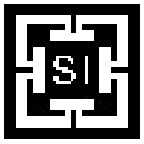
UBS

A Desktop for Co-operative Work
OO Experiences From a Large Banking Project

Daniel Megert
daniel.Megert@acm.org

UBS AG
P.O. Box
8098 Zurich
Switzerland



CHOOSE Forum '99 on
Object-Oriented
Software Architecture

Agenda

- Background
- Major challenges
- How we met these challenges
- Development process and team organization
- Lessons learned
- Q & A

Background

- UBS AG
- Corporate Desktop

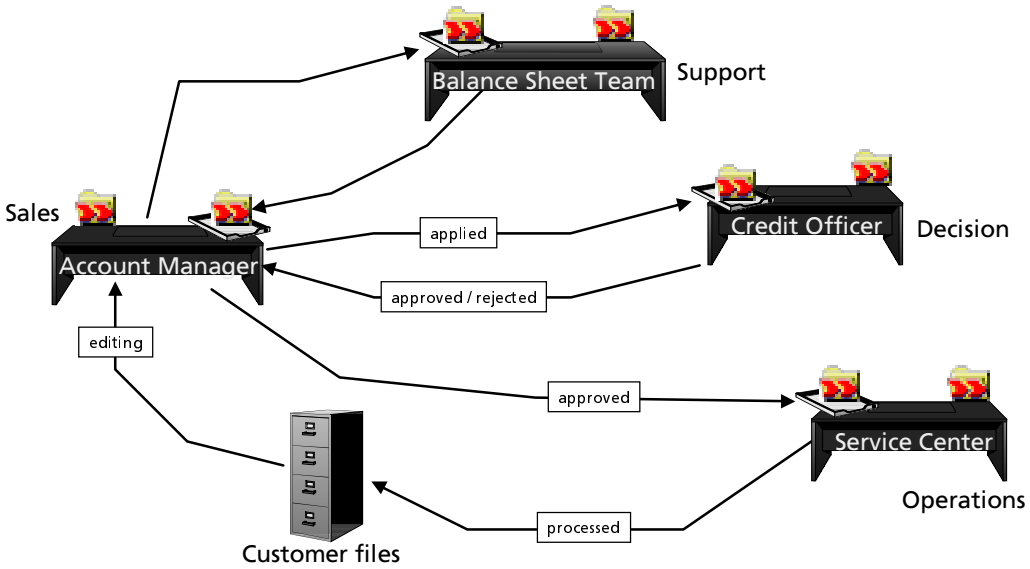
Background - UBS AG

- SBC and UBS legally merged to UBS AG last June
  - Merger is taking place right now and consumes a lot of resources
- 50'000 employees, 4'000 working in the IT dept.
  - One of the biggest IT development centre in Switzerland
- Market capitalization of USD 75 billion
- USD 1'033 billion assets managed

Background - Corporate Desktop

- Motivation
  - Save money
  - Standardize the credit process
  - Large legacy system with very old UI (no GUI)
  - Many tools involved for a single business TRX
  - Implement new analyzing and rating rules
- Process characteristics
  - Long duration of a Business TRX (days to weeks)
  - Parallel handling of many business TRX
  - Interruptions (e.g. by a customer)
  - Standard workflow with slight variations

Background - Corporate Desktop



Background - Corporate Desktop

- Purpose
  - Credit risk management and loan administration
  - Support complex work processes
  - Integrate different tools in a single workplace system
- System characteristics
  - 500 concurrent users right now (>1000 authorised)
  - Developed by 25 people in 18 months
  - Survived merger of the two banks
  - Purely object-oriented
  - Framework already reused by 3 other projects

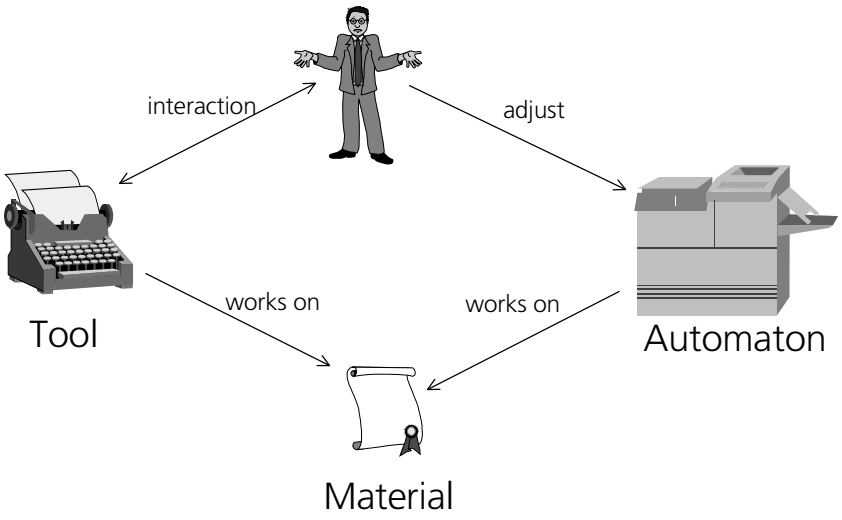
Major Challenges

- Seamless integration of
  - mainframe into a pure o-o client/server environment
  - office documents
- Flexible support of workflow
- Offer *one* use-metaphor with a consistent UI
- Prevent re-entry of the same data
- Reduce paper work
- Build an extensible system that fits the user's needs and can easily be adapted to new business requirements

How We Met these Challenges

- *Tools & Materials* and the *Desktop* metaphor
- Business transaction folders (BTF)
- Integration of
  - Legacy systems and mainframes
  - Office documents
- Our architecture

Tools & Materials in a Nutshell

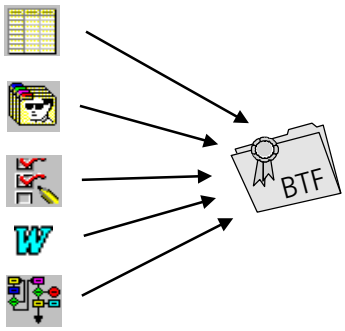


Tools & Materials in a Nutshell

- Developed at GMD and University of Hamburg
- Proven by the GEBOS system (RWG, Stuttgart)
- Materials, e.g. a note
- Tools, e.g. a typewriter or text-editor
- Automaton, e.g. photocopier
- Folders to give the materials a unique place to live
  - keep and manage materials
  - can be private or public
  - can be used by one person or shared

The Business Transaction Folder (BTF)

- Pure materials
- Materialized mainframe data
- Hybrid materials
- Office documents
- Images



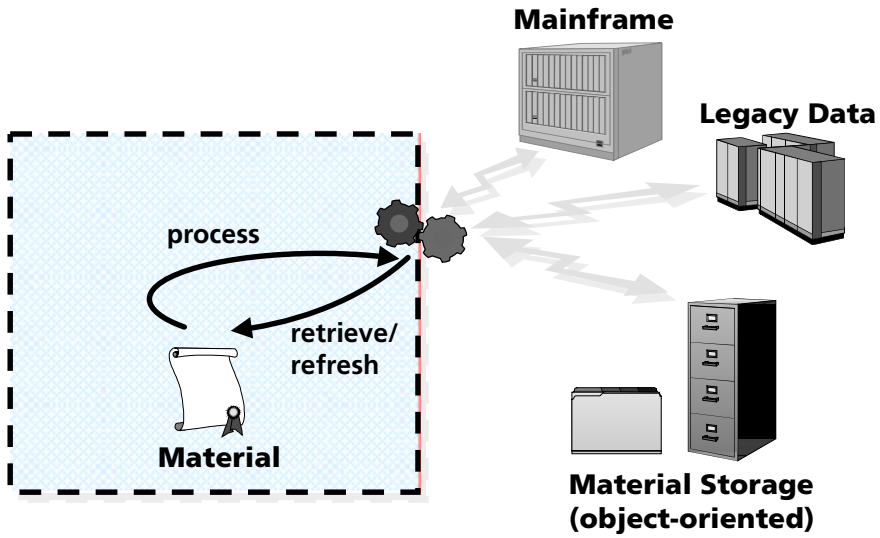
The Business Transaction Folder (BTF)

- All you need at your fingertips
  - manage all materials (business objects) that a user needs for a business transaction in one place
  - no more searching in different systems and places
- Easily generated by an electronic assistant
- Good base for archiving
- Context for final processing on mainframe
- Supports processes either with a weak or hard workflow
  - send BTF to people and roles (shared inboxes)

The Business Transaction Folder (BTF)

- Not an ordinary folder
  - knows about a specific process type
  - has a state
  - has an associated checker
- Easy configuration and extension of the current system
  - new materials can be inserted into a BTF
  - workflow can either be attached to the BTF and/or be enforced by a tool

Integration of Legacy Systems and Mainframes



Integration of Legacy Systems and Mainframes

- Result of a read transaction is a *material* which holds the data at the moment the transaction took place
  - can live in the desktop system without a permanent link to legacy system or mainframe
  - can be updated with new data (refresh)
- An automaton accepts such a material and knows how (with which data) to update the mainframe
- Works with all kinds of systems
  - elegant and easy integration
  - no need to model the whole old system as objects

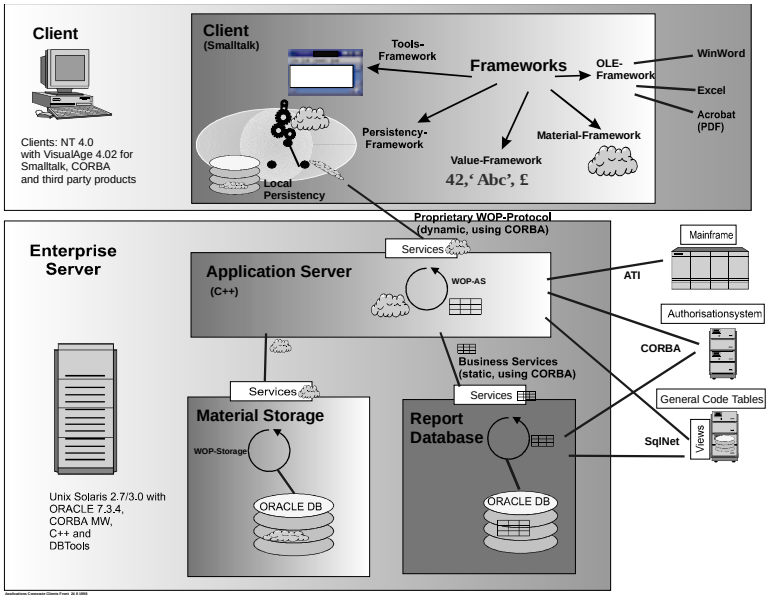
Integrating other Documents

- One of the key benefits for our users
  - insert new documents through a menu choice into a folder or a BTF
  - open the right tool for each document (e.g. Word) out of our Corporate Desktop environment
  - store the compressed documents in a relational DB
  - Word, Excel and Acrobat are already integrated
- Planed for the next release
  - drag and drop directly from Windows explorers
  - generic document support

Architecture

- Overview
- Development Statistics
- Modelling the database

System Architecture of Corporate Desktop



Architecture - Overview

- Classical three-tier service architecture using CORBA as a transport layer for our own WOP protocol
  - client calls services (no hard-coded server connection)
  - objects by value (not possible with current CORBA versions)
  - less time lost with IDL-changes
  - more flexible and dynamic
  - as of yet no performance problems
- Middle-Fat Client
  - allows easier switching to a laptop version
  - less network traffic

Corporate Desktop - Development Statistics

	Framework (Cal)	Application (Gf4)
Business Objects	30	25
Users	25	1500
Classes (client)	534	677
Methods (client)	8564	10607
LOC (client)	28959	67905
Runtime Image Size	-	5.7 MB
Classes (server)	376	26
Methods (server)	2426	342
Database Tables	54	81

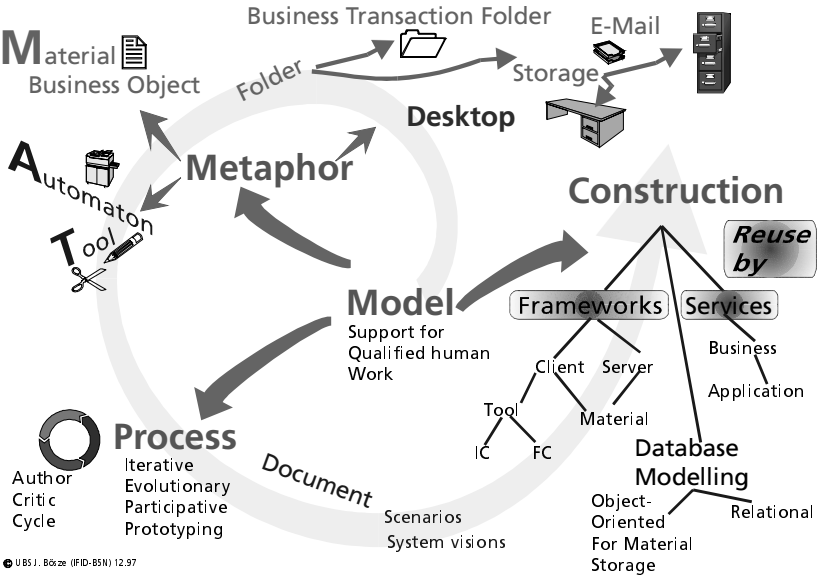
Modelling the Database

- Adhering to company standards
  - no o-o databases allowed
  - business data must be stored in 3NF
- Managing the impedance mismatch between relational and object-oriented world
  - used a rolled-down approach
  - container relation table with rolled-up attributes
- Managing the impedance gap between data access and data analysis
  - used two different databases

Development Process and Team Organization

- Development process
- Team Organization
- Impedance mismatch between client and server development

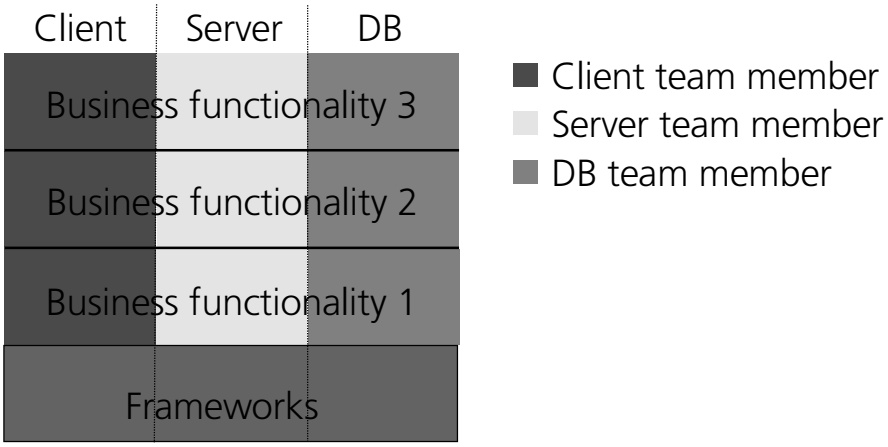
Development Process for Corporate Desktop



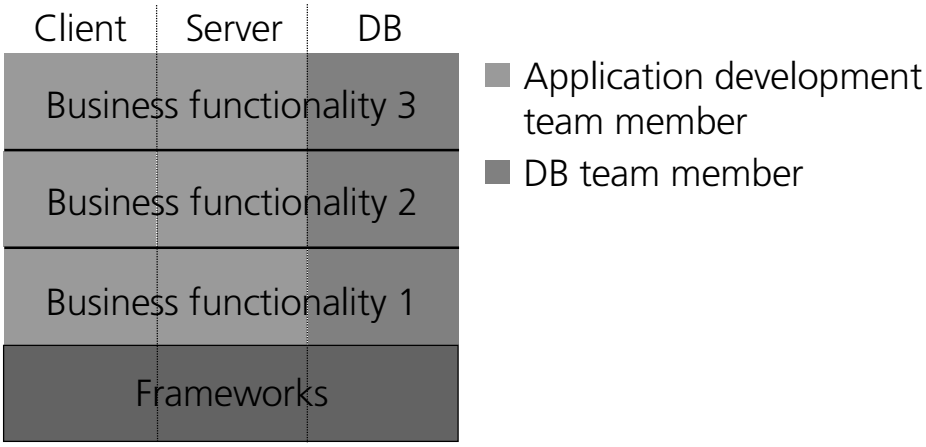
Team Organization

Role	Head count
Project Manager	1
QA Manager	1
Configuration Manager	1
Lead Architect	1
Database Manager	2
Framework Developer (client)	4
Framework Developer (server)	1-2
Application Developer (client)	8
Application Developer (server)	5

C/S Development Yesterday



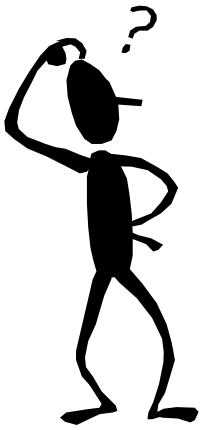
C/S Development Today



Lessons Learned

- Tools & Materials make developers think and program like the users, therefore more useful products emerge
- BTF wrap all a user needs for a specific business transaction and allows the definition of either hard or weak workflow
- Let the system do the integration and not the user
- Keep the interfaces between client and server programmers small
- Let the developers have fun (common lunch, grill parties, events...)

Q & A



Wanted: Software Engineer in my team

Tools & Materials: References

- [Bäumer et al. 97] D. Bäumer, G. Gryczan, R. Knoll, C. Lilienthal, D. Riehle, H. Züllighoven: *Framework Development for Large Systems*. Communications of the ACM, October 1997, Vol. 40, No. 10, pp. 52-59.
- [Budde et al. 92a] R. Budde, M.-L. Christ-Neumann, K.-H. Sylla: *Tools and Materials, an Analysis and Design Metaphor*. In: Proceedings of the TOOLS 7, New York, London: Prentice-Hall, 1992.
- [Budde & Züllighoven 92] R. Budde, H. Züllighoven: *Software Tools in a Programming Workshop*. In: Ch. Floyd, H. Züllighoven, R. Budde, R. Keil-Slawik (eds.): Software Development and Reality Construction. Berlin, Heidelberg: Springer-Verlag, 1992.
- [Lilienthal & Züllighoven 96] C. Lilienthal, H. Züllighoven: *Techniques and Tools for Continuous User Participation*. In: Proceedings PDC-Conference, Boston, November 13-15, 1996. p. 153-159.
- [Lilienthal & Züllighoven 97] C. Lilienthal, H. Züllighoven: *Application-Oriented Use Quality, The Tools and Materials Approach*. Interactions Magazine, ACM, November+December 1997, Volume IV.6.
- [Riehle & Schnyder 94] D. Riehle, M. Schnyder. *Design and Implementation of a Smalltalk Framework for thr Tools and Materials Metaphor*. Technical Report 95.7.1, UBILAB Zürich, Switzerland: Union Bank of Switzerland, 1995.
- [Riehle & Züllighoven 94b] D. Riehle, H. Züllighoven: *A Pattern Language for Tool Construction and Integration Based on the Tools&Materials Metaphor*. In: Proceedings of the First Conference on Pattern Languages of Programs (PLOP '94), Monticello, Illinois, August 4-6, 1994, Paper B.
- [Riehle & Züllighoven 95] D. Riehle, H. Züllighoven: *A Pattern Language for Tool Construction and Integration Based on the Tools and Materials Metaphor*. In: J.O. Coplien, D.C. Schmidt (eds.): Pattern Languages of Program Design. Reading, Massachusetts: Addison Wesley, 1995. Chapter 2, pp. 9-42.
- [Wulf et al. 96] M. Wulf, G. Gryczan, H. Züllighoven: *Process Patterns - Supporting Cooperative Work in the Tools & Materials Approach*. In: Pro-ceedings of IRIS 19 1, 10.-13. August 1996, Lökeberg, Schweden. pp. 445-459.
- [Züllighoven 98] H. Züllighoven: *Das objektorientierte Konstruktionshandbuch*. dpunkt.verlag, 1998.