



# Flow Graph Manipulator (FGM) 3.0 – Reverse-Engineering-Werkzeug für komplexe Softwaresysteme

**Anja Beier**

pro et con

**Innovative Informatikanwendungen GmbH**

11. Workshop Software-Reengineering  
04.-06. Mai 2009, Bad Honnef



# Agenda

- Ausgangssituation
- Spezifikation der Erweiterungen
- Programmanalyse
- Architekturanalyse
- Demonstration ausgewählter Funktionalitäten
- Fazit und Ausblick



# Ausgangssituation (1)

- FGM - Flow Graph Manipulator
  - Bereits seit 10 Jahren kommerzielles Produkt
  - Erfolgreich in verschiedenen Migrationsprojekten eingesetzt
    - eCom Logistik (HPNonStop-COBOL-Anwendung)
    - Amadeus Germany (BS2000-Migration, SPL nach C++)
  
- Eigenschaften
  - Verschiedene Frontends (COBOL, PL/I, SPL, ...)
  - Analyse von eingebetteten Systemen (SQL, DL/I, CICS)
  - Detaillierte Analyse einzelner Quellprogramme
  - Feingranulare Informationen
    - Datenfluss
    - Steuerfluss
    - Metriken und Statistikdaten

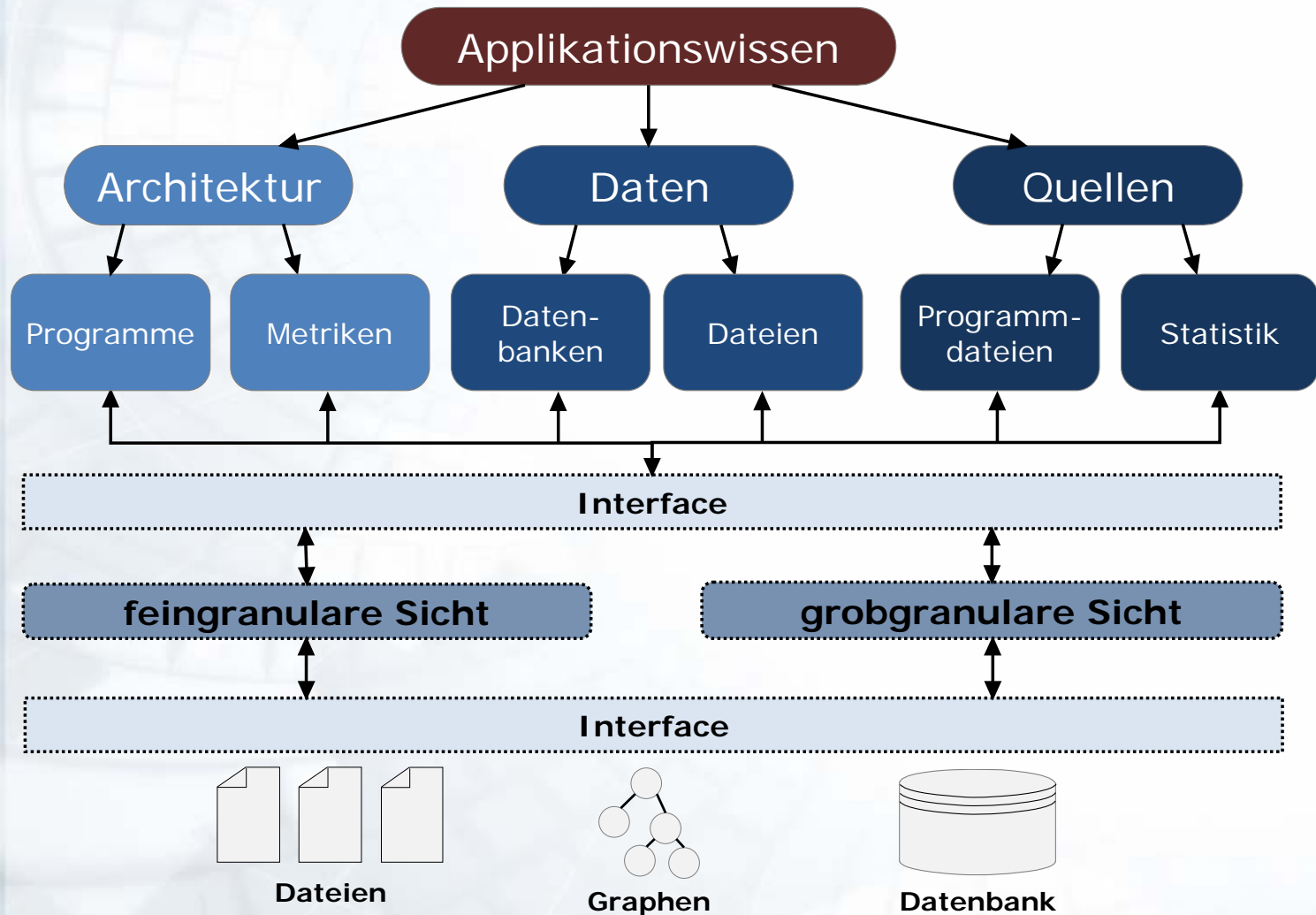


## Ausgangssituation (2)

- Motivation für Erweiterung
  - Veraltete Benutzeroberfläche
    - **Ablösung durch modernes Oberflächenkonzept**
  - Bisherige Funktionalität unvollständig für die Analyse komplexer Softwareprojekte
  - Problem: Gewinnung programmübergreifender Informationen
    - Programmaufrufe
    - Aussage verschiedener Metriken bezogen auf das Gesamtprojekt
  - Sicht auf die Gesamtapplikation ist wesentlicher Aspekt für Reverse Engineering und Redokumentation
    - **Neue Funktionalität "Applikationswissen"**

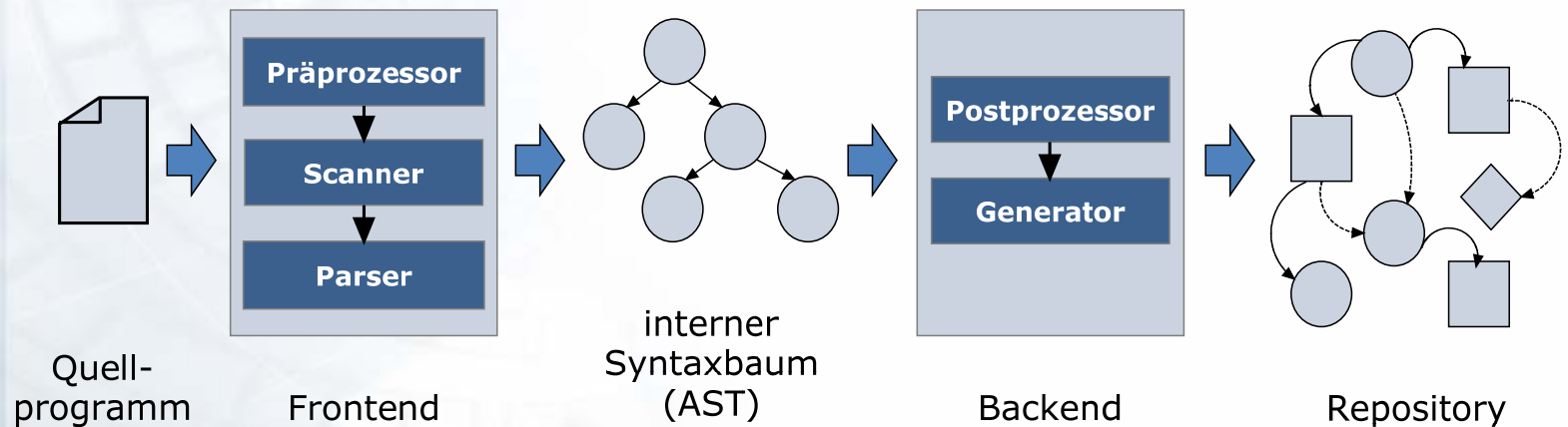


# Spezifikation der Erweiterungen



# Programmanalyse (1)

- Ermittlung detaillierter Informationen eines Programms



- Repository – interne Programmrepräsentation
  - Objekte (Variablen, Strukturen, Dateien, ...)
  - Beziehungen (Anweisungen, Includes, ...)

# Programmanalyse (2)

## ➤ Repository-Graph:

- Individueller Aufbau für jede Programmiersprache
  - ca. 20 Objekttypen, ca. 60 Beziehungstypen
- Durchschnittliches COBOL-Programm:
  - ca. 10.000 Objekte, ca. 15.000 Beziehungen

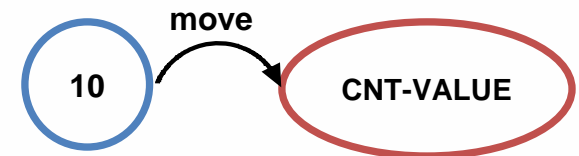
## ➤ Beispiel:

```
MOVE 10 TO CNT-VALUE.
```

```
(cobol_data 6417 CNT-VALUE (loc 2 497 9) ...)
```

```
(cobol_constant 6420 "10" (loc 2 721 23))
```

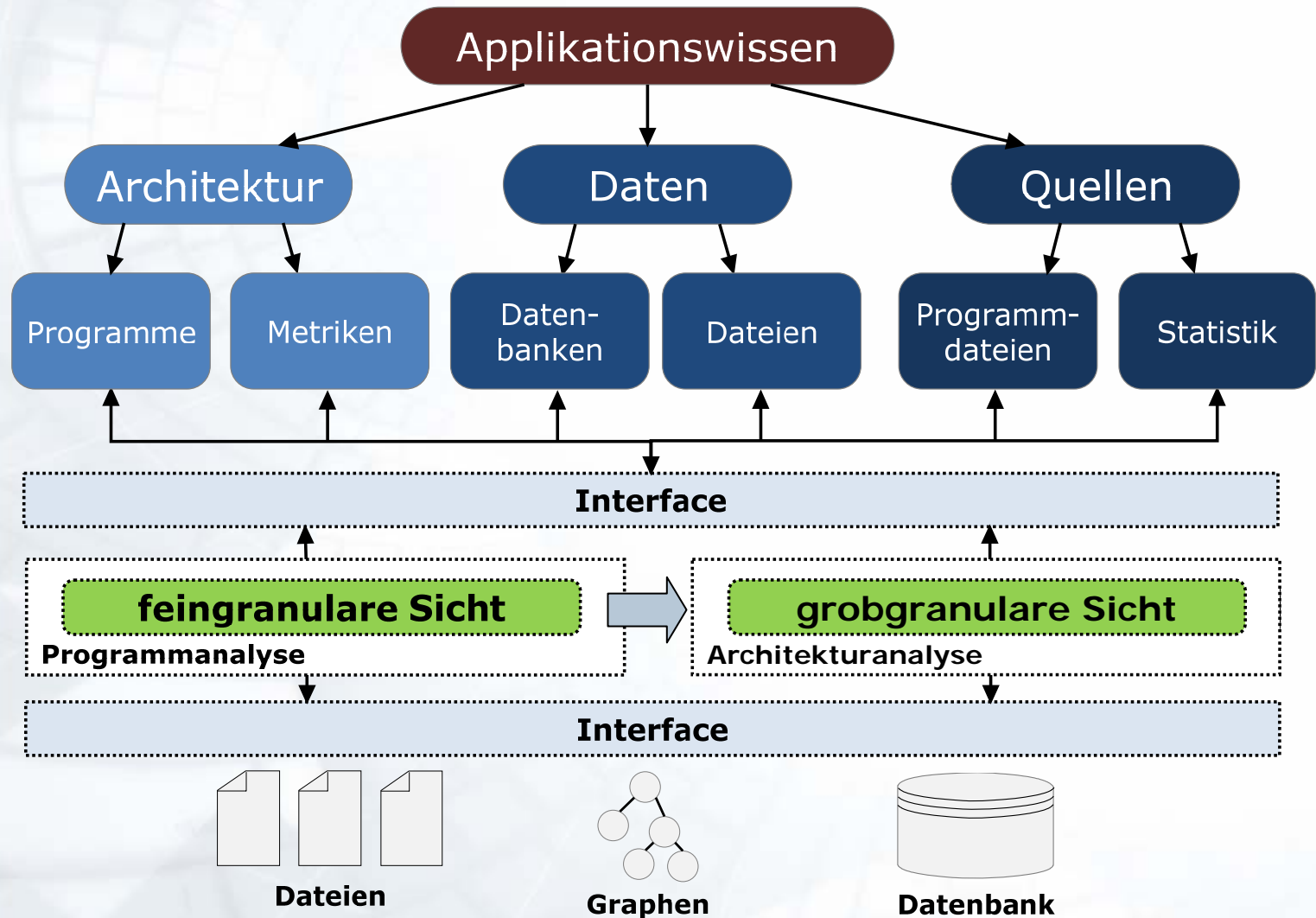
```
(cobol_move 4024 (loc 2 721 18) (6417) (6420))
```



## ➤ Abfragesprache FGM-LANguage



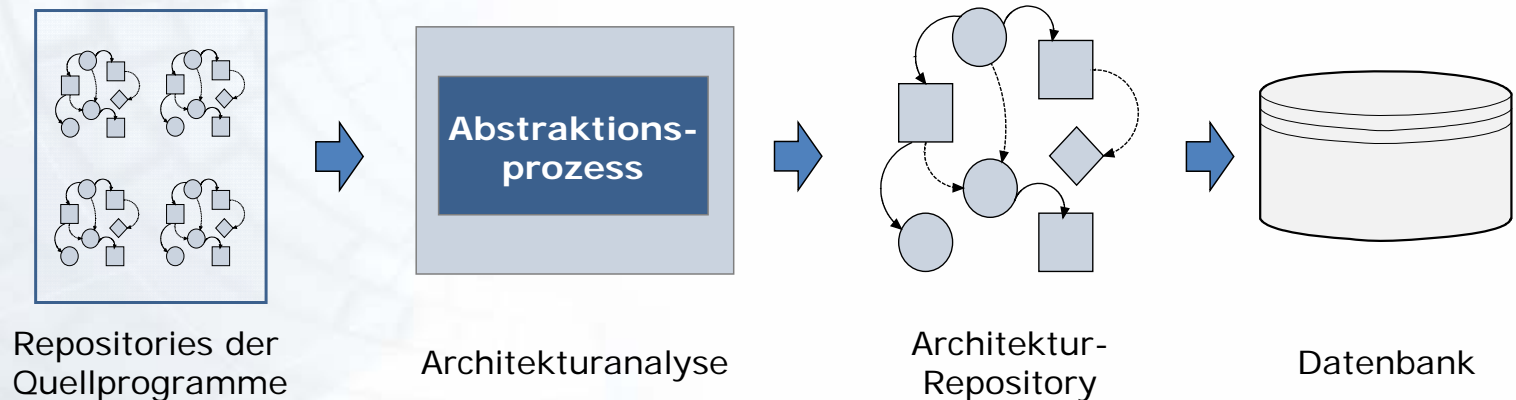
# Architekturanalyse





# Architekturanalyse - Repository

- Abstraktion von der feingranularen zur grobgranularen Sicht

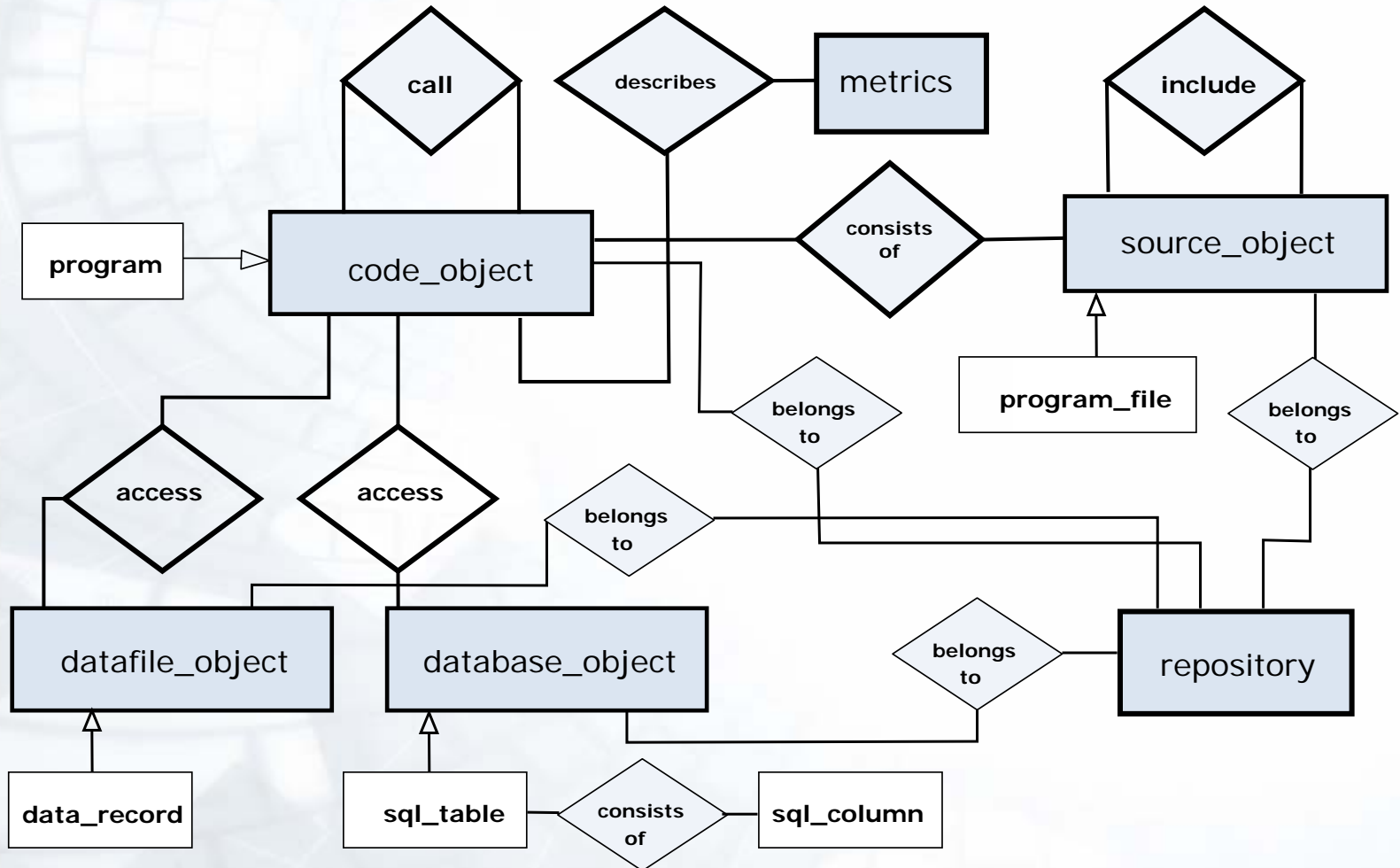


- **Architektur-Repository**

- Programmübergreifend
- Sprachübergreifend
- ➔ **Entwurf eines Architekturgraph-Modells**

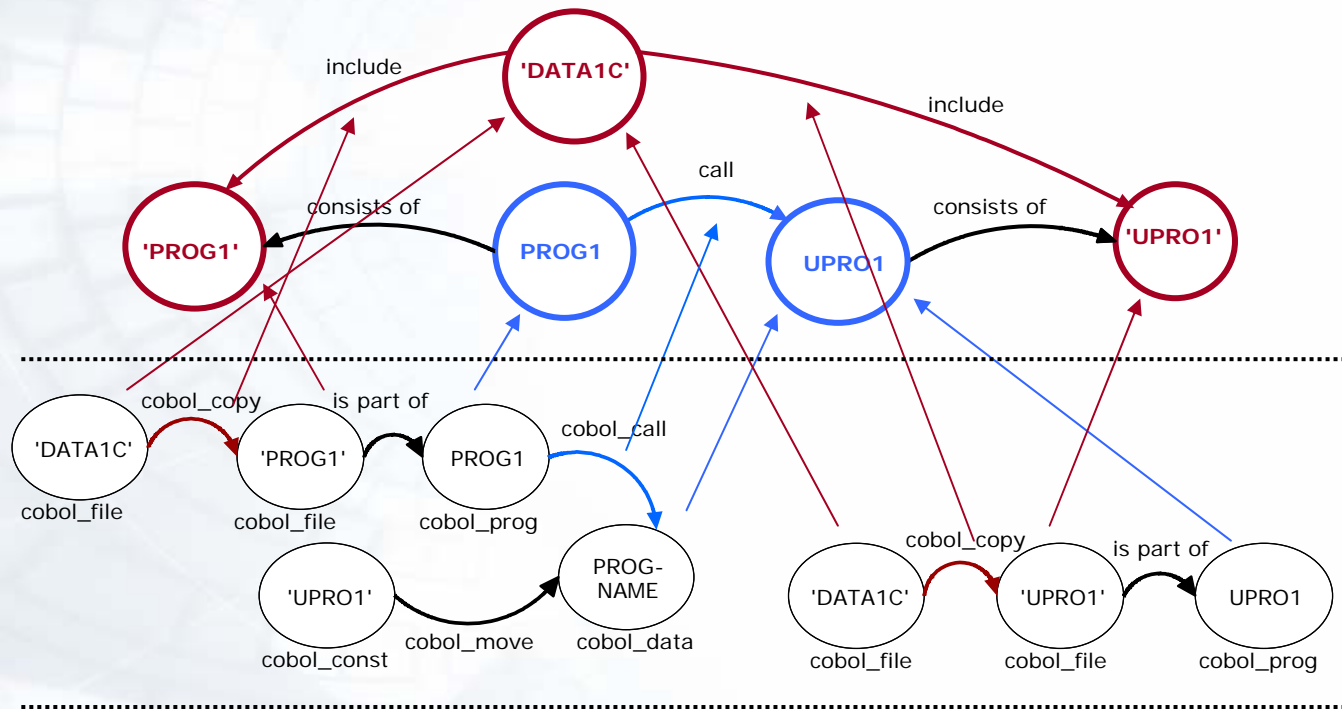


# Architekturanalyse - Graphmodell





# Architekturanalyse - Abstraktion



```

PROGRAM-ID. PROG1.
...
COPY DATA1C.
...
MOVE 'UPRO1' TO PROG-NAME.
CALL PROG-NAME.
    
```

Datei "PROG1"

```

PROGRAM-ID. UPRO1.
...
COPY DATA1C.
    
```

Datei "UPRO1"

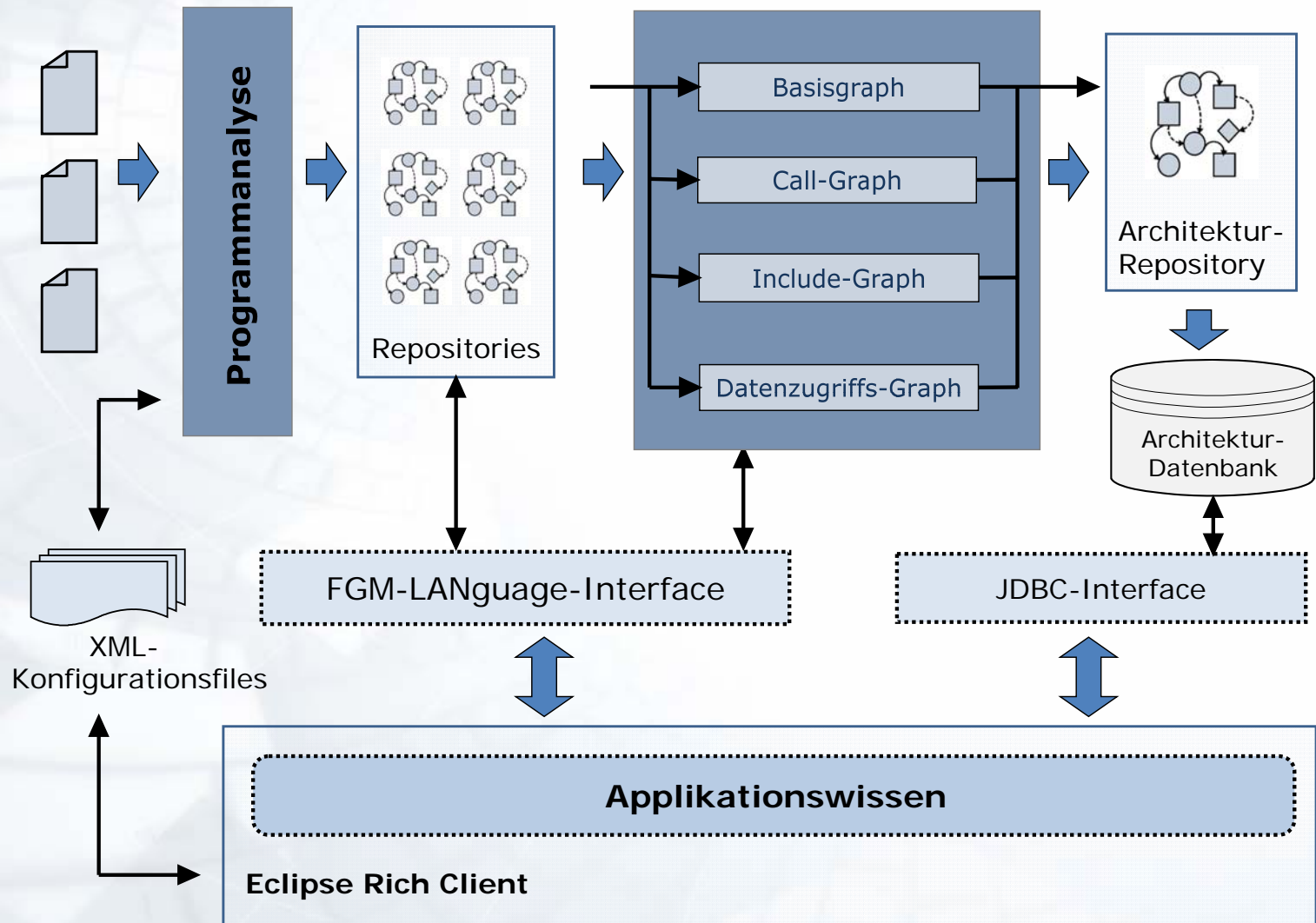
Architekturgraph  
(SQL-Datenbank)

Datenflussgraph  
(Repositories)

Quellprogramme



# Systemarchitektur



# Demonstration

FGM - C:/Programme/proetcon/FGM\_3.0/samples/cobol/cobol\_mf/COBOLDEMO

Datei Analyse Ansicht Hilfe

Architekturgraph

Explorer

- src
  - DMUCA01
  - DMUCA08
  - FAZCA15
  - FBUCD99
  - H0061
  - DMUCA02
  - FADCA17
  - FAZCA14E
  - U7005
  - ZOUCA02
  - ZOUCA06
  - ZOUCA16
  - ZOUCA17

Meldungen

Dateiname	Meldung	Zeile	Pfad
H0061	cannot find File DMAA01CC in ANY LIBRARY	0	
H0061	INIT-ZSL is not defined	530	C:\Programme\p



## Fazit und Ausblick

- Realisierung der Funktionalität "Applikationswissen" als zentrale Komponente des FGM 3.0
- Übernahme und Integration bewährter Module und Werkzeuge
  - Frontends
  - Repository-Modell
  - FGM-LANguage-Interface
- Erweiterungen vorgesehen



**Vielen Dank**

**pro et con**

Innovative Informatikanwendungen GmbH

**Analyse - Reengineering - Migration**

**Anja Beier**

[Anja.Beier@proetcon.de](mailto:Anja.Beier@proetcon.de)

[www.proetcon.de](http://www.proetcon.de)

Tel.: 0371/5347-324

Fax: 0371/5347-345