INFO

DELTA
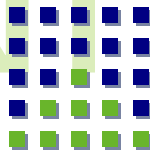### software technology

case study
SUVA

SUVA uses generative programming tools from Delta Software Technology to maximize return on investment when integrating legacy COBOL applications in a distributed thin-client architecture.

Suva is the Swiss national accident insurance organization covering 100,000 companies and 1.9 million people – more than 50% of the working population.

Their challenge was to migrate critical applications from OS/2 fat clients to a distributed thin-client architecture.

Read in this detailed case study how Suva used generative development tools from Delta Software Technology to create components by reusing 5,000,000 lines of COBOL.

SCORE® Adaptive Bridges™ for Intelligent Service Enablement

"SCORE Adaptive Bridges handled our complex legacy architecture. We chose the right solution."

Robert Koch
CIO, Suva

SOLUTION OVERVIEW

customer

Suva is the Swiss national accident insurance organisation covering 100,000 companies and 1.9 million people – over 50% of the working population.

challenge

Maximize return on investment on legacy COBOL application migrating from fat-client to distributed thin-client architecture.

Delta Software Technology solutions

- SCORE® Adaptive Bridges™ (formerly SCORE® Integration Suite™)

- SCOUT²™ Development Platform

- PBE Pattern By Example™

- ANGIE Frame Generator™

why Delta Software Technology?

Enabled Suva to reuse more than 5,000,000 lines of existing COBOL code, creating components with service interfaces suitable for Java GUI clients.

Key business benefits

- Maximized ROI on existing COBOL legacy applications.

- Integrate with Java GUI today and J2EE/EJB etc tomorrow, with no extra effort.

- Generation of native code supports high transaction workloads.

- Cross-platform support avoids lock-in.

- Generative programming techniques for flexible solutions to project requirements.

- Consistent 400% to 600% improvement in developer productivity.

business partner

SAXOS Informatik AG
www.saxos.ch

"We set an aggressive schedule for this project. We were impressed that using SCORE Adaptive Bridges we managed to consistently achieve productivity improvements of 400% to 600%."

Thomas Müller Manager Application Architecture and Methods, Suva

PROJECT TIMELINE

1998

Suva starts migration project planning.

Early 1999

Suva identifies Delta Software Technology as potential supplier for the project by searching the Web with the keywords "COBOL migration".

June 1999

Suva establishes contact with SAXOS.

Late 1999

First prototype to validate SCORE concept. Suva's final deployment platform not decided. Prototype generated for various platforms and infrastructure combinations.

Early 2000

SCORE Adaptive Bridges selected for project. Pilot project to validate approach and create supporting infrastructure, tools and methodology. Suva decided on architecture with thin clients, Citrix MetaFrame, Windows 2000 servers, Java GUI clients, Oracle Tuxedo middleware (formerly BEA) and WebLogic Enterprise application server, IBM AIX servers. Database remains IBM DB2 on IBM S/390 mainframes accessed using DB2 Connect.

Late 2000 to July 2002

Main migration project with up to 60 developers creates more than 30,000 source files with approaching 12,000,000 lines of code.

August 2002

Migrated application in trial production with 200 users. No problems are encountered. Performance is so good that Suva puts development version of the application into production with all test code included so any problems are easily diagnosed.

October 2002

Migrated application in full production with 2,000 users. Workload averages 2,000,000 hits per hour to the central DB2 databases. This level of production workload is easily handled by the new system.

"We received excellent support from SAXOS and Delta Software Technology throughout all phases of this large and complex project."

*Robert Koch
CIO, Suva*

"We are excited that customers such as Suva are achieving exceptional savings in time and effort using our technology"

*Rüdiger Schilling
CEO,
Delta Software Technology*

THE CHALLENGE

The Swiss National Accident Insurance Fund (Suva) is an independent, non-profit company under public law and is based in Lucerne with 20 agencies throughout Switzerland. Its main concerns are the prevention of accidents and occupational diseases, accident insurance and rehabilitation.

Suva is financially independent and does not receive any state subsidies. It currently provides insurance cover for more than half of the Swiss workforce – more than 1,9 million men and women.

Suva decided in 1998 to undertake a series of strategic projects to modernize their core IT applications and technical infrastructure to better support the requirements of a dynamic and increasingly networked organization. Integration was required with an increasing number of external partners. The legacy technical architecture did not have the openness and flexibility that was clearly going to be required.

Suva's legacy architecture was more than 2,000 fat-client IBM OS/2 workstations deployed in the head offices and across 20 agencies. A complex LAN and WAN infrastructure connected the OS/2 workstations to a central IBM DB2 database running on IBM S/390 mainframes at Suva's head office in Lucerne.

The legacy applications were implemented in COBOL, deployed as OS/2 fat-clients. The GUI used "Dialog System" from Micro Focus. Database access used native SQL from a data tier implemented in COBOL running locally on OS/2.

The following table gives an overview of the size of the legacy application in terms of the number of source files and lines of code.

OS/2 Fat Client	Source Files	Lines of Code
GUI Windows	410	
Business Logic Tier	789	2,250.000
Data Access Tier	2,700	1,500,000
Architecture Modules	100	475,000
COBOL Copybooks	5,600	775,000

The legacy architecture presented some major obstacles to meeting Suva's strategic business requirements. IBM was no longer developing OS/2 and the total cost of ownership (TCO) was rapidly becoming unacceptable.

Non-standard tools caused disappointing levels of developer productivity. The lack of Intranet and Internet support made it difficult to integrate with external users and applications. It was clear to Suva that dramatic changes were required.

the new architecture

After carefully investigating a number of alternatives, Suva decided on a thin-client architecture. This would be simple to manage and provide "anyone with access from anywhere". This was an important business driver for Suva as it became increasingly necessary to communicate with a wide range of external partners.

The new architecture would introduce Java for GUI development. The presentation tier would use thin clients connected to Windows 2000 servers with Citrix MetaFrame. The middle tier with the migrated COBOL business logic would use Oracle WebLogic Enterprise application servers running on AIX servers from IBM. Oracle Tuxedo would be used for communication between clients and application servers. The existing COBOL data tier would be migrated to AIX and connect to the existing DB2 databases on the S/390 mainframes using DB2 Connect from IBM.

The challenge for Suva was how to migrate from the legacy OS/2 fat client architecture to the new strategic architecture. The business case was clear: there had to be a clear return on investment for the substantial time and money that would need to be invested in this migration. At the same time, retraining costs for the substantial user base had to be kept to the absolute minimum.

A direct migration from the OS/2 to Windows using the Micro Focus Dialog System was ruled out as it offered no real benefits.

The challenge was therefore how to create the new GUI in Java while maximizing the ROI for the considerable investment that had already been made in the legacy business and data tiers. These tiers were implemented in COBOL, but they were an exact fit to Suva's business requirements. Ideally the COBOL could be retained "as is" and the new Java GUI added as a new presentation tier.

It was clear to the Suva project team that it would not be possible to simply replace the existing Micro Focus Dialog System GUI with a Java equivalent. While the COBOL applications had been carefully designed and followed a strict three tier architecture, they were designed before object-oriented and component-based methodologies had became mainstream.

A way was needed to intelligently wrap the reusable parts of the COBOL code and so make it directly useable from the object-oriented Java GUI.

"The Suva application was an excellent basis for a successful integration project with SCORE Adaptive Bridges. Standards were not just defined in manuals – they were implemented in the actual code."

*Norbert Nigg
CEO, SAXOS*

An open integration solution was important to Suva not just for the current migration of the legacy applications; it was also necessary to be open for future developments. At the start of the project Suva had not decided which technology to use for new development. An open integration solution was therefore vital to Suva.

One of many complications with the legacy fat clients was the interface between the GUI and business tier was always a local call with access to shared memory. The new architecture required the Java GUI and the COBOL business tier to run on different platforms. The introduction of distributed systems and a middleware component to the new application architecture was a further complication to take into account.

The Solution –
SCORE Adaptive Bridges

After evaluating a number of vendors – including Forté, Dynasty and Micro Focus – Suva selected SCORE Adaptive Bridges (formerly: SCORE Integration Suite) from German software tools specialist Delta Software Technology.

SCORE Adaptive Bridges is a cost-effective and highly productive solution for exploiting the value of existing legacy applications and data stores by service-based integration. SCORE Adaptive Bridges combines integrated software tools, processes and related professional services to create adaptive services with interface intelligence, service agility and service quality.

In this project SCORE Adaptive Bridges provided Suva with a quick and easy way of taking their existing COBOL business logic and data access tiers and directly reusing them within their very different new architecture.

intelligent service enablement

SCORE Adaptive Bridges gave the Suva project team an integrated workflow with three key processes – Discovery, Composition and Production – each supported by intelligent tools seamlessly interoperating around a shared XML-based repository.

Discovery

Business tier components were defined using Component Definition Files and loaded into the XML-based Composition Repository. The Suva project team automated much of the discovery work due to the well-designed architecture of the legacy applications.

Automatic analysis of COBOL copybooks saved considerable time and effort, while ensuring 100% consistency with the legacy COBOL business tier.

Composition

The Suva project team used the composition tool to design more fine-grained interfaces and methods for exposure to the GUI clients, as well as any special mapping of parameters, data conversion etc. The tool was also used in later stages of the project to declare optimization requirements.

Production

SCORE Adaptive Bridges generated all code required to interface the new Java GUI clients with the legacy COBOL business tier client-side proxies and custom adapters for Java, server-side component frames for COBOL, plus all code on both client and server sides to handle the Tuxedo middleware. As Suva need to add additional platforms such as J2EE/EJB, .NET and Web services then this is simply a matter of selecting the required platform adapters to generate all the necessary code. No additional specification or design work is required.

CREATING COMPONENTS AND SERVICES

Suva knew from previous experience that it was not enough to just map the existing COBOL interfaces 1:1 – this would just be creating technical wrappers and be a dead-end. Suva wanted to create real components with service-oriented interfaces that could be easily integrated, both with the new Java GUI and yet unknown future developments. SCORE Adaptive Bridges provided Suva with exactly the right functionality to meet this goal.

The OS/2 fat client had GUI and business logic tiers running on the same platform with access to shared memory. There had therefore been little need to manage the amount of data passed between tiers. In many cases more than 100K bytes of data were passed on each call from GUI to business tiers.

Large interfaces were not a major performance issue in the fat client environment; however, this was never going to perform acceptably in a distributed environment. A Java GUI client could not realistically pass 100K bytes through Tuxedo on each server call.

SCORE Adaptive Bridges neatly solved this challenge for Suva by adding a component frame around the existing COBOL business tier. This frame defines the operations and the interfaces that are to be exposed to the GUI client.

"Delta's technology handled all the integration details, allowing us to maintain our critical systems while building new adaptive interfaces."

*Thomas Müller
Manager
Application
Architecture
and Methods,
Suva*

Instead of a server having a single "wide" interface it could now have a series of interfaces and operations that define only the data that is actually needed. Better for performance and much easier to maintain.

SCORE Adaptive Bridges knows "both sides" of the interface and automatically generates all the required code. There is no change to the interface of the existing COBOL business tier. This is a non-invasive approach and means Suva did not need to modify (or retest) the core business or data access tiers.

Following considered but pragmatic interface redesign by a member of the Suva team each server component typically exposed from 5 to 10 operations as services to the GUI clients. In some exceptional cases servers that previously had extremely large interface structures resulted in more than 30 operations being exposed. Fine-grained interfaces made it much easier for Suva's GUI client development team to understand and use the business services provided by such "giant" servers.

Suva Benefits from Generative Programming

Suva planned on zero training costs for the migrated application –
the new Java GUI had to be 100% visibly and semantically
compatible with the legacy Dialog System solution on OS/2.

Suva's previous experience with application architectures meant that
when they looked at designing and implementing the Java GUI for
their new architecture they took a few steps back to consider the best
approach to take. The result was a custom Java GUI framework that
handled a lot of the common functionality across the large number of
Java windows and dialogs. A main goal of this framework was to
enable the visibility and semantic consistency with the legacy
system.

The creation of the Suva Java GUI framework raised the question of
the best way to integrate this custom framework with the standard
Java proxies generated by SCORE Integration Suite. Proxies are
client-side representations of the functionality provided by the
COBOL business logic tier. As is common with most distributed
systems, the proxies isolate the client from all technical details for
communicating with back-end servers.

SCORE Adaptive Bridges is architectured and implemented
following the principles of generative programming. It was therefore
simple for the Suva project team to support integration of this custom
GUI framework.

Using an early version of the PBE Pattern By Example tool from Delta Software Technology, the project team was able to code generation patterns to map the standard SCORE Integration Suite proxy interfaces to those needed by the Suva client framework. The patterns obtained all required information about the proxies from the standard composition repository.

The custom code generation patterns were integrated into the standard proxy generation process using Delta's ANGIE frame-based generation technology. Every time a proxy was generated a corresponding Suva framework adapter was generated. This approach delivered the Suva project team 100% correct Java code that was guaranteed to always be synchronized with the standard SCORE Adaptive Bridges proxies and the back-end servers.

code generation patterns for test drivers

With a project of this size an automated testing environment quickly becomes an important requirement. The Suva project team was again able to leverage the generative programming foundations of SCORE Integration Suite to help meet this requirement.

This time Pattern By Example and ANGIE worked together with the standard SCORE Adaptive Bridges code generation patterns to generate dedicated Java test drivers.
These non-GUI Java clients could be easily tested from the command line and provided the team with a quick, easy and scriptable way to verify the functionality of the back-end servers.

The automatically generated test drivers gave the Suva project team the ability to test the servers for basic functionality before the Java GUI clients had been developed. This allowed the project to proceed with the client and server teams running in parallel.

As the server specifications developed and changed over time the test drivers were automatically regenerated along with the standard SCORE Adaptive Bridges client proxies and Suva framework adapters. Again, the Java code was always 100% correct and fully synchronized with the rest of the application.

aspect-oriented programming for testing

Suva's new distributed application architecture was more complicated than the legacy fat client OS/2 solution. The experience gained over that time in developing and testing fat clients was not necessarily a good basis for the new application architecture.

Ease of testing and debugging were highlighted as essential contributors to the overall success of the project. The project team therefore decided to create a sophisticated trace system to assist with debugging and testing. Trace functionality would be added to all aspects of the system and could be enabled dynamically.

To implement the designed trace functionality required that standard trace code be included at numerous points in the application – including the different component and middleware frames, proxies, adapters etc. Using more traditional tools it can be very difficult to develop, test and maintain such infrastructure code. Although "trace" is a single aspect, the code to implement it needs to be created "all over the place" within the application.

SCORE Adaptive Bridges directly supported the project team with various elements of aspect-oriented programming including aspect-weaving and code-weaving. The code associated with the trace "aspect" was simply defined in a central location using code patterns. The ANGIE frame-based generation technology was used to define the points in the generated application code the trace code should be generated.

A major advantage of the SCORE Adaptive Bridges architecture is that all information about the component to be generated is held in the composition repository. The code generators therefore have a broad view of what needs to be generated. The feature-oriented code is easily "weaved" to the right location, in the right component source file, in the right generated source object. The generator gets it right – each and every time.

DEVELOPMENT TOOL INTEGRATION WITH SCOUT²

Suva's project used 8 different development tools on three different platforms – including SCORE Adaptive Bridges, Visual Age for Java, WebLogic Enterprise, COBOL and MKS Source Integrity.

The SCOUT² Development Platform from Delta Software Technology was easily able to integrate the Suva project team's development tools and workflow processes, creating a seamless platform-independent view across Suva's development, test and production environments.

SCOUT² managed all compilation, linkage etc. of the platform-specific source code for the services, together with change control and other common infrastructure tasks.

SCORE Adaptive Bridges is fully integrated with SCOUT², allowing members of the Suva project team to discover, compose and produce their new business services in a familiar, flexible and extensible development environment.

project metrics

The following table gives an overview of the size of the project in terms of the number of components generated by Suva using SCORE Adaptive Bridges. For information the number of generated lines of code is also shown. 100% of this code is generated from the composition repository.

System Component	Source Files	Language	Lines of Code *Note: Includes generated test framework*
Business Tier Server Components	407	SCORE Integration Suite	
Generated Server-Side Frames	395	COBOL	3,500,000
Generated Suva GUI Framework Adapters	393	Java	168,000
Generated Client-Side Proxies	391	Java	500,000

Key Business Benefits

Migrating the legacy COBOL business and data access tiers to the new architecture with SCORE Adaptive Bridges delivered a wide range of tangible business benefits to Suva, including:

investment protection

The considerable investments Suva had made in the legacy COBOL business and data access tiers were fully protected. SCORE Integration Suite allowed Suva to directly reuse around 5,000,000 lines of existing COBOL code.

developer productivity

SCORE Adaptive Bridges takes away from the developers the need to worry about middleware and common infrastructure features such as interface versioning, data conversion, parameter mapping etc.

By automatically generating all such code SCORE Adaptive Bridges allows Suva's developers to focus on the core business functionality of the application. This is a "win-win" situation as the developers have more time to dedicate to business functionality – and – Suva applications have a leading-edge infrastructure for no additional effort.

non-invasive integration

The functionality to support service-based integration was added non-invasively. This ensured that Suva reduced to the minimum any additional support or maintenance costs for the core business functionality. The core COBOL code remained totally unchanged.

"Our experience with SCORE Adaptive Bridges in the prototype and pilot phase were confirmed when we scaled up to the real project – 60 developers, 30,000 source files and 12,000,000 lines of code. SCORE Adaptive Bridges delivers what it promises."

*Robert Koch
CIO, Suva*

flexible client integration

The Suva project team defined the interfaces to their business services in a neutral manner in the composition repository. From here the complete client interface code is generated for the selected deployment platform.

Currently the Suva application is using a standard Java GUI. Initial testing has already started with J2EE/EJB clients. SCORE Adaptive Bridges enables Suva to support such new clients from the existing server specification. The server frames generated by SCORE Adaptive Bridges can support multiple client technologies in parallel.

At no time do Suva developers need to be concerned with any of the technology details of connecting clients to servers.

platform independence

All information on components and deployment platforms are held in platform-neutral format in the XML-based Composition Repository. As Suva introduces additional deployment platforms the existing component specification can be reused to automatically generate all the code for the new platform.

Full platform independence gives Suva both the flexibility to meet new requirements and to avoid potential lock-in by any single platform vendor.

"We generated services for diverse middleware and platform combinations – SCORE Adaptive Bridges handled this effortlessly. We are now working on our strategy for future projects with Delta."

Thomas Müller
Manager
Application
Architecture and
Methods, Suva

open standards

SCORE Adaptive Bridges builds on open standards from the Object Management Group (OMG) and other standards bodies. All specifications, repositories and code generation patterns are open to Suva – nothing is "hidden" in proprietary file formats or "secret" databases.

All communication between clients and servers use standard Internet and Intranet protocols. There is no proprietary communication protocol at SCORE Adaptive Bridges. All generated code is well structured and documented for ease of understanding.

service-based integration

The legacy OS/2 fat client architecture was not designed for deployment in a distributed environment. Even so, SCORE Adaptive Bridges has allowed Suva to take the existing COBOL business tier and deploy it according to the latest thinking on service-based architectures.

Rather than simply exposing the legacy "wide" interface structures to the new GUI clients, Suva took the time to design fine-grained interfaces with only the minimally required data fields. SCORE Adaptive Bridges generates all the code to map between these "internal" and "external" interfaces.

The clearly defined and documented services help Suva to reduce maintenance complexity. Suva is also well positioned to integrate business functionality for new channels such as Web services and portals.

The legacy OS/2 fat client architecture was inherently stateful with data being retained in OS/2 memory across transactions. The new architecture called for a stateless design to get the most out of the BEA WebLogic Enterprise application servers.

A stateless design provides considerably more flexibility in designing load balancing and server fail-over for a robust and reliable deployment environment.

SCORE Adaptive Bridges solved this challenge for Suva by automatically generating all state management code. This code respects the architectural constraints of the WebLogic Enterprise environment while ensuring that the GUI clients do not have to be concerned with any state management issues.

Suva has the best of both worlds – stateless system architecture for increased availability with all state management being handled transparently.

SCORE Adaptive Bridges' automatic optimization feature dramatically reduces network traffic and application server loads compared to typical distributed system implementations.

Using the Composition Manager Suva can automatically "batch" operation calls, reducing the interactions between clients and servers. All the code for client-side caching and operation sequencing control is generated fully automatically. The client developer is not aware of the optimization.

multiple client versions

SCORE Adaptive Bridges supports multiple versions of each component interface. This feature allows Suva to support multiple versions of their client application from the same server instance.

This is a valuable benefit that considerably simplifies rollout of new application functionality without requiring all clients be updated in parallel. It also reduces system management overhead in the WebLogic Enterprise deployment environment.

generative approach

As with any large project things do not always follow the original plan. It is therefore important that the tools and methods that are being used are flexible enough to react to changes in requirements.

The Suva project showed how the core SCORE Adaptive Bridges were quickly and easily updated to meet Suva's specific code generation requirements. For example, creating an additional framework adapter layer for the GUI clients.

Using Delta Software Technology's innovative generative development tools, including PBE Pattern By Example and ANGIE Frame Generator, the Suva project team was able to make many of these changes themselves without waiting for the next product update.

This flexibility to react to project specific requirements is a hallmark of the generative approach implemented by Delta Software Technology.

"With SCORE Adaptive Bridges we see the results of our long-term commitment to generative programming and emerging technologies.

Rüdiger Schilling CEO, Delta Software Technology

Business Partner

Suva partnered with SAXOS Informatik AG (www.saxos.ch), distributor of Delta Software Technology's products and services in Switzerland and Liechtenstein.

Based in Zurich, SAXOS has more than 20 years experience of Delta products and specializes in the use of advanced software tools for software development, integration and migration.

Senior consultants from SAXOS worked closely with Suva throughout the project. A key role was working with Suva's architecture group to establish the migration concept, define the architecture and implement the required tools and automation for SCORE Integration Suite.

SAXOS also worked closely with Delta Software Technology's product architects and developers to create project specific enhancements for Suva using PBE Pattern By Example and ANGIE Frame Generator.

"We are delighted with the results of the Suva project and are looking forward to working with Suva on new projects."

*Norbert Nigg
CEO, SAXOS*

DELTA SOFTWARE TECHNOLOGY

Delta Software Technology is a specialist for generative development tools that automate the modernisation, inte¬gration, development and maintenance of individual IT applications.

We understand the enterprise IT as a living organism that is continuously changing. Our automated solutions help you to quickly and safely adapt your applications to new business requirements, architectures, technologies and technical infra¬structures.

Delta has a more than 30-year track record of successfully delivering advanced software technology to Europe's leading organisations, including AMB Generali, ArcelorMittal, Deutsche Telekom, Hüttenwerke Krupp Mannesmann, Gothaer Versicherungen, La Poste, RDW, Suva and UBS.

AMELIO® Modernization Platform™
The tailor-made factory for the modernisation of large IT applications: 100% automatically and that's why it is safe, reliable and error-free.

SCORE® Adaptive Bridges™
Intelligent service enablement for the reuse of proven applications with modern technologies: flexible, profitable and non-invasive.

SCORE® Data Architecture Integration™
Data as real business services: fast, easy and independent of data architectures and management systems.

SCOUT²™ Development Platform
Optimized and integrated development processes across all software components, tools and platforms: Stop the "fight against the infrastructure".

ADSplus™ Application Development
Platform-independent development for future-proof back-end applications.

www.d-s-t-g.com