Komponentenprogrammierung und Middleware

Sommer 2002,
Prof. Andreas Polze

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Zielsetzung

- Entwicklung einer Sprüche- und Zitat-Verwaltung
- Komponentenbasiertes System
- Nutzung von CORBA 2 und Java
- Verwaltung von E-Mail-Anhängseln
- Nutzung eines einfachen Prozessmodells

Signaturen und Quotes (1)

- Signatur
  - Hier: an E-Mail oder Usenet-Posting angehängte Textzeilen
  - Trennzeichen (typisch „−− “)
  - Kontaktinformation
  - Sinnspruch
- Quote Of The Day / Spruch des Tages
  - Ursprünglich: Einfaches TCP/IP-Protokoll, welches auf entsprechende Anfrage Zitat zurückliefert
  - Zitat wird i.d.R. aus Datenbank zufällig ausgewählt

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Signaturen und Quotes (2)

Komponenten-Programmierung und Middleware

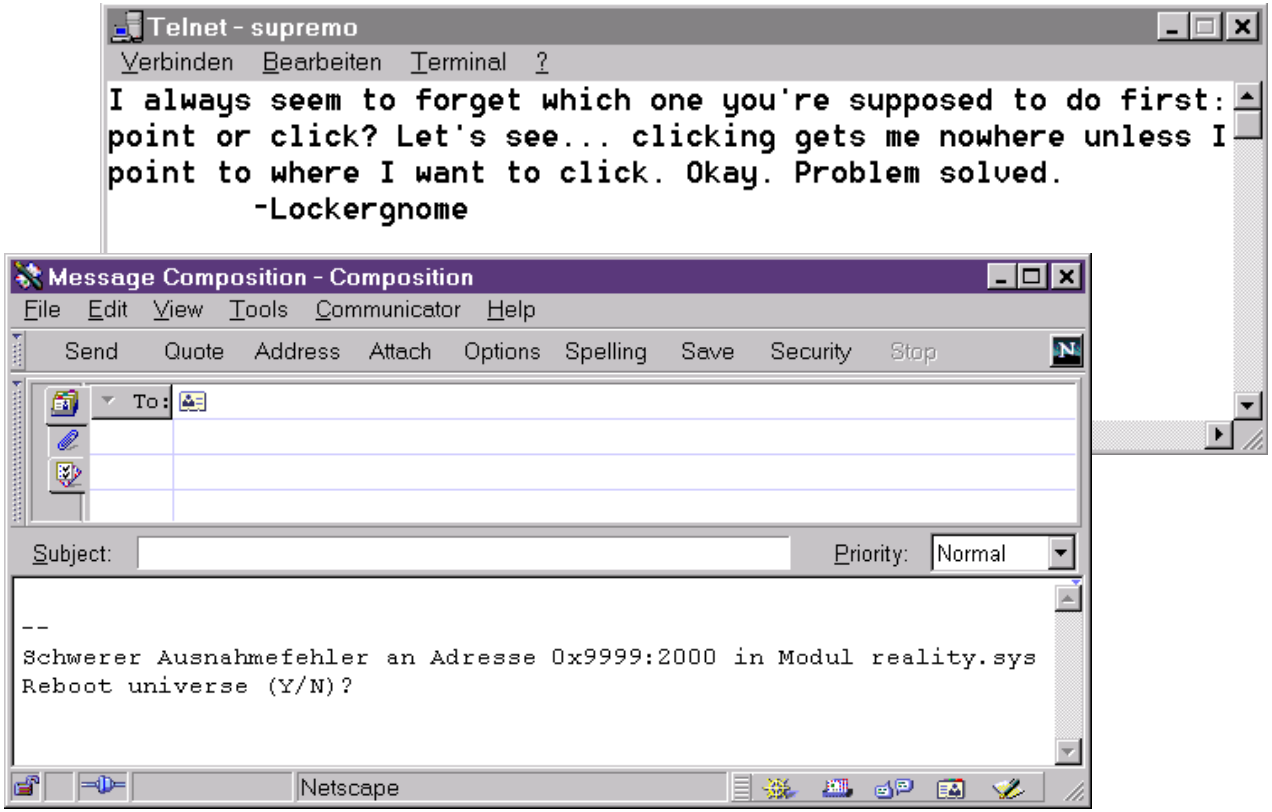
Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

- QOTD-“Session“ und Signatur in Email



JSig - Grundfunktionalität

JSig Funktionalität
Use-Case: Typische Benutzerinteraktionen
JSig- Aufbau
Persistenz und Personalisierung
Vorteile einer Datenbank-Nutzung
Datenbankschema
Klassendiagramm: Gesamtsystem
Klassendiagramm: Grundfunktionalität
Demo – Teil 1

JSig Funktionalität

- „Quote Of The Day“-Dienst über Java
- Anzeige zufälliger Signaturen
- Personenbezogene Profile
  - Login mit Passwort und Nutzernamen
  - Optionale Kontaktinformationen
- Optionaler Separator-String
- Persistente Speicherung von Quotes und Signaturen
- Versenden der Signaturen/Quotes per Mail

Use-Case: Typische Benutzerinteraktionen

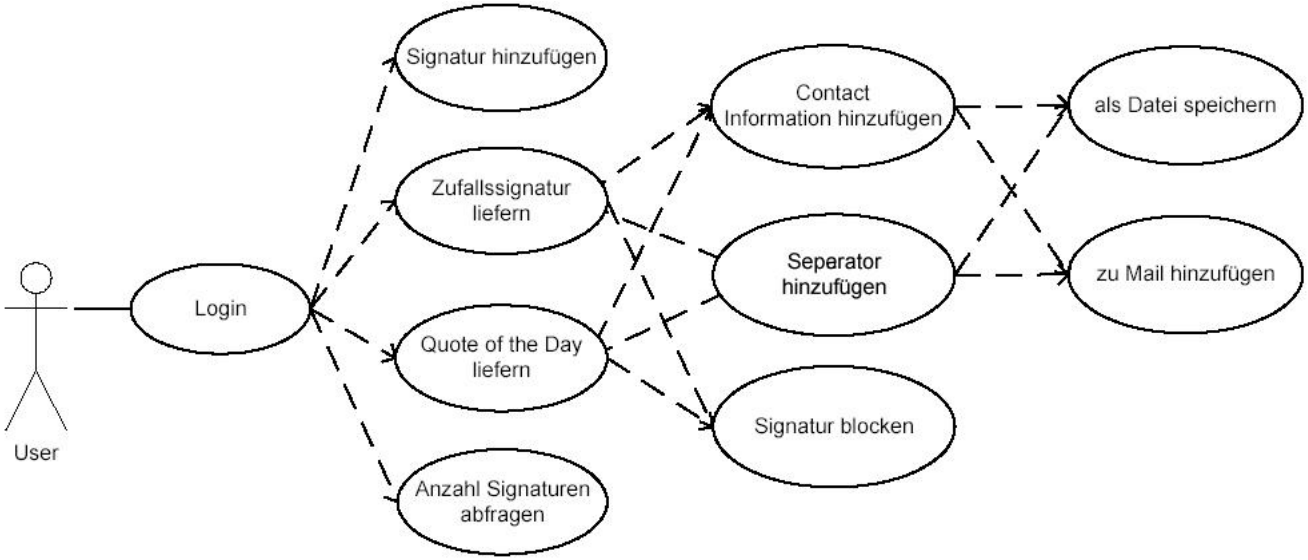
Komponenten-Programmierung und Middleware

Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig



JSig- Aufbau

Komponenten-Programmierung und Middleware

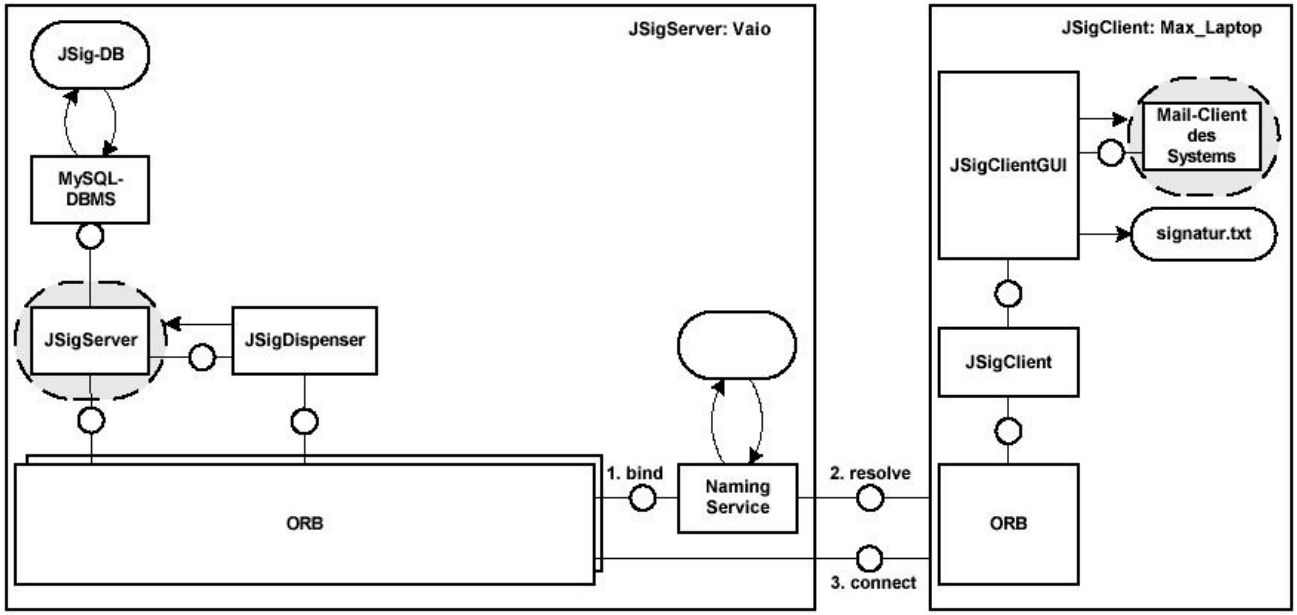
Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

- Aufbau des Grundsystems



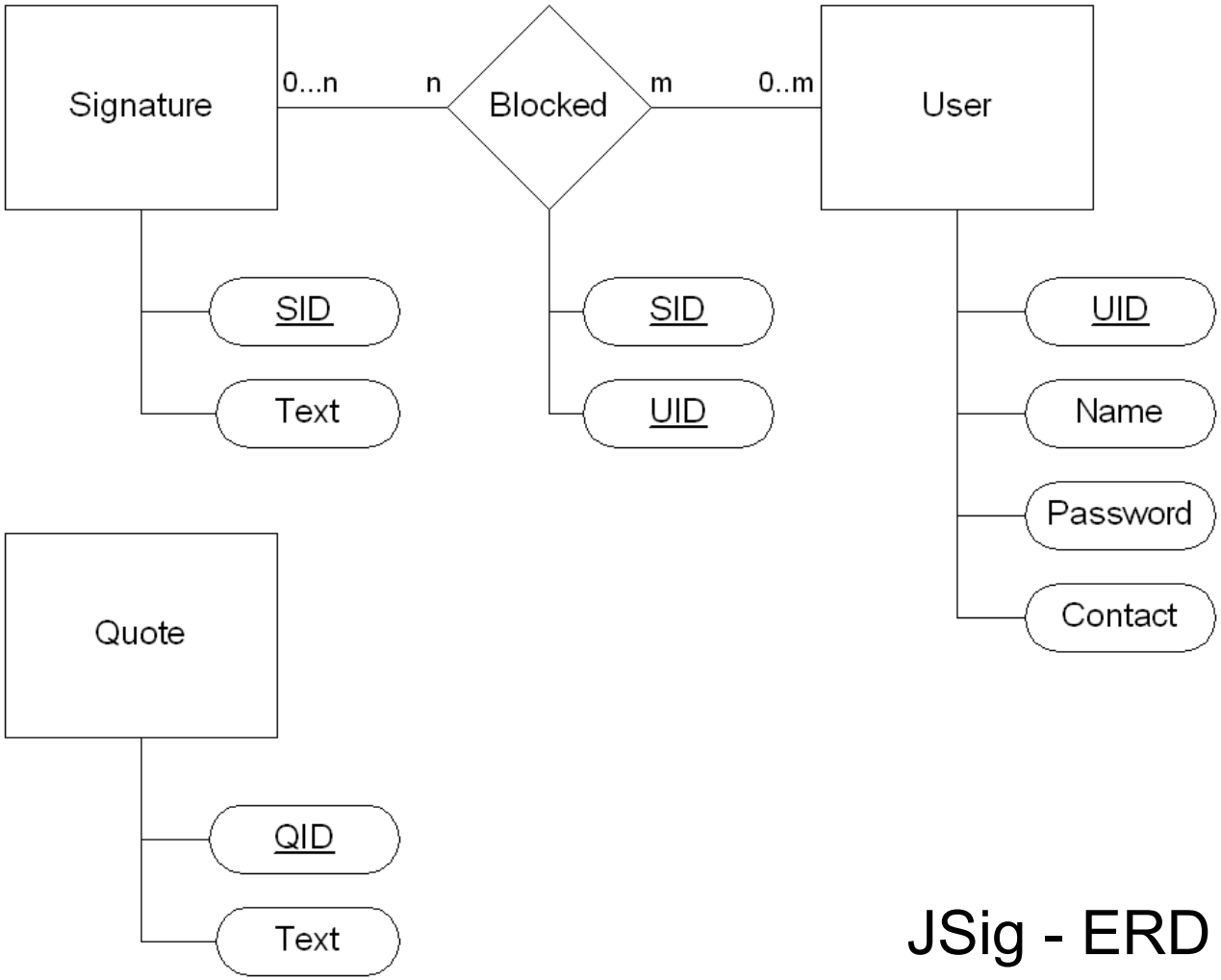
Persistenz und Personalisierung

- Passwort sowie Kontaktinformationen sind benutzerspezifisch
- Persistenz durch Datenbankanbindung realisiert
- Verhalten ist im Datenbankmodell direkt umgesetzt durch Abbildung der Informationen auf DB-Schema

Vorteile einer Datenbank-Nutzung

- DBMS erleichtert Programmierung
  - Konflikte beim Zugriff auf die Daten werden durch DBMS gelöst
  - Persistenz gegeben
  - Nutzung von SQL-Funktionen (`COUNT(*)` etc.)
  - I.d.R. kein Datenverlust bei Programmierfehler
  - Vereinfachung durch grafische Tools
- Daten können auf einfache Weise von anderen Diensten genutzt werden
  - Webservices etc. mit Nutzung gleicher Daten
- JSig: Nutzung von Suns **JDBC-Schnittstelle** und SQL-Datenbank

Datenbankschema



JSig - ERD

Komponenten-Programmierung und Middleware

Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

Klassendiagramm: Gesamtsystem

Komponenten-Programmierung und Middleware

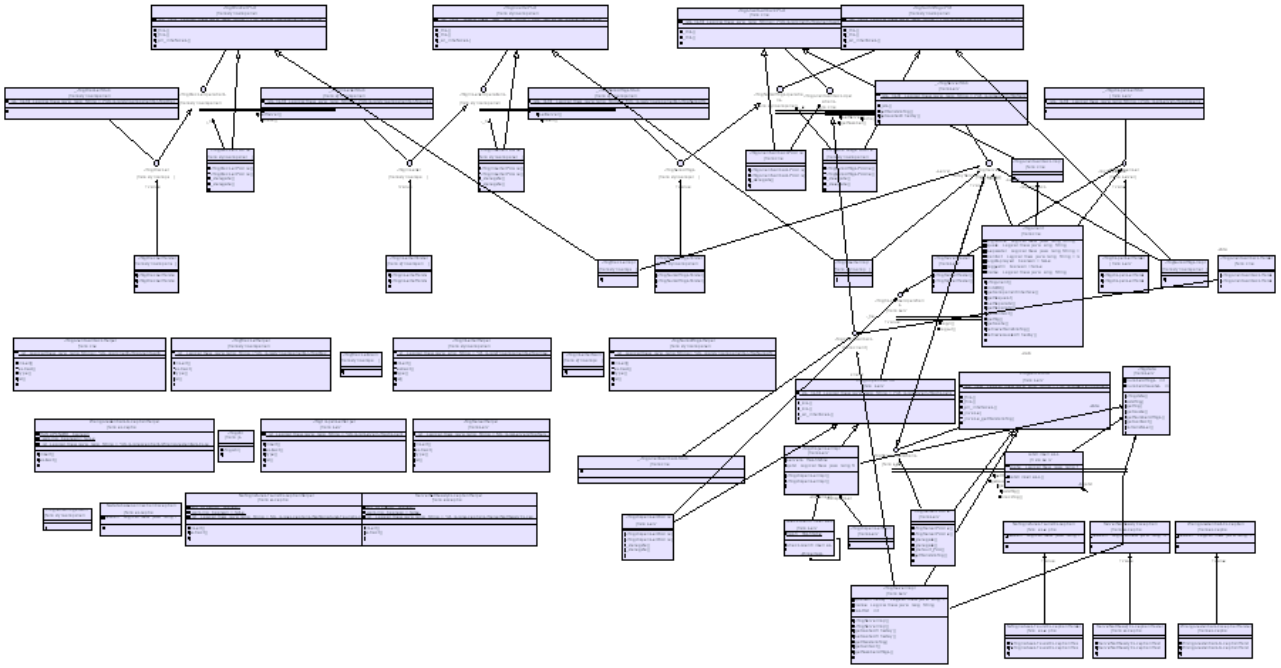
Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

- JSig: Klassendiagramm die Erste....



Klassendiagramm: Grundfunktionalität

Komponenten-Programmierung und Middleware

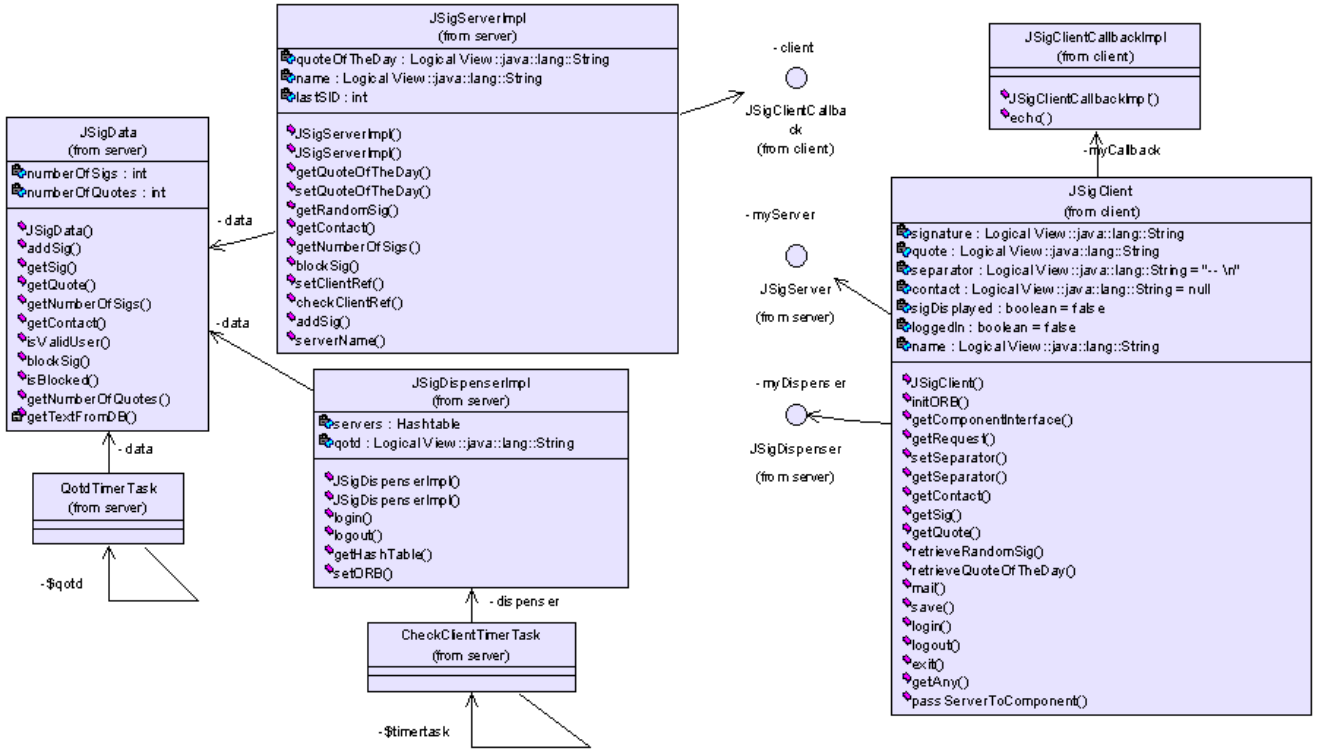
Sommer 2002,
Prof. Andreas Polze

17.06.2002

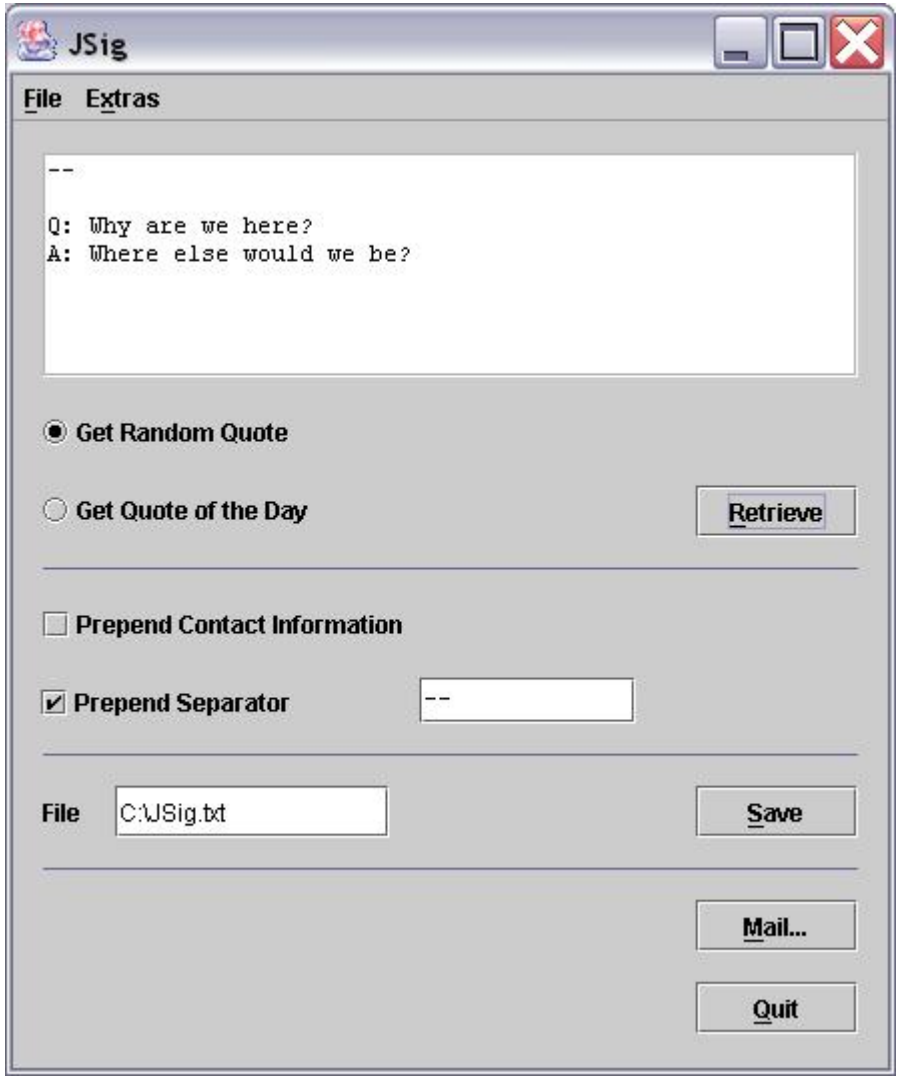
Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

- JSig: Die wichtigsten Java-Klassen



Demo – Teil 1



Komponenten-Programmierung und Middleware

Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

JSig - Dynamische Komponenten

Dynamische Komponenten
Interface Repository IR
Dynamic Invocation Interface DII
JSig - Ablauf
JSig - Aufbau erweitert
Dynamische Komponenten
Klassendiagramm
Demo – Teil 2

Dynamische Komponenten

- JSigBlocker
  - Signaturen können von einem Benutzer einzeln geblockt werden
  - Die geblockten Signaturen werden dann an *diesen* User nicht mehr ausgegeben
- JSigInserter
  - Signaturen können zum Dienst hinzugefügt werden
- JSigNumOfSigs
  - Ermittelt die Anzahl aller verfügbaren Signaturen

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

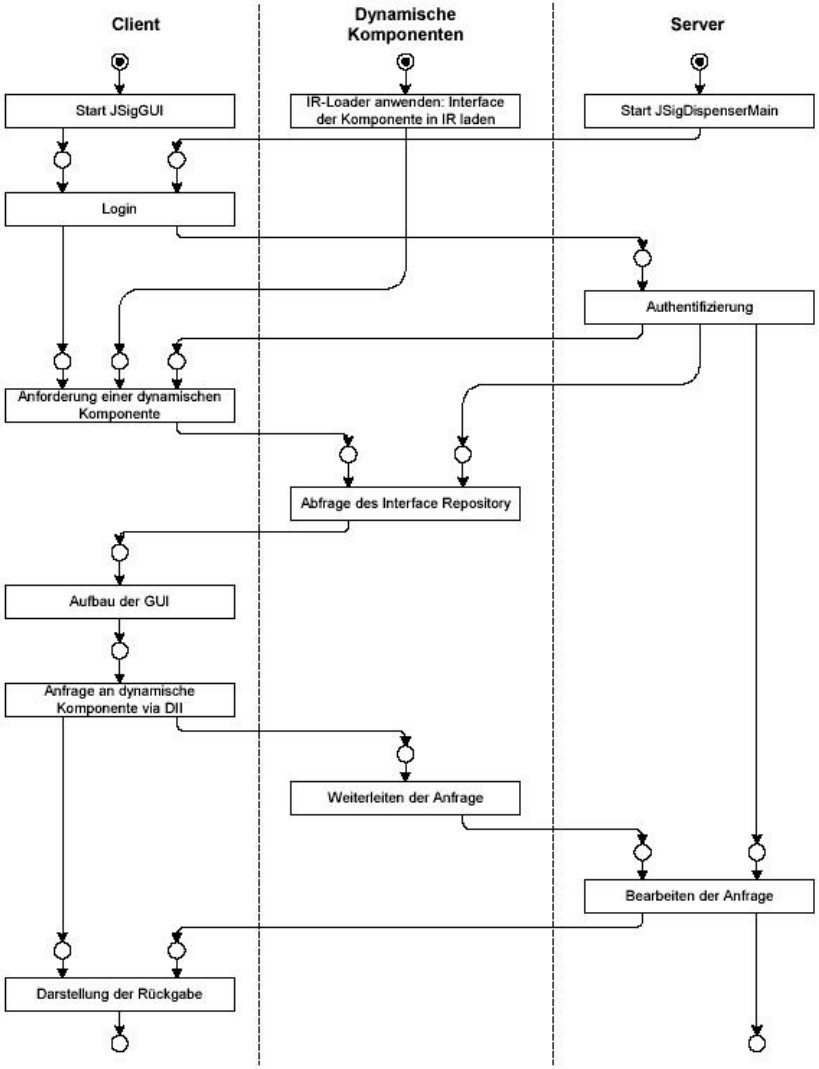
Interface Repository *IR*

- Stellt binäre Interface-Informationen bereit, die zur Compile-Zeit unbekannt waren
- Nicht von allen ORBs unterstützt
- Tool (Loader) liest IDL-Definitionen ein und speichert die Daten im IR
- Inhalt des IR stellt Objekt-Hierarchie dar
- OpenORB: Hierarchie kann mit dem Management-Board visualisiert werden

Dynamic Invocation Interface *DII*

- CORBA bietet das Dynamic Invocation Interface zur Verwendung von Interfaces zur Laufzeit an
- Ersatz für die vorkompilierten Client-Stubs
- Nicht nur Abfrage der existierenden Interfaces einer Komponente
  - Auch Methoden und Parameter der Interfaces
- GUI-Aufbau unseres Clients anhand der Methodensignaturen aus IR
- Ähnliche Techniken
  - **Java:** Introspection / Reflection
  - **COM:** Abfrage unterstützter IF durch IUnknown

JSig- Ablauf



Komponenten-Programmierung und Middleware

Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

JSig- Aufbau erweitert

Komponenten-Programmierung und Middleware

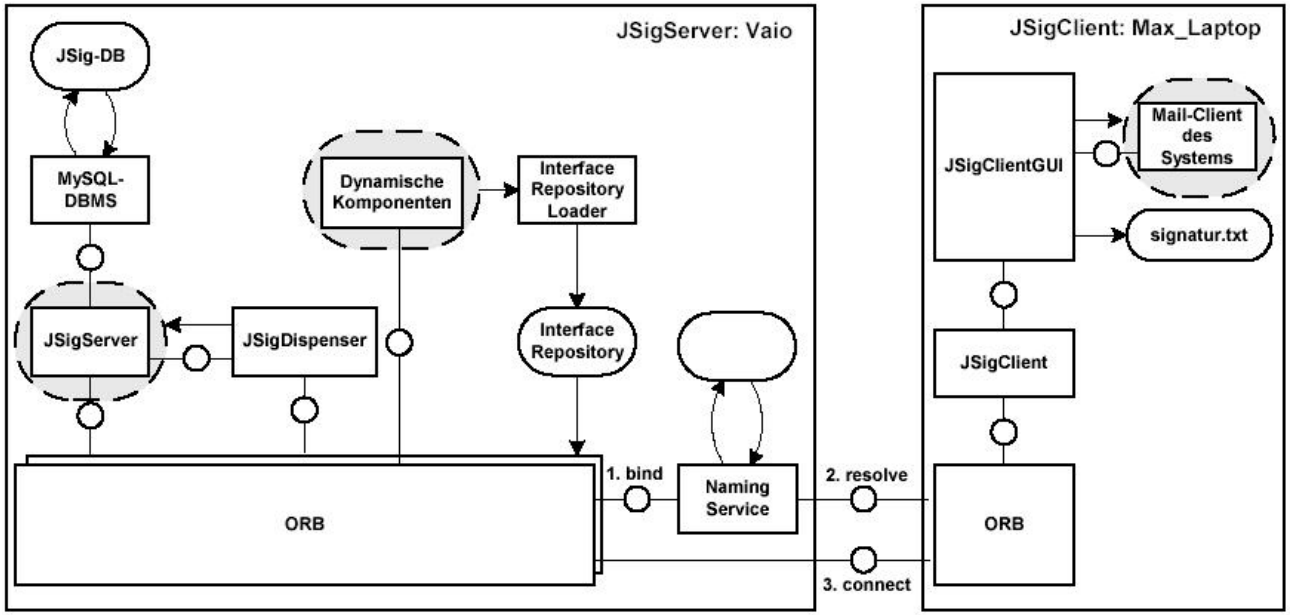
Sommer 2002,
Prof. Andreas Polze

17.06.2002

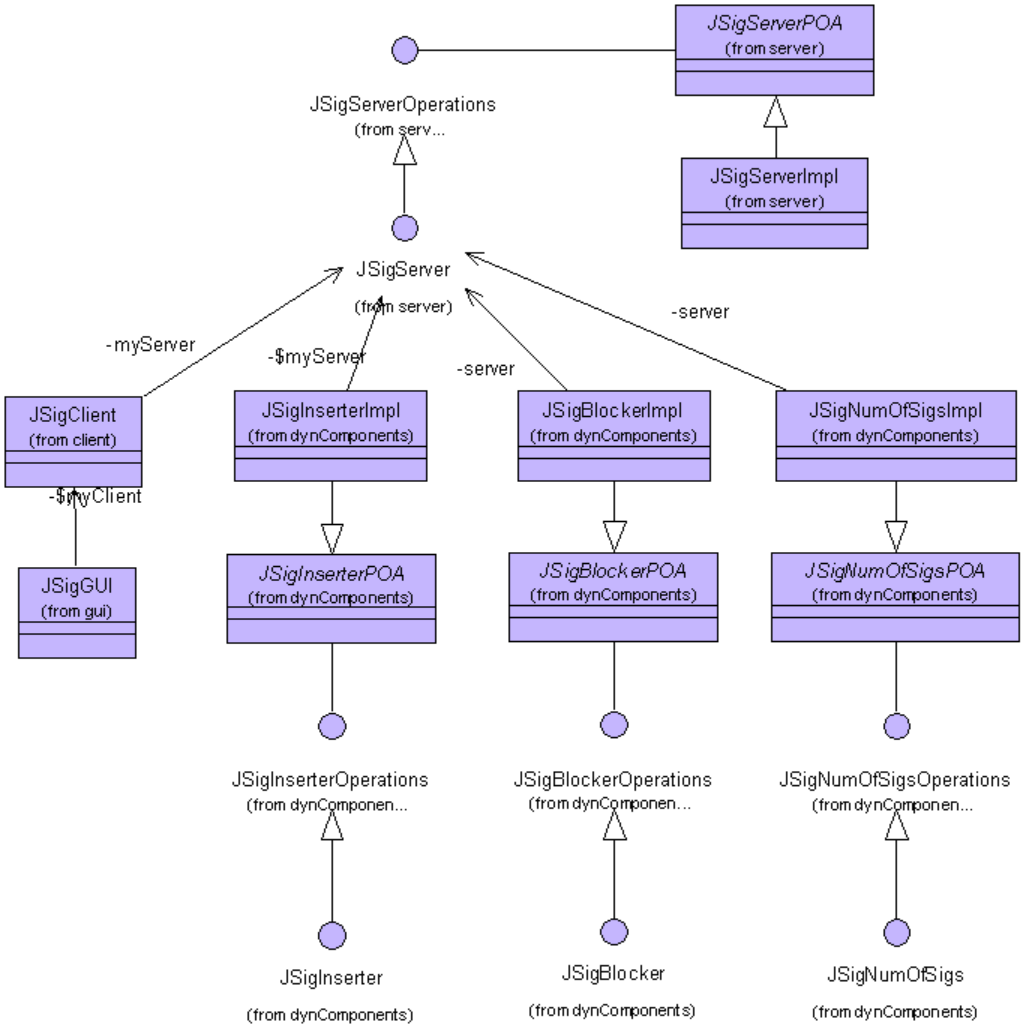
Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

- Aufbau mit dynamischen Komponenten



Dynamische Komponenten Klassendiagramm



Komponenten-Programmierung und Middleware

Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

Demo – Teil 2

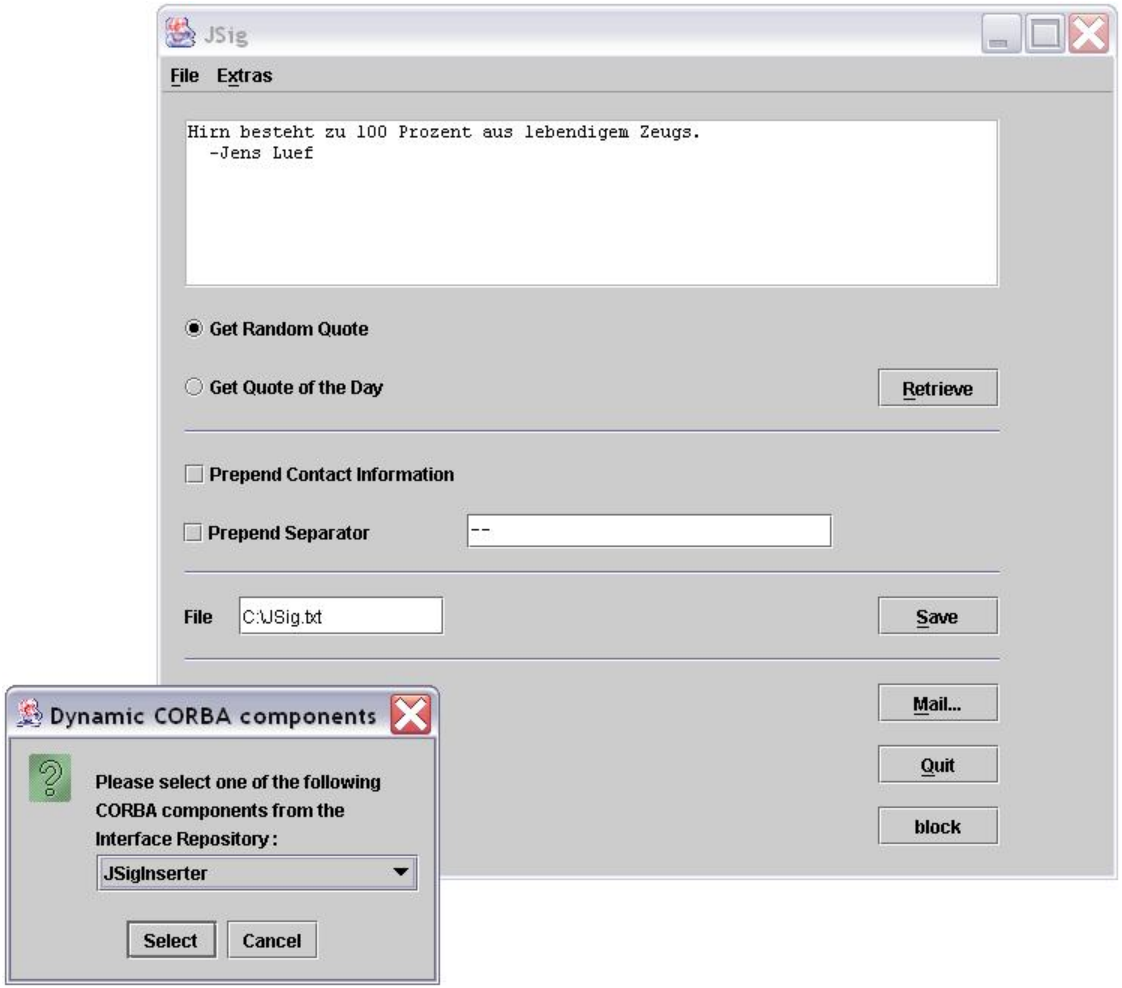
Komponenten-Programmierung und Middleware

Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig



Design Patterns

Model View Controller MVC
Singleton
Factory
Observer
Weitere Design Patterns

Model View Controller *MVC*

- Trennung von Darstellung und Datenverwaltung
- Verwendung indirekt durch Swing API
  - Bei Java Swing kein reines MVC
  - Funktionalität von View und Controller wird in UI Delegate zusammengefasst

Singleton

- Stellt sicher, dass von einer Klasse höchstens ein Objekt erzeugt wird
- Verwendung in den Klassen CheckClientTimerTask und QotdTimerTask

Factory

- Ermöglicht die Delegation der Objekt-Erzeugung an Unterklassen
- Verwendung in JSigData (`Class.forName()`)

Observer

- Änderungen des Zustands eines Objektes führen zur Benachrichtigung und Aktualisierung aller abhängigen Objekte
- Verwendung in JSigDispenser (TimerTasks)

Weitere Design Patterns

- Mediator (Vermittler)
  - Definiert ein Objekt, welches das Zusammenspiel einer Menge von Objekten in sich kapselt
  - Verwendung in JSigDispenser
- Composite
  - Fügt Objekte zu Baumstrukturen zusammen um Teil-Ganzes-Hierarchien zu präsentieren
  - Verwendung implizit durch Verwendung von Swing-Komponenten wie JFrame, JButton usw.

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Verwendete Tools und Systeme

CORBA und Java
Concurrent Versions System CVS
OpenORB: Der Ersatz für den JDK-ORB
MySQL und phpMyAdmin
Rational Rose
Entwicklung und Dokumentation
Tests und Integration

Wahl von CORBA und Java

- Hohe Funktionalität
  - Aber stellenweise auch entsprechend komplex
- Breite Werkzeugunterstützung
- Betriebssystem-Unabhängigkeit
  - Und mit CORBA auch andere Programmiersprachen möglich
- Aber nicht nur fachliche Gründe...
  - In der Praxis gegenwärtig sehr verbreitet
  - Einsatz von Java/CORBA in Bachelorprojekt
  - Bereits gesammelte Erfahrungen mit CORBA
- Zufriedenheit mit gewähltem Framework

Concurrent Versions System *CVS*

- Etabliertes Versionierungssystem
- Starke Vereinfachung der parallelen Entwicklung
  - Besonders bei mehreren Standorten
  - Gute Erfahrungen mit Concurrency
  - Fordert aber Disziplin
- Server: cvsnt unter Windows XP mit DSL
  - Windows-Version mittlerweile stabil und schnell
- Clients:TortoiseCVS, WinCVS, Sun Forte 4
- Bessere Ergebnisse und einfachere Einrichtung als Konkurrenzprodukte

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Concurrent Versions System *CVS* contd.

Komponenten-Programmierung und Middleware

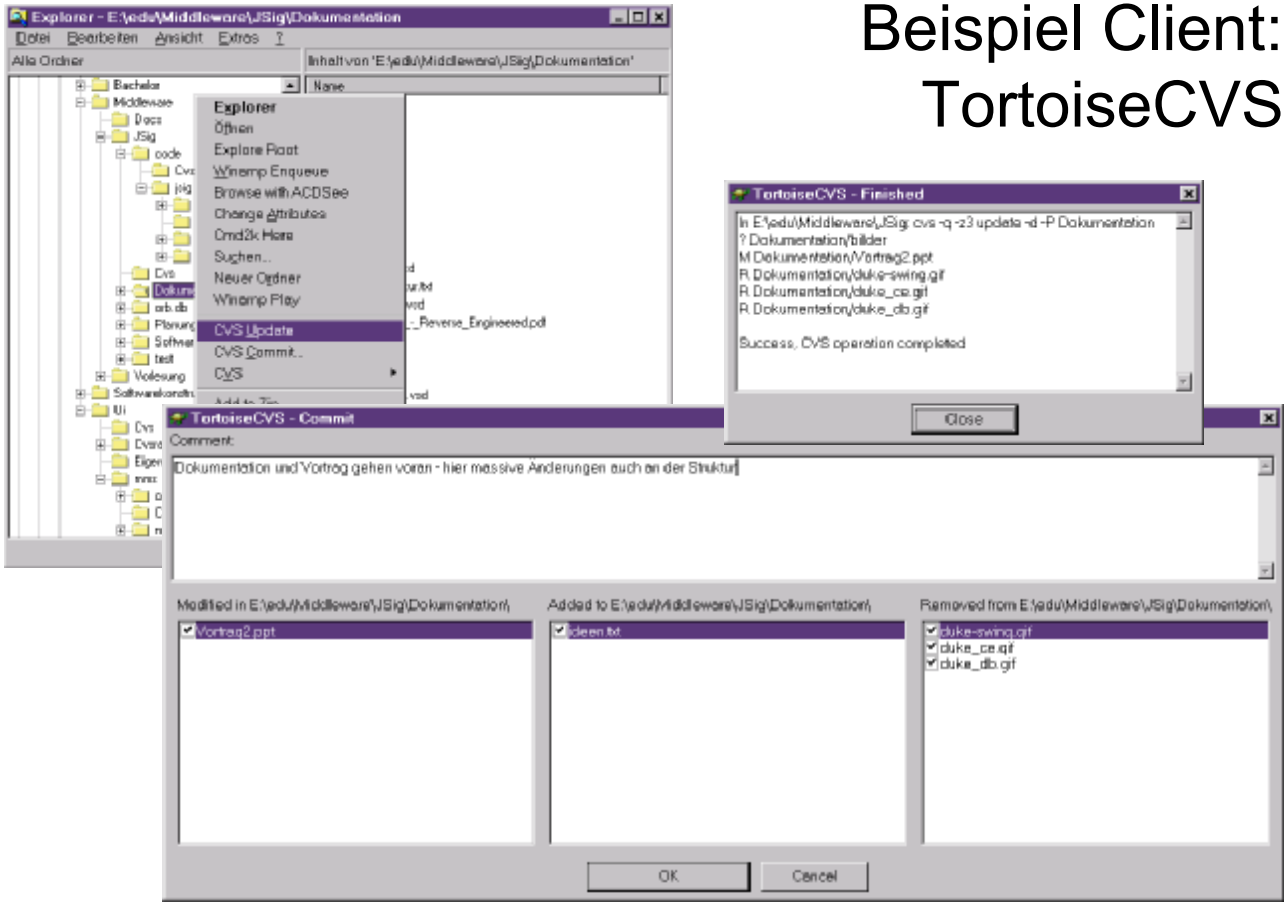
Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

Beispiel Client: TortoiseCVS



OpenORB: Der Ersatz für den ORB aus dem JDK

- Der mit Suns JDK 1.4 gelieferte ORB ist zuverlässig, aber einfach gehalten
  - Bietet kaum Zusatzfunktionalität
- Ersatz musste her: **OpenORB 1.2.1**
  - Java-Implementation von OMG-CORBA-Spezifikation 2.4.2
  - Momentan neun CORBAServices (Trading, Transaction etc.)
  - Weitere Werkzeuge verfügbar
  - Zuverlässiger und relativ einfach zu konfigurierender Request Broker

OpenORB: Vorteile

- Bietet die zusätzlichen, für JSig benötigten CORBAServices
- Kostenlos, OpenSource
- Beispiele verfügbar
- Vorteile gegenüber evaluiertem JacORB von Gerald Brose et al (FU Berlin):
  - Mit OpenORB trotz hoher Leistung und Funktionalität einfachere Anwendung und Konfiguration
  - OpenORB ist Package-orientiert (Module)

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

MySQL und phpMyAdmin

- **MySQL**: Schnelles, zuverlässiges DBMS
  - Weite Verbreitung, viele Tools
  - In JSig keine Transaktionen notwendig
- Anbindung an Java mit MySQL-Treiber **MM.MySQL** über **JDBC**-Schnittstelle
  - OpenSource
  - Nicht vollständig JDBC-konform
- **phpMyAdmin**: Verwaltung von MySQL-System und Datenbanken
  - Sammlung von PHP-Skripten
  - Webserver notwendig
  - Benutzerverwaltung integriert

Rational Rose (1)

- Rapid Development Tool
- Modellierung der Klassen(-hierarchie) und Packages mit UML
- Positive Erfahrungen mit der *Anwendung*
  - Forward Engineering der Javaklassen
  - Reverse Engineering der IDL-Definition
  - Integration und Code-Erzeugung durch Rose
- Nach Einführung schnell „in Produktion“
  - Aber nur Nutzung eines kleinen Teils der Funktionalität von Rose

Rational Rose (2)

Komponenten-Programmierung und Middleware

Sommer 2002,
Prof. Andreas Polze

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

Rational Rose - JSigModellierung.mdl - [Class Diagram: Logical View / jsig]

File Edit View Format Browse Report Query Tools Add-Ins Window Help

getNumberOfQuotes
getTextFromDB
JSigDispenserImpl (JSig
JSigGUI (JSigGUI)
JSigGUIAddComponent (
JSigGUILogin (JSigGUIL
JSigInserterImpl (JSigIns
JSigNumOfSigsImpl (JSi
JSigServerHolder (JSigS
JSigServerImpl (JSigSer
JSigUtil (JSigUtil)
QotdTimerTask (QotdTir
Associations
Component View
jsig
client
exceptions
server
JSigData

Der großartige JSig-Client auf Benutzerseite. Redet vor allem mit den Interfaces.

JSigDispenser
JSigBlockerImpl
JSigInserterImpl
JSigBlocker
JSigInserter
JSigClient
JSigGUIAddComponent
JSigGUI

16:10:54| [[Update Model Properties]]
16:10:54| [[Update Model Properties]]
16:10:54| [[Update Model Properties]]
16:10:55| [[Update Model Properties]]
16:10:55| [[Update Model Properties]]

Log

For Help, press F1 Default Language: Analysis NUM

Entwicklung und Dokumentation

- IDEs:
  - Sun Forte 4 (Sun ONE Studio)
  - Eclipse SDK
- In beide IDEs Sun JDK 1.4 eingebunden
- IDL-Compiler
  - Aus Sun JDK 1.4 (idlj)
  - OpenORB IDL-Compiler
- Codedokumentation
  - JavaDoc
  - Meist direkt bei der Modellierung mit Rational Rose eingegeben
  - Gute Unterstützung durch verwendete IDEs

Tests und Integration

- Code-Reviews und Inspektionen
  - Unregelmäßig, kein festgelegtes Verfahren
- Rudimentärer Einsatz von JUnit
- Manuelle Tests: Skripte, Debug-Ausgaben
- Datenbank: vorab getestete SQL-Statements
- Integration von beiden Seiten: Top-Down und Bottom-Up
  - Testtreiber und Testdummys
  - Integrationstests sukzessive

Erfolge, Probleme, Erkenntnisse

Lauffähiges System entspr. der Zielsetzung
Nutzung von bereits erworbenen Kenntnissen
Wiederverwendung von Code
Probleme während der Entwicklung
Erkenntnisse für das Projektteam

Lauffähiges System entspr. der Zielsetzung

- Erreicht wurden:
  - ü Ausgabe von zufälligen Signaturen
  - ü Ausgabe von Quote Of The Day
  - ü Persistentes Blockieren einer Signatur
  - ü Hinzufügen einer Signatur
  - ü Ausgabe der Signaturanzahl
- Nicht erreicht wurden:
  - ➙ Suchen nach einer Signatur

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Nutzung von bereits erworbenen Kenntnissen

- Modellierungs- und Designtechniken
- Datenbanken
  - Datenbankentwurf
  - Datenbankadministration durch Webservice / Apache
- Softwarebauelemente I und II: Implementationsunabhängigkeit durch Interfaces
- Bedienwissen für die wichtigen Werkzeuge und Systeme
- Alternative Realisierung durch Web Service

Wiederverwendung von Code

- Eines der Ziele der OO
- Wichtiger Grund für komponentenbasierte Systeme
  - Re-Use von Code in binärer Form
- JSig:
  - Verwendung von Util- und Exception-Klassen
  - Prinzip des Dispensers aus ähnlichem Projekt (Übernahme großer Codestücke)
  - Keine Wiederverwendung von bereits bestehenden Komponenten

Probleme während der Entwicklung

- Prozessmodell wurde teilweise nicht eingehalten
- Wechsel des ORBs
- Umstellung der GUI durch dynamische Generierung
- Personenbasierte Datenhaltung führte zum Dispenser
- Drohender Verlust der Übersicht und der gemeinsamen Basis führte zur Verwendung von Rational Rose
  - Ergebnis u.a.: Getrenntes Programmieren von Klassen ermöglicht, die am Ende nahezu problemlos integriert werden konnten.

Erkenntnisse für das Projektteam

- Bedienwissen zu weiteren Werkzeugen
- Kardinalitäten im UML-Klassendiagramm fördern das Systemverständnis
- Sichtbarkeit von Methoden und Attributen muss in Produktionssystemen sorgfältig modelliert werden
- Ein Versionierungssystem vereinfacht das Dateimanagement erheblich

Ausblick

Sicherheit
CORBASec – Security Conformance Levels
CORBASec – Role based Security
Nachteile unserer CORBA-Wahl
Realisierung mit anderen Frameworks
Mögliche Realisierung ähnlicher Funktionalität
Deployment und Installation
Andere Plattformen und Sprachen mit JSig
Mögliche Erweiterungen und andere Träumereien

Sicherheit

- Verschlüsselung der Passwörter in der Datenbank wäre durch SQL-Funktionen einfach möglich (`PASSWORD(text)`)
  - Aber fragwürdig, da die Übertragung zum DBMS unverschlüsselt ist
  - Nur gewisser Schutz bei unberechtigtem Zugriff auf Datenbank
- Verschlüsselung des Passworts im Client denkbar
  - gleicher Mechanismus zum Erstellen notwendig
- OpenORB bietet SSL-Extension
  - entspricht der **SSLIOP**-Spezifikation der OMG

Sicherheit: CORBASec

- Teil der CORBAServices
- CORBA Security Specification
  - Identifikation und Authentifizierung
  - Authorisierung und Zugriffskontrolle
  - Sicherheitsaufzeichnungen
  - Kommunikationssicherheit
  - Nichtzurückweisbarkeit (Non-Repudation) einer getätigten Aktion
  - Administration von sicherheitsrelevanten Informationen

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

CORBASec – Security Conformance Levels

- Level 1:
  - Security-Unaware Applications
  - Trusted Computing Base
  - Entry Level Dienste
- Level 2:
  - Security-Aware Applications
  - Security Interfaces

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

CORBASec – Role based Security

- Domains:
  - Gruppierte Objekte
  - Sicherheitsregeln (Policies)
  - Hierarchische Struktur
- Role:
  - Rechte
  - Pflichten
  - Funktionen
  - Interaktionen

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

CORBASec – Role based Security

- Access Control Policy
  - Erlaubte Operationen (positive Policies)
  - Verbotene Operationen (negative Policies)
  - Beispiel:
    Objekte einer Domain dürfen Operationen op1, op2 auf Objekt Typ1 und Operationen op1, op3 auf Objekt Typ2 ausführen
  - Zeitbeschränkungen
  - Verschlüsselung symmetrisch oder asymmetrisch

17.06.2002

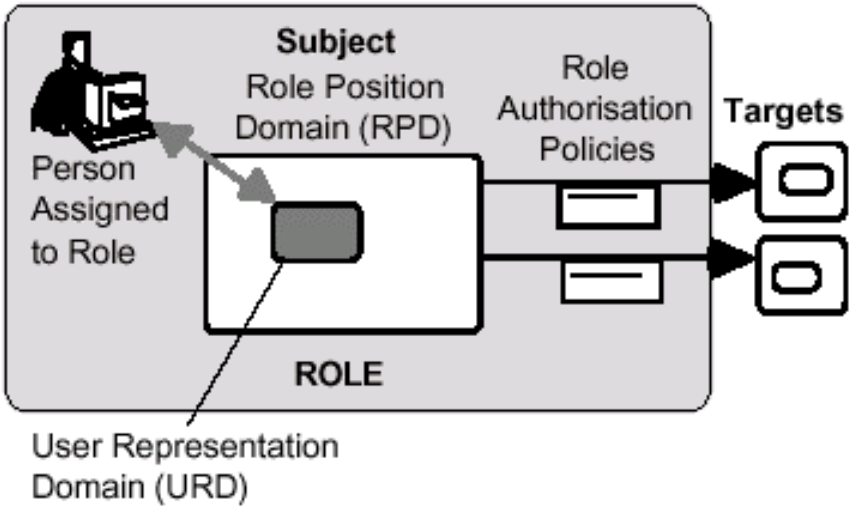
Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

CORBASec – Role based Security

Komponenten-Programmierung und Middleware

Sommer 2002,
Prof. Andreas Polze

- Roles
  - Role Position Domain (RPD) + Access Control und Obligation Policies
  - RPD = Status, Position
  - Beziehungen (Interaktionsprotokolle)



17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Projekt JSig

CORBASec – Role based Security

- Simultaneous Role Sessions
  - Login: Objekt-Adapter zwischen Benutzer und System
  - Rollen für Benutzer getrennt
  - Security System **à** unterscheidet welche Rolle für welchen Benutzer
  - URD außerhalb der RPD

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Nachteile unserer CORBA-Wahl

- Implementierung relativ aufwendig
  - Großer Overhead
  - Hohe Komplexität
- CORBA-Dokumentation
  - Marketing-Dokumente zu ungenau
  - Spezifikation für unsere Bedürfnisse zu umfangreich
- Zu OpenORB nur mangelhafte Dokumentation verfügbar

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Realisierung mit anderen Frameworks

- JSig mit COM, DCOM, ActiveX
  - Stark an Windows gebunden
  - Damit herstellerabhängig **à** keine Portabilität
  - Nutzung der Win32 API
  - Nutzung des Windows Thread Model

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Mögliche Realisierung ähnlicher Funktionalität

- Realisierung als Web Service mit verschiedenen aktuellen Protokollen
  - Schnittstellenbeschreibung via WSDL möglich
  - Aufrufe können über SOAP oder XML-RPC ablaufen
  - UDDI (global) bzw. WSIL (lokal) kann als Repository verwendet werden
  - Einfachere Implementierung mit Tools (Web Service Toolkit, IBM)

Deployment und Installation

- Installation des jetzigen JSig-Systems auf anderen Maschinen mit hohem Aufwand verbunden
  - OpenORB-Installation und Konfiguration
  - Einrichtung des MySQL-DBMS, der JSig-Datenbank und der Treiber
  - Pfade und Startskripte
- Vereinfachung des Setups notwendig
  - Clientseitig Jar-Archiv und Konfiguration in Text- oder XML-Datei
  - Serverseitig SQL-Skripte für DB-Einrichtung (bereits bei Entwicklung und Test verwendet)

Andere Plattformen und Sprachen mit JSig

- Andere Betriebssysteme möglich
  - Test des aktuellen Systems auf Unix- und Unix-verwandten Betriebssystemen
- Client-Implementierung in anderer Programmiersprache
  - Weiterer C++-Client, der mit bestehendem Java-Server kommuniziert
- Ohne Änderungen am Code können JSig-Server und DBMS auf getrennten Rechnern betrieben werden
  - Bestehende Datenabstraktion bietet überladenen Konstruktor mit Datenbank-Host als Argument

Mögliche Erweiterungen und andere Träumereien

- Weitere Komponenten
  - Dynamische Administrations-Komponente
    - Hinzufügen von Nutzern und Quotes
    - Löschen von Quotes und Signaturen aus DB
  - Komponente zum Suchen von Signaturen
  - Signature-Checker
    - Assistent zur Überprüfung von Zeilenlänge etc.
- Anbindung anderer Datenbanken
  - Mit SQL(-92)-Datenbanken minimaler Aufwand zur Umstellung: Tausch des Treibers
  - Aber schlechte Erfahrungen mit der einfachsten Möglichkeit zur Realisierung: JDBC-ODBC-Brücke

Fragen

Anhang

Glossar
Weiterführende Literatur
Quellen
Quellen – CORBA
Links – verwendete Software
Weitere Links

Glossar

- DBMS - DataBase Management System
- SSL - Secure Socket Layer
- SQL - Structured Query Language, Sprache für Datenbankzugriff und Modifikation
- ORB - Object Request Broker
- CORBA - Common Object Request Broker
- JDBC - inoffiziell: Java DataBase Connection, Suns Standard zum Zugriff auf Datenbanksysteme
- ODBC - Open DataBase Connectivity, Standard zum Zugriff auf SQL-

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Weiterführende Literatur

- CVS: http://www.cvshome.org/
- CORBA: OMG
- Java 1.4, speziell JDBC und Datenbanktreiber (Beispiel zur Programmierung eines eigenen Treibers): Java ist eine Insel

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Quellen

- Folien zur Vorlesung
- David Flanagan, Java in a Nutshell, O‘Reilly & Associates, 2002
- Christian Ullenboom, Java ist eine Insel, Gallileo Press, 2001
- Corba and Java Programming
- Beispiel M. Dirska
- Java API 1.4 Docs
- OpenORB 1.2.1-Examples

17.06.2002

Natalia Peters
Antje Rogotzki
Maximilian von Spiegel
Simon Staiger

Quellen – CORBA (1)

- G. Brose, A. Vogel and K. Duddy: Java Programming with CORBA, 3rd Edition, OMG Press
- www.omg.org (Bildquelle)
- http://miss.wu-wien.ac.at/~mitloehn/se/w98/projekte/j9250905/Corba%20Einf%81hrung.htm

Quellen – CORBA (2)

- http://www.informatik.fh-muenchen.de/~schieder/seminar-java-ss98/corba/ausarbeitung/corba.html
- http://chili.wifo.uni-mannheim.de:54581/~iiop/corba.html
- http://www.hta-be.bfh.ch/~fischli/kurse/oot/corba/skript.pdf
- Yialelis, Lupu, Sloman: Role-Based Security for Distributed Object Systems, 1996
- Stand 30. Juni 2002

Links – verwendete Software

- Newsgroup zum Testen der Signaturen: news:de.alt.fan.aldi
- Community OpenORB: http://openorb.sourceforge.net/
- MySQL AB: http://www.mysql.com/
- MM.MySQL-Treiber: http://mmmysql.sourceforge.net/
- phpMyAdmin: http://phpwizard.net/projects/phpMyAdmin

Weitere Links

- Eclipse IDE: http://www.eclipse.org/
- Sun ONE Studio 4: http://wwws.sun.com/software/sundev/jde/buy/index.html
- cvsnt: http://www.cvsnt.org/
- TortoiseCVS: http://www.tortoisecvs.org/
- PHP: http://www.php3.de/