

COBOL RE-ENGINEERING

DAS RULAMAN® PRODUKT GRANUM

Auf einen Blick

Die Mehrzahl der Programme, die produktiv auf großen Servern oder Mainframes laufen, ist immer noch in COBOL geschrieben. Der Code wurde oftmals durch Jahrzehnte der Wartung unleserlich.

IT Modernisation bietet mit dem Lösungspaket GRANUM¹⁾ einen Strauß von Möglichkeiten zur Optimierung des COBOL Codes: Re-Engineering, Refactoring, Restrukturierung, Formatierung und Sanierung sind nur einige der wichtigsten Schlagworte. Der Wert bestehender Anwendungssysteme wird signifikant erhöht, darüber hinaus können die Programme vollautomatisch modularisiert in Service-Bus Architekturen eingesetzt werden.

DIE HERAUSFORDERUNG

COBOL entstand in den frühen 1960er Jahren und wurde seitdem immer weiter ausgebaut und angepasst. Die Programme – oft geprägt durch Hardwareanforderungen der „ersten Jahre“ – wurden dabei immer unleserlicher und passen nicht mehr in moderne strategisch positionierte IT-Landschaften.

➤ Junge Informatiker wollen COBOL nur widerstrebend lernen; sie betrachten es in ihrer Vita oft als Negativposten. Entsprechend haben die Unternehmen große Mühe, geeignetes Personal für die Wartung zu finden. Was tun?

➤ Die Programme neu zu schreiben ist wirtschaftlich nicht vertretbar.

➤ Eine vollautomatische Transformation in moderne Sprachen, ist eine phantastische Alternative. Wird das Ergebnis jedoch wirklich in eine OO-Welt passen?

➤ Die bestehenden Programme umfassen oft Zehntausende von Codezeilen. In dieser Form

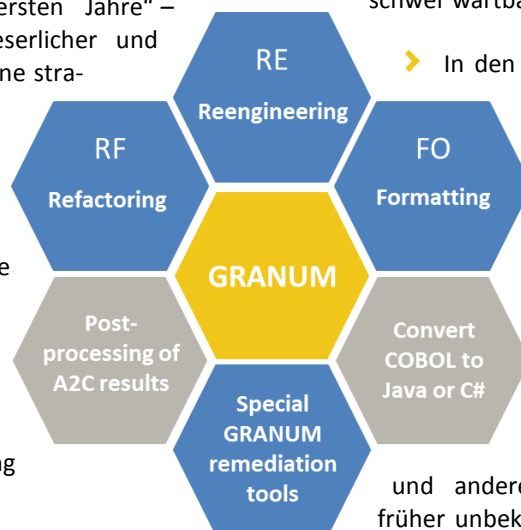
können Sie niemals in Serviceorientierte Architekturen (SOA) eingebettet werden.

➤ Der Code ist durch jahrelange Wartung von unterschiedlichen Personen unleserlich und schwer wartbar geworden.

➤ In den Programmen befinden sich Überreste von alten Code-Generatoren oder Pre-Compilern. Diese müssen endlich entfernt werden.

➤ **Alle Modernisierungs-Maßnahmen müssen preiswert und risikolos sein!**

➤ Der Code ist uneinheitlich und schlecht strukturiert. Scope Terminatoren, das EVALUATE und andere moderne Sprachmittel waren früher unbekannt. Der Code sollte dringend in diese Richtung saniert werden.



Gleichgültig, ob Sie lediglich COBOL Wartungsprobleme in den Griff bekommen wollen, ob Sie eine Umformung in OO-Sprachen planen oder ob Sie die Anwendungen in eine SOA einbetten wollen: die Programme müssen massiv bereinigt, neu strukturiert, neu formatiert, von totem Code befreit und modularisiert werden.

DAFÜR WURDE GRANUM GESCHAFFEN.

¹⁾ GRANUM, lat. für Korn

Die GRANUM Komponenten

GRANUM ist wie alle Produkte der IT Modernisation als eine Kollektion von *Rules* und *Rule Sets* implementiert. Diese Regeln sind in der Spezialsprache SPL (Sourcecode Processing Language= geschrieben, die Teil des RULAMAN®-Frameworks ist.

- Einführung von *Scope Terminatoen*
- Umsetzen undruckbarer (verborgener) Zeichen in Literalen und VALUE.
- Einführung von figurativen Konstanten anstelle von Literalen oder VALUE Definitionen.
- Umsetzen von *Intrinsic Functions* anderer Hersteller.

RE – RE-ENGINEERING

Die wichtigsten Funktionen von GRANUM/RE:

- **SANIERUNG.** RE formt Sprachkonstrukte, die nicht mehr unterstützt werden oder bei Einsatz eines modernen Compilers zu Fehlern führen, vollautomatisch in legale Statements um.
- **RE-ENGINEERING.** CobRE strukturiert den Code neu, entfernt Sprungbefehle, führt Scope Terminators (END-Klauseln) ein und schafft mit vielen ausgeklügelten Algorithmen ein Programm, das gut lesbar ist und so z.B. auch für eine maschinelle Umformung in objektorientierte Sprachen geeignet ist.
- **DIALEKTANPASSUNG.** Es gibt über 300 verschiedene, meist inkompatible COBOL Dialekte: Versionen und Varianten geprägt von Betriebssystem und Hardware-Architektur oder einfach nur - von der Marketingabteilung. Auch wurde beim Übergang von COBOL-68 auf COBOL-74 wurde eine Vielzahl von Statements oder Klauseln geändert oder durch neue Konstrukte ersetzt, die von den Compilern immer noch unterstützt werden, obwohl sie in keinem Handbuch zu finden sind. Beliebige Dialekte können so in einander zu überführt oder auf ein „normiertes“ COBOL portiert werden.

Nachfolgend weitere Funktionen, manche davon als Optionen wählbar:

- Auflösen von sog. Spaghetti-Code
- Entfernen vieler oder aller GO TO Befehle
- Strukturierung in SECTIONS und Paragraphen
- Erkennen und Auslagern von Unterprogrammen
- IF Sequenzen und Kaskaden in EVALUATE Strukturen umformen
- Entfernen von totem prozeduralem Code
- Eliminieren ungenutzter Felder
- Entfernen nicht sinntragender Punkte

RF – REFACTORING

GRANUM/RF ermöglicht es, große COBOL Programme in beliebige kleine Module aufzuspalten. Dabei werden die „Schnitte“ von einem Entwickler an der Grenze von Paragraphen oder Sections gekennzeichnet.

GRANUM/RF führt folgende Aktionen aus:

- Die gekennzeichneten Abschnitte werden aus dem Primärprogramm herausgelöst und in vollständige COBOL Module („Sekundärprogramme“) transformiert.
- Datenfelder des Primärprogramms, die im Sekundärprogramm angesprochen werden, übergibt RF über die LINKAGE SECTION.
- Die Werte dieser Felder werden vor dem Aufruf des Sekundärprogramms gesammelt und via CALL übergeben.
- Nach der Rückkehr werden die Felder des Primärprogramms entsprechend aufgeladen.
- Der Code zur Datenverbindung ist im Quellcode der Programme enthalten: es gibt keine Laufzeitroutinen!
- Sprungbefehle, die aus dem Sekundärprogramm ins Primärprogramm weisen, werden durch eine entsprechend generierte Logik unterstützt.

Anm.: Zu viele derartige Sprünge erschweren Lesbarkeit und Verständnis. Deshalb sollte GRANUM/RE vorher die Sprungbefehle reduzieren oder ganz eliminieren.

- Das Sekundärprogramm ist „schlank“. Das bedeutet, dass nur diejenigen Datendefinitionen übernommen werden, die tatsächlich notwendig sind. Falls z.B. für REDEFINES ganze Strukturen übernommen werden müssen, so sind alle nicht referenzierten Bereiche als FILLER zusammengefasst.

Der Entwickler macht folgende Angaben:

1. Beginn und Ende des auszulagernden Codes
2. Den gewünschten Namen des Sekundärprogramms
3. Eine Beschreibung des Programms in Form von COBOL Kommentaren
4. Name des Entwicklers

Die so entstehenden Primär- und Sekundärprogramme sind sofort ablauffähig und erbringen die Funktion des ursprünglichen Programms.

Bei entsprechender Granulierung sind solche Module bestens geeignet, als Service in einem Enterprise Service Bus eingesetzt zu werden.

FO – FORMATTING

Programmierstandards gab es früher oftmals nicht oder nur rudimentär. Als diese später eingeführt oder erweitert wurden, blieb selten die Zeit, den bestehenden Code nachzuarbeiten.

Darüber hinaus brachte jeder Programmierer seine eigenen Vorstellungen von Einrückungen und formaler Strukturierung ein.

GRANUM/FO erlaubt es, COBOL Standards auf einfache Weise zu definieren und bestehende Programme vollautomatisch neu zu formatieren. Die Funktionalität wird dadurch selbstredend nicht verändert.

Die Programme gewinnen durch diese Komponente deutlich an Lesbarkeit.

GRANUM REMEDIATION TOOLS

Viele Produkte wurden in den vergangenen Jahrzehnten innerhalb von COBOL eingesetzt, um

- die Produktivität der Entwickler zu verbessern
- einheitlich strukturierten Code zu erzeugen
- die Standardprobleme der Listenverarbeitung wie Gruppenwechsel, Sortierung, Zwischensummen usw. zu lösen
- Dokumentation direkt aus dem Quellcode heraus zu erzeugen
- Wartungsaktivitäten zu dokumentieren
- Change Management Informationen im Quellcode abzulegen

- Entscheidungstabellen innerhalb des Programms in einer Art Kurzschrift zu definieren
- Code über Makros zu erzeugen

Die betreffenden Produkte sind meist nur noch deshalb in Betrieb, weil man den betreffenden Code nicht automatisiert und risikolos zu vernünftigem nativen COBOL Code transformieren konnte.

Ein weiteres Thema ist die Pflege von Programmen, die aus 4GL-Sprachen wie CSP oder PACBASE generiert wurden, aber später auf der Basis des generierten Codes gewartet werden mussten, weil die 4GL-Sprache nicht mehr verfügbar war.

All diese Themen adressieren spezielle GRANUM Remediation Tools, die in die GRANUM-Suite eingebettet werden.

Für aktuelle Kundensituation entstanden beispielsweise GRANUM Zusätze, die

- SWT01 eliminieren
- VORELLE ablösen
- PACBASE COBOL umsetzen
- Aus COBOL Datenstrecken herauslesen und diese in Datenbankbeschreibungen umformen

Weitere Tools sind in Arbeit. **Sprechen Sie uns an:** Aufbauend auf den GRANUM Komponenten lassen sich mit Hilfe von RULAMAN/SPL mit vergleichbar wenig Aufwand beliebige weitere *Remediation Tools* entwickeln.

WEITERE ANWENDUNGEN

Wir wollen noch auf zwei weitere interessante Möglichkeiten hinweisen

1. Nachdem Assembler Programme mit dem IT-M Produkt A2C in COBOL transformiert und getestet wurden, bietet es sich an, gerade die großen Programme mittels GRANUM weiter zu verarbeiten und somit ihren Wert zu steigern. Nach einem Refactoring mit GRANUM/RF können logisch abgegrenzte Teile eines ehemaligen Assembler Programms als Service Routinen in einer SOA Architektur verwendet werden.
2. Vor einer voll- oder halbautomatischen Umsetzung von COBOL nach Java oder C# muss der Code unbedingt nach gewissen Regeln umgeformt werden, da sonst kaum lesbarer OO-Code entsteht. Bei der Entwicklung von GRANUM haben wir entsprechende Optionen berücksichtigt.

Die Vorteile der Modernisierung mit GRANUM

Mit Bezug auf die eingangs erwähnten Herausforderungen, die sich aus der riesigen Menge von COBOL Programmen und – wichtiger noch – aus dem Wert der darin enthaltenen Geschäftslogik ergeben, bietet GRANUM eine Reihe von eleganten Antworten.

- Sanierter COBOL Code, der gut lesbar und sauber strukturiert ist, der keine Reminiszenzen aus Technologien und Gewohnheiten des letzten Jahrhunderts enthält, wird auch von jungen Informatikern verstanden und gerne weitergepflegt und -entwickelt.

Dies kann die Nachwuchsproblematik in der Anwendungsentwicklung deutlich entschärfen.

- Die maschinelle Umstellung von COBOL auf eine OO-Sprache ist möglich, wenn der Code entsprechend positioniert wurde. Derartig umgesetzte Programme lassen sich dann innerhalb von reinrassigen OO-Umgebungen besser pflegen.

Notabene: Hier gilt es zuerst zu prüfen, welche Programme für diese Umsetzung in Frage kommen: Millionen von COBOL Zeilen unbesehen und stur in Java oder C# zu transformieren, ergibt wirtschaftlich und technisch keinen Sinn.

- Die bestehenden COBOL Monolithen werden automatisiert in handhabbare Teile zerlegt. Diese können dann leicht als externe Subroutinen zukünftiger Programme verwendet oder als Services in serviceorientierte Architekturen eingebettet werden.
- Der sanierte und in vielerlei Hinsicht verbesserte Code erleichtert und beschleunigt Fehlersuche, Wartung und Erweiterung ganz wesentlich.
- Die Abhängigkeit von antiquierten Drittprodukten wie Pre-Compilern, Codegeneratoren, Makrosystemen, Entscheidungstabellengeneratoren usw. ist vollständig verschwunden. Einige dieser Produkte sind noch lizenzpflichtig: diese Kosten entfallen jetzt.

Wichtiger aber ist die Eliminierung der Risiken von Drittprodukten, deren Urheber längst vom Markt verschwunden sind.

- Das Risiko der Umsetzung ist minimal, weil menschliche Fehler und Ermüdungserscheinungen ausgeschlossen sind.
- Ein Code Freeze entfällt, weil wir auch Millionen von Zeilen innerhalb weniger Stunden umsetzen können.
- Die Kosten der Transformation mit GRANUM betragen nur einen Bruchteil derjenigen einer manuellen oder halbautomatischen Umsetzung.

Wie können Sie die Möglichkeiten von GRANUM nutzen?

Wir bieten die Leistungen von GRANUM als Service zum Festpreis an.

Das bedeutet, dass wir Ihren Code gesamthaft oder in Paketen verarbeiten.
Die Modalitäten und genauen Parameter der Transformation vereinbaren wir vorher gemeinsam.

Bitte sprechen Sie uns auf Ihre Wünsche und Vorstellungen an:

GRANUM@it-modernisation.eu
oder telefonisch unter 040 / 8398 4130.