XCMR: Ein generisches Reengineering-Werkzeug basierend auf einem relationalen Quellcode-Repository.

Friedhelm Kühn, IT-Berater
friedhelm.kuehn@t-online.de

Abstract.

Unternehmen haben immense Summen in die Entwicklung ihrer Anwendungssysteme investiert und hängen vom zuverlässigen und wirtschaftlichen Betrieb dieser Anwendungen ab. Und doch erschließt sich die Mehrzahl der im zugrundeliegenden Quellcode enthaltenen Fakten den Mitarbeitern der IT-Abteilungen nur schwer. Mit Hilfe von XCMR (Extensible Code Mining Repository), einem generischen Software-Reengineering-Werkzeug, können Informationen aus Quellcodes gewonnen und großflächige Änderungen automatisiert/regelbasiert vorgenommen werden. Einsatzschwerpunkt für XCMR sind projekt- und sprachübergreifende (3GL) Entwicklungs-, Wartungs- und Modernisierungsszenarien, in denen unternehmensweite Repository-Instanzen aufgebaut werden. Das breiteste Wirkungsspektrum und der höchste Nutzen wird beim Einsatz im Umfeld Z/OS erreicht.

Einleitung.

Anwendungsquellcodes sind nicht nur gewöhnliche Texte, sondern unterliegen den semantischen und syntaktischen Regeln der zugrundeliegenden Programmiersprache(n). Ihre Befehle beschreiben unterschiedliche Objekte und setzen diese Objekte in bestimmten Reihenfolgen miteinander in Beziehung. Die Gesamtheit der Einzeldefinitionen und Beziehungen stellt eine Wissensbasis dar, die es mit Hilfe entsprechend spezialisierter Werkzeuge zu erschließen und zu verwerten gilt. Bei XCMR geschieht das auf der Basis eines Prozesses, bestehend aus den nachfolgend beschriebenen Einzelschritten.

1. Code Explorer

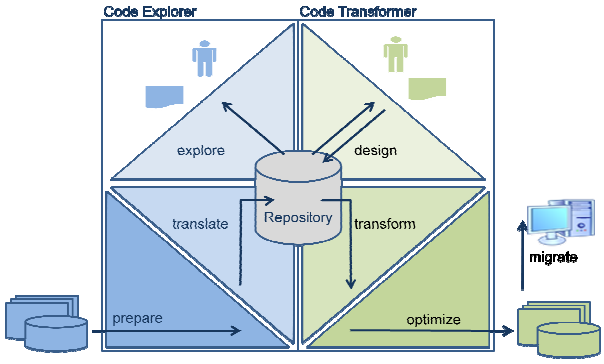
1.1 Prepare.

Die Konsistenz eines Quellcode-Repository ist wesentlich. Dafür ist Vollständigkeit und Redundanzfreiheit erforderlich, die im Rahmen einer Bestandsaufnahme durch Bereinigung der Quellcode-Basis erreicht wird.

Außerdem können bei dieser Gelegenheit offensichtlich nicht mehr benutzte Programme entfernt werden, um das Quellcode-Repository nicht von vornherein durch ‚toten Code' zu belasten.

1.2 Translate.

Im relationalen Repository von XCMR entsprechen Datenbank-Tabellen den verschiedenen Definitions- und Verarbeitungs-Befehlen der unterschiedlichen Programmiersprachen. Die Befüllung des Repositories geschieht durch maschinelle Übersetzung (translate) der vorliegenden Quellcode-Member, anhand der durch die Programmiersprachen vorgegebenen syntaktischen Regeln. Erzeugt werden Inhalte der zugehörigen Datenbanktabellen. Dabei wird die textuelle Darstellung eines Sachverhalts in strukturierte Informationselemente gleicher Bedeutung überführt und diese im Repository abgelegt.



1.3 Explore.

Die in der Datenbank abgelegten Informationen werden mit SQL-Abfragen analysiert. Auf diesem Wege werden diese für die Mittel/Werkzeuge der klassischen Informationstechnik erschlossen. Bereits in dieser Phase können vielfältige Fragestellungen mit programmierten Regeln zur ‚semantischen Analyse' beantwortet werden. XCMR enthält bereits mehrere hundert Standardauswertungen und es können mit geringem Aufwand neue Regeln für spezifische Fragestellungen erstellt und als Auswertung ergänzt werden. Gefundene Informationen oder Zusammenhänge werden in adäquater Form dargestellt.

XCMR verfügt über eine browserbasierte Dialogoberfläche sowie Dialogoberflächen in Form von Eclipse-Plugins.

Wird XCMR ausschließlich zur Wissensbeschaffung, sprich zur Optimierung und Kostensenkung im Bereich der Software-Wartung, -Pflege und -Erweiterung eingesetzt, werden die nachfolgenden Schritte nicht durchlaufen.

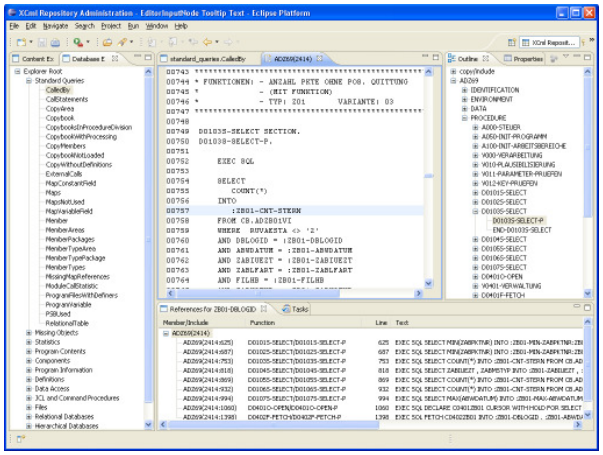
2. Code Transformer

Für (großflächige) regelbasierte Veränderung des zugrundliegenden Quellcodes sind die nachfolgenden Schritte von Bedeutung.

2.1 Design.

Wenn bei einem Projekt in größerem Umfang spezifische Änderungen am vorliegenden Quellcode vorgenommen werden müssen, bietet sich die automatisierte Quellcodeänderung mittels entsprechender XCMR-Regelwerke an. Dafür ist es erforderlich die zu ändernden Stellen durch ‚Suchregeln' genau zu spezifizieren. Dies können einerseits ‚Start-Regeln' sein, mittels derer bestimmte Felder/Programmstellen anhand spezifischer Feld-Layouts oder besonderer Quellcode-Kontexte identifiziert werden. Andererseits können betroffene Stellen ausgehend von bereits bekannten Stellen durch Betrachtung von ‚Datenflüssen' identifiziert werden. Wichtig ist die vollständige Identifikation aller relevanten Stellen.

Auf der Basis einer Kategorisierung der betroffenen Programm-Kontexte werden Änderungsvorschriften, d.h. Regeln für die Art der Anpassungen, erarbeitet.



2.2 Transform.

Diese Regeln werden in Programmlogik für die automatisierte Änderung der entsprechenden Quellcodestellen umgesetzt. Die Programmierung dieser Änderungsregeln ist wenig aufwendig, so dass sich eine Automatisierung bereits bei mittlerer Häufigkeit des Vorkommens lohnt. Die Regeln können iterativ verfeinert werden, da automatisierte Änderungen wiederholbar sind.

Die Anwendung der entsprechenden Transformationsregeln auf die in der Wissensbasis gekennzeichneten Informationsobjekte erstellen das gewünschte Ergebnis. Des Weiteren werden in dieser Phase eventuell erforderliche Programme zur Datenüberleitung erstellt.

2.3 Optimize.

Einige Programmstellen bleiben für die manuelle Änderung im Umfeld der normalen Entwicklungsumgebung, z.B. wenn sich die Automatisierung nicht lohnt. Bei komplexen Migrationsprojekten mit Architekturwechsel bzw. Sprachkonvertierung kann eine manuelle Sichtung und Anpassung zur Verbesserung der Code-Qualität bzw. Wartbarkeit erforderlich sein. Durch Test der Anwendung wird deren Funktionalität verifiziert (vergleichender Test).

2.4 Migrate.

Die Produktivschaltung der geänderten Anwendung, inklusive einer eventuellen Datenmigration schließt das Änderungs- bzw. Modernisierungs-Projekt ab. Je nach Umfang der Änderungen/Anwendungsgröße kann die Einführung in einem (Big-Bang) oder mehreren Schritten (paketweise) erfolgen.

3. Besonderheiten.

Das dem XCMR-Repository zugrundeliegende relationale Datenmodell wird im Kundenumfeld offengelegt. Somit können die enthaltenen Informationen auch mit anderen, nicht zu XCMR gehörenden Hilfsmitteln genutzt werden.

XCMR unterstützt über die Programmiersprachen zur Programmlogik hinausgehend auch andere Quellcodeformen (Maskenbeschreibung, Job-Control, DB2-DDL, IMS-DBD/-PSB, RACF, …). Damit werden insbesondere nicht nur die Software-Entwickler, sondern auch Datenbankadministratoren und sonstiges projektübergreifend tätige Personal produktivitätssteigernd unterstützt.

XCMR zielt bei der Bewältigung vorliegender Aufgabenstellungen auf optimale Lösungen und optimalen Wirkungsgrad. Entsprechend ist es üblich und gewollt, die vorhandene XCMR-Funktionen innerhalb von bzw. begleitend zu entsprechenden Projektmaßnahmen aufgaben- bzw. kundenspezifisch anzupassen bzw. zu erweitern.

4. Zusammenfassung

Der Einsatz von XCMR erhöht die Effektivität bei der Software-Pflege/-Änderung existierender Anwendungen und senkt die damit verbundenen Kosten wesentlich. Die wirtschaftliche und risikoarme Modernisierung existierender Anwendungen wird durch automatisierte Quellcode-Änderungen auf der Basis spezifizierter Regeln möglich. Die bei der Entwicklung der erforderlichen Analyse- und Änderungsregeln erreichbare Produktivität stellt kunden- und projektspezifisch optimierte Quellcode-Anpassungen bzw. Codegenerierung innerhalb des laufenden Modernisierungsprojektes sicher. XCMR wurde bereits für die Wartungsunterstützung großvolumiger Anwendungssysteme eingesetzt und hat sich in vielfältigen Software-Reengineering Projekt-Szenarien (Anwendungsmigration, Sprachkonvertierung, Datenbankumstellung, Datenmigration und Plattformwechsel) bewährt.