

Erweiterung bestehendes Datenflussanalyse- Werkzeug für prozedurweite Datenflussanalyse

- Bachelorthesis
- Masterthesis

Aufgabenstellung

In nahezu jedem größeren Unternehmen bilden betriebliche Informationssysteme (BIS) geschäftskritische Prozesse ab.

In ihre Erstellung und Weiterentwicklung werden jedes Jahr Milliardenbeträge investiert und viele Systeme werden über Jahrzehnte weiterentwickelt und gewartet. Dabei wird die eingangs erstellte Dokumentationen oftmals nur unzureichend gepflegt und ursprünglich befolgte Architekturprinzipien geraten mit der Zeit in Vergessenheit oder werden - manchmal bewusst - übergangen. Die bestehende Codebasis wird dadurch zunehmend unübersichtlich und ihre Wartung zunehmend aufwändig. Analysewerkzeuge wie der DependencyAnalyzer von itestra können dabei helfen, in solchen Codebasen technische und fachliche Zusammenhänge nachzuvollziehen.

Der DependencyAnalyzer ist ein itestra-eigenes Werkzeug zur statischen Programmanalyse, das u.a. der Analyse von Kontroll- und Datenflüssen in Programmen verschiedener Programmiersprachen dient (darunter COBOL, JAVA, C). Da langjährig erfolgreiche, geschäftskritische Systeme oftmals auf einer COBOL-Codebasis beruhen, liegt der Fokus dieser Arbeit auf COBOL-Systemen. Batch-Verarbeitungen in COBOL-Systemen sind häufig so aufgebaut, dass ein gegebenes Programm eine Datei als Verarbeitungsergebnis schreibt, die in einem Folgeschritt von einem weiteren Programm eingelesen und weiterverarbeitet wird.

Die Reihenfolge der ausgeführten Programme sowie die verwendeten Ein- und Ausgabedateien werden durch Prozeduren definiert, die ihrerseits in einer eigenen Prozedursprache geschrieben sind, z.B. JCL (IBM z/OS) oder SDF (Fujitsu BS2000). Die Verarbeitung der so gelesenen bzw. geschriebenen Dateien erfolgt in der Regel sequentiell. Die Struktur der einzelnen Records wird dabei in COBOL durch Datenstrukturen fixer oder variabler Länge definiert.

Diese Datenstrukturen sind in verschiedenen Programmen nicht zwangsläufig identisch strukturiert und benannt, selbst wenn sie auf denselben Dateien arbeiten. Die Nachverfolgung einzelner Werte über mehrere Programme hinweg wird dadurch erschwert, ist jedoch für die Analyse und Wartung notwendig.

Aktuell berücksichtigt der DependencyAnalyzer keine Prozedursprachen für eine programmübergreifende Datenflussanalyse, sondern ist auf Datenflüsse innerhalb einzelner Programme beschränkt.

Zielsetzung der Arbeit

Diese Arbeit zielt darauf ab, diese Lücke zu füllen: Der DependencyAnalyzer soll befähigt werden, die Prozedursprachen JCL und SDF so weit zu parsen, dass er hierüber definierte Abhängigkeiten zwischen Programmen bei seiner Analyse berücksichtigen kann. Ziel ist, Datenflüsse über Programme hinweg und auch bei wechselndem Aufbau der verwendeten Ein- und Ausgabestrukturen nachverfolgen zu können.

Betreuung (itestra GmbH)

Philip Rauwolf (rauwolf@itestra.de)