300 at one blow!

modernizing a NonStop application at Elmer Dienstleistungs GmbH & Co. KG

September 26, 2012

Juergen Depping
© CommitWork GmbH
300 at one blow!

The headline relates to a fairy tale ...

... modernizing 300 ScreenCOBOL dialogues
agenda

• introduction of Elmer Group

• motivation

• course of the project
  – consulting, workshops, prototyping
  – infrastructure setup
  – advanced concepts & solutions
  – implementing, testing and impressions (samples of new features)
  – [ release management, if enough time ]
  – next steps

• peek into the crystal ball

• conclusion
Elmer Group

- specialized retailer for sanitary, heating and air conditioning in Germany
- modern, mid sized and family led group of firms
- trans-regionally operating provider
- closeness to customers reflected by more than 60 distribution- and exhibition locations
motivation
initial situation

• snapshot of existing “Legacy“ application software
  – dialogues implemented in ScreenCOBOL
  – Pathway servers implemented in COBOL
  – data sets mainly based on Enscribe
  – manual source code management, history managed via Backup

• findings
  – software development and user interfaces definitely not „state of the art“
  – integration and interaction with other IT systems critical
  – modernization projects piling up
objectives

• state of the art graphical dialogue interface

• modernized software development

• open minded about new stuff
constraints

• modernizing embraces more than 300 ScreenCOBOL dialogues

• first modernizing step is not touching COBOL servers and not changing any interfaces

  approach allows parallel operation with existing software

  planned „step by step“ rollout to the branches

• Elmer IT staff getting involved in software development

  aiming at independent maintenance and future development of new applications
course of the project

1. CommitWork: consulting, workshops, prototyping
CommitWork consulting – initial topics

• customer questions:
  – rich or thin client?
  – what for is OmnivoBase?
  – Open Source Tomcat or NSJSP?
  – experiences from other projects e.g.
    • Thyssen Krupp Steel AG
    • Rasselstein GmbH
what for is OmivoBase? an application environment – desktop view

dynamic menus

dialogue selection

dialogue panel

open dialogues list

choose between desktop- or single window view

= resizable panels
OmnivoBase – user & dialogue management

- adaptable menus, user privileges, groups, standardized groups, open window design, invoking external programs from menu, ...
CommitWork Consulting: OmnivoBase – architecture

- protecting investment - exchangeable GUI builder
- protecting investment - exchangeable comm layer
prototyping

- A sample part of order management application was implemented at this project stage.

- Performance tests and analysis were carried out by using actually available “Wyse thin clients”.
Migration effort analysis

• later, only analysis samples were taken from existing systems and existing source code was rated following a certain scoring

• criteria included:
  – amount of lines of code
  – amount of fields to display
  – portion of business logic / plausibility
  – amount of server calls
  – amount of calls to sub-dialogues

• analysis findings:
  – Min: 7 hours per dialogue
  – Max: 16 hours per dialogue
  – Average: 9 hours per dialogue
course of the project

2. infrastructure setup
infrastructure

• hardware:
  – NonStop NS2000 with NonStop OS release J06.09
  – installing virtual SuSE Enterprise Server for CVS, Nexus & Jira

• software:
  – CVS repository
  – Maven repository Nexus
  – Jira for project monitoring and tracking
  – Java on NonStop
  – initially Tomcat, later NSJSP
  – OmnivoBase

• Elmer staff seminars individually and along the project, covering Java, CVS, Maven, NSJSP & OmnivoBase
course of the project

3. follow up concepts and solutions
modularizing

- without breaking down into modules, a monolythic construct is arising, possibly leading to a system collapse, when growing rampant

- the whole application is split up into 24 separate projects, each consisting of many modules

- by utilizing vertical partitioning, single projects out of the total application can be deployed to different NSJSP clusters (an option supported by OmnivoBase)

this kind of separation is transparent for developers and users
vertical partitioning of applications  (high availability)

shutdown of separate application clusters for software updates is feasible
(NSJSP Cluster)!
course of the project

4. implementing, testing & impressions
implementation and testing

- Elmer IT implemented appr. 125 dialogues and CommitWork took appr. 175 dialogues

- state of the art tools like JIRA were utilized to cover project-, task- and error management

- migration of dialogues very close to “old“ dialogues in some cases combining dialogues or implementing ergonomic enhancements
„300 at one blow!“ - code conversion in a very short time

Screen-COBOL screen

Omnivo ScreenCOBOL-Wizard

client

presentation with OmnivoSwing

behaviour

client & server

transport class

DDL

server

HP DDL2JAVA class

Omnivo DDL2JAVA-Wizard

CommitWork
Omivo Screen-COBOL: 3000 questões blow!
ScreenCOBOL-Inspector – generated view
ScreenCOBOL-Code – CDS from Commitwork
Java code – 1:1 manual conversion
this finished dialogue
OmnivoScreenCOBOL & OmnivoSwing

• OmnivoScreenCOBOL Wizard:
  – generating view classes from ScreenCOBOL screens
    utilizing OmnivoSwing elements for user interface design
  – generating code fragments for behavior classes

• OmnivoSwing
  – based on Java Swing; currently containing 79 classes
  – enhanced functionality:
    • binding to transport objects
    • validation
    • data mapping and conversion
intermediate conclusion –  
– OmnivoScreenCOBOL & OmnivoSwing

• sticking to planned efforts by utilizing OmnivoScreenCOBOL & OmnivoSwing

• till today no generators available, providing maintainable, object oriented Java code from ScreenCOBOL code

only views, validations and bindings are generated

everything else is coded manually

source code therefore maintainable and ready for enhancements

developers get along well with the new Java source code!

• some impressions ...
impressions

- new design elements like “input date“ field
- pop up menu: selection list
- other design elements

- dialogue selection for input fields
impressions (2)

- validation, error messages, ...

- pop up menu for presenting history of messages
impressions (3)

- dialogue sequences as tab pages
  no field changes allowed, but fields are visible!
• integrating an online HELP function

pressing “?” button invokes a PDF Help file for the appropriate dialogue
impressions (5)

- compared to ScreenCOBOL, more table entries can be listed and processed

additionally available are sorting, filtering and different column selections
course of the project

5. introducing Release Management
introducing a new Release Management

• current versioning products used in parallel:
  
  – Maven's release management works with libraries
different versions are filed in POM files
distinguishes between “Snapshot“ and real “Release“

  – CVS' release management allows assigning “Release Tags“
also development in parallel is feasible (so called “Branches“)

  – OmnivoBase provides client- & server version checking

    • Major Release   - interface changes force new Major Release
    • Minor Release   - other changes
introducing a new Release Management

• the finally selected release management covers every kind of versioning and is focused at:
  
  – development
  – acceptance
  – production

• to avoid interferences between enhancements and bug fixing for later, there is a distinction between hot fixes for production (CVS Branch) and application enhancements (CVS HEAD)
course of the project

6. next steps
next steps

• decision for deployment of Terminal Servers
  optimizing software distribution, currently based upon Java Webstart

• followed by a step by step rollout to the 60 locations of Elmer Group
peek into the crystal ball
peek into the crystal ball

- migrating database from Enscribe to SQL
- coupling and integrating application to other systems
- modernizing services as pure Java servers
- introducing Java based reports (Jasper Reports)
- adapting to TAPI interfaces
- ...
peek into the crystal ball

open for most everything

NonStop SOA stack
ITP-Webserver
NonStop JSP

Pathway
existing services

Java Beans
new services

3rd party

Microsoft .net
Internet Explorer
Java
OmnivoBase App
CommitWork
conclusion
requirements accomplished

- modern graphical dialogue interface
  - enhanced user management
  - enhanced management of user privileges
  - higher comfort using dialogues and dialogue flow

accomplished by:

- OmnivoBase: user control, user privileges, menu
  - very granular user privileges
    user can only get access to dedicated dialogues

- utilizing OmnivoSwing

enhancements:

- buttons instead of function keys
- utilizing new GUI elements
- dialogue flow using „Tab Cards“, online HELP functions, HELP dialogues,...
requirements accomplished (2)

- modernizing software development
  - COBOL development using modern IDE
  - source code control using repository (also for COBOL)
  - enhanced release management

accomplished by:

- CommitWork Developer Suite (CDS – COBOL)
- using CVS within CDS
  (revision proof source code management)
- new release management concept by using
  CVS, Maven and OmnivoBase
requirements accomplished (3)

- open for new stuff
  - integration in interactions with other systems
  - modernizing server software
  - utilizing state of the art extensions, available in a Java environment

accomplished by:

- various features of a Java environment
  - web services
  - EJB
  - access to remote databases via JDBC
deploying OmnivoBridge, …

- pure Java migration of existing services

- utilizing commercial and „Open Source“ Java 3rd party products
additional enhancements

- vertical partitioning (by modularizing) allows almost seamless growth of applications – no limits

- OmnivoBase's architecture ensures full investment protection, allowing to simply exchange middleware products

  existing customers already changed middleware environments easily:
  - CORBA → NSJSP
  - Oracle WebLogic (previously BEA) → NSJSP

- project management using JIRA
modernizing, but properly done!

need more information?

Juergen Depping
depping@commitwork.de
+49 231 94116912