

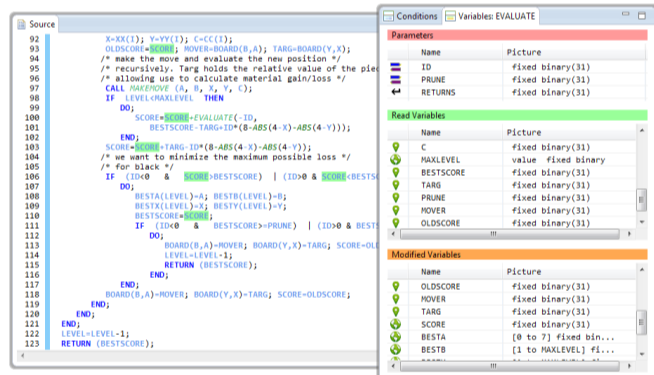
### Prozeduranalyse

- Ermittelt im Programm enthaltene Prozeduren und ihre Aufrufbeziehungen.
- Darstellung der Prozeduren und ihrer Abhängigkeiten als Graphen. Zusätzlich gibt es eine Baumdarstellung der gefundenen Prozeduren.
- Zu jeder Prozedur werden Details wie deren Kontrollfluss in Form eines Activity-Diagrams, ihre Parameter und die gelesenen und modifizierten Variablen dargestellt.

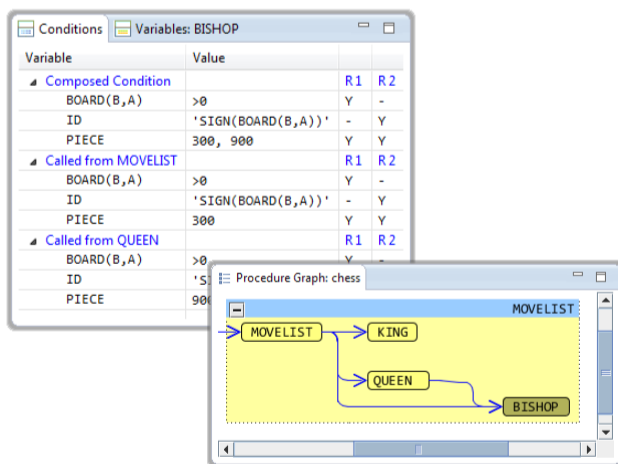


### Erkennung von Composites

- Prozeduren, die logische Einheiten bilden, werden in Composites zusammengefasst.
- Einbettung der Composites und ihrer Abhängigkeiten in den Prozedurgraph. Zusätzlich ist eine Baumdarstellung der logischen Programmstruktur inklusive der Composites und der darin enthaltenen Prozeduren verfügbar.

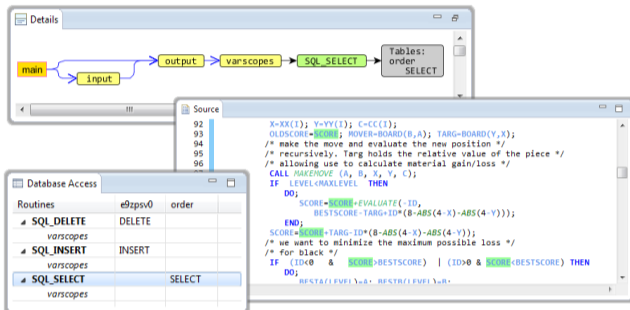


### Bedingungsanalyse



- Ermittlung der Bedingungen, unter denen ein Unterprogramm ausgehend vom Hauptprogramm aufgerufen wird. Ebenso Analyse der Bedingungen, unter denen ein Unterprogramm ein anderes aufruft.
- Unter welcher aggregierten Bedingung ausgehend vom Programmstart wird eine Prozedur aufgerufen, sowie unter welcher aggregierten Bedingung ruft eine Prozedur eine andere Prozedur auf?
- Darstellung der einzelnen und zusammengefassten Bedingungen in Form von Bedingunstabellen.

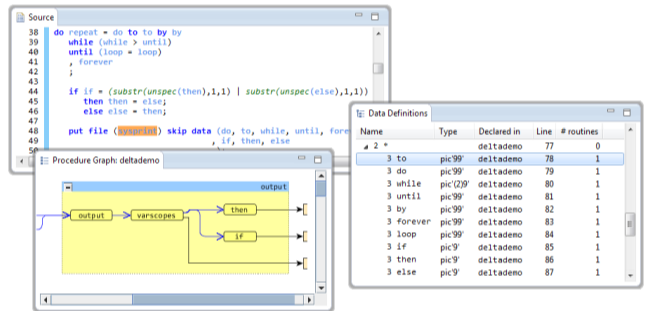
## I/O-Analysen



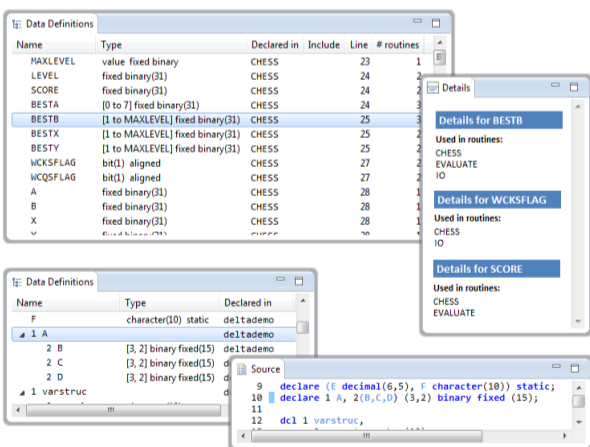
- Analysiert, auf welche Tabellen und Files innerhalb eines Programms zugegriffen wird.
- Welche Prozedur greift lesend oder modifizierend auf eine bestimmte Tabelle(ns palte) oder ein File zu und wer ruft wiederum diese Prozedur auf?
- Erkennung potenzieller CRUD-Prozeduren.

## Schlüsselwort, Funktion oder Variable

- Analyse, ob es sich bei einem Statement um ein Schlüsselwort, eine Funktion oder um eine Variable handelt.
- Ergebnisse werden graphisch dargestellt, in Listen ausgegeben und durch Chroma-Coding im Code repräsentiert.



## Datendefinitionsanalyse

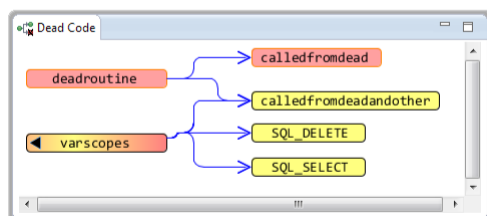


- Ermittlung aller Datenelemente, die im Programm definiert werden, sowie Auflösung von Fakturierungen. Pro Datenelement werden Details wie der Datentyp (auch bei impliziten Deklarationen), sein Gültigkeitsbereich und seine Verwendung ausgegeben, sowie welche Datenelemente von welchen anderen redefiniert (based, Pointer) oder mittels like kopiert werden.
- Include-Aufrufe werden aufgelöst, so dass auch Datenelemente, die aus einem Include stammen, ermittelt werden.

## Analysen zur Code-Bereinigung

### Dead Code Analyse

- Ermittelt Prozeduren, die nie oder nur aus anderen toten Prozeduren aufgerufen werden.
- Analysiert, ob Prozeduren Code enthalten, der nie ausgeführt werden kann.



## Analyse überflüssiger Datendefinitionen

- Liefert eine Aufstellung der Datenelemente, die nie oder nur innerhalb von totem Code verwendet werden.
- Analysiert für Gruppen, ob diese oder ihre untergeordneten Elemente je verwendet werden.

Name	Type	Declared in	Include	Line
F	character(10) static	deltademo		9
1 A	deltademo			10
1 varstruc	deltademo			12
1 glbstructure	deltademo	deltademoinclude		2
2 var2	fixed binary(15)	deltademo	deltademoinclude	3
2 varlike	like varstruc	deltademo	deltademoinclude	4
1 neverused	deltademo			20
1 usedindead	deltademo			23
2 var3	fixed binary(31)	deltademo		24
1 *	union	input		57
sysprInt	file print	deltademo		73
repeat	fixed binary(31)	deltademo		74
1 *	union	deltademo		75
1 *	union	output		85
aValue	character(20)	output		102

## Include-Analyse

Line	Code	#exec
1	DECLARE MAXLEVEL VALUE (5) FIXED BINARY;	6
2	DECLARE LEVEL FIXED BINARY (31);	6
3	DECLARE SCORE FIXED BINARY (31);	6
4	DECLARE BESTB(1:MAXLEVEL) FIXED BINARY (31);	0
5	DECLARE NDX FIXED BINARY (31);	3
6	DECLARE PCECE FIXED BINARY (31);	4
7	DECLARE WCKSPFLAG BIT(1) ALIGNED;	0

- Ermittelt für jedes verwendete Include, ob es Code-Anteile enthält, die in allen analysierten Programmen toten Code oder tote Datendefinitionen darstellen.
- Stellt fest, welche Anteile des Includes nur in einzelnen Programmen benötigt werden und deshalb besser dort definiert werden sollten.

## Reports und Exports

- Analyseergebnisse können als HTML-Reports ausgegeben werden.
- Alle Analysen können zusätzlich in verschiedenen Dateitypen exportiert werden.

## Prädikat „BEST OF 2015“ für AMELIO Logic Discovery

Die Initiative Mittelstand zeichnete AMELIO Logic Discovery bei der Verleihung des Innovationspreises mit dem Prädikat „BEST OF 2015“ aus.



## Keine „one size fits all“-Lösung



**AMELIO Logic Discovery** ist konfigurierbar, so dass die Menge der gelieferten Analysen an das jeweilige Einsatzszenario angepasst werden kann. Die vorgestellten Analysen stellen nur eine Auswahl dar, weitere Analysen sind standardmäßig verfügbar. Zusätzliche kundenspezifische Analysen können hinzugefügt werden. Weitere Informationen erhalten Sie unter: [www.delta-software.com/amld](http://www.delta-software.com/amld)

### Delta Software Technology

Delta Software Technology ist Spezialist für generative Software-Werkzeuge, die die Modernisierung, Integration, Entwicklung und Wartung individueller IT-Anwendungen automatisieren.

Unsere Lösungen helfen Ihnen, Ihre Anwendungen schnell und sicher an neue Geschäftsanforderungen, Architekturen, Technologien und technische Infrastrukturen anzupassen.

#### AMELIO® Modernization Platform™

Maßgeschneiderte Factory für die Modernisierung großer IT-Anwendungen: 100% automatisch und deshalb sicher, zuverlässig und fehlerfrei.

#### HyperSenses®

Integriertes System für modellgetriebene Entwicklung von DSLs und Software-Generatoren.

#### SCORE® Adaptive Bridges™

Intelligentes Service Enablement für die Wiederverwendung bewährter Anwendungen mit modernsten Technologien: Flexibel, rentabel und non-invasiv.

#### SCORE® Data Architecture Integration™

Daten als echte Business Services: Schnell, einfach und unabhängig von Datenarchitekturen und Speicherungsformen.

#### ADS™ Application Development for COBOL and PL/I

Plattformunabhängige Entwicklung für zukunftssichere Back-End-Anwendungen.

Delta liefert seit mehr als 35 Jahren erfolgreich fortschrittliche Software-Technologie an Europas führende Organisationen, zu denen u.a. AMB Generali, ArcelorMittal, Deutsche Telekom, Hüttenwerke Krupp Mannesmann, Gothaer Versicherungen, La Poste, RDW, Suva und UBS gehören.



Delta Software Technology GmbH  
Eichenweg 16  
57392 Schmallenberg

phone +49 2972 9719-0  
fax +49 2972 9719-60  
e-mail [info@delta-software.com](mailto:info@delta-software.com)

[www.delta-software.com](http://www.delta-software.com)

Copyright © 2015 Delta Software Technology GmbH. Alle Rechte vorbehalten.

Delta, SCORE, ObjectBridge, SCOUT², AMELIO, HyperSenses und das Delta Software Technology Logo sind registrierte Warenzeichen und SCORE Adaptive Bridges, SCORE Data Architecture Integration, Model Driven Legacy Integration, Integration in Motion, SCORE Transformation Factory, AMELIO Modernization Platform, AMELIO Logic Discovery, ADS, ANGLE und Active Intent sind Warenzeichen der Delta Software Technology GmbH in Deutschland und/oder anderen Ländern. Alle anderen eingetragenen Warenzeichen, Warenzeichen, Handelsnamen oder Dienstleistungsmarken sind Eigentum ihrer jeweiligen Besitzer.

Bestellnummer: MT 11\*078.02 – März 2015