

Wartbare, Plattformunabhängige Grafische Benutzeroberflächen Schnell entwickeln

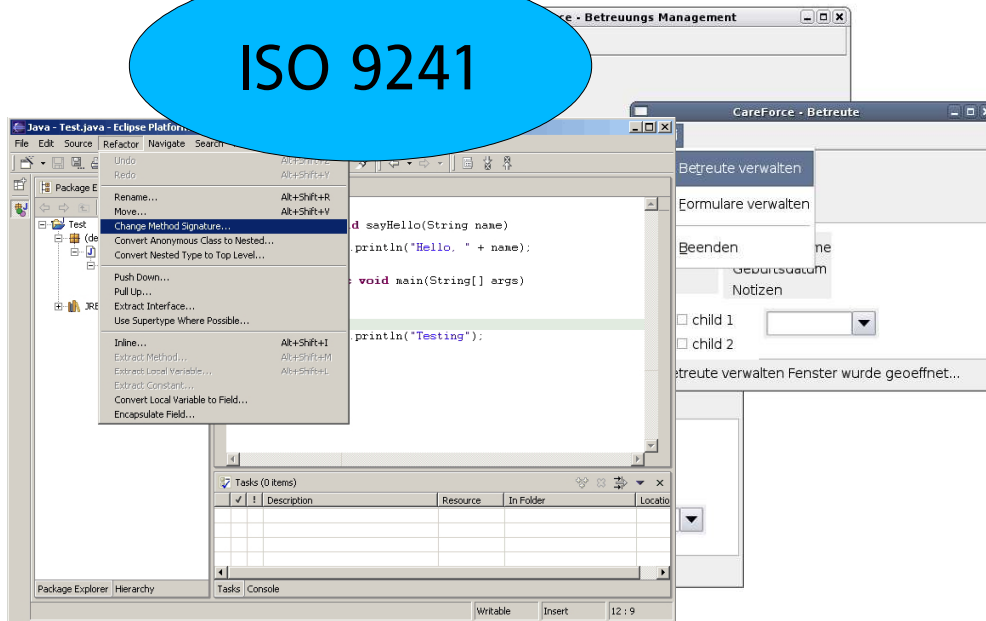
Abschlusspräsentation zur Bachelorarbeit
von Axel Pospischil
bei Prof. Dr. B. Humm, Prof. Dr. S. Schneider
Hochschule Darmstadt

Grafische Benutzeroberflächen - Komplexität reduzieren - vorhandenes Wissen nutzen.

Abschlusspräsentation zur Bachelorarbeit
von Axel Pospischil
bei Prof. Dr. B. Humm, Prof. Dr. S. Schneider
Hochschule Darmstadt

GUI's - Kontext im Kleinen

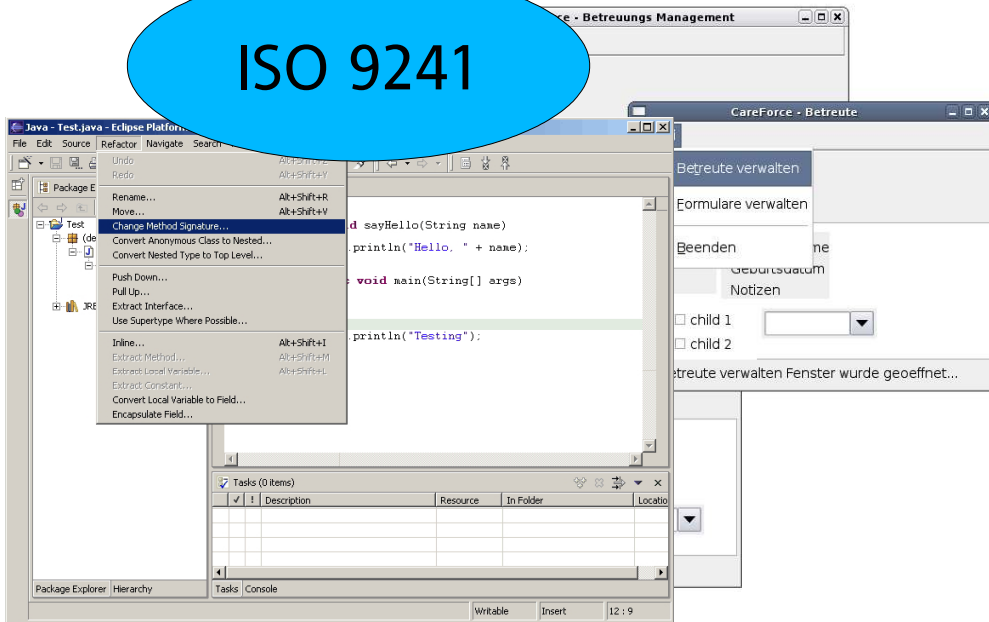
ISO 9241



- Benutzerschnittstelle
 - Benutzerfreundlichkeit, Barrierenfreiheit
 - Gebrauchsnutzen, Ergonomie

GUI's - Kontext im Kleinen

ISO 9241



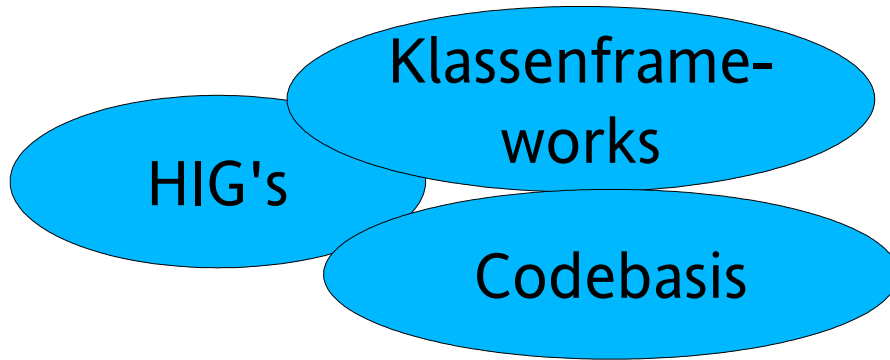
• Benutzerschnittstelle

- Benutzerfreundlichkeit, Barrierenfreiheit
- Gebrauchsnutzen, Ergonomie

• Softwareprojekte

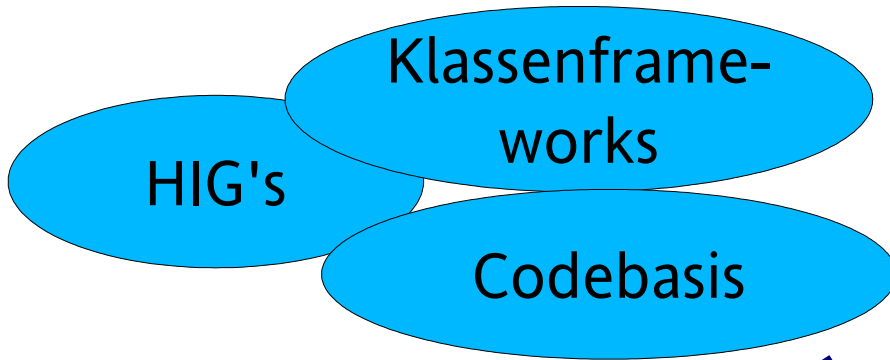
- 10-25% in Entwicklungsphase - 75-90% in Wartung
- Entwicklung / Wartung
- Robust, wartbar, schnell entwickelt, dabei plattformunabhängig
- Teamarbeit, Dokumentation, Skillserwerb
- Hersteller(un)abhängigkeit, Zukunftssicherheit
- Plattformunabhängig

GUI's - Kontext im Kleinen



- Konzeptionelle Umsetzung
 - Komplexitätstreiber identifizieren
 - Anforderungsmanagement
 - Expertenwissen erlernen und dokumentieren

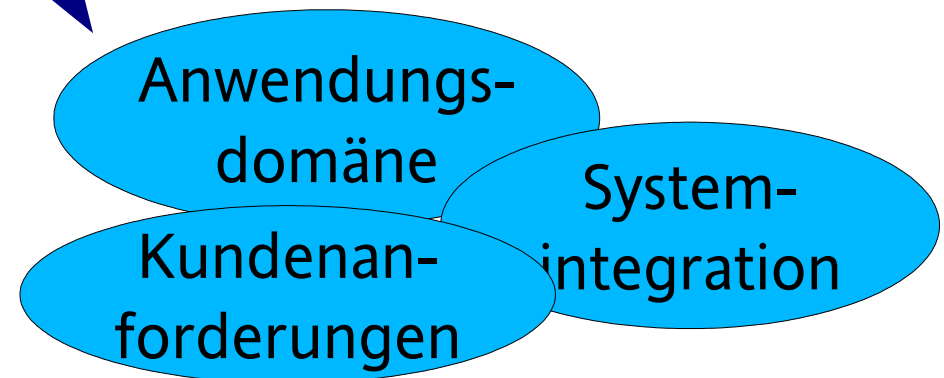
Kontext im Grossen



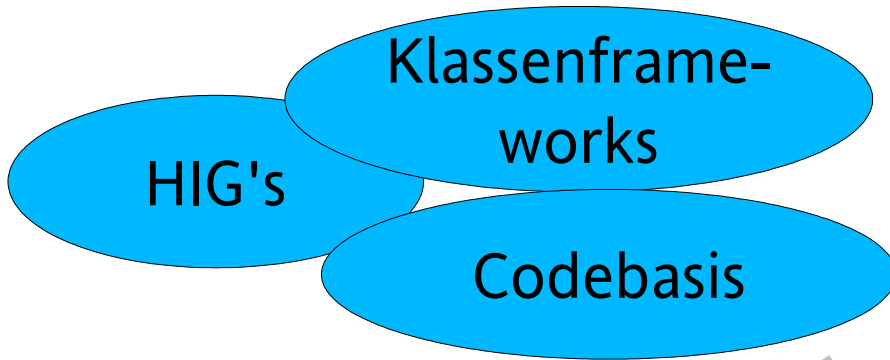
● Konzeptionelle Umsetzung

- Komplexitätstreiber identifizieren
- Anforderungsmanagement
- Expertenwissen erlernen und dokumentieren

- Einordnung, Integration, Design der
 - Anwendungsdomäne
 - Anwendungslandschaft
 - Vorgegebene Prozesse
 - Transaktionen, Persistenz, Verteilung



Kontext im Grossen

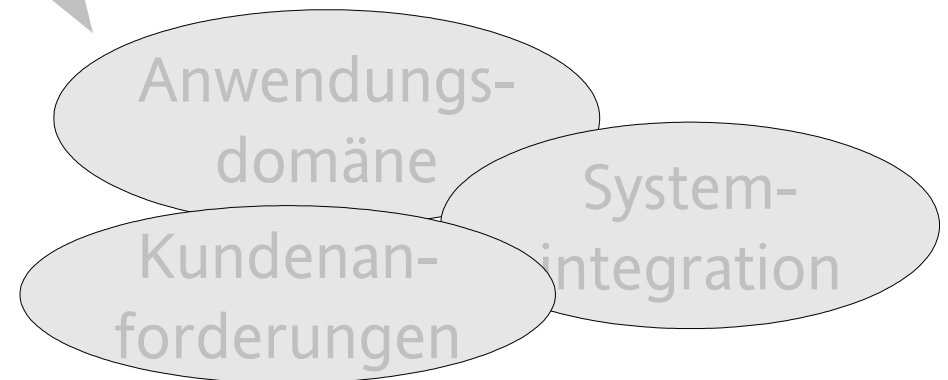


● Konzeptionelle Umsetzung

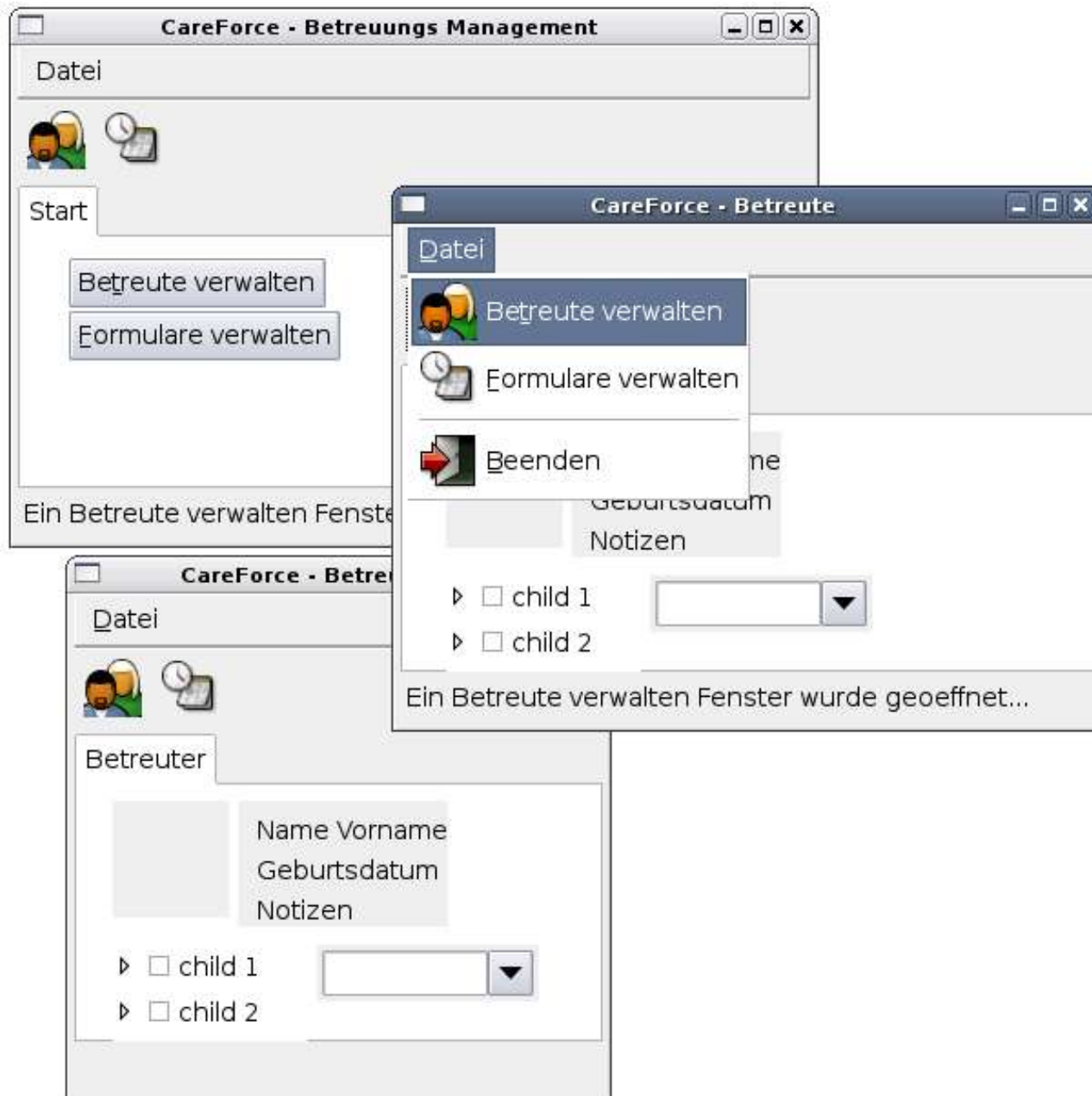
- Komplexitätstreiber identifizieren
- Anforderungsmanagement
- Expertenwissen erlernen und dokumentieren

- Einordnung, Integration in und Design der
 - Anwendungsdomäne
 - Anwendungslandschaft
 - Vorgegebene Prozesse
 - Transaktionen, Persistenz, Verteilung

Ausblick



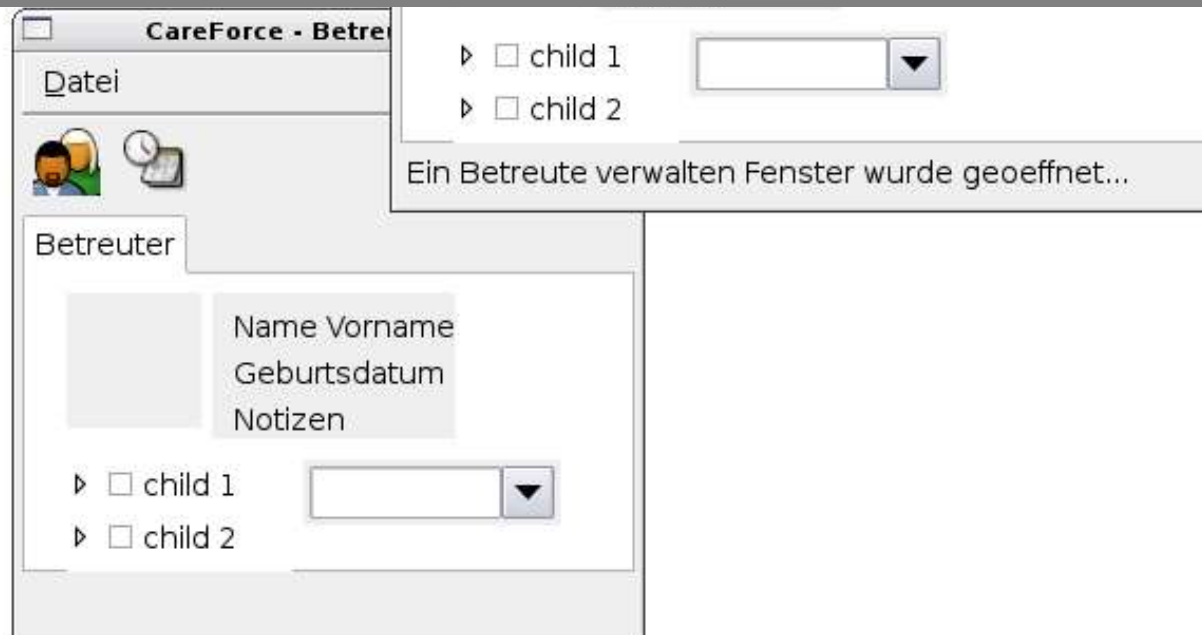
Komplexität der Anforderungen



Komplexität der Anforderungen



Wer sind die Experten für ein konkretes GUI Projekt ?



Komplexität der Anforderungen

Wer sind die Experten für ein konkretes GUI Projekt ?

- Allgemeine Anforderungen
 - Visuelle Darstellung
 - Funktion
 - Ergonomie
 - Anwenderfreundlichkeit
 - Gebrauchsnutzen
 - Plattformunabhängigkeit

Komplexität der Anforderungen

Wer sind die Experten für ein konkretes GUI Projekt ?

- Allgemeine Anforderungen
 - Visuelle Darstellung
 - Funktion
 - Ergonomie
 - Anwenderfreundlichkeit
 - Gebrauchsnutzen
 - Plattformunabhängigkeit
- Sicht des Softwarearchitekten
 - Wartungsfreundlichkeit
 - Erweiterbarkeit
 - Trennung von GUI-Code und Fachlogik
 - Werkzeuge

GUI's - Expertenwissen – Konzepte

- Wo ist das Erfahrungswissen gespeichert?

...

- Wie kann man dieses Erfahrungswissen im Team erlernen und verfügbar machen?

...

- Welche Voraussetzungen sind dazu nötig?

...

- Welche Hilfsmittel stehen zur Verfügung?

...

GUI's - Expertenwissen – Konzepte

- **Wo ist das Erfahrungswissen gespeichert?**

...

- Wie kann man dieses Erfahrungswissen im Team erlernen und verfügbar machen?

...

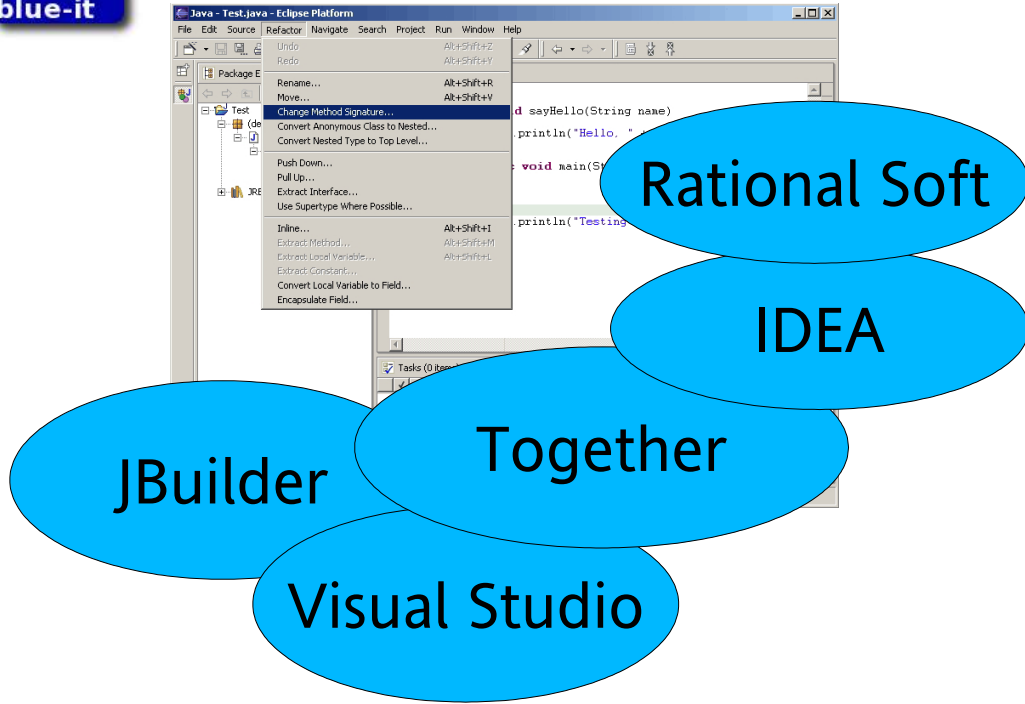
- **Welche Voraussetzungen sind dazu nötig?**

...

- Wie kann man dieses Erfahrungswissen im Team erlernen und verfügbar machen?

...

Antworten ?



- All in One Entwicklungsumgebungen
- Programmierumgebung

Antworten ?

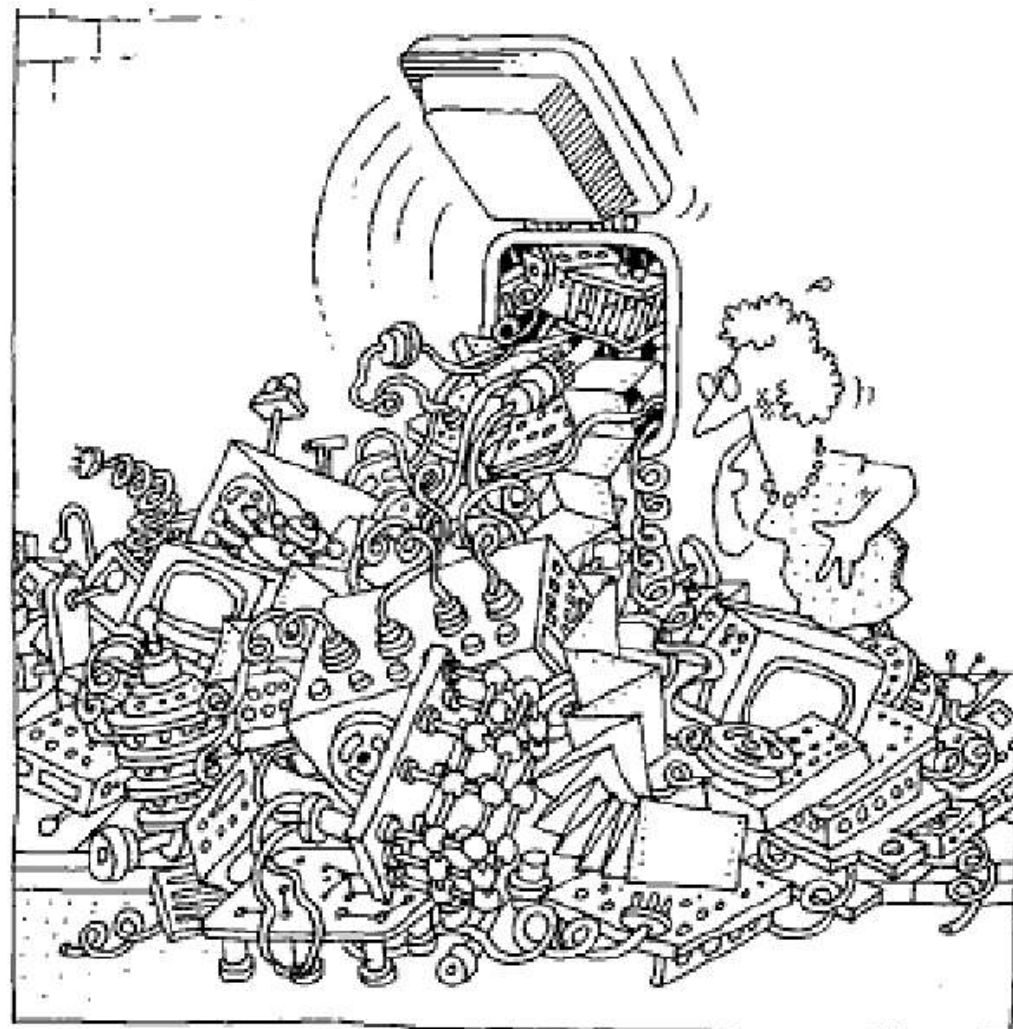
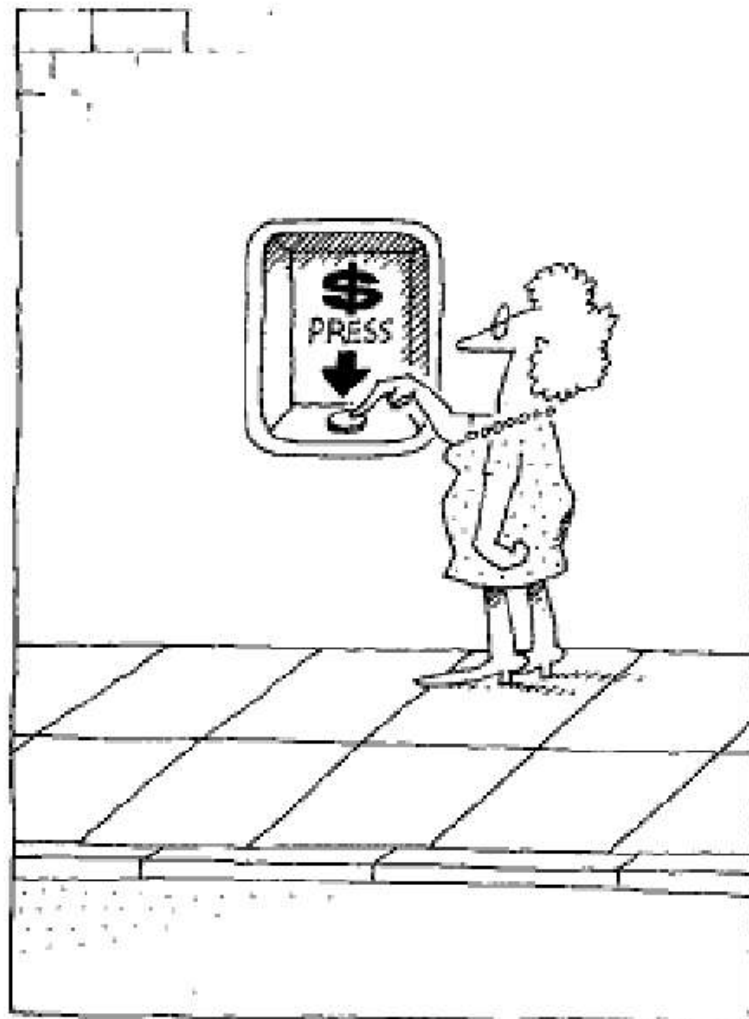
IDEA Rational Soft

Together

Visual

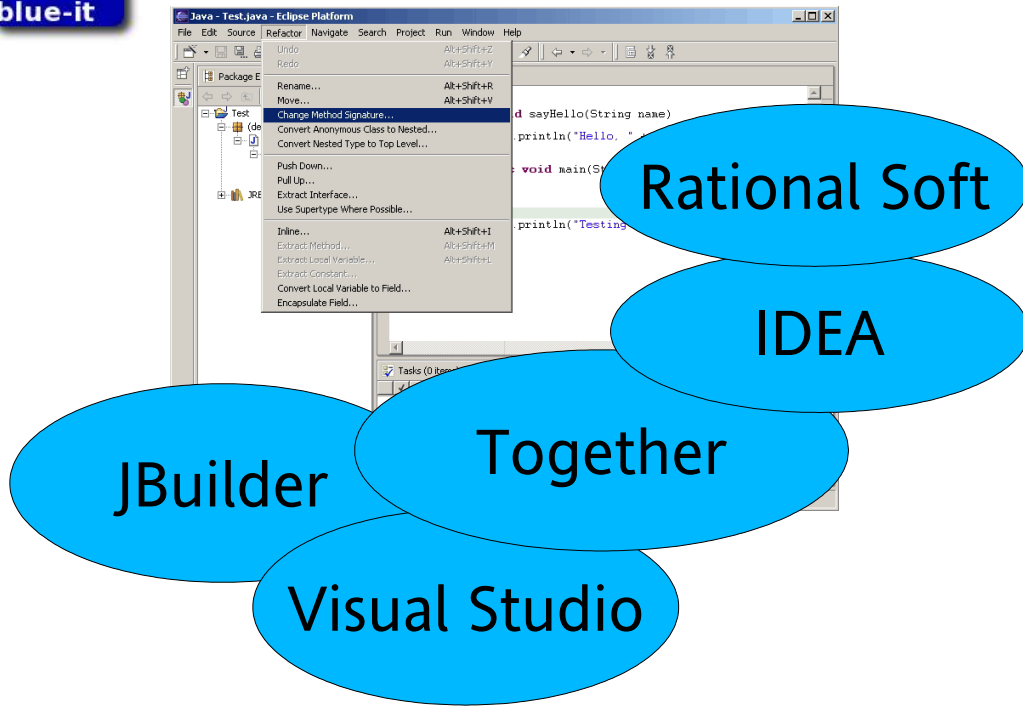
JBu

nach UN(!)-sachgemäßem Gebrauch.



Quelle: Booch, Objekt-Oriented Analysis and Design, 1994, 2000

Antworten ?



- All in One Entwicklungsumgebungen
- Programmierumgebung

- Unzureichende Trennung von Designcode und GUI Code (Teamarbeit)
- Stabilitätsprobleme der GUI nach Codeveränderungen (Robustheit)
- Codegenerierung (Bindung an Werkzeug)
- Kundenanforderungen (Wartung/Erweiterung)

Lösung Abstraktion

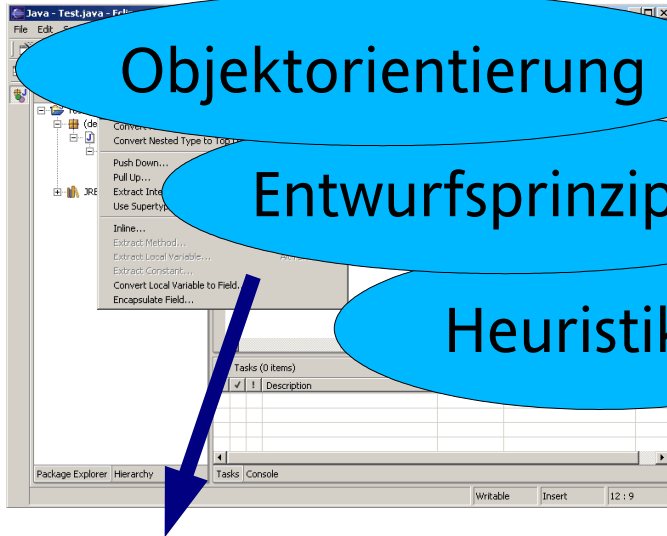
Objektorientierung

Entwurfsprinzipien

Heuristiken

Patterns

Muster-
kataloge



Lösung Abstraktion

Objektorientierung

Entwurfsprinzipien

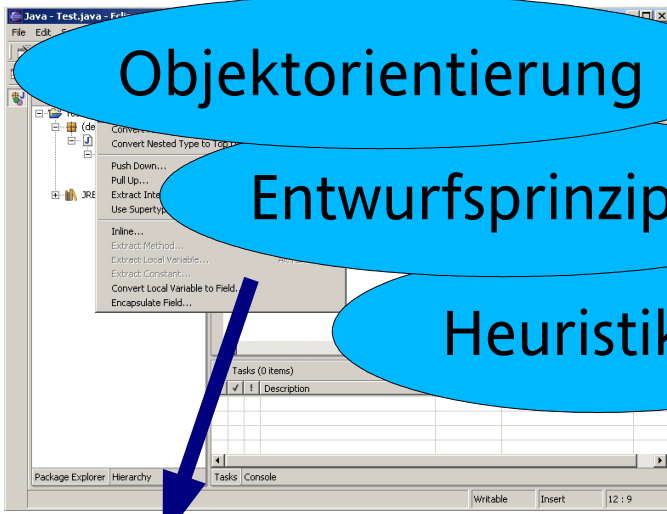
Heuristiken

Patterns

Muster-
kataloge

- Heuristiken
 - für Klassen, Interfaces, Pakete, Vererbungsbeziehungen

Lösung Abstraktion



Objektorientierung

Entwurfsprinzipien

Heuristiken

Patterns

Muster-
kataloge

- Heuristiken

- für Klassen, Interfaces, Pakete, Vererbungsbeziehungen

- Entwurfsprinzipien

- Identifiziere Aspekte, die sich ändern
- Programmiere auf eine Schnittstelle, nicht auf eine Implementierung
- Ziehe Komposition der Vererbung vor
- Führe den Entwurf auf die Anforderungen zurück [...]

GUI's: Expertenwissen – Konzepte

- Wo ist das Erfahrungswissen gespeichert?
 - Entwurfsmuster, Musterkataloge als Bausteine
- Wie kann man dieses Erfahrungswissen im Team erlernen und verfügbar machen?
...
- Welche Voraussetzungen sind dazu nötig?
 - Objektorientierte Vorgehensweise
- Welche Hilfsmittel stehen zur Verfügung?
...

GUI's: Expertenwissen – Konzepte

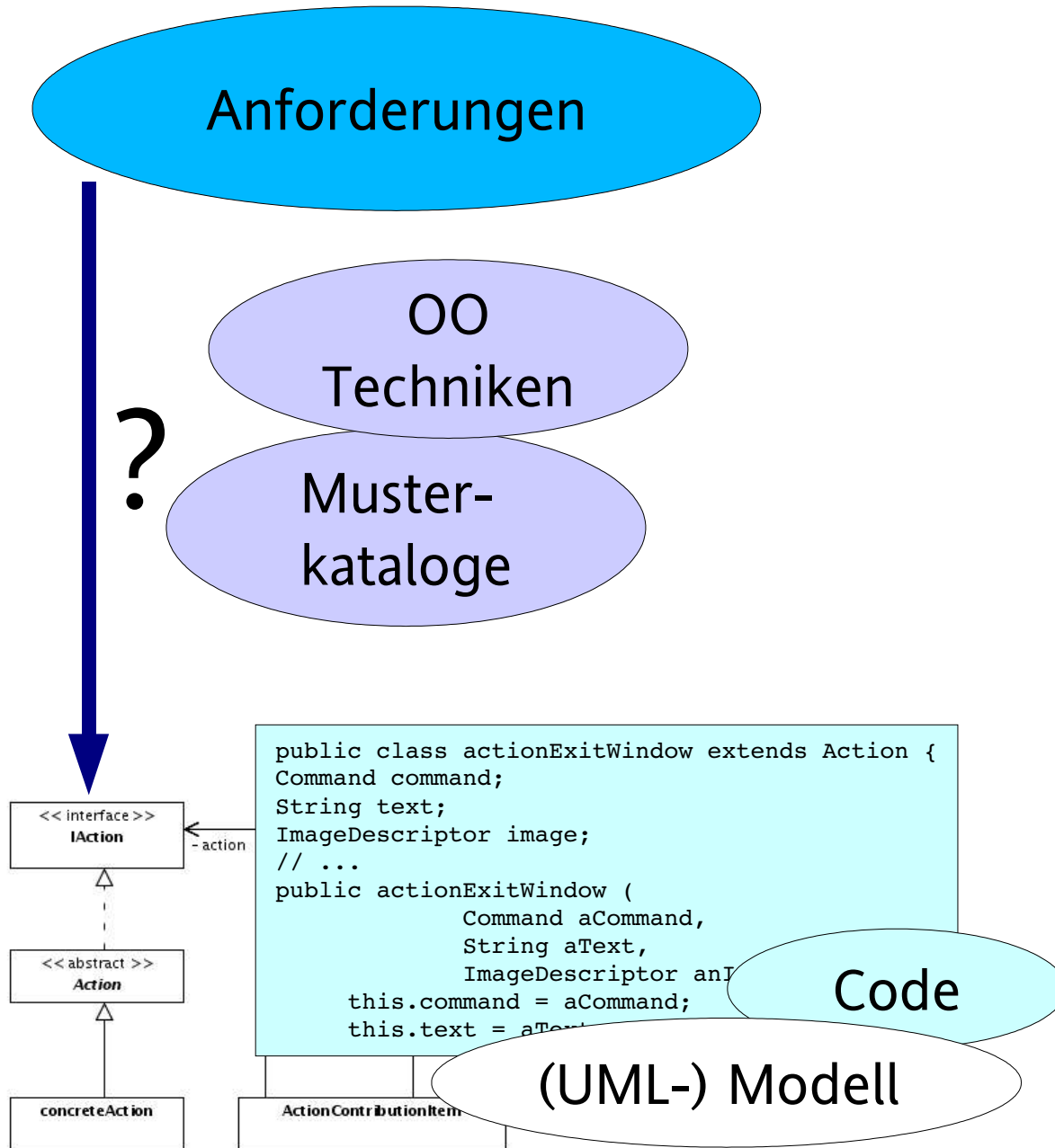
- Wo ist das Erfahrungswissen gespeichert?
 - Entwurfsmuster, Musterkataloge als Bausteine
- **Wie kann man dieses Erfahrungswissen im Team erlernen und verfügbar machen?**

...

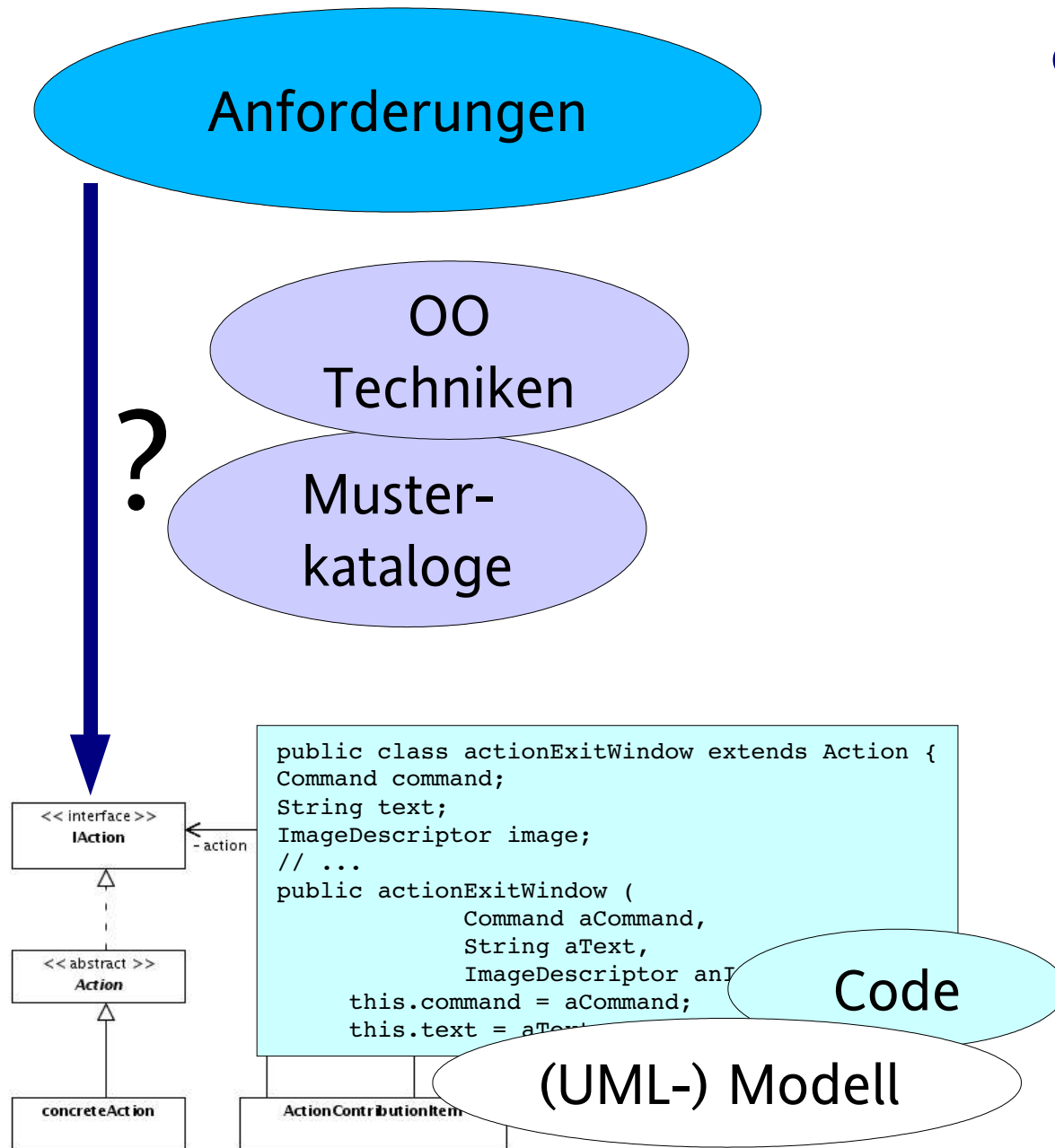
- Welche Voraussetzungen sind dazu nötig?
 - Objektorientierte Vorgehensweise
- Welche Hilfsmittel stehen zur Verfügung?

...

Entwurfsmuster identifizieren

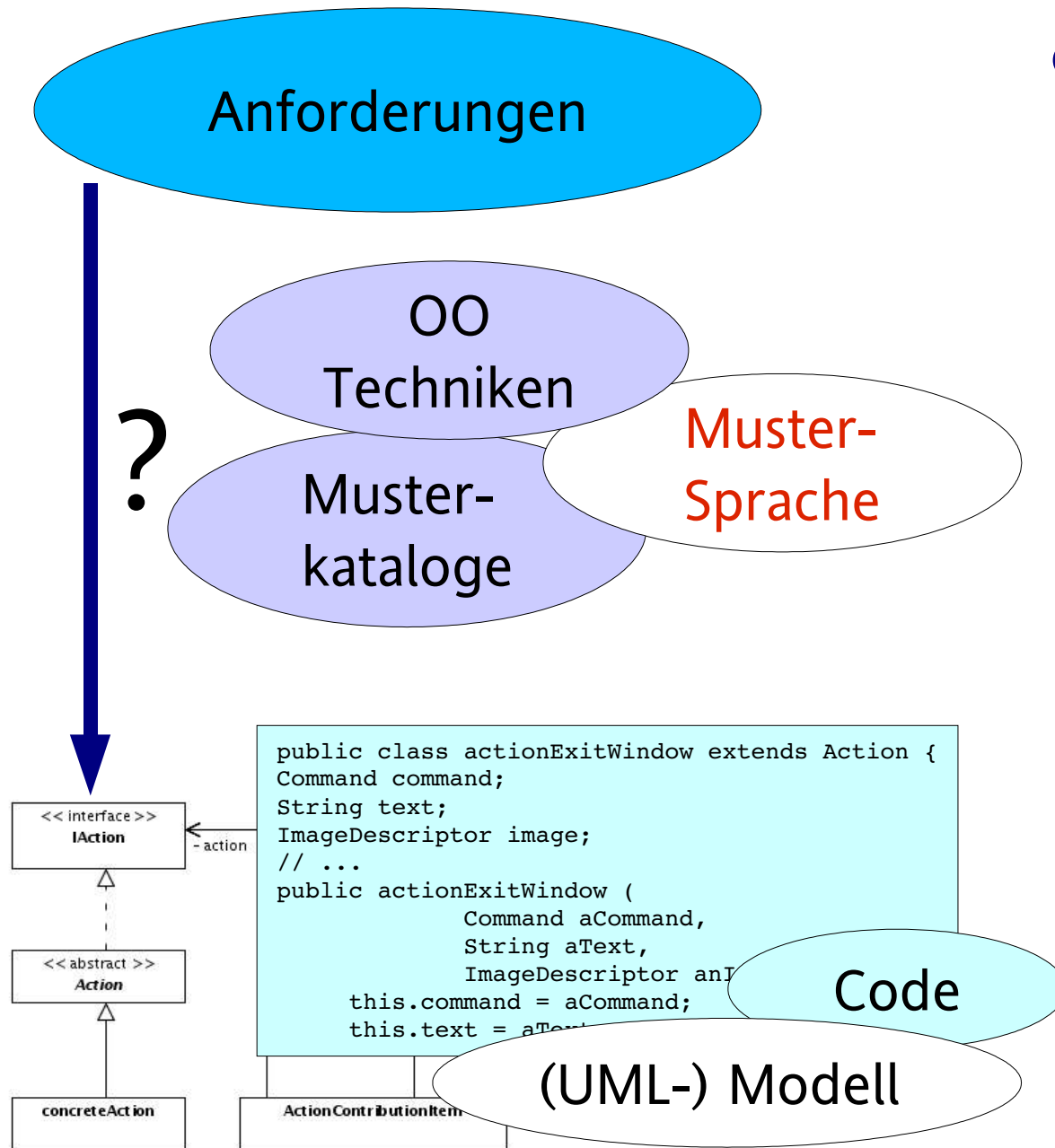


Entwurfsmuster identifizieren



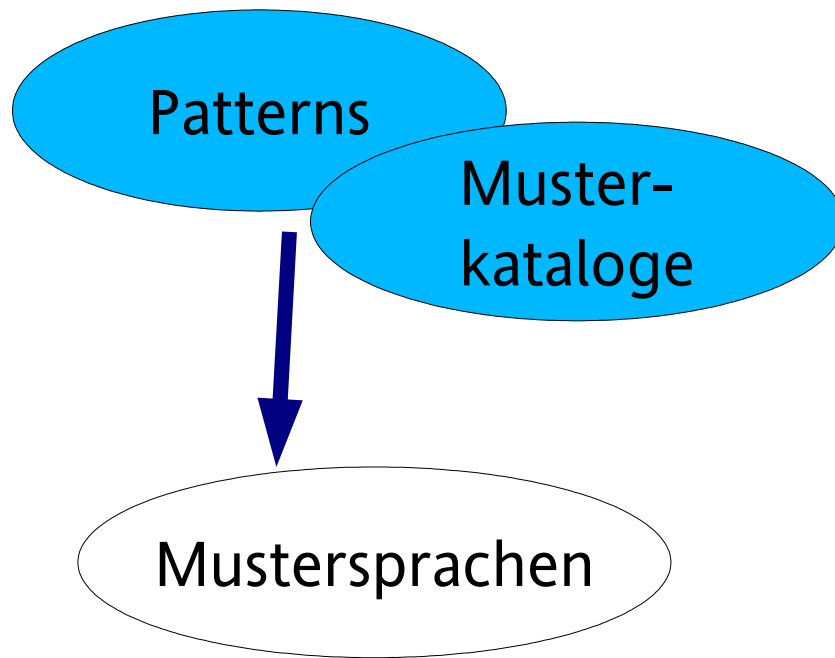
- „Mustermapping“
 - Anforderungen feststellen
 - Informationssammlung (Extraktion)
 - Adaptieren und Implementieren
 - Testen
 - Dokumentieren
 - Review

Entwurfsmuster identifizieren



- „Mustermapping“
- Anforderungen feststellen
- Informationssammlung (Extraktion)
- Adaptieren und Implementieren
- Testen
- **Dokumentieren**
- Review

Human Computer Interaktion

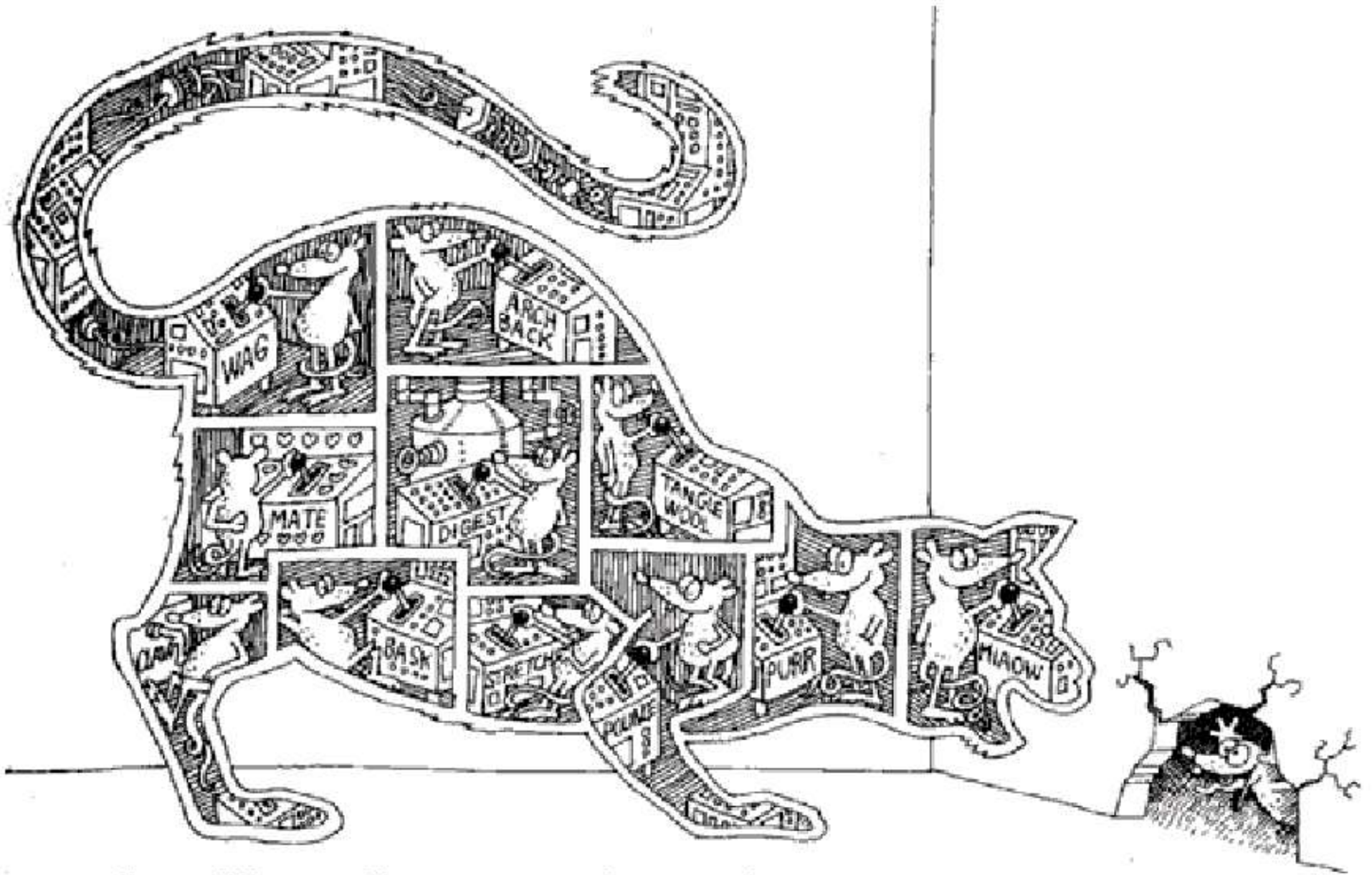


- HCI (syn. „CHI“)
- Anforderungen an Benutzerschnittstellen dokumentieren
- Beschreibung
- Analog zur Patternbeschreibung
- Beschreibung benutzerzentrierter Aspekt
 - Problem (-kontext)
 - Konflikte
 - Usability
- Keine Implementierungsdetails (Nachteil?!)
- Standard ? Zugänglich ?!

GUI's: Expertenwissen – Konzepte

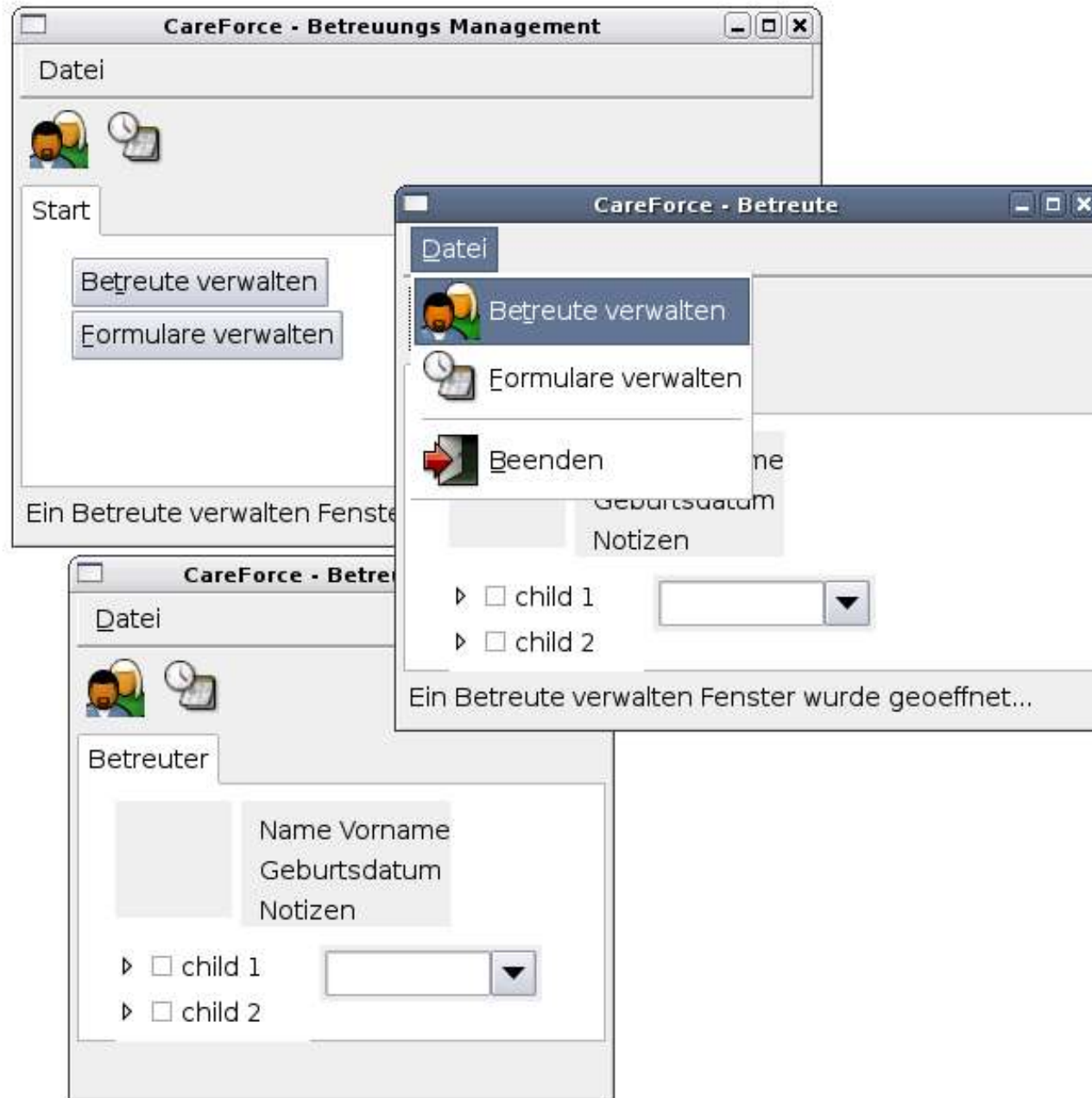
- Wo ist das Erfahrungswissen gespeichert?
 - Entwurfsmuster, Musterkataloge als Bausteine
- Wie kann man dieses Erfahrungswissen im Team erlernen und verfügbar machen?
 - Musterkataloge, Mustersprachen, ...
- Welche Voraussetzungen sind dazu nötig?
 - Objektorientierte Vorgehensweise
- Welche Hilfsmittel stehen zur Verfügung?
 - UML Werkzeug, Text, Code Werkzeug, ...

„Muster“ - lösungen finden



...allows different objects to act at the same time

„Muster“ - lösungen finden



„Muster“ - Lösungen extrahieren

Open Source
GUI Framework

```
public class Command {  
    String text;  
    ImageDescriptor image;  
    // ...  
    public actionExitWindow (  
        Command aCommand,  
        String aText,  
        ImageDescriptor anImage ) {  
        this.command = aCommand;  
        this.text = aText;  
    }  
}
```



Entwurfs-
prinzipien

Heuristiken

Patterns

Muster-
Sprache

UML Modell
& Code

„Muster“ - Lösungen extrahieren

Open Source
GUI Framework

```

public class Command {
    String text;
    ImageDescriptor image;
    // ...
    public actionExitWindow (
        Command aCommand,
        String aText,
        ImageDescriptor anImage ) {
        this.command = aCommand;
        this.text = aText;
    }
}
    
```

Entwurfs-
prinzipien

Heuristiken

Patterns

?

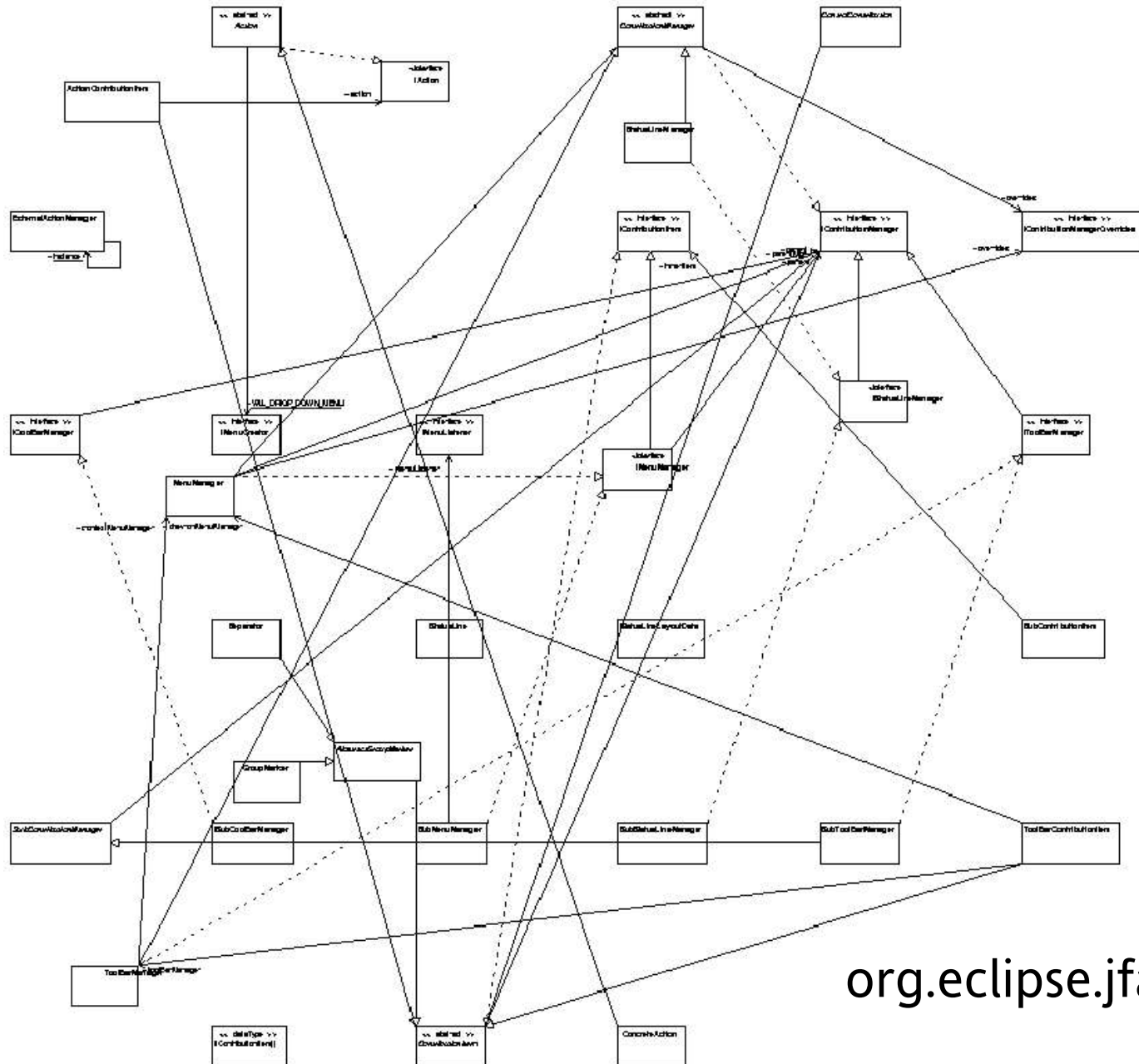
Muster-
Sprache

UML Modell
& Code

• Reverse Engineering

- Dokumentation des Frameworks erfassen
- Reverse Engineering des Paketes (UML)
- Reduzierung des Modells auf relevante Klassen
- Analyse Statik
- Analyse Dynamik
- Prozess Mustermapping (ab Adaptieren und Implementieren)

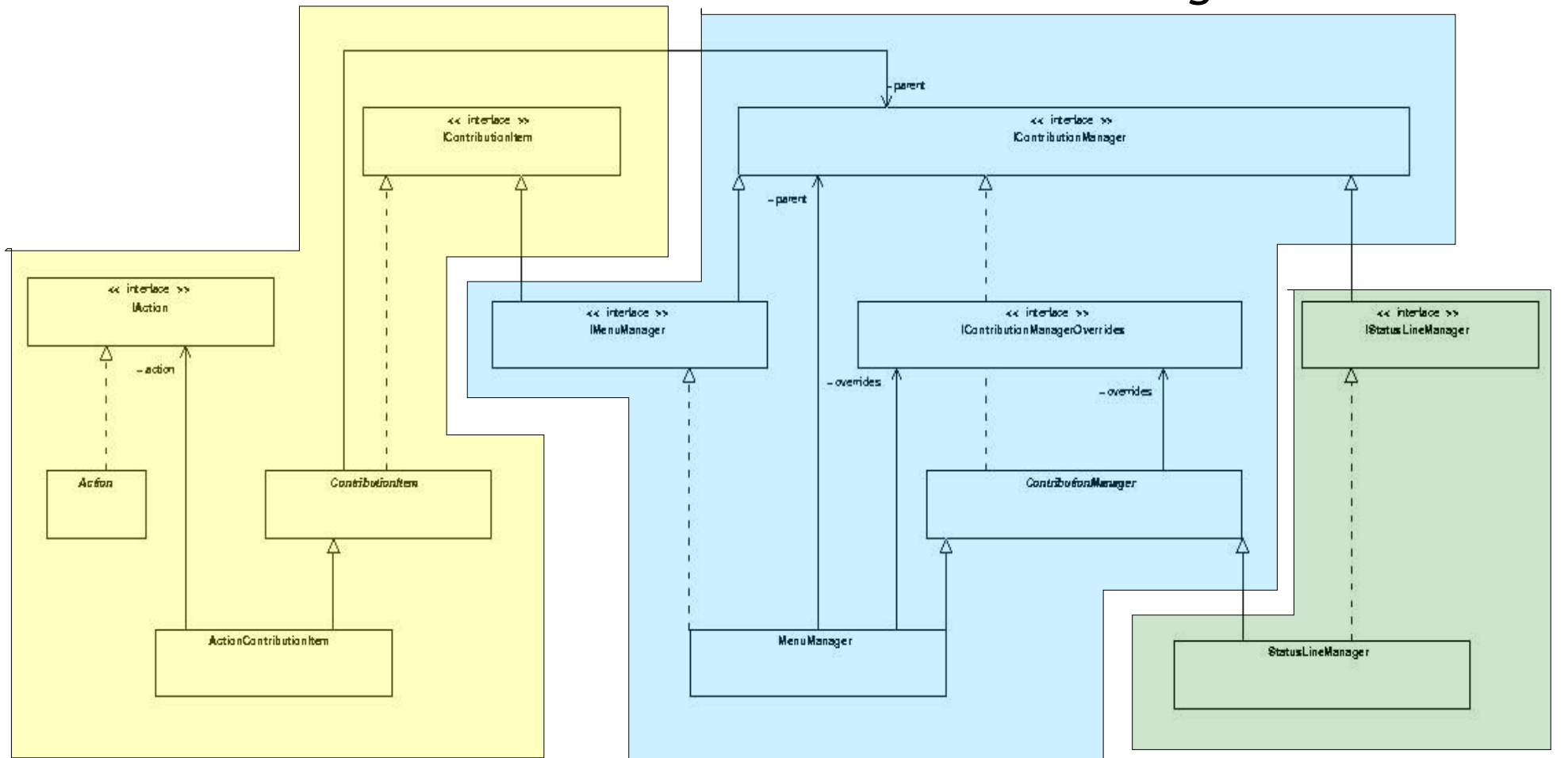
„Muster“ - Lösungen extrahieren



org.eclipse.jface.action

„Muster“ - Lösungen extrahieren

Menü Management



Benutzeraktion

Statuszeilen Management

Beispiel: org.eclipse.jface.action

„Muster“ - Lösung: Benutzeraktion

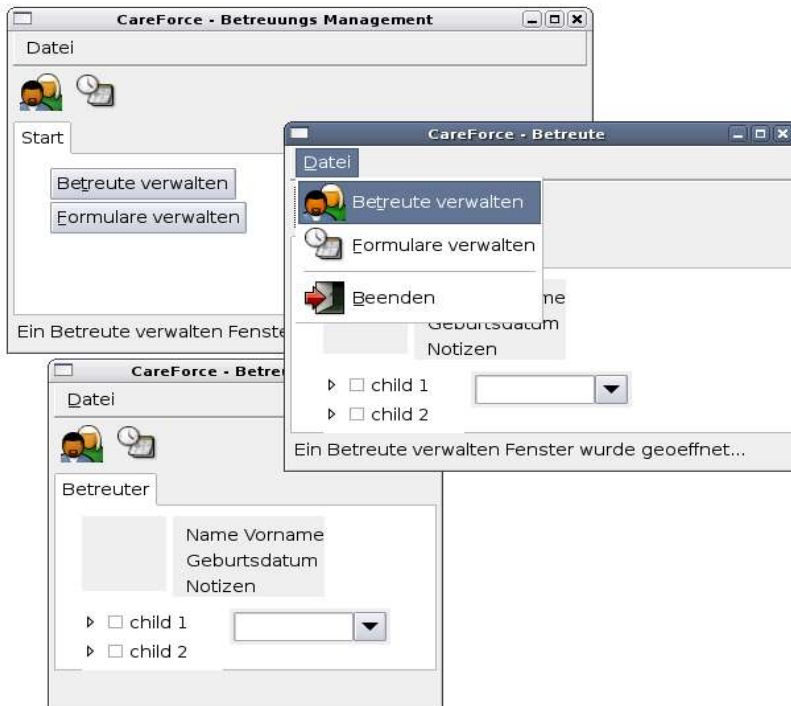
Name, Autor, Lizenz	<p>„Actions and Contributions“ Implementierung: org.eclipse.jface.actions Lizenz: CPL (opensource.org/licenses/cpl.php)</p>
Problem	Benutzerinteraktion programmübergreifend auf verschiedene Weise zugänglich machen.
Gebrauchsprinzip	Benutzerinteraktion
Kontext	Fensterapplikation mit Menüleisten, Kontextmenüs, Tollbars, Buttons, Statusleiste
Konflikte	Erwartungen: Standards, Neigung, Workflow
Lösung	<p>Stelle dem Benutzer per default alle geforderten Zugänge für eine Interaktion zur Verfügung. Gruppiere die Benutzeraktionen an Hand eines Benutzerworkflows</p>
Gebrauchsnutzen	<p>Ermöglicht die gewohnte Arbeitsweise (per Icon - visuell, Menüleiste – textuell) beizubehalten. Bietet eine standardisierte Umgebung. Ermöglicht die Adaption auf eigene Gewohnheiten.</p>
	...

„Muster“ – Lösung: Benutzeraktion

Name, Autor, Lizenz

„Actions and Contributions“
 Implementierung: org.eclipse.jface.actions
 Lizenz: CPL (opensource.org/licenses/cpl.php)

Beispiel

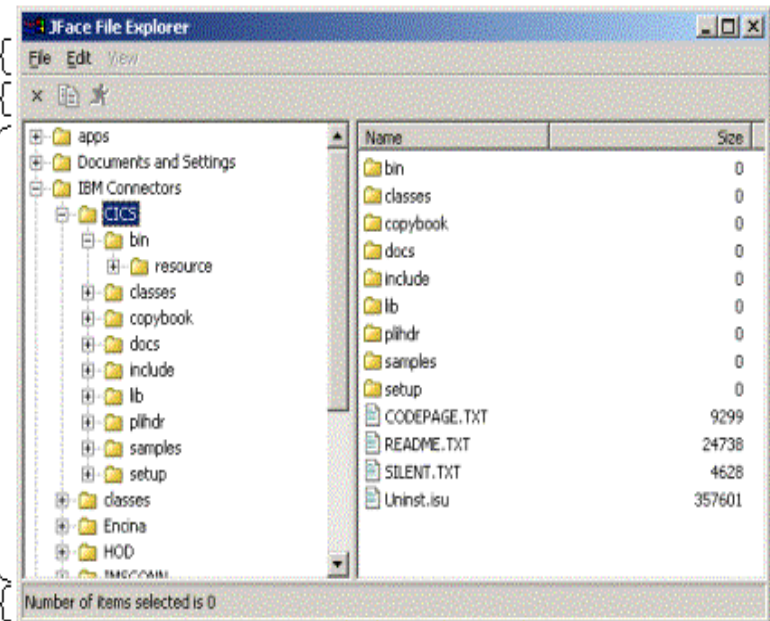


Menu Bar

Tool Bar

Contents

Status Line



Bekannte Anwendung

- KDE: Actions
- Eclipse.org / Jface: Actions and Contributions

Komplexe GUI's

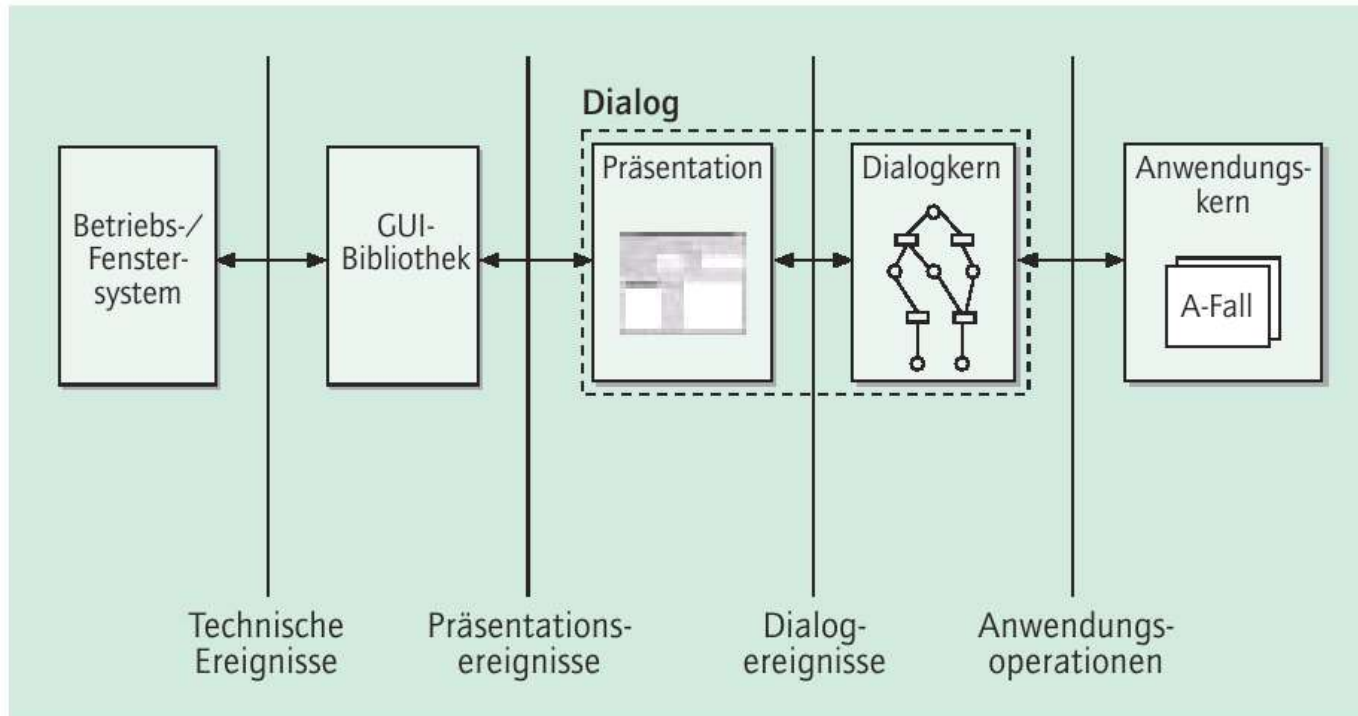
- Basis: zusammengesetzte Muster
 - Observer (syn. Publisher/Subscriber) – Synchronisation
 - Model View Controller, Presentation Model, Model View Presenter
- Referenzimplementierungen
 - JFace (Menü's, Wizards, Präferenzen, ...)
 - Eclipse.org (Komplexe Anwendungen)
 - Hibernate (Datenpersistenz)

Ausblick: Komplexe Anwendungen

- Open Source Referenzplattformen
 - Eclipse.org
 - ursprünglich für die Entwicklung integrierter Entwicklungsumgebungen gedacht
 - grundsätzlich offen für beliebige Anwendungen
 - Rich Client Plattform - Unterstützung komplexer Anwendungen (Deployment, Wartung, ...)
 - Plugin Architektur – Erweiterbarkeit / Offenheit
 - JFace - Standalone Anwendungen

Ausblick: Komplexe Landschaften

- Open Source Referenzarchitekturen
 - z.B. Quasar - Beispiel für GUI Referenzarchitektur
 - Standardarchitekturen im Grossen
 - z.B. für Transaktionen, Persistenz, GUI, Verteilung



Zusammenfassung

- Open Source GUI Frameworks warten noch „Entzauberung“
- Standards sind schwierig auszumachen
- GUI Development ist komplex
- Langfristige Sicht:
 - Musterkataloge, Mustersprachen
 - Referenzimplementierungen
 - Referenzarchitekturen

Quellenangaben

- GUI's entwickeln (Work in Progress)
 - <http://www.blue-it.org/de/documents>
- Eclipse
 - <http://help.eclipse.org>
- Patterns
 - u.a. <http://www.fowler.com>
- Quasar
 - <http://www.sdm.de/de/unternehmen/fe/>