

# Das Debeka-Projekt MiKe – Migration der Debeka-Kernanwendungen von Bull/GCOS8 auf AIX

Johannes Bach, Martin Schulze

Debeka-Hauptverwaltung, Abteilung BO/Q, D-56058 Koblenz  
{johannes.bach,martin.schulze}@debeka.de

**Zusammenfassung:** In diesem Beitrag wird das bislang größte IT-Projekt in der Geschichte der Debeka vorgestellt: die Migration der Versicherungs- und Bauspar-Kernanwendungen von einem Bull-/GCOS8-Großrechner auf eine AIX-Plattform.

Aufgrund verstärkter Zweifel an der Zukunftssicherheit der GCOS8-Großrechnerplattform der Firma Bull fiel im Jahr 2002 die Entscheidung, die Kernanwendungen der Debeka auf eine möglichst herstellerunabhängige Plattform zu migrieren und mit den Vorarbeiten zu beginnen.

Wegen des zu sichernden Investitionsvolumens von mehr als eintausend Personenjahren kam nur eine 1:1-Migration in Frage, bei der sowohl die Programm- und Job-Sourcen als auch die Daten automatisiert konvertiert werden. Bedingt durch eine phasenweise Vorgehensweise, in der einzelne Anwendungen paketweise migriert werden, entstand für eine Übergangszeit ein Hybridsystem, in dem das alte GCOS-System und das neue AIX-System durch Brücken miteinander verbunden sind. Obwohl ein auf dem Gebiet von GCOS8-Migrationen erfahrener Dienstleister gewonnen werden konnte, ergaben sich im Lauf des Projekts unerwartete Herausforderungen beispielsweise bei der Abbildung der migrierten CODASYL-Datenbank auf eine relationale Datenbank mit anderen Sperrmechanismen, welche jedoch zeitnah bewältigt werden konnten.

Das Projekt wurde zusätzlich genutzt, einen neuen Softwareentwicklungsprozess auf Basis einer modernen Entwicklungsumgebung im Unternehmen einzuführen und damit sowohl die Qualität als auch die Produktivität der Softwareentwicklung entscheidend zu verbessern.

## 1 Unternehmen

Die *Debeka-Gruppe* gehört zu den Top-Ten der deutschen Versicherungsbranche und besteht im Wesentlichen aus mehreren rechtlich eigenständigen Unternehmen: dem Krankenversicherungsverein a.G., dem Lebensversicherungsverein a.G., der Allgemeinen Versicherung AG, der Pensionskasse AG und der Bausparkasse AG. Im Gegensatz zu vielen anderen Versicherungskonzernen betreibt die Debeka schon von Anfang an ein übergreifendes Rechenzentrum für alle Teilunternehmen und entwickelt ihre Software in einer Hauptabteilung der Hauptverwaltung in Koblenz.

Zentraler Bestandteil der IT-Landschaft sind die *Debeka-Kernsysteme*, die in COBOL geschrieben sind, auf einem CODASYL-Datenbanksystem beruhen und auf einem Bull-Großrechner mit Betriebssystem GCOS8 laufen.

## 2 Die Kernsysteme der Debeka

Bei diesen Debeka-Kernsystemen handelt es sich um ein *hochintegriertes Anwendungssystem*, das die einzelnen *Fachanwendungen* (Bestands-/Vertragsverwaltung, Leistungsbearbeitung etc.) der einzelnen Unternehmen beinhaltet und miteinander verbindet (Abbildung 1). Mit diesen Anwendungen werden ca. 12 Mio. Versicherungs- und Bausparverträge von ca. 6 Mio. Kunden verwaltet.

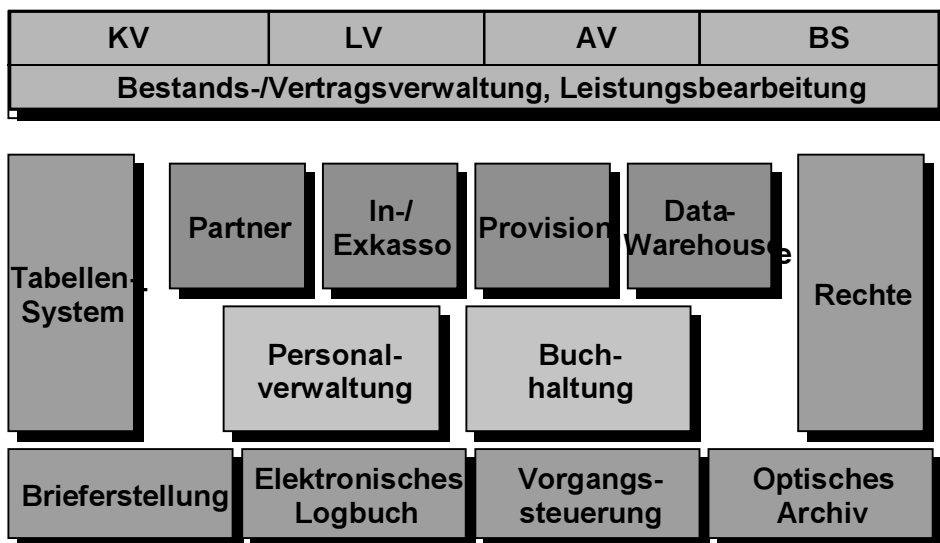


Abbildung 1: Debeka-Kernsysteme: Fach- und Serviceanwendungen

Zentrale *Serviceanwendungen* stellen den Fachanwendungen unternehmensübergreifende allgemeine Funktionen (Verwaltung von Mitgliederdaten, Inkasso und Exkasso, Personalverwaltung, etc.) über entsprechende Zugriffsroutinen (COBOL-Unterprogramme) zur Verfügung (Abbildung 1). Die Kernsysteme werden von ca. 5500 gleichzeitigen Online-Usern verwendet, die mehr als 2 Mio. Transaktionen pro Tag durchführen. Insgesamt handelt es sich um ca. 8000 COBOL-Programme mit 8 Mio. LOC und 10000 Jobs mit 1 Mio. LOC; die Datenbanken haben ein Volumen von ca. 2 TByte.

### 3 Gründe für die Migration

Aufgrund zunehmender Unsicherheit hinsichtlich der wirtschaftlichen Rahmenbedingungen und der damit verbundenen langfristigen Überlebensfähigkeit der GCOS8-Plattform wurde eine *Neuausrichtung der Debeka-IT* erforderlich. Indizien hierfür waren unter anderem:

- Aufkündigung der CMOS-basierten Großrechner-Produktlinie,
- Wegfall von Wissen bzw. Wissensträgern (insbesondere in Deutschland),
- schwindende Kundenbasis,
- auslaufender oder fehlender Support durch den Hersteller,
- Ressourcenmangel und Skalierungsprobleme.

Auf Empfehlung des IT-Managements beschloss der Debeka-Vorstand daraufhin im Jahr 2002, die GCOS8-Plattform abzulösen und mittelfristig auf eine neue, zukunftssichere und möglichst herstellerunabhängige Technologieplattform zu migrieren.

Dabei war den an der Entscheidung beteiligten Personen durchaus bewusst, dass die Migration

- Einfluss auf fast alle Geschäftsbereiche der Debeka haben wird,
- höchste Priorität gegenüber anderen geplanten Vorhaben und Projekten erfordert und diese damit verzögern oder bei nicht verzögerbaren Vorhaben (z.B. aufgrund gesetzlicher Vorgaben) zu einem erhöhten Ressourcenbedarf führen wird,
- zumindest zeitweise einen beträchtlichen Teil der technischen und fachlichen Ressourcen aus den verschiedenen Bereichen binden wird.

Deshalb wurde das Bewusstsein geschaffen, dass das Migrationsprojekt kein normales Projekt der IT-Abteilungen ist, sondern ein *Projekt des Debeka-Gesamtunternehmens*. Es wurde klar, dass die Migration *lebensnotwendig* ist, auch wenn sie aus betriebswirtschaftlicher Sicht kurz- bis mittelfristig keine Vorteile bringt, sondern im Gegenteil extrem kostenintensiv ist, wertvolle Ressourcen bindet und enorme Risiken birgt.

### 4 Vorarbeiten

Aufgrund der Vorgaben für die künftige Zielplattform wurden im Rahmen einer *Vorstudie* im Jahr 2003 zunächst verschiedene Szenarien offener Systemplattformen (AIX- und Linux-Varianten mit verschiedenen Komponentenkombinationen) sowie deren prinzipielle Eignung für den Betrieb der Debeka-Kernanwendungen untersucht. Wesentliches Ergebnis dieser Vorstudie war eine Empfehlung für eine Unix-Plattform auf Basis eines kommerziellen Unix-Derivates (z.B. IBM pSeries unter AIX) in Verbindung mit dem Einsatz einer relationalen Datenbank (z.B. ORACLE), eines Transaktionsmonitors (z.B. BEA Tuxedo) sowie eines J2EE-Applikationsservers (z.B. BEA Weblogic).

Darauffin wurde in einer *Migrationsstudie* zunächst eine Entscheidungsvorlage zur Auswahl der technischen Zielplattform erarbeitet, in der die vorgeschlagene Unix-Variante als Alternative einer klassischen Großrechnerplattform (IBM zSeries unter z/OS) gegenübergestellt und aufgrund aller vorliegenden Informationen bewertet wurde. Im Herbst 2004 fiel die Entscheidung für die AIX-Variante.

Auf Basis der getroffenen Plattformsentscheidung wurde Ende 2004 im Rahmen der Migrationsstudie eine detaillierte Migrationsstrategie erstellt. In ihr sind der Migrationsprozess und die einzelnen Migrationsschritte konkret beschrieben:

- Untersuchung und Bewertung verschiedener Dienstleister und derer Werkzeuge zur Konvertierung der Anwendungen und der Daten,
- Beschreibung des Vorgehensmodells für die Migration der Anwendungen und der Datenbestände sowie Festlegung der Reihenfolge der Migration,
- Festlegung der Maßnahmen für die Pilotphase sowie durchzuführende Tests, ob die Ergebnisse der Pilotphase die Weiterführung des Projekts erlauben,
- Ausarbeitung eines Masterplans für das gesamte Migrationsvorhaben.

Im Januar 2005 begannen dann die Arbeiten im eigentlichen *Migrationsprojekt*, das bis zum Herbst 2008 abgeschlossen sein soll.

## 5 Aufgabenstellung

Zur Erarbeitung einer Migrationstrategie war eine detaillierte und umfassende *Analyse* der zu migrierenden Anwendungs- und der bestehenden Systemlandschaft notwendig. Zunächst wurde die Ist-Situation analysiert und die softwaretechnische und fachliche *Architektur* der Kernsysteme dokumentiert. Dabei entwickelte sich das für alle Beteiligten notwendige Verständnis der Aufgabenstellung.

Schließlich konnten folgende *Aufgabenpakete* identifiziert und beschrieben werden:

- Wechsel von Hardware und Betriebssystem,
- Wechsel des Datenbankmanagementsystems und des damit verbundenen DB-Paradigmas,
- partielle Neuentwicklung systemnaher Infrastrukturkomponenten für das Zielsystem sowie für das Hybridsystem während der paketweisen Migration,
- Konvertierung und Test aller GCOS8-Geschäftsanwendungen,
- Übernahme und Konvertierung sämtlicher Datenbestände (Datenbanken, Dateien und Bandbestände),
- Änderung der Betriebsprozesse und der Organisationsstrukturen im Systembetriebsbereich,
- Wissensaufbau und Schulung von Systemadministratoren und Betriebsmannschaft,
- Konzeption, Bereitstellung und Schulung neuer Softwareentwicklungsprozesse und -werkzeuge.

Zur Bewältigung dieser Aufgaben wurde eine *Projektstruktur* gewählt, in der die einzelnen Aufgabenblöcke Teilprojekten zugeordnet wurden (Abbildung 2). Die inhaltliche Koordination der Aufgaben und Ergebnisse obliegt dabei neben der Projektleitung – unterstützt durch eine QS – insbesondere dem Teilprojekt PAP (Koordination und Abstimmung der Migrationsaufgaben und –aktivitäten pro Phase und Anwendungspaket).

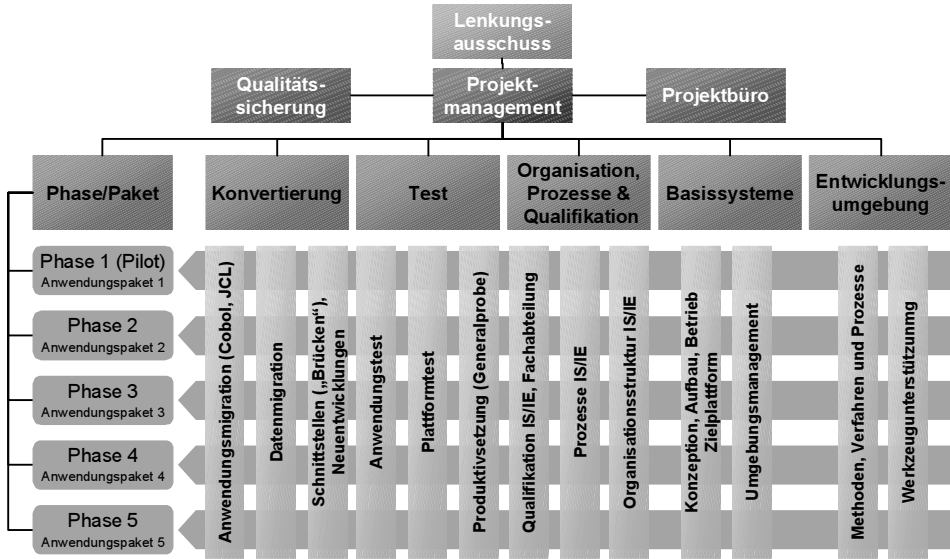


Abbildung 2: Projektaufteilung in Teilprojekte und Phasen

Die Leitung des Gesamtprojekts und die Leitung der einzelnen Teilprojekte besteht ausnahmslos aus Debeka-Mitarbeitern. Zur Verstärkung wurden externe Mitarbeiter der Unternehmensberatung Accenture bedarfsgerecht hinzugezogen. Die Aufgabe der Konvertierung von Anwendungscode und Daten wurde in die Hände des französischen Unternehmens Metaware gelegt. Beim Aufbau der neuen AIX-basierten Systemumgebungen unterstützt der langjährige Systempartner Bull.

## 6 Migrationsstrategie und Vorgehensweise

In der Migrationsstudie wurde nach Abwägung aller Einfluss- und Risikofaktoren beschlossen, die Migration von Code und Daten in einer klassischen *1:1-Migration in mehreren Phasen* durchzuführen. Dabei sollte die Konvertierung weitestgehend automatisiert durch Werkzeuge erfolgen. Andere als die für eine automatisierte Konvertierung unbedingt erforderlichen technischen und funktionalen Änderungen und Anpassungen an Programmen, Daten und Prozessen werden im Rahmen des Projektes nicht durchgeführt.

Die vorhandenen COBOL-Programme unter GCOS8 werden soweit wie möglich unverändert nach AIX überführt. Die IDS-II Datenbank auf GCOS8 wird in eine relationale ORACLE-Datenbank überführt, wobei Datenfelder und Zeichensätze konvertiert werden und die bisherigen Inline-Datenbankzugriffe durch syntaktisch ähnliche Aufrufe entsprechender COBOL-Routinen auf Basis einer generierten Zugriffsschicht ersetzt werden (Beispiel 1). Die GCOS-Batchjobs (JCL) werden weitgehend in Shellskripte (KSH) unter AIX überführt, wobei die Struktur erhalten bleibt, aber die JCL-Anweisungen durch einen vorgegebenen Satz von ksh-Skripten ersetzt werden (Beispiel 2).

```
...
MOVE ZERO TO FEHL-C
FIND NEXT <record> WITHIN <set>
IF DB-STATUS ...
```

```
...
MOVE ZERO TO FEHL-C
* FIND NEXT <record> WITHIN <set>
MOVE "NEXT" TO NAVI-MODE
MOVE "<record>" TO REC-NAME
CALL "FIND-WITHIN-SET" USING ...
IF DB-STATUS ...
```

Beispiel 1: Inline-Datenbankzugriffe im COBOL-Code vor der Migration (oben) und Aufruf von COBOL-Callroutinen auf Basis der Zugriffsschicht nach der Migration (unten)

```
$ *
$ KL5252.
$ SELECT KL-OBJECT/KV/KV7/KL5252O
$ EXECUTE DUMP
$ LIMITS 100,200K,-5K
$ PRMFL SA,R,R,KL-WORK/KL/PKUES05
$ FILE SZ,X1S,200R

$ SYSOUT P*,*Z
$ SYSOUT J*,*Z
$ *
```

```
# *
(KL5252)
m_FileAssign -m R -a R SA ${DKS_DATA}/work/kl/pkues05.dat
m_FileAssign -m W -a X SZ ${MT_TMP}/${MT_JOB_PID}_X1_${MT_JOB_NAME}
# *
m_ProgramExec kl5252
JUMP_LABEL=K5221L
;;
```

Beispiel 2: JCL-Batchjob vor der Migration (oben) und ksh-Skript nach der Migration (unten)

Die zeichenorientierte *Benutzeroberfläche* der Hostanwendungen wurde bereits Ende der 90er Jahre auf ein vorgelagertes Java-basiertes GUI (JeT – Java enhanced Terminal) umgestellt und war somit nicht migrationsrelevant. Das dazu gehörende GCOS8-basierte Masken-/Dialogsystem SOLD wurde im Rahmen der Migration auf einem ebenfalls vorgelagerten J2EE-Applikationsserver neu implementiert, wobei ein Switch eingerichtet werden konnte, der die Benutzeranfragen auf das jeweilige Zielsystem weiterleitet.

Weiterhin wurde im Vorfeld der Migration der GCOS8-Job-Scheduler durch den systemübergreifenden *Scheduler* UC4 abgelöst, wodurch systemübergreifende Job-abläufe möglich wurden.

Während sich der produktive Code eines Migrationspakets zur Konvertierung beim externen Dienstleister befindet, sind Änderungen im Rahmen der normalen Wartung unvermeidlich. Diese Änderungen müssen gegen Ende der jeweiligen Konvertierungsphase und vor der Produktivsetzung auf der Zielplattform im Rahmen so genannter „*Delta-Migrationen*“ in den konvertierten Code eingepflegt werden. Um den Aufwand und die Risiken, die mit den Delta-Migrationen verbunden sind, zu reduzieren, sollten Parallelentwicklungen auf das absolut Notwendige reduziert werden („Wartungsarmes Intervall“).

Die *phasenweise Migration* wurde vor allem zur Reduktion des Risikos sowie zur Entzerrung des notwendigen Personalbedarfs angestrebt. Die Migration in mehreren Paketen ermöglicht darüber hinaus eine bessere Abstimmung des Projektes auf die Bedürfnisse und Möglichkeiten der Debeka sowie eine flexiblere Reaktion auf eventuelle Veränderungen des Geschäftsumfeldes. Eine Einschnitt-Migration („Big-Bang“) schien – aufgrund der Größe der Anwendungslandschaft und der sich daraus ergebenden hohen Komplexität, des hohen Personalbedarfs sowie der langen Dauern von Konvertierung, Test und Produktivsetzung – nicht sinnvoll. Darüber hinaus würde sich das Risiko – bei einer fehlgeschlagenen Produktivsetzung bzw. bei Problemen danach – auf alle Unternehmensbereiche der Debeka-Gruppe ausweiten.

Die zur Umsetzung der phasenweisen Migration übergangsweise notwendigen *Datenbrücken* (zur ihrer Identifikation siehe [BDS2006]) zwischen den GCOS8- und AIX-Systemen wurden mit Hilfe einer eigens für die Debeka entwickelten Bull-Technologie XDBSP (eXtended Database Satellite Processing) und weiteren Standardtechniken wie der Flowbus Message Queue und FTP realisiert. Der phasenweise Migrationsansatz stützt sich stark auf die prinzipielle Machbarkeit und ausreichende Leistungsfähigkeit dieser Bull XDBSP-Daten- und Anwendungsbrücken. Ein entsprechender Nachweis mittels eines Prototyps wurde in der Vorbereitungsphase erbracht.

Zwischen dem Produktionsstart der konvertierten Anwendungen der ersten Migrationsphase im November 2007 und dem Abschluss des Migrationsprojektes voraussichtlich im Herbst 2008 werden die existierende GCOS-Plattform und die neue AIX-Plattform als *Hybridsystem* parallel betrieben (Abbildung 3). In dieser Zeit werden

die mit dem heutigen Systembetrieb betrauten Debeka-Mitarbeiter sukzessive in Betriebsaufgaben der neuen Plattform eingebunden. Allerdings wird während des Parallelbetriebs mehr Personal benötigt, um die beiden Produktiv- sowie auch zusätzliche Test- und Entwicklungsumgebungen zu betreuen.

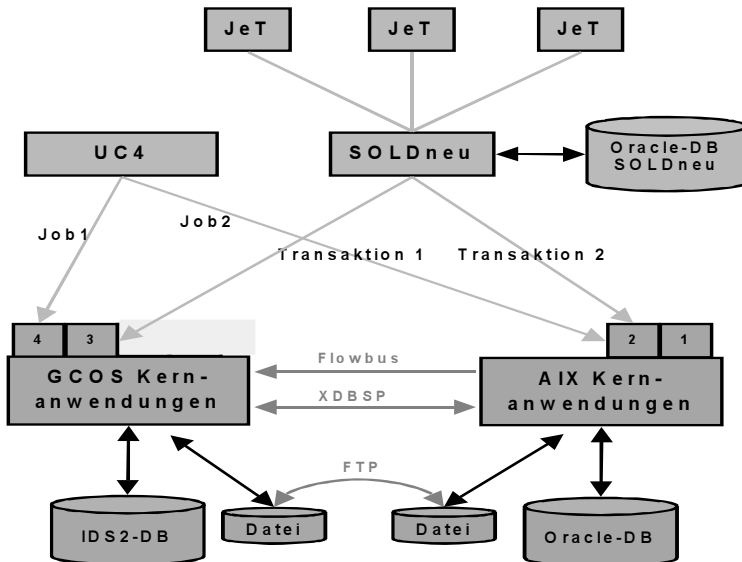


Abbildung 3: Hybridsystem während der Migrationsdauer

Schließlich ergab sich folgende *Vorgehensweise*:

- Phasenübergreifende Vorarbeiten:
  - Programm- und Datenqualifizierung
  - Bereinigung der Anwendungssysteme
  - Erstellung eines Prototyps und Justieren der Konvertierungswerkzeuge und -Prozesse
- Arbeiten pro Phase/Paket:
  - Konvertierung des Anwendungspakets und Bereitstellung der (Test-) Umgebungen
  - „Wartungsarmes Intervall“ mit ausführlichen Tests der konvertierten Anwendungen (Dauer: einige Wochen)
  - „Wartungsfreies Intervall“ (Dauer: zwei Wochen)
    - Konvertierung der Änderungen (Delta-Migration)
    - (Re-)Tests
    - Auf-/Umbau der produktiven Zielumgebung
  - Umstellung („switch over“, Dauer: ein Wochenende)
    - Produktivsetzung des Pakets auf dem Zielsystem
    - Abschaltung auf dem GCOS-System
    - Konvertierung der Datenbanken und Dateien des Pakets
    - Inbetriebnahme der Brückenmodule



Die ursprüngliche Planung zur Durchführung der Migration in fünf Phasen wurde zwischenzeitlich aufgrund der gewonnenen Erfahrungen auf zwei Phasen reduziert (Abbildung 4).

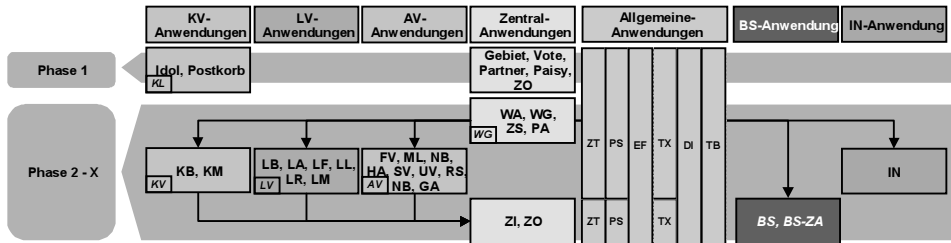


Abbildung 4: Verteilung der Anwendungssysteme auf Projektphasen

## 7 Besondere technische Herausforderungen

Durch die intensive Vorbereitung und die umfangreiche Pilotphase gelang es, die meisten konzeptionellen und technischen Risiken im Vorfeld aufzudecken und auszuschließen. Trotzdem gab es natürlich kleinere „Überraschungen“, die jedoch in vertretbarer Zeit gelöst werden konnten. Berichtenswert sind drei ernstere Probleme:

- Für die Tests vor Produktivsetzung der ersten Phase sowie für die Wartung des laufenden Hybridsystems bis zum Abschluss der Migration mussten auch entsprechende *Test-Umgebungen* zur Verfügung gestellt werden. Insbesondere bis zur ersten Phase wurde der Test-Großrechner aber auch für die Tests in der laufenden Entwicklung des GCOS-Systems benötigt. Somit war ein längerer Test unter den realen Bedingungen nach der Migration nur an speziellen Testwochenenden möglich. Der Auf- bzw. Umbau der finalen Entwicklungs- und Produktionsumgebungen am Switch-Over-Wochenende war extrem aufwändig und komplex.
- Zur Migration der CODASYL-Datenbank auf die relationale Oracle-Datenbank wurden Regeln zur Abbildung netzwerkartiger Datenbankstrukturen auf relationale Strukturen definiert. Daraus konnte ein Generierungsframework erstellt werden, das neben der Oracle-DDL auch die entsprechenden *Zugriffsschichten*, welche die vorhandene Zugriffslogik auf elementare SQL-Befehle abbildet, generiert. Nur so konnten manuelle Eingriffe in die Anwendungsprogramme vermieden werden. Ein Nachteil dieses Ansatzes besteht jedoch darin, dass auf die so erstellten Oracle-Datenbanken ohne Verwendung der Zugriffsschicht (COBOL-Unterprogramme) nicht zugegriffen werden kann.
- Aufgrund der unterschiedlichen Sperrparadigmen von GCOS8 und Oracle (Fehlen des Isolation Levels „Repeatable Read“) entschied man sich, die Datenintegrität durch die Verwendung von SQL-SELECT-FOR-UPDATE-Statements sicherzustellen. Dadurch traten im Rahmen der ersten Lasttests verstärkt Konflikte bei konkurrierenden Zugriffen auf gemeinsame Datenbanktabellen auf. Eine saubere Lösung für dieses prinzipielle Problem hätte dann doch noch einen Eingriff in die

Anwendungsprogramme bedeutet. Um dies zum umgehen, entschied man sich für eine Steuertabelle, die der Zugriffsschicht pro Transaktion und DB-Tabelle mitteilt, ob ein normales SQL-SELECT- oder ein SQL-SELECT-FOR-UPDATE-Statement zu verwenden ist.

## 8 Aufbau einer neuen Softwareentwicklungsumgebung

Das Teilprojekt SEU (Softwareentwicklungsumgebung) hatte die Aufgabe, eine neue *Softwareentwicklungsumgebung* für die AIX-Plattform zu entwickeln und einzuführen. Ausgehend von der vorhandenen GCOS8-Entwicklungsumgebung, welche über Jahrzehnte gewachsen war, galt es nun die damit verbundenen und etablierten Softwareentwicklungsprozesse zu analysieren und den veränderten Rahmenbedingungen anzupassen. Dabei wurden die folgenden Aspekte und Anforderungen berücksichtigt und sinnvoll gegeneinander abgewogen:

- bewährte und von den Entwicklern gewohnte Verfahren sollten beibehalten werden, um den Lernaufwand gering zu halten,
- zwingend notwendige Änderungen aufgrund der neuen Gegebenheiten mussten eingeführt werden,
- sinnvolle Verbesserungen sollten bei einem vertretbaren Lern- und Arbeitsaufwand für die Entwickler eingeführt werden,

wobei die verfügbare Zeit bis zur Produktivsetzung der Anwendungspakete der ersten Phase für die Neukonzeption und Einführung berücksichtigt werden musste.

Das Team sah in dieser Aufgabenstellung die einmalige Chance, eine Entwicklungsumgebung zu schaffen, die zum einen zeitgemäß, zum anderen aber auch flexibel neuen Anforderungen gewachsen ist, die in den nächsten Jahren oder sogar Jahrzehnten entstehen werden. Herausragende Merkmale dieser neuen Softwareentwicklungsumgebung sind:

- alle Entwicklungswerkzeuge (textuelle und graphische Editoren für Quelltexte, Datenbankschemata, Jobpläne, ...) sind als *Eclipse-Plugins* realisiert. Dadurch steht den Entwicklern eine integrierte Umgebung zur Verfügung, die die Fähigkeiten von Eclipse (individuelle Gestaltung der Oberfläche, Syntax-Hervorhebung, Eingabeunterstützung, Navigation in und zwischen Dokumenten, Darstellung von Syntaxfehlern, ...) ausnutzt (Abbildungen 5 und 6),
- zu jedem Entwicklungsartefakt existieren *lesbare Sourcecode-Dateien* im Text- oder XML-Format, die beim Wechsel zwischen Entwicklungs-, Test- und Produktionsumgebung in Subversion revisionssicher versioniert werden,
- alle *Entwicklungsprozesse* (für Programme, Scripte, Datenbanken, Jobpläne, Masken, ...) wurden möglichst vereinheitlicht,
- alle Entwicklungstätigkeiten werden über *ZEDER*, das zentrale Repository der Debeka ([BS2006]), in einem Auftragsverwaltungssystem verwaltet und revisionssicher dokumentiert,



Abbildung 5: COBOL-Workbench auf Basis von Eclipse

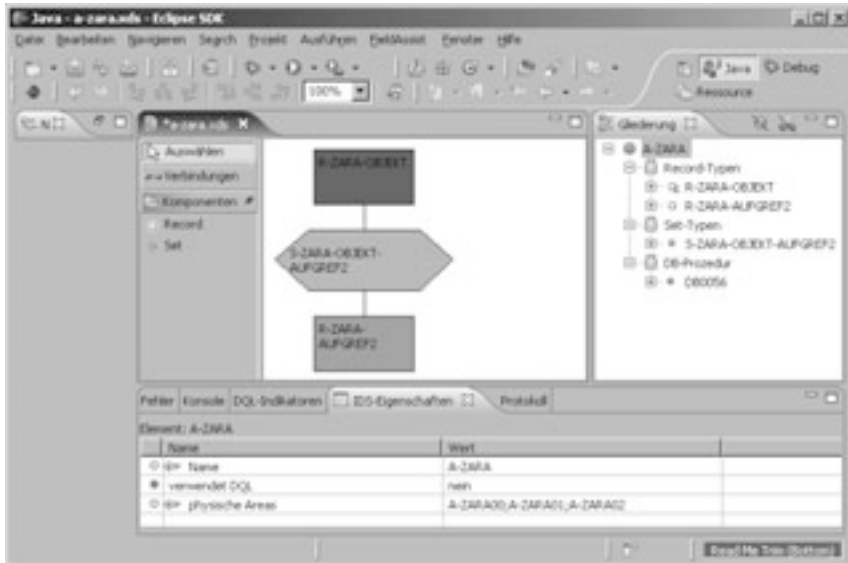


Abbildung 6: IDS-Designer für virtuelle IDS-Datenbanken

- der Entwicklungsprozess ist weiterhin *modulbasiert*, wobei tägliche Produktionsübernahmen durchgeführt werden und keine wochen- oder monatelangen Releasezyklen entstehen,
- neben der Produktionsumgebung und einer integrierten Standardtestumgebung arbeiten die Entwickler in privaten *Entwicklungsumgebungen* oder in individuell konfigurierbaren *Teamtestumgebungen*. Test- und Entwicklungsumgebungen können isoliert gehalten werden oder Module aus „höheren“ Umgebungen referenziell verwenden (Abbildung 7).

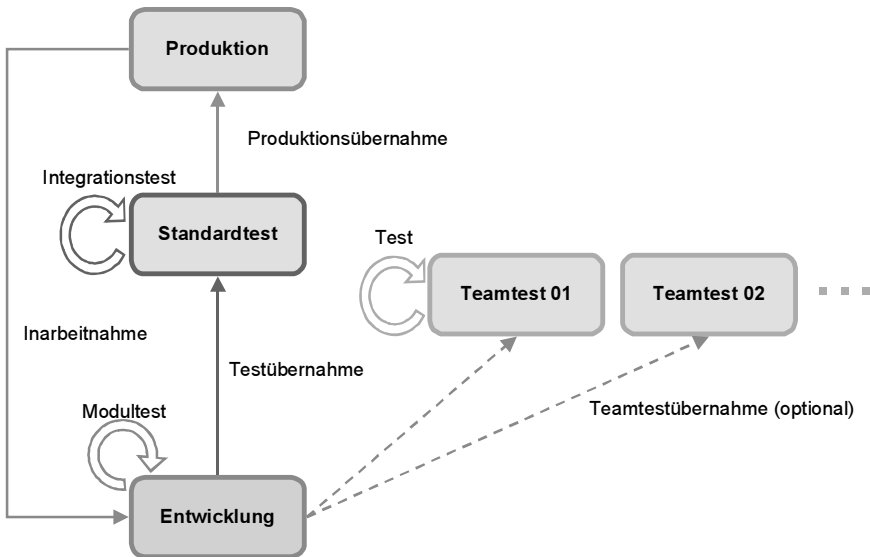


Abbildung 7: Produktions-, Test- und Entwicklungsumgebungen

## 9 Ausblick auf weitere Arbeiten im Projekt und nach dem Projekt

Im *November 2007* wurde das erste Anwendungspaket – und damit auch die neue Entwicklungsumgebung – erfolgreich auf der neuen AIX-Plattform produktiv gesetzt und mit der GCOS-Umgebung über entsprechende Brücken verbunden. Die Produktivsetzung der restlichen Anwendungssysteme und die Abschaltung des GCOS8-Systems ist für *Herbst 2008* geplant. Diese geänderte Vorgehensweise hat insbesondere den Vorteil, dass für die finale Zielumgebung – und damit auch für die für Phase 2 benötigten Testumgebungen – keine Hybridumgebung benötigt wird, was die Komplexität erheblich reduziert.

Nach Abschluss der eigentlichen Migration „warten“ jedoch bereits weitere Aufgaben, bei denen es darum geht, die Nutzenpotentiale der „offenen“ AIX-Plattform zu erschließen. Das Teilprojekt SEU beschäftigt sich zur Zeit beispielsweise mit der Frage,

wie die *relationale Datenbanktechnologie* nativ eingesetzt und entsprechend von den COBOL-Anwendungsprogrammen genutzt werden soll.

Weitere Nutzenpotentiale werden vor allem in der besseren Integration der Debeka-Kernsysteme mit den *Randsystemen* gesehen. Die bisherigen, teilweise proprietären Lösungen zur Kopplung der Systeme sollen abgelöst und zukünftig durch Standard-Technologie ersetzt werden.

## 10 Fazit

Aufgrund eines sehr hohen Integrationsgrades der einzelnen Anwendungssysteme (wie es bei den Debeka-Kernsystemen der Fall ist) wächst der Aufwand für die Realisierung von Brücken und für den Betrieb der Hybridumgebungen exponentiell. Damit werden die generellen Vorteile eines phasenweisen Vorgehens bei der Migration durch den größeren Aufwand beim Betrieb eines Hybridsystems durchaus in Frage gestellt.

Entgegen der ursprünglichen Auffassung, dass die größte Herausforderung eines solchen Migrationsprojektes in der Konvertierung des Programmcodes und der Daten der Anwendungssysteme liegt, kam man im Laufe des Projektes zur Erkenntnis, dass dies das geringere Problem darstellt. Aufgrund des hohen Automatisierungsgrades der Konvertierungswerkzeuge und der Erfahrung des externen Dienstleisters entstanden dadurch keine größeren Projektrisiken. Schwieriger war auf der technischen Seite die zusätzliche *Komplexität* durch die Einführung der Brückentechniken und der Einsatz der Programme unter den unterschiedlichen Sperrmechanismen.

Auch auf der organisatorischen Seite ergab sich eine besondere Herausforderung. Bisher wurden neue Technologien vor ihrer Einführung bei der Debeka intensiv untersucht, zunächst in einem überschaubaren Bereich eingeführt und dann aufgrund erster Erfahrungen aus der Praxis generell freigegeben. Bei der Migration der Kernsysteme war dies nicht möglich – aufgrund des engen Zeitrahmens im Projekt musste eine Fülle neuer Technologien, Systeme und Prozesse geschult und *in kurzer Zeit parallel* eingeführt werden.

## Literaturverzeichnis

- [BDS2006] Bach, Johannes; Doppler, Max; Schulze, Martin: *Repositorygestützte Erkennung von Schnittstellen in einem hochintegrierten Anwendungssystem*. In: Kaiser, Uwe; Kroha, Petr; Winter, Andreas: 3. Workshop Reengineering Prozesse (REPRO 2006). Informatik-Bericht 2/2006, Institut für Informatik, Universität Mainz, 2006.
- [BS2006] Bach, Johannes; Schulze, Martin: *Migration des Debeka-Software-Repositorys auf ein RDBMS*. In Rainer Gimnich, Andreas Winter (Hg.): 8. Workshop Software-Reengineering, Softwaretechnik-Trends, Band 26(2), 2006.