

Fallstudie SUVA

Zur Maximierung ihres Return on Investment setzt die SUVA bei der Integration ihrer COBOL-Legacy-Anwendungen in eine verteilte Thin-Client-Architektur die generativen Entwicklungswerkzeuge von Delta Software Technology ein.

Die Suva, die Schweizerische Unfallversicherungsanstalt, versichert als nationaler Unfallversicherer in der Schweiz 100.000 Unternehmen und 1,9 Millionen Arbeiter und Angestellte – mehr als 50 % der erwerbstätigen Bevölkerung.

Das Unternehmen sah sich der Herausforderung gegenüber, kritische Anwendungen von OS/2 Fat-Clients auf eine verteilte Thin-Client-Architektur zu migrieren.

In diesem Bericht erfahren Sie, wie es der Suva mit Hilfe generativer Entwicklungswerkzeuge von Delta Software Technology gelang, durch die Wiederverwendung von 5.000.000 Zeilen COBOL-Code Service-Komponenten zu erstellen.

„SCORE Adaptive Bridges hat unsere komplexe Legacy-Architektur gemeistert. Wir haben uns für die richtige Lösung entschieden.“

Robert Koch
CIO, Suva



ÜBERBLICK ÜBER DIE LÖSUNG

Kunde

Die Suva, die Schweizerische Unfallversicherungsanstalt, versichert 100.000 Unternehmen und 1,9 Millionen Personen – über 50 % der erwerbstätigen Bevölkerung der Schweiz.

Herausforderung

Maximierung der Investitionsrentabilität (Return on Investment) bei der Migration einer COBOL-Legacy-Anwendung von einer Fat-Client- auf eine verteilte Thin-Client-Architektur.

Lösungen von Delta Software Technology

- SCORE® Adaptive Bridges™ (früher: SCORE® Integration Suite™)
- SCOUT2™ Development Platform
- PBE Pattern By Example™
- ANGIE Frame Generator™

Warum Delta Software Technology?

Die Suva wurde in die Lage versetzt, bei der Schaffung von Komponenten mit Service-Schnittstellen für Java GUI-Clients mehr als 5.000.000 Zeilen vorhandenen COBOL-Code wiederzuverwenden.

Wichtige Vorteile für den Kunden

- Maximierung des ROI für bestehende COBOL-Legacy-Anwendungen.
- Integration mit Java GUI heute und mit J2EE/EJB usw. morgen – ohne zusätzlichen Aufwand.
- Generierung von Native Code unterstützt hohe Transaktionsvolumen.
- Plattform-übergreifender Support verhindert Bindung an eine Plattform.
- Generative Programmieretechniken ermöglichen flexible Lösungen für Projektanforderungen.
- Ständige Steigerung der Produktivität von Entwicklern um 400 % bis 600 %.

Partnerunternehmen

SAXOS Informatik AG
www.saxos.ch

„Für dieses Projekt setzten wir einen sehr engen Zeitplan. Wir waren beeindruckt, dass wir durch die Verwendung von SCORE Adaptive Bridges eine ständige Produktivitätssteigerung von 400 % bis 600 % erreicht haben.“

*Thomas Müller,
Leiter Anwendungsarchitektur und Methoden,
Suva*

PROJEKT-ZEITPLAN

1998	Die Suva beginnt mit der Projektplanung für die Migration.
Anfang 1999	Die Suva identifiziert über eine Internet-Suche mit den Suchbegriffen „COBOL Migration“ Delta Software Technology als potentiellen Lieferanten für das Projekt.
Juni 1999	Die Suva nimmt Kontakt mit SAXOS auf.
Ende 1999	Erster Prototyp zur Prüfung des SCORE-Konzepts. Die letztendliche Einsatzplattform der Suva steht noch nicht fest. Der Prototyp wird für verschiedene Plattformen und Infrastrukturkombinationen generiert.
Anfang 2000	SCORE Adaptive Bridges wird für das Projekt ausgewählt. Pilotprojekt zur Prüfung der Vorgehensweise und zur Entwicklung der unterstützenden Infrastruktur, Tools und Methoden. Suva entscheidet sich für eine Architektur mit Thin Clients, Citrix MetaFrame, Windows 2000-Servern, Java GUI-Clients, Oracle Tuxedo-Middleware (früher: BEA) und WebLogic Enterprise-Anwendungsservern sowie IBM AIX-Servern. Die Datenbank bleibt IBM DB2 auf IBM S/390-Großrechnern, auf die über DB2 Connect zugegriffen wird.
Ende 2000 bis Juli 2002	Das Haupt-Migrationsprojekt mit bis zu 60 Entwicklern erstellt über 30.000 Quelldateien mit nahezu 12.000.000 Codezeilen.
August 2002	Die migrierte Anwendung wird versuchsweise mit 200 Benutzern in Betrieb genommen. Es treten keine Probleme auf. Die Leistung ist so gut, dass die Suva die Entwicklungsversion der Anwendung mit dem gesamten Testcode in die Produktion übernimmt, so dass alle Probleme leicht diagnostiziert werden können.
Oktober 2002	Die migrierte Anwendung geht komplett mit 2.000 Benutzern in den Produktionsbetrieb über. Die Auslastung beträgt im Durchschnitt 2.000.000 Zugriffe pro Stunde auf die zentralen DB2-Datenbanken. Das neue System verarbeitet dieses Volumen problemlos.

„Wir hatten einen hervorragenden Support von SAXOS und Delta Software Technology in allen Phasen dieses umfangreichen und komplexen Projekts.“

*Robert Koch
CIO, Suva*

„Wir sind begeistert davon, dass Kunden wie Suva mit unserer Technologie so außerordentliche Einsparungen an Zeit und Aufwand erreichen.“

*Rüdiger Schilling
CEO,
Delta Software Technology*

DIE HERAUSFORDERUNG

Die Schweizerische Unfallversicherungsanstalt (Suva) ist eine unabhängige Organisation des öffentlichen Rechts ohne Erwerbscharakter mit Sitz in Luzern und 20 Niederlassungen in der Schweiz. Hauptziele der Organisation sind die Verhütung von Arbeitsunfällen und Berufskrankheiten, die Bereitstellung einer Unfallversicherung und die Rehabilitation.

Die Suva ist finanziell unabhängig und erhält keine staatlichen Subventionen. Zurzeit versichert sie über die Hälfte der erwerbstätigen Bevölkerung in der Schweiz – mehr als 1,9 Millionen Männer und Frauen.

1998 beschloss die Suva, eine Reihe strategischer Projekte zur Modernisierung ihrer Kern-IT-Anwendungen und ihrer technischen Infrastruktur durchzuführen, um den Anforderungen einer dynamischen und zunehmend vernetzten Organisation besser gerecht zu werden. Die Integration mit einer wachsenden Zahl externer Partner war erforderlich. Die vorhandene technische Architektur verfügte nicht über die Offenheit und Flexibilität, die hierfür eindeutig notwendig war.

Die alte Architektur

Die vorhandene Legacy-Architektur der Suva bestand aus über 2.000 Fat-Client IBM OS/2-Workstations, die in der Hauptverwaltung und in den 20 Niederlassungen eingesetzt waren. Über eine komplexe LAN- und WAN-Infrastruktur waren die OS/2-Workstations mit einer zentralen IBM DB2-Datenbank verbunden, die auf IBM S/390-Großrechnern in der Hauptverwaltung der Suva in Luzern lief.

Die Legacy-Anwendungen waren in COBOL implementiert und als OS/2 Fat-Clients eingesetzt. Die grafische Benutzeroberfläche verwendete „Dialog System“ von Micro Focus. Datenzugriffe waren mit nativem SQL in einer in COBOL implementierten Datenebene, die lokal unter OS/2 ausgeführt wurde, realisiert.

Die folgende Tabelle gibt einen Überblick über die Größe der Legacy-Anwendung in Bezug auf die Anzahl der Quelldateien und Codezeilen.

OS/2 Fat Client	Quelldateien	Codezeilen
GUI-Fenster	410	
Geschäftslogikschicht	789	2.250.000
Datenzugriffsschicht	2.700	1.500.000
Architekturmodule	100	475.000
COBOL Copybooks	5.600	775.000

Die alte Architektur wies für die Suva einige wesentliche Nachteile hinsichtlich der Erfüllung ihrer strategischen Geschäftsziele auf. IBM entwickelte OS/2 nicht mehr weiter, und die Betriebskosten (Total Cost of Ownership, TCO) wurden zunehmend inakzeptabel.

Nicht standardisierte Tools führten zu einer schlechten Produktivität bei den Entwicklern. Die fehlende Intranet- und Internet-Unterstützung machte die Integration von externen Benutzern und Anwendungen schwierig. Für Suva stand fest, dass grundlegende Veränderungen notwendig waren.

Die neue Architektur

Nach der sorgfältigen Prüfung einiger Alternativen entschied die Suva sich für eine Thin-Client-Architektur. Diese wäre einfach zu verwalten und würde „jedem den Zugang von überall“ ermöglichen. Diese Tatsache war ein wichtiger Geschäftsfaktor für die Suva, da es in zunehmendem Maße wichtig wurde, mit zahlreichen externen Partnern zu kommunizieren.

Die neue Architektur sollte die Entwicklung einer Benutzerschnittstelle in Java vorsehen. Die Darstellungsschicht sollte Thin Clients verwenden, die mit Windows 2000-Servern mit Citrix MetaFrame verbunden sein sollten. Die mittlere Schicht mit der migrierten COBOL-Geschäftslogik sollte Oracle WebLogic Enterprise-Anwendungsserver verwenden, die auf AIX-Servern von IBM ausgeführt werden. Oracle Tuxedo sollte für die Kommunikation zwischen Clients und Anwendungsservern eingesetzt werden. Die bestehende COBOL-Datenschicht sollte auf AIX migriert werden und über DB2 Connect von IBM mit den bestehenden DB2-Datenbanken auf den S/390-Großrechnern verbunden sein.

**Legacy-
Migration
für einen
maximalen
ROI**

Die Herausforderung bestand für die Suva in einer Lösung für die Frage, wie man von der alten OS/2 Fat-Client-Architektur zur neuen strategischen Architektur migrieren könne. Der Geschäftsfall war klar: Es musste einen eindeutigen Return on Investment geben – die nicht unerheblichen Aufwendungen an Zeit und Geld, die in die Migration investiert werden müssten, sollten sich lohnen. Gleichzeitig mussten die Kosten für die Schulung der großen Anzahl an Benutzern auf einem absoluten Minimum gehalten werden.

Eine direkte Migration von OS/2 auf Windows über das Micro Focus Dialog System war ausgeschlossen, da dies keine echten Vorteile gebracht hätte.

Die Herausforderung bestand daher darin, die neue Benutzeroberfläche in Java zu erstellen und gleichzeitig den ROI für die erhebliche Investition zu maximieren, die bereits in den bestehenden Geschäfts- und Datenschichten steckte. Diese Schichten waren in COBOL implementiert, passten aber genau zu den Geschäftsanforderungen der Suva. Idealerweise sollte das COBOL in seinem Zustand erhalten bleiben und die neue in Java entwickelte Oberfläche als neue Darstellungsschicht hinzugefügt werden.

Dem Suva-Projektteam war klar, dass es nicht möglich sein würde, die vorhandene Micro Focus Dialog System-GUI einfach durch eine Java-Entsprechung zu ersetzen. Auch wenn die COBOL-Anwendungen sorgfältig entwickelt waren und einer strengen 3-Schichten-Architektur folgten, waren sie doch konzipiert worden, bevor objektorientierte und komponentenbasierte Methoden gängig waren.

Es musste ein Weg gefunden werden, um die wiederverwendbaren Teile des COBOL-Codes intelligent zu kapseln und so direkt über die objektorientierte Java-GUI nutzbar zu machen.

„Die Suva-Anwendung war eine hervorragende Basis für ein erfolgreiches Integrationsprojekt mit SCORE Adaptive Bridges. Standards waren nicht nur in Handbüchern definiert, sondern im tatsächlichen Code implementiert“

*Norbert Nigg
CEO, SAXOS*

Eine offene Integrationslösung war für die Suva nicht nur mit Hinblick auf die anstehende Migration der Legacy-Anwendungen wichtig; sie sollte auch Raum für zukünftige Entwicklungen lassen. Zu Beginn des Projekts hatte die Suva noch nicht entschieden, welche Technologie für neue Entwicklungen verwendet werden sollte. Deshalb war eine offene Integrationslösung für die Suva von zentraler Bedeutung.

Eine von vielen Komplikationen mit den alten Fat-Clients bestand darin, dass die Schnittstelle zwischen der grafischen Benutzeroberfläche und der Geschäftsschicht immer ein lokaler Aufruf mit Zugriff auf einen gemeinsamen Speicher war. Die neue Architektur sah vor, dass die grafische Java-Benutzerschnittstelle und die COBOL-Geschäftsschicht auf unterschiedlichen Plattformen ausgeführt werden sollten. Eine weitere zu beachtende Schwierigkeit war die Einführung verteilter Systeme und einer Middleware-Komponente in die neue Anwendungsarchitektur.

DIE LÖSUNG – SCORE ADAPTIVE BRIDGES

Nach der Auswertung einer Reihe von Anbietern – unter ihnen Forté, Dynasty und Micro Focus – wählte die Suva SCORE Adaptive Bridges (früher: SCORE Integration Suite) des deutschen Spezialisten für Software-Tools Delta Software Technology.

SCORE Adaptive Bridges ist eine kostengünstige und in hohem Maße produktive Lösung zur Nutzung des Werts vorhandener Legacy-Anwendungen und -Datenbestände durch Service-basierte Integration. SCORE Adaptive Bridges kombiniert integrierte Entwicklungswerkzeuge, Prozesse und zugehörige professionelle Dienstleistungen zur Erstellung adaptiver Services mit Interface Intelligence, Service Agility und Service Quality.

Bei diesem Projekt bot SCORE Adaptive Bridges der Suva einen schnellen und einfachen Weg zur direkten Wiederverwendung der bestehenden COBOL-Geschäftslogik und Datenzugriffsschichten in der völlig anders konzipierten neuen Architektur.

Intelligentes Service Enablement

SCORE Adaptive Bridges bot dem Suva-Projektteam einen integrierten Workflow mit drei Schlüsselprozessen – Discovery, Composition und Production, jeder unterstützt durch intelligente Tools, die nahtlos über ein gemeinsames XML-basiertes Repository interoperieren.

Discovery

Komponenten der Geschäftsschicht wurden über Component Definition Files definiert und in das XML-basierte Composition Repository geladen. Auf Grund der gut durchdachten Architektur der Legacy-Anwendungen konnte das Suva-Projektteam einen großen Teil der Discovery-Arbeit automatisieren. Die automatische Analyse der COBOL-Copybooks sparte viel Zeit und Mühe und stellte gleichzeitig eine 100%-ige Konsistenz mit der COBOL-Geschäftsschicht sicher.

Composition

Das Suva-Projektteam verwendete das Composition Tool zur Entwicklung fein granularer Schnittstellen und Methoden für die Kommunikation mit den GUI-Clients sowie für spezielle Abbildungen von Parametern, für die Datenumwandlung usw. Dieses Tool wurde in späteren Phasen des Projekts auch zur Angabe von Optimierungsanforderungen verwendet.

Production

SCORE Adaptive Bridges generierte den gesamten erforderlichen Code für die Schnittstelle zwischen den neuen Java-GUI-Clients und der vorhandenen COBOL-Geschäftsschicht – Client-seitige Proxies und spezifische Java-Adapter, Server-seitige Komponentenrahmen für COBOL sowie den gesamten Code, sowohl Server- als auch Client-seitig, für die Arbeit mit der Tuxedo-Middleware. Sollte die Suva zusätzliche Plattformen wie J2EE/EJB, .NET und Webdienste hinzufügen wollen, muss lediglich der erforderliche Plattformadapter ausgewählt werden, um den gesamten erforderlichen Code zu generieren. Zusätzliche Spezifikationen und Entwicklungsarbeiten sind nicht erforderlich.

ERSTELLEN VON KOMPONENTEN UND SERVICES

Aus früheren Erfahrungen wusste man bei der Suva, dass es nicht ausreichte, die vorhandenen COBOL-Schnittstellen nur 1:1 zu übernehmen – dies würde lediglich zu technischen Wrappern und somit in eine Sackgasse führen. Die Suva wollte echte Komponenten mit Service-orientierten Schnittstellen erstellen, die leicht zu integrieren sein sollten, und zwar sowohl mit der neuen Java-GUI als auch mit bislang noch unbekanntem zukünftigen Entwicklungen. SCORE Adaptive Bridges bot der Suva exakt die hierzu benötigte Funktionalität.

Der OS/2 Fat-Client führte GUI- und Geschäftslogikschicht auf der gleichen Plattform aus, mit Zugriff auf den gemeinsam genutzten Speicher. Daher hatte es wenig Bedarf gegeben, die zwischen den Schichten übertragenen Datenmengen zu reduzieren. In vielen Fällen wurden über 100 KBytes an Daten bei jedem einzelnen Aufruf von der GUI- an die Geschäftsschicht übergeben.

Umfangreiche Schnittstellen spielten in der Fat-Client-Umgebung keine große Rolle für die Leistung des Systems. In einer verteilten Umgebung könnte so jedoch niemals eine annehmbare Performance erreicht werden. Ein Java GUI-Client könnte nicht bei jedem Serveraufruf 100 KBytes über Tuxedo weitergeben.

SCORE Adaptive Bridges hat dieses Problem für die Suva sauber gelöst, indem der bestehenden COBOL-Geschäftsschicht ein Komponentenrahmen hinzugefügt wurde. Dieser Rahmen definiert die Operationen und Schnittstellen, die dem GUI-Client gegenüber kenntlich gemacht werden müssen.

„Deltas Technologie behandelte alle Details der Integration und ermöglichte es uns, unsere kritischen Systeme zu pflegen während wir gleichzeitig neue adaptive Interfaces entwickelten.“

*Thomas Müller
Leiter Anwendungsarchitektur und Methoden,
Suva*

Anstelle eines Servers mit einer einzigen „breiten“ Schnittstelle war nun eine ganze Reihe von Schnittstellen und Operationen möglich, die nur die tatsächlich benötigten Daten definieren. Dies ist besser für die Systemleistung und sehr viel einfacher zu verwalten.

SCORE Adaptive Bridges kennt „beide Seiten“ der Schnittstelle und generiert automatisch den gesamten erforderlichen Code. Es gibt keine Änderung an der Schnittstelle zur bestehenden COBOL-Geschäftsschicht. Dabei handelt es sich um einen nicht-invasiven Ansatz, was bedeutet, dass die Suva ihre Kern-Geschäfts- oder Zugriffsschichten nicht modifizieren (oder neu testen) musste.

Nach dem sorgfältigen, aber pragmatischen Design der neuen Schnittstellen durch ein Mitglied des Suva-Teams stellte jede Serverkomponente in der Regel 5 bis 10 Operationen als Services für die GUI-Clients zur Verfügung. In einigen Ausnahmefällen führte dies dazu, dass Server, die vorher extrem umfangreiche Schnittstellenstrukturen hatten, nun mehr als 30 Operationen anboten. Fein granulare Schnittstellen machten es für das GUI-Client-Entwicklungsteam der Suva leichter, die von solchen „Server-Giganten“ bereitgestellten Geschäftsfunktionen zu verstehen und zu nutzen.

SUVA PROFITIERT VON GENERATIVEM PROGRAMMIEREN

Adapter für Suvas GUI- Framework

Die Suva hatte für die migrierte Anwendung keine Schulungskosten vorgesehen – die neue Java-GUI musste folglich optisch und semantisch zu 100 % mit der alten Dialog System-Lösung auf OS/2 kompatibel sein.

Frühere Erfahrungen der Suva mit Anwendungsarchitekturen bewegten das Projektteam dazu, bei der Entwicklung und Implementierung der Java-GUI für ihre neue Architektur einige Schritte zurück zu gehen, um den bestmöglichen Ansatz zu finden. Das Ergebnis war ein unternehmensspezifisches Java-GUI-Framework, das einen Großteil der allgemeinen Funktionen für die große Anzahl der Java-Fenster und -Dialoge verarbeitete. Ein Hauptziel dieses Framework war es, die optische und semantische Konsistenz mit dem Legacy-System zu ermöglichen.

Das Erstellen des Suva Java-GUI-Framework warf die Frage nach der besten Möglichkeit zur Integration dieses unternehmensspezifischen Frameworks mit den von SCORE Adaptive Bridges generierten Java-Proxies auf. Proxies sind Client-seitige Darstellungen der Funktionen, die von der COBOL-Geschäftsschicht bereitgestellt werden. Wie dies bei den meisten verteilten Systemen der Fall ist, isolieren die Proxies den Client von allen technischen Details für die Kommunikation mit Back-End-Servern.

SCORE Adaptive Bridges ist nach den Prinzipien des generativen Programmierens (Generative Programming) aufgebaut und implementiert. Daher war es für das Suva-Projektteam kein Problem, die Integration dieses unternehmensspezifischen GUI-Frameworks zu unterstützen.

Code- Generierungs- muster für Testtreiber

Mit Hilfe einer frühen Version des PBE Pattern By Example-Tools von Delta Software Technology war das Projektteam in der Lage, Generierungsmuster zu codieren, um so die Standard-Proxy-Schnittstellen der SCORE Adaptive Bridges auf die vom Suva-Client-Framework benötigten Schnittstellen abzubilden. Die Muster enthielten sämtliche erforderlichen Informationen über die Proxies aus dem Standard-Composition Repository.

Die unternehmensspezifischen Code-Generierungsmuster wurden mit ANGIE, Deltas Frame-basierten Generator, in den Standard-Proxy-Generierungsprozess integriert. Jedes Mal, wenn ein Proxy generiert wurde, wurde auch ein entsprechender Adapter für das Suva-Framework erstellt. Diese Vorgehensweise lieferte dem Suva-Projektteam zu 100 % korrekten Java-Code, der garantiert immer mit den Standard-Proxies der SCORE Adaptive Bridges und den Back-End-Servern synchronisiert war.

Bei einem Projekt dieser Größenordnung wird eine automatisierte Testumgebung schnell zu einer wichtigen Anforderung. Auch hier war das Suva-Projektteam in der Lage, die mit SCORE Adaptive Bridges verfügbaren Funktionen für Generative Programming zu nutzen, um diese Anforderung zu erfüllen.

Diesmal arbeiteten Pattern By Example und ANGIE bei der Generierung von speziellen Java-Testtreibern mit den Standard-Code-Generierungsmustern von SCORE Integration Suite zusammen.

Diese nicht zur GUI gehörigen Java-Clients konnten problemlos über die Befehlszeile getestet werden und lieferten dem Team einen schnellen, einfachen und skriptfähigen Weg zur Überprüfung der Funktionsweise.

Aspekt-orientiertes Programmieren zu Testzwecken

Die automatisch generierten Testtreiber versetzten das Suva-Projektteam in die Lage, die Server auf ihre grundlegenden Funktionen hin zu testen, bevor die Java GUI-Clients entwickelt wurden. So war es möglich, dass die Client- und Server-Teams gleichzeitig an dem Projekt arbeiten konnten.

Entsprechend der Entwicklung der Server-Spezifikationen und ihrer nach und nach durchgeführten Änderungen wurden die Testtreiber zusammen mit den Standard-Client-Proxies von SCORE Adaptive Bridges und den Suva-Framework-Adaptern automatisch neu generiert. Wieder war der Java-Code immer zu 100 % korrekt und mit dem Rest der Anwendung vollständig synchronisiert.

Die neue verteilte Anwendungsarchitektur der Suva war komplizierter als die alte Fat-Client-Lösung unter OS/2. Die Erfahrungen, die während der Entwicklung und dem Testen der Fat-Clients gemacht wurden, bildeten nicht unbedingt eine geeignete Basis für die neue Anwendungsarchitektur.

Einfaches Testen und Debuggen wurden als wesentliche Faktoren für den Gesamterfolg des Projekts hervorgehoben. Deshalb entschied sich das Projektteam, ein ausgeklügeltes Überwachungssystem zu erstellen, welches das Testen und Debuggen unterstützen sollte. Die Überwachungsfunktion sollte allen Aspekten des Systems hinzugefügt und dynamisch aktiviert werden.

Zur Implementierung der entwickelten Überwachungsfunktionalität musste der Standard-Überwachungscode an zahlreichen Stellen innerhalb der Anwendung hinzugefügt werden, z. B. in den verschiedenen Komponenten- und Middleware-Frames, Proxies, Adaptern usw. Mit herkömmlichen Werkzeugen kann es sehr schwierig sein, einen solchen Infrastruktur-Code zu entwickeln, zu testen und zu pflegen. Auch wenn es sich bei der Überwachungsfunktion um ein Einzelkonzept handelt, muss der Code zu seiner Implementierung überall innerhalb der Anwendung erstellt werden.

SCORE Adaptive Bridges unterstützte das Projektteam mit verschiedenen Elementen der Aspekt-orientierten Programmierung, einschließlich Aspekt- und Code-Weaving. Der mit dem Aspekt Überwachung verbundene Code wurde einfach unter Verwendung von Code-Mustern in einem zentralen Speicherort definiert. Die Frame-basierte ANGIE-Generortechnologie wurde verwendet, um diejenigen Punkte im generierten Anwendungscode zu definieren, an denen der Überwachungscode generiert werden sollte.

Ein wichtiger Vorteil der Architektur von SCORE Adaptive Bridges ist die Tatsache, dass alle Informationen über die zu generierende Komponente im Composition Repository gespeichert sind. Somit haben die Code-Generatoren einen breiten Überblick über die zu generierenden Elemente. Der funktionsorientierte Code wird problemlos zum korrekten Speicherort in der korrekten Komponenten-Quelldatei und im korrekten generierten Quellobjekt zugewiesen. Der Generator kann daher durchweg fehlerfrei arbeiten.

ENTWICKLUNGSTOOL-INTEGRATION MIT SCOUT²

Für das Suva-Projekt wurden 8 verschiedene Entwicklungswerkzeuge auf drei verschiedenen Plattformen verwendet – unter ihnen SCORE Adaptive Bridges, Visual Age for Java, WebLogic Enterprise, COBOL und MKS Source Integrity.

Die SCOUT² Development Platform von Delta Software Technology war ohne Probleme in der Lage, die Entwicklungstools und Workflow-Prozesse des Suva-Projektteams zu integrieren, wodurch ein nahtloser, plattformunabhängiger Überblick über die Entwicklungs-, Test- und Produktionsumgebungen der Suva entstand.

SCOUT² Development Platform verwaltete die gesamte Kompilierung, Linkage usw. des plattformspezifischen Quellcodes für die Services zusammen mit der Änderungskontrolle und anderen üblichen Infrastrukturaufgaben.

SCORE Adaptive Bridges ist vollständig mit SCOUT² integriert und ermöglichte es somit den Mitgliedern des Suva-Projektteams, ihre neuen Geschäftsservices in einer vertrauten, flexiblen und erweiterungsfähigen Umgebung festzulegen, zusammenzustellen und zu produzieren.

Projekt- Metrik

Die folgende Tabelle gibt einen Überblick über die Größe des Projekts in Bezug auf die Anzahl der Komponenten, die Suva mithilfe von SCORE Adaptive Bridges generieren konnte. Die Anzahl der generierten Codezeilen ist ebenfalls aufgeführt. 100% dieses Codes wurde aus dem Composition Repository generiert.

System-komponente	Quell-dateien	Sprache	Codezeilen <i>Hinweis: beinhaltet generierte Test- Frameworks</i>
Business-Tier Server Komponenten	407	SCORE Adaptive Bridges	
Generierte Server-seitige Services	395	COBOL	3.500.000
Generierte Suva GUI Framework Adapter	393	Java	168.000
Generierte Client-seitige Proxies	391	Java	500.000

WICHTIGE GESCHÄFTSVORTEILE

Die Migration der alten COBOL-Geschäfts- und Datenzugriffsschichten auf die neue Architektur mit SCORE Adaptive Bridges brachte für die Suva eine ganze Reihe greifbarer Vorteile:

Investitionsschutz

Die erheblichen Investitionen, welche die Suva in die alten COBOL-Geschäfts- und Datenzugriffsschichten getätigt hatte, wurden vollständig geschützt. SCORE Adaptive Bridges ermöglichte es der Suva, rund 5.000.000 Zeilen vorhandenen COBOL-Codes direkt wiederzuverwenden.

Entwicklerproduktivität

Mit SCORE Adaptive Bridges müssen sich Entwickler nicht mehr um Middleware- und allgemeine Infrastruktur-Features wie Anpassung der Schnittstellenversionen, Datenkonvertierung, Zuweisung von Parametern usw. kümmern.

Durch die automatische Generierung all dieser Codes ermöglicht es SCORE Adaptive Bridges den Suva-Entwicklern, sich auf die Kern-Geschäftsfunktionen der Anwendung zu konzentrieren. Dies ist eine „Win-Win“-Situation, da einerseits die Entwickler mehr Zeit haben, sich den Geschäftsfunktionen zu widmen, und andererseits die Suva-Anwendungen ohne Mehraufwand eine optimale Infrastruktur erhalten.

Nicht invasive Integration

Die Funktion zur Unterstützung der Service-basierten Integration wurde nicht invasiv hinzugefügt. So wurde sichergestellt, dass die Suva mögliche zusätzliche Support- und Wartungskosten für die Kern-Geschäftsfunktionen auf ein Minimum reduzieren konnte. Der COBOL-Kerncode blieb vollständig unverändert.

„Unsere Erfahrungen mit SCORE Adaptive Bridges in der Prototypen- und Pilotphase wurden bestätigt, als wir auf das tatsächliche Projekt erweiterten – 60 Entwickler, 30.000 Quelldateien und 12.000.000 Codezeilen. SCORE Adaptive Bridges hält, was es verspricht.“

*Robert Koch
CIO, Suva*

Flexible Client- Integration

Das Suva-Projektteam definierte die Schnittstellen zu den Geschäftsservices in neutraler Art und Weise im Composition Repository. Von dort aus wird der gesamte Client-Schnittstellencode für die ausgewählte Einsatzplattform generiert.

Derzeit verwendet die Suva-Anwendung eine Standard-Java-GUI. Erste Tests mit J2EE/EJB-Clients wurden bereits begonnen. SCORE Adaptive Bridges ermöglicht es der Suva, solche neuen Clients aus der bestehenden Server-Spezifikation heraus zu unterstützen. Die von SCORE Adaptive Bridges generierten Server-Frames können mehrere Client-Technologien nebeneinander unterstützen.

Suva-Entwickler müssen sich dabei nicht mit den technischen Details für die Anbindung von Clients an Server beschäftigen.

Plattform- unabhängige Strukturen

Alle Informationen über Komponenten und Einsatzplattformen werden in einem Plattform-unabhängigen Format im XML-basierten Composition Repository bereitgehalten. Fügt die Suva weitere Einsatzplattformen hinzu, kann die vorhandene Komponentenspezifikation zur automatischen Generierung des gesamten Codes für die neue Plattform wiederverwendet werden.

Die vollständige Unabhängigkeit von bestimmten Plattformen gibt der Suva die Flexibilität, neuen Anforderungen gerecht zu werden und eine potenzielle Bindung an einen einzigen Plattformanbieter zu vermeiden.

Offene Standards

SCORE Adaptive Bridges baut auf offenen Standards der Object Management Group (OMG) und anderer Standardisierungsorganisationen auf. Alle Spezifikationen, Repositories und Code-Generierungsmuster sind für die Suva offen – nichts ist in geschützten Dateiformaten oder „geheimen“ Datenbanken versteckt.

Die gesamte Kommunikation zwischen Clients und Servern verwendet standardmäßige Internet- und Intranet-Protokolle. Es gibt kein proprietäres Kommunikationsprotokoll bei SCORE Adaptive Bridges. Der gesamte generierte Code ist zum besseren Verständnis wohl strukturiert und dokumentiert.

„Wir haben Services für verschiedene Middleware- und Plattform-Kombinationen entwickelt – SCORE Adaptive Bridges ist damit problemlos fertig geworden. Zurzeit arbeiten wir an unserer Strategie für zukünftige Projekte mit Delta.“

*Thomas Müller
Leiter Anwen-
dungsarchitektur
und Methoden,
Suva*

Service- basierte Integration

Die alte OS/2 Fat-Client-Architektur war nicht für den Einsatz in einer verteilten Umgebung ausgelegt. SCORE Adaptive Bridges hat es der Suva ermöglicht, die bestehende COBOL-Geschäftsschicht weiter zu verwenden und sie mit den neuesten Technologien auf dem Gebiet der Service-basierten Architekturen einzusetzen.

Statt einfach die alten „breiten“ Schnittstellenstrukturen mit den neuen GUI-Clients zu verbinden, nahm sich die Suva die Zeit, fein granulare Schnittstellen zu entwickeln, die lediglich die minimal erforderliche Anzahl an Datenfeldern enthalten. SCORE Adaptive Bridges generiert den gesamten Code für das Mappen zwischen diesen „internen“ und „externen“ Schnittstellen.

Die klar definierten und dokumentierten Services unterstützen die Suva bei der Reduzierung des Wartungsaufwands. Die Suva ist außerdem bestens für die Integration von Geschäftsfunktionalitäten für neue Kanäle wie Webdienste und Portale gerüstet.

System- Verlässlich- keit und Verfügbar- keit

Die alte OS/2 Fat-Client-Architektur war naturgemäß zustandsbehaftet (stateful), wobei Daten über Transaktionen hinweg im OS/2-Speicher bereitgehalten wurden. Die neue Architektur erforderte ein zustandsloses (stateless) Design, um die bestmögliche Leistung aus den Oracle WebLogic Enterprise-Anwendungsservern zu holen.

Ein zustandsloses Design bietet deutlich mehr Flexibilität beim Erstellen von Lastverteilung und Server Fail-Over-Funktionen für eine stabile und zuverlässige Einsatzumgebung.

SCORE Adaptive Bridges hat diese Herausforderung für die Suva gemeistert, indem automatisch der gesamte Code für die Statusverwaltung generiert wurde. Dieser Code berücksichtigt die architekturbedingten Einschränkungen der WebLogic Enterprise-Umgebung und stellt gleichzeitig sicher, dass die GUI-Clients nicht mit Aufgaben der Statusverwaltung belangt werden müssen. Auf diese Weise kann die Suva die Vorteile beider Ansätze nutzen – eine zustandslose Systemarchitektur für erhöhte Verfügbarkeit bei transparenter Statusverwaltung.

Netzwerk- optimierung

Die automatische Optimierungsfunktion von SCORE Adaptive Bridges reduziert den Netzwerkverkehr und die Auslastung von Anwendungsservern im Vergleich zu gängigen Implementationen verteilter Systeme drastisch.

Mit Hilfe des Composition Managers kann die Suva automatisch Operationsaufrufe „stapeln“ und so die Interaktionen zwischen Clients und Servern reduzieren. Der gesamte Code für das Client-seitige Caching und die Steuerung der Operationssequenzen wird vollautomatisch generiert. Der Client-Entwickler bemerkt diese Optimierung nicht.

Verschie- dene Client- Versionen

SCORE Adaptive Bridges unterstützt mehrere Versionen jeder Komponentenschnittstelle. Diese Funktion ermöglicht es der Suva, verschiedene Versionen ihrer Client-Anwendung über die gleiche Server-Instanz zu unterstützen.

Dies ist ein wertvoller Vorteil, der die Einführung neuer Anwendungsfunktionen deutlich erleichtert, da nicht sämtliche Clients parallel upgedatet werden müssen. Außerdem reduziert dies den Verwaltungsaufwand der WebLogic Enterprise-Einsatzumgebung.

Generativer Ansatz

In umfangreichen Projekten läuft nicht immer alles nach Plan. Deshalb ist es wichtig, dass die verwendeten Werkzeuge und Methoden flexibel genug sind, um auf Änderungen der Anforderungen reagieren zu können.

Das Suva-Projekt zeigte, wie schnell und einfach der Kern von SCORE Adaptive Bridges angepasst werden konnte, um die speziellen Anforderungen der Suva an die Code-Generierung zu erfüllen. Ein Beispiel hierfür ist die Erstellung einer zusätzlichen Framework-Adapterschicht für die GUI-Clients.

Durch den Einsatz der innovativen generativen Entwicklungswerkzeuge von Delta Software Technology wie PBE Pattern By Example und ANGIE Frame Generator war das Suva-Projektteam in der Lage, viele dieser Änderungen selbstständig durchzuführen, ohne auf das nächste Produkt-Update warten zu müssen.

Diese Flexibilität bei der Reaktion auf projektspezifische Anforderungen ist ein Kennzeichen des generativen Ansatzes, wie er von Delta Software Technology umgesetzt wird.

„Mit SCORE Adaptive Bridges ernten wir die Früchte unserer langfristigen Strategie der Nutzung generativer Programmierung und neuester Technologien.“

*Rüdiger Schilling
CEO,
Delta Software Technology*

PARTNER

Die Suva hat mit der SAXOS Informatik AG (www.saxos.ch) zusammengearbeitet, dem Vertriebspartner von Delta Software Technology in der Schweiz und Liechtenstein.

Das Unternehmen mit Sitz in Zürich hat mehr als 20 Jahre Erfahrung mit Delta-Produkten und ist auf die Anwendung fortschrittlicher Software-Tools für die Entwicklung, Integration und Migration von Software spezialisiert.

Erfahrene Mitarbeiter von SAXOS arbeiteten während des gesamten Projekts eng mit der Suva zusammen. Eine der wichtigsten Aufgaben war dabei die Kooperation mit der Suva-Architekturgruppe bei der Erstellung des Migrationskonzeptes, der Definition der Architektur sowie der Implementierung der erforderlichen Werkzeuge und Automatisierungsfunktionen für SCORE Adaptive Bridges.

SAXOS arbeitete auch intensiv mit den Produktarchitekten und Entwicklern von Delta Software Technology bei der Erstellung projektspezifischer Verbesserungen für die Suva mit PBE Pattern By Example und ANGIE Frame Generator zusammen.

„Wir sind von den Ergebnissen des Suva-Projekts begeistert und freuen uns darauf, mit der Suva an weiteren Projekten zu arbeiten.“

*Norbert Nigg
CEO, SAXOS*

DELTA SOFTWARE TECHNOLOGY

Delta Software Technology ist Spezialist für generative Software-Werkzeuge, die die Modernisierung, Integration, Entwicklung und Wartung individueller IT-Anwendungen automatisieren.

Wir begreifen die Unternehmens-IT als einen lebendigen Organismus, der sich ständig verändert. Unsere automatisierten Lösungen helfen Ihnen, Ihre Anwendungen schnell und sicher an neue Geschäftsanforderungen, Architekturen, Technologien und technische Infrastrukturen anzupassen.

Delta liefert seit mehr als 30 Jahren erfolgreich fortschrittliche Software-Technologie an Europas führende Organisationen, zu denen u.a. AMB Generali, ArcelorMittal, Deutsche Telekom, Hüttenwerke Krupp Mannesmann, Gothaer Versicherungen, La Poste, RDW, Suva und UBS gehören.

AMELIO® Modernization Platform™

Maßgeschneiderte Factory für die Modernisierung großer IT-Anwendungen: 100% automatisch und deshalb sicher, zuverlässig und fehlerfrei.

SCORE® Adaptive Bridges™

Intelligentes Service Enablement für die Wiederverwendung bewährter Anwendungen mit modernsten Technologien: Flexibel, rentabel und non-invasiv.

SCORE® Data Architecture Integration™

Daten als echte Business Services: Schnell, einfach und unabhängig von Datenarchitekturen und Speicherungsformen.

SCOUT²™ Development Platform

Optimierte und integrierte Entwicklungsprozesse über alle Software-Komponenten, Werkzeuge und Plattformen: Stoppt den „Kampf gegen die Infrastruktur“.

ADSplus™ Application Development

Plattformunabhängige Entwicklung für zukunftssichere Back-End-Anwendungen.

www.d-s-t-g.com

Copyright © 2002–2010 Delta Software Technology GmbH. Alle Rechte vorbehalten.

Delta, SCORE, ObjectBridge, AMELIO und das Delta Software Technology Logo sind registrierte Warenzeichen, und SCORE Adaptive Bridges, SCORE Integration Suite, Model Driven Legacy Integration, Integration in Motion, AMELIO Modernization Platform, SCOUT², ADSplus, ANGIE und PBE Pattern By Example sind Warenzeichen der Delta Software Technology GmbH in Deutschland und/oder anderen Ländern. Alle anderen eingetragenen Warenzeichen, Warenzeichen, Handelsnamen oder Dienstleistungsmarken sind Eigentum ihrer jeweiligen Besitzer.

Bestellnummer: MT 11'010.07 – Oktober 2010