

Das Cobus-Projekt: COBOL-Bestandsanalyse und Sanierung bei der Debeka

Andreas Fuhr*, Jürgen Ebert, Volker Riediger

Institut für Softwaretechnik
Universität Koblenz-Landau

Marcus Rausch†, Johannes Bach, Max Doppler, Sven Kühner, Sandra Wilbert

Abteilung BO/S
Debeka-Hauptverwaltung Koblenz

Abstract: Cobus ist ein Kooperationsprojekt zwischen der Debeka Versicherungsgruppe und der Universität Koblenz-Landau. Ziel des Projekts ist die Sicherstellung der Wartbarkeit und der Zukunftsfähigkeit des COBOL-Kernsystems der Debeka. In drei Arbeitsbereichen werden die interne Qualität von COBOL-Modulen erhoben, die Architektur redokumentiert, sowie daraus abgeleitete Sanierungsmaßnahmen ermittelt. Zu den Ergebnissen des Projekts zählen u. a. eine Werkzeugkette zur Berechnung von Software-Metriken und ein Dashboard zur Visualisierung der Metriken, die Spezifikation von elf Architektur-Perspektiven, sowie Werkzeuge für deren Erzeugung und die Integration von Code-Refaktorisierungsmaßnahmen in die Entwicklungsumgebung der Debeka.

1 Einleitung

Auch über 50 Jahre nach Einführung der Programmiersprache COBOL laufen in vielen Firmen noch Systeme, die in dieser Sprache implementiert sind. Darin sind häufig Kernprozesse realisiert, die für die Firmen einen hohen Wert darstellen und die für die nächsten Jahrzehnte erhalten werden müssen.

Cobus (COBOL-Bestandsanalyse und Sanierung) ist ein Kooperationsprojekt der Debeka-Versicherungsgruppe in Koblenz und der Universität Koblenz-Landau. Federführend entwickeln die Abteilung BO/S der Debeka und das Institut für Softwaretechnik der Universität Methoden und Werkzeuge zur Analyse und Qualitätsbewertung des COBOL-Kernsystems der Debeka. Aus den Ergebnissen der Bewertung sollen geeignete Sanierungsmaßnahmen abgeleitet werden. Das Projekt unterstützt somit die Aufrechterhaltung der Wartbarkeit und der Zukunftsfähigkeit des Debeka-Kernsystems.

* A. Fuhr, Universität Koblenz-Landau, Institut für Softwaretechnik, Universitätsstr. 1, 56070 Koblenz
Mail: afuhr@uni-koblenz.de, Telefon: (0261) 287-2705

† M. Rausch, Debeka Hauptverwaltung, Abteilung BO/S, Ferdinand-Sauerbruch-Str. 18, 56058 Koblenz
Mail: marcus.rausch@debeka.de, Telefon: (0261) 498-2243

Das Projekt umfasst drei Kern-Aufgabenbereiche: (1) die Beurteilung der Qualität einzelner COBOL-Programme (interne Qualität), (2) die Redokumentation der Software-Architektur und (3) das Ableiten und Durchführen von Sanierungsmaßnahmen. Ausgewählte Ergebnisse der drei Bereiche werden in Abschnitt 2 vorgestellt.

2 Projekt-Ergebnisse

Im Rahmen des Cobus-Projekts werden sowohl Methoden entwickelt als auch Software-Werkzeuge, um diese Methoden zu unterstützen. Ausgewählte, wichtige Ergebnisse werden nach den Aufgabenbereichen geordnet dargestellt.

2.1 Interne Qualität *(abgeschlossen)*

Die Bewertung der internen, detaillierten Qualität einzelner COBOL-Programme wurde durch die Berechnung von Software-Metriken umgesetzt. Mit Hilfe des Goal-Question-Metric-Ansatzes [SBCR02] wurden *21 atomare Basis-Metriken* definiert, die speziell auf die Bedürfnisse der Debeka zugeschnitten sind. Der Katalog der Metriken umfasst dabei sowohl bekannte Metriken (z. B. Lines of Code) als auch individuelle Maße (z. B. Logische Verschachtelungstiefe von Sections, berechnet anhand von Debeka-Namenskonventionen). Um die Basis-Metriken mit den Qualitätseigenschaften Komplexität, Kopplung und Kohäsion als Repräsentanten für die Wartbarkeit von COBOL-Programmen in Beziehung setzen zu können, wurden in einer Umfrage von 24 Mitarbeitern aus allen sechs Debeka-Entwicklungsabteilungen COBOL-Programme nach diesen Eigenschaften bewertet. Anschließend wurden *drei statistische Modelle entwickelt*, die anhand der Basis-Metriken die Qualitätseigenschaften vorhersagen können [FRB⁺12].

Um die Metriken-Bewertungen kontinuierlich berechnen zu können, wurde eine Werkzeugkette entwickelt: die *Cobus-Metrik-Toolsuite (CMTS)*. Sie besteht aus folgenden Komponenten. Der *COBOL-Parser* wurde von der Debeka selbst entwickelt und parst COBOL-Module in einen abstrakten Syntaxgraphen, der mit zusätzlichen Informationen für die Metrikberechnung angereichert wird. Der *COBOL-Checker* ist ein von der Debeka entwickeltes Werkzeug, das Verstöße gegen die COBOL-Programmierrichtlinien der Debeka erfasst. Diese fließen in die Metrikberechnung mit ein. Das *Metrik-Tool* berechnet die 21 Basis-Metriken und die 3 Qualitätsmetriken und speichert die Ergebnisse in eine Datenbank. Die Toolsuite erlaubt sowohl die kontinuierliche, tagesaktuelle Berechnung der Metriken als auch die nachträgliche Bewertung der Systemhistorie durch Zugriff auf ein Versionsverwaltungssystem. Somit ist ein Metrik-Datenbestand entstanden, der den gesamten Systemzustand seit Oktober 2010 widerspiegelt und der täglich aktualisiert wird.

Zur Auswertung und Visualisierung der Daten wurde ein *Metrik-Dashboard (Metrik-Cockpit)* entwickelt. Basierend auf der BIRT-Engine [EBP13] zur Erstellung von Reports wurde ein Webtool entworfen, welches Daten sowohl in tabellarischer Form als auch in Form von Diagrammen darstellen kann (z.B. Balken- oder Kreisdiagramme). Zusätzliche Metrik-Reports lassen sich dank BIRT schnell und unkompliziert ergänzen.

2.2 Architektur-Redokumentation (laufend)

Für ein besseres Verständnis des Systems wird die Software-Architektur des Kernsystems redokumentiert und analysiert. Um zu ermitteln, welche Perspektiven auf die Architektur für das Projekt relevant sind, wurde ein strukturierter Prozess definiert, der anhand von Nutzungsszenarien der Architekturbeschreibung solche Perspektiven erhebt. Als Ergebnis wurden 11 Debeka-spezifische Perspektiven ermittelt [Fuh13].

Aktuell werden Werkzeuge entwickelt, die die 11 Architektur-Perspektiven ausgehend von COBOL-Programmen, Dokumentation und weiteren Artefakten berechnen können.

2.3 Sanierungsmaßnahmen (laufend)

Als erste Sanierungsmaßnahme wurde beschlossen, die Einhaltung der Debeka-Programmierrichtlinien besser zu unterstützen. Dazu wurde der COBOL-Checker in die Debeka-eigene Eclipse-Entwicklungsumgebung integriert, so dass Richtlinienverstöße je nach Wichtigkeit als Information, Warnung oder Fehler markiert werden. Die Entwickler nutzen diese Funktion, um ihre Programme richtlinienkonform zu refaktorisieren.

Aktuell werden die Metrik-Ergebnisse daraufhin untersucht, ob sie Rückschlüsse auf notwendige Sanierungsmaßnahmen zulassen. Im weiteren Projektverlauf sollen zudem Maßnahmen aus den Ergebnissen der Architektur-Redokumentation abgeleitet werden.

3 Gewonnene Erkenntnisse

Die Erfahrungen aus dem Kooperationsprojekt sind durchweg positiv. Als Herausforderung stellte sich die Einführung der Metriken in den Entwickleralltag dar. Hier haben sich vertrauensbildende Maßnahmen, sowie die frühzeitige Einbeziehung aller Beteiligten in den Entwicklungsprozess als essentiell erwiesen. Bei Einführung der Richtlinienüberprüfung hat eine Unterstützung der Entwickler durch gut integrierte Hilfestellungen dazu geführt, dass Entwickler freiwillig auf die Einhaltung der Programmierrichtlinien achten.

Literatur

- [EBP13] Eclipse BIRT Project, 2013. <http://www.eclipse.org/birt/phoenix>.
- [FRB⁺12] A. Fuhr, M. Rausch, J. Bach, D. Bildhauer, J. Ebert, M. Doppler, J. Haas, D. Höh, V. Riediger und M. Schulze. Einführung von COBOL-Wartbarkeits-Metriken bei der Debeka. *ST-Trends*, 32(2):65–66, 2012.
- [Fuh13] A. Fuhr. Identifying Architectural Viewpoints: A Scenario-Based Approach. *ST-Trends*, 33(2):57–58, 2013.
- [SBCR02] R. van Solingen, V. R. Basili, G. Caldiera und H. D. Rombach. Goal Question Metric (GQM) Approach. In J. J. Marciniak, Hrsg., *Encyclopedia of Software Engineering*. John Wiley, New York, 2002.