



Vorsprung
durch Software.
IT-Kosten einsparen.
Zukunft sichern.

Schwäbisch Hall 
Auf diese Steine können Sie bauen

”

Das Team der PKS habe ich während des kompletten Projektverlaufs auf allen Ebenen, vom Entwickler bis zum Geschäftsführer, als fachlich kompetenten und fairen Projektpartner kennen- und schätzen gelernt.

“

Bernd Schürle
Programmleiter
Bausparkasse Schwäbisch Hall

■ Das Unternehmen

Die Bausparkasse Schwäbisch Hall ist mit einem Marktanteil von rund 32 Prozent und über 7 Mio. Kunden die größte Bausparkasse in Deutschland. Mehr als 7.000 Mitarbeiter im Innen- und Außendienst sorgen für qualifizierte Beratung und optimalen Kundenservice. Seine Backend-Kernsysteme entwickelt das Unternehmen seit über 30 Jahren im eigenen Hause mit erfahrenen Entwicklern auf IBM Mainframe Systemen. Dabei kommen vor allem die Programmiersprachen Assembler, S-Cobol (AMB) und Cobol zum Einsatz. Durch die sehr hohe Flexibilität der Inhouse-Softwareentwicklung in Kombination mit den „Unique Features“ der Mainframe-Architektur konnten alle Anforderungen der Fachbereiche hinsichtlich Verfügbarkeit, Stabilität und Skalierbarkeit der Anwendungen stets bestens bedient werden.

■ Einblick

Im Zuge einer allgemeinen Konsolidierung der Entwicklungslandschaften unter z/OS und der Reduzierung von Risiken bei Host-Anwendungen hatte die Bausparkasse Schwäbisch Hall (BSH) die Aufgabe, die unter Microfocus AMB (AppMaster Builder) entwickelten S-COBOL Programme in die strategisch gesetzte Sprache M-COBOL zu transformieren. Als Zielentwicklungsplattform wurde IBM RDz mit dem BSH-eigenen MTOC-Plugin sowie einem neu bereitzustellenden Editor-Plugin für MFS-Masken definiert. Der Umfang der betroffenen Anwendungen umfasste etwa 600 IMS/DC-Programme, die mit der AMB Dialogkomponente OnlineExpress entwickelt wurden. Weitere ca. 1.200 Subprogramme und ca. 600 Batchprogramme wurden mit AMB entwickelt. Um die Entwickler von alltäglichen Aufgaben zu entlasten und eine entsprechend höhere Produktivität zu erreichen, hatte die BSH im AMB Umfeld etwa 200 User Makros im Einsatz. Diese Makros ermöglichen dem Anwendungsentwickler die Wartung des Codes an zentraler Stelle. Das Gesamtvolumen der Anwendungen umfasste 2.365 Programme und 9.164 Copy-Module, bestehend aus ca. 5,1 Mio Kommentarzeilen und ca. 12 Mio Codezeilen.

■ BSH stand daher vor den folgenden drängenden Fragen:

- Wie kann man einen komplexen Programmbestand von insgesamt über 15 Mio Zeilen vollautomatisiert und mit höchster Güte bezüglich der Wartbarkeit des Zielcodes transformieren?
- Mit welcher Konvertertechnologie kann man den mit GOTOs durchsetzten Code so restrukturieren, dass alle (!) von AMB generierten GOTO-Befehle durch strukturierte Anweisungen ersetzt werden und der Zielcode dadurch wieder wartbar wird?
- Wie kann man die hohe Entwicklerproduktivität mit AMB im Bereich der Online-Entwicklung auch unter IBM RDz gewährleisten?
- Können auch Teile der bisher bei der BSH halbmanuell durchgeführten Qualitätssicherung in der Softwareentwicklung („Q-Scan“) über eine Transformations-Engine vollautomatisiert erfolgen?

■ Ausblick

Ein spezifisches Transformationsverfahren auf Basis der PKS Konvertertechnologie und ein bewährtes Vorgehensmodell bieten der BSH eine 100%ig automatisierte Lösung für die Migration von S-Cobol nach M-Cobol bei kalkulierbaren Kosten und Risiken sowie einer sehr guten Wartbarkeit des Zielcodes.

Durch langjährige Erfahrung im Compilerbau und damit verbundenen Detailkenntnissen verfügt die PKS mit YYOP über die Technologieplattform, welche komplexe Syntaxstrukturen vollumfänglich analysieren, interpretieren und gemäß den Wartbarkeitsanforderungen des Kunden in die Zielsyntax generieren kann. Mit einem speziellen Tooling findet eine vollautomatische Restrukturierung der Sourcen statt. Dabei werden zunächst über umfangreiche Parsingläufe die von AMB generierten Cobolsourcen gelesen und alle Artefakte, Metainformationen und Abhängigkeiten in ein Repository eingetragen. Ein spezielles Pattern-Matching-Verfahren erkennt dabei mehrfach vorkommende Strukturen und kapselt diese in Copystrecken. Anhand der Zielcodedefinition, die gemeinsam mit dem Kunden festgelegt und von den Experten der PKS in einem Regelwerk implementiert wurde, erfolgt die Generierung in die Zielsprache M-Cobol.

■ Fazit

Damit die Auswahl der relevanten Fachtests optimal unterstützt werden kann, bediente sich das Projektteam außerdem an den eXplain Coverage Tools. Mit diesen kann der jeweilige Abdeckungsgrad von Syntaxstrukturen ermittelt und damit sichergestellt werden, dass gleiche Codekonstellationen in unterschiedlichen Programmen nicht unnötigerweise mehrmals den Fachttest durchlaufen müssen.

Für die Weiterentwicklung im Online-Bereich sorgt ein bedienfreundliches RDz-Plugin, der Screenpainter, das von PKS implementiert und nahtlos in den Software-Lifecycle bei BSH integriert wurde. Als Vorgehensmodell kam das sogenannte BigBang-Light-Verfahren zum Einsatz. Bei diesem Verfahren werden zunächst alle Sourcen transformiert und an den Kunden compilefähig übergeben. Die eigentliche Produktivsetzung und der zuvor notwendige Fachttest erfolgen dann im Rahmen der fortlaufenden Weiterentwicklung der Anwendungen. Dadurch schlagen weder Testkosten noch -zeiten direkt auf das Migrationsprojekts durch.

Natürlich erfolgt während dieser sogenannten Stabilisierungsphase bei Änderungsanforderungen durch die BSH eine unmittelbare Re-Konvertierung des Sourcebestands bei der PKS.

” PKS überzeugte uns mit der hohen Konverterkompetenz im Prototypen. Ich hätte nicht geglaubt, dass es möglich wäre, eine 100% automatisierte Transformationsengine bereitzustellen. Damit war für uns sichergestellt, dass das Vorhaben machbar sein wird.

Bernd Schürle
Programmleiter
Bausparkasse Schwäbisch Hall



■ Vorteile auf einen Blick

- ✓ Mit YYOP gelingt eine 100% automatisierte Transformation und Re-Engineering der Cobol-Sourcen mit einer sehr guten Wartbarkeit des Zielcodes
- ✓ Das Ersetzen aller von AMB generierten GOTO-Befehle durch strukturierte Anweisungen (Abfragen und Schleifen) ermöglicht eine sehr gute Lesbarkeit des Zielcodes. Neue Mitarbeiter finden sich nun sehr rasch im Code zurecht und können anfallende Wartungs- und Änderungsarbeiten zügig und fehlerfrei erledigen.
- ✓ Die Architektur des Systems wird verbessert und dadurch werden die Kosten bei der zukünftigen Wartung und Weiterentwicklung der Anwendungen minimiert. Dies geschieht durch die Trennung von Anwendungsschicht zu Technologieschicht und die Kapselung in Makros.
- ✓ Mit dem Screenpainter wurde ein Plugin zur Verfügung gestellt, das den AMB Screenpainter vollständig ersetzt und gleichzeitig die Wartbarkeit der Online-Programme gewährleistet.
- ✓ Die Wartungskosten für die AMB-Werkzeuge entfallen vollständig. Zukünftig muss kein Knowhow bzgl. AMB bei nachrückenden oder neuen Anwendungsentwicklern aufgebaut werden.
- ✓ Ein Prototyp, bei dem exemplarisch ein ausgewähltes Subset der Anwendung von PKS transformiert wurde, sicherte die technische Machbarkeit für die BSH bereits vor Projektbeginn zu 100% ab. Die zu erwartenden Ergebnisse konnten damit frühestmöglich mit Entwicklern diskutiert und bewertet werden.
- ✓ Auf Basis des Prototypen wurde das Projekt zum Festpreis von der PKS durchgeführt. Für die BSH ergab sich dadurch eine volle Kostentransparenz bis zum Projektende.
- ✓ Der bei einer Anwendungstransformation erforderliche kosten- und zeitintensive Fachtest konnte dank eXplain Coverage Tools minimiert werden.
- ✓ Der Umfang des Sourcecodes nach allen Optimierungsmaßnahmen ist sehr überschaubar und bewegt sich im Bereich Faktor 1,2-1,5 gegenüber dem originalen AMB Code und beträgt nur ca. 1/4 des von AMB generierten Codes (statt 15 Mio. Lines of Code nun 4 Mio. Lines of Code)

■ Tools und Verfahren

- YYOP
- zNative for Microfocus AMB
- eXplain Coverage Tools
- Dz Screenpainter for MFS