

e x|xcellent solutions

| **Quality** is our business.

eXXcellent Project experience

Andreas Lux - eXXcellent solutions

Thema

Der Einsatz von Together in einem
kommerziellen Projekt
(eXXcellent Project experience)

Inhalte

Der Vortrag beinhaltet:

- x| Projektbeschreibung
- x| Verwendung von Together bei der
 - Analyse
 - Spezifikation
 - Implementierung
- x| Vorgenommene Anpassungen und Erweiterungen

Oberfläche

paper globe
PAPER · PRINT · PACKAGING

Willkommen bei paperglobe!

Wir sind für Sie da:

News
Rund ein Dutzend unabhängige Fachredaktionen berichten tagesaktuell direkt aus den Branchen Papier, Druck und Verpackung.

Events

- Schulungen, Workshops, Seminare und Fachmessen.
- Alle relevanten und branchenspezifischen Kongresse.

paperglobe liefert alle Termine und Kontaktadressen – schnell, einfach übersichtlich.

Jobs
Alle aktuellen und brandheißen Stellenanzeigen aus sechs

Events

- Engineering/Finishing & Converting.....
- Englisch einmal anders: Anwendung in der Druck- und Medienbranche
- MacSolutions
- Farbmessung in der Drucktechnik
- 44. DFTA-Fachtagung mit ProFle

Jobs

- Industriemeister Papiererzeugung
- Verkaufs-/Vertriebsprofi
- Erfahrener Verkäufer

Newsletter
Durch Ihre Registrierung erhalten Sie die neuesten Informationen der Papier-, Druck- und Verpackungsindustrie tagesaktuell per eMail.

REGISTRIERUNG

Registriert?
Durch eine Registrierung erhalten Sie Zugriff auf weitere Informationen in paperglobe.

Projektübersicht



Verwendete Technologien

Für die Umsetzung des Projekts wurden folgende Technologien verwendet:

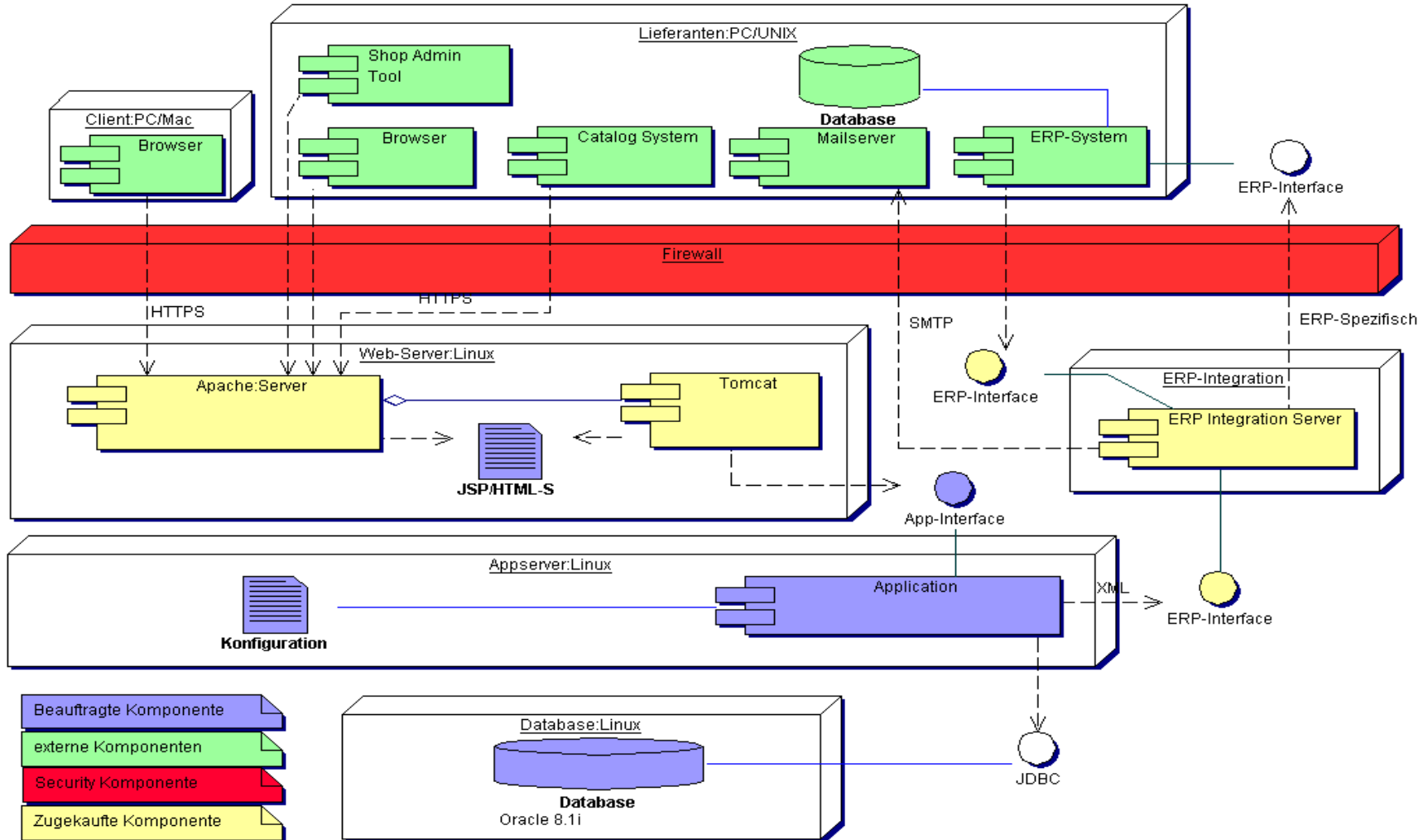
x| Frontend

- | Reiner HTML-Client
- | Geschachtelte JSP-Seiten
- | Organisiert über konfigurierbares Framework

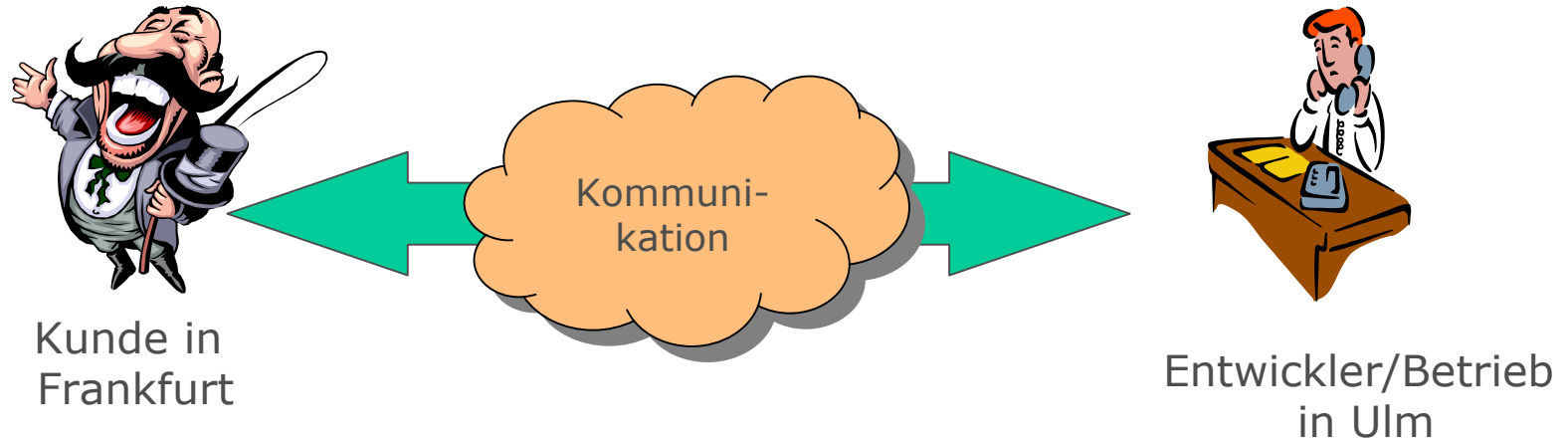
x| Backend (Linux)

- | EJB ´s auf jBoss Applikationsserver
- | Apache Webserver
- | Oracle 8.1i

Übersicht Infrastruktur



Rahmenbedingungen



- x| Kunde hat Sitz in Frankfurt
- x| Entwicklerteam in Ulm
- x| Betrieb der Plattform in Ulm

Probleme aufgrund der Infrastruktur

Aus der gegebenen Infrastruktur ergeben sich folgende Probleme:

- x| Kommunikation erfolgte ausschließlich auf elektronischem Weg
- x| Die Verteilung von physikalischen Gegenständen kann mehrere Tage dauern
- x| Die Veranstaltung von Meetings wird sehr aufwendig
- x| Manuelle Information des Kunden über den Projektfortschritt

Vorteile aufgrund der Infrastruktur

Aus der gegebenen Infrastruktur ergeben sich auch diverse Vorteile:

- x| Informationsaustausch erfolgt meist schriftlich
- x| Meetings sind i.d.R. zielgerichteter
- x| Meetings sind detaillierter vorbereitet

Einsatzgebiete von Together

Together wurde eingesetzt in:

- x| Spezifikation
- x| Design
- x| Implementierung
- x| Dokumentationsgenerierung

Einsatz bei der Spezifikationsphase

In der Spezifikationsphase wurde Together für folgende Aufgaben eingesetzt:

- x| Anforderungsanalyse

mit Use-Case-Diagrammen

- x| Prozessmodellierung

Beschreibung von Regelwerken anhand von Activity Diagrammen

- x| Beschreibung des Screenflows

Screen Definition Diagramme

- x| Beschreibung der Seiteninhalte

Screen Definition Diagramme

Vorgehensmodell

Es wurde ein eigenes, sehr schlankes Vorgehensmodell für Web-basierte Anwendungen verwendet.

- xl Verwendet die UML als Notation
- xl Anforderungen erwarten eine Beschreibungen als Aktivitäten
- xl Aktivitäten werden in User- und Systemaktivitäten unterteilt
- xl Aktivitätsdiagramm beschreibt den zugehörigen Controller
- ⇒ Einheitliche Implementierung der gesamten Businesslogik

Anforderungsanalyse

Für die Anforderungsanalyse wurde Together in der folgenden Weise verwendet:

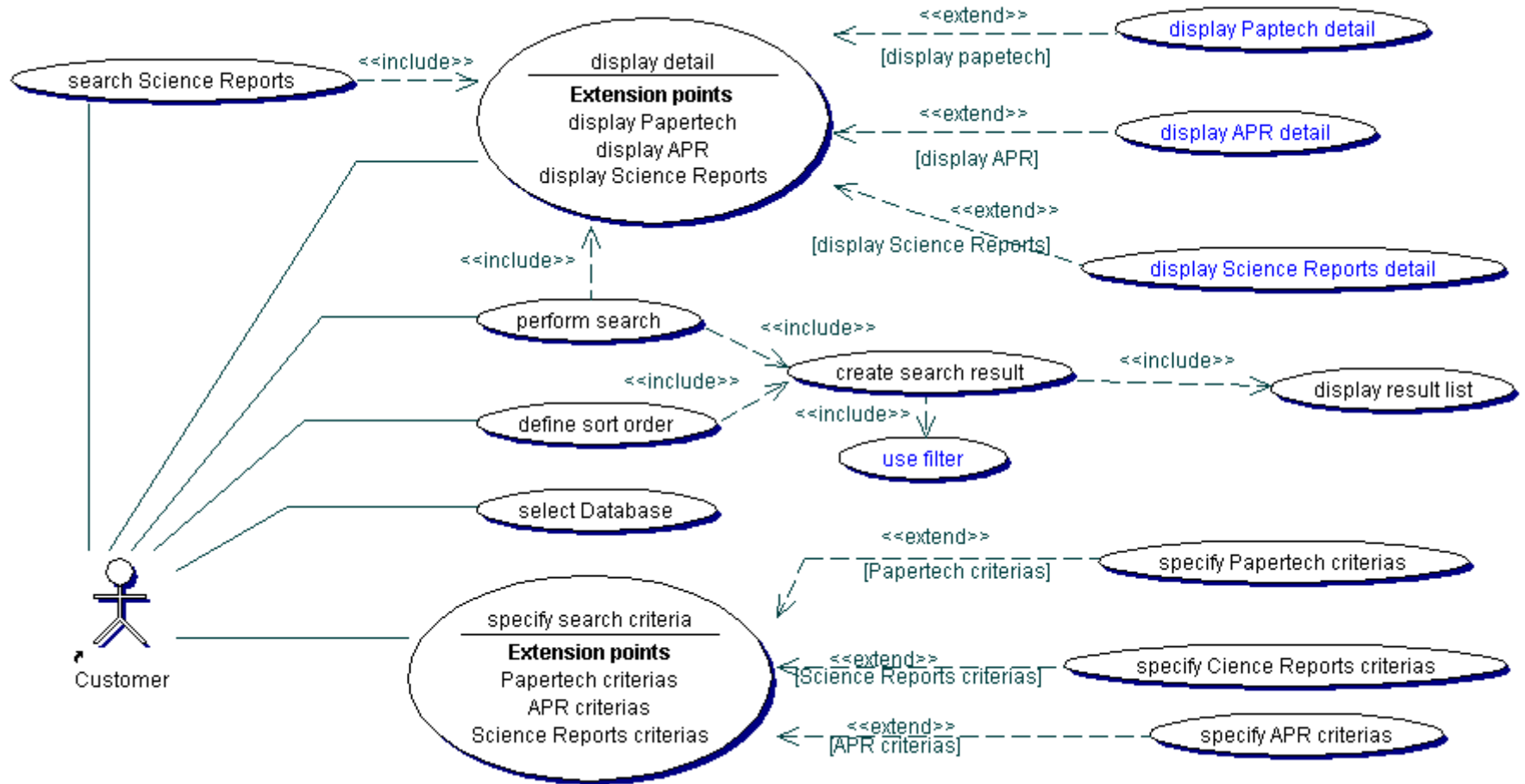
- x| Use-Case-Diagramme zur Erfassung der Kundenanforderungen
- x| Activity-Diagramme zum Erfassen der Businessregeln
- x| Trennung der Aktivitäten in User- und System- Aktivitäten

Verwendung der Use-Case-Diagramme

Use-Case-Diagramme werden in der Weise verwendet, dass

- xl alle Anforderungen modelliert werden, die dem Kunden als wichtig erscheinen
- xl die Granularität von Use-Cases nicht nach Zeitrastern (5 UC pro Mannjahr) modelliert werden
- xl sowohl Anwendungsfälle als auch plattformübergreifende Anforderungen mit Use-Cases modelliert werden
- xl Aktoren werden zentral in einem Diagramm beschrieben

Beispiel für Use-Case-Modellierung

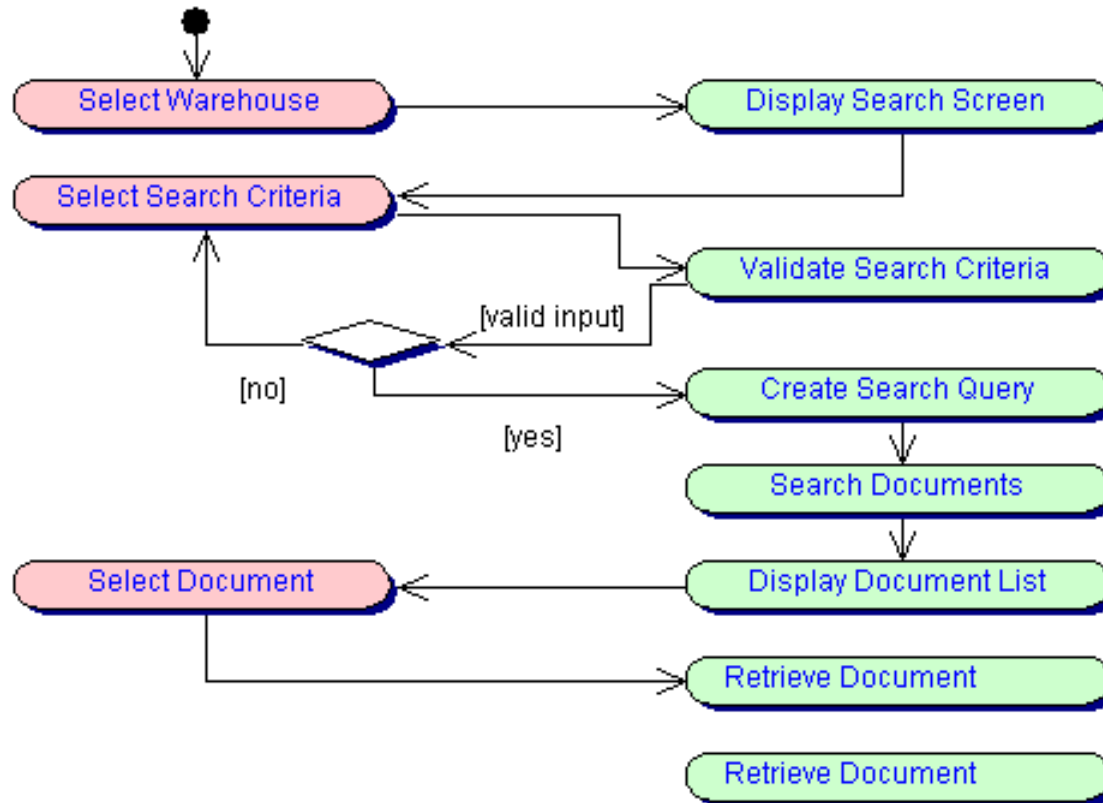


Verwendung von User-Activities

User-Activities dienen zur Definition der Aktivitäten eines Benutzers und zur Definition der Schnittstellen zum System.

- x| User-Aktivitäten sind Bedienungsschritte des Benutzers
- x| User-Aktivitäten provozieren Systemaktivitäten oder setzen diese voraus

Beispiel für User-Activity-Modellierung



Notwendige Anpassungen in Together

Folgende Anpassungen mussten vorgenommen werden:

x| User-Activity-Diagramme

- Neue Stereotypes (User-Activity und System Activity)
- Hintergrundfarbe anhand des gesetzten Stereotypes

Spezifikation

Für die Spezifikation wurde Together in folgender Art und Weise verwendet:

- x| Vervollständigung der Businesslogik
- x| Erstellung des Navigationskonzepts
- x| Funktionaler Entwurf der Oberflächen
- x| Generierung der Spezifikation als RTF-Dokument

Notwendige Anpassungen in Together

- x| Einführung eines neuen Diagrammtyps „Screen Definition“
- x| Erstellung eines neuen Symbols für die Darstellung der Seitenabfolge
- x| Seitensymbole müssen verbunden werden können
- x| Erstellung neuer Symbole für die Darstellung von Eingabe- und Ausgabe-Controls
- x| Erweiterung der Ausgabe-Controls um die Eingabemöglichkeit von Texten

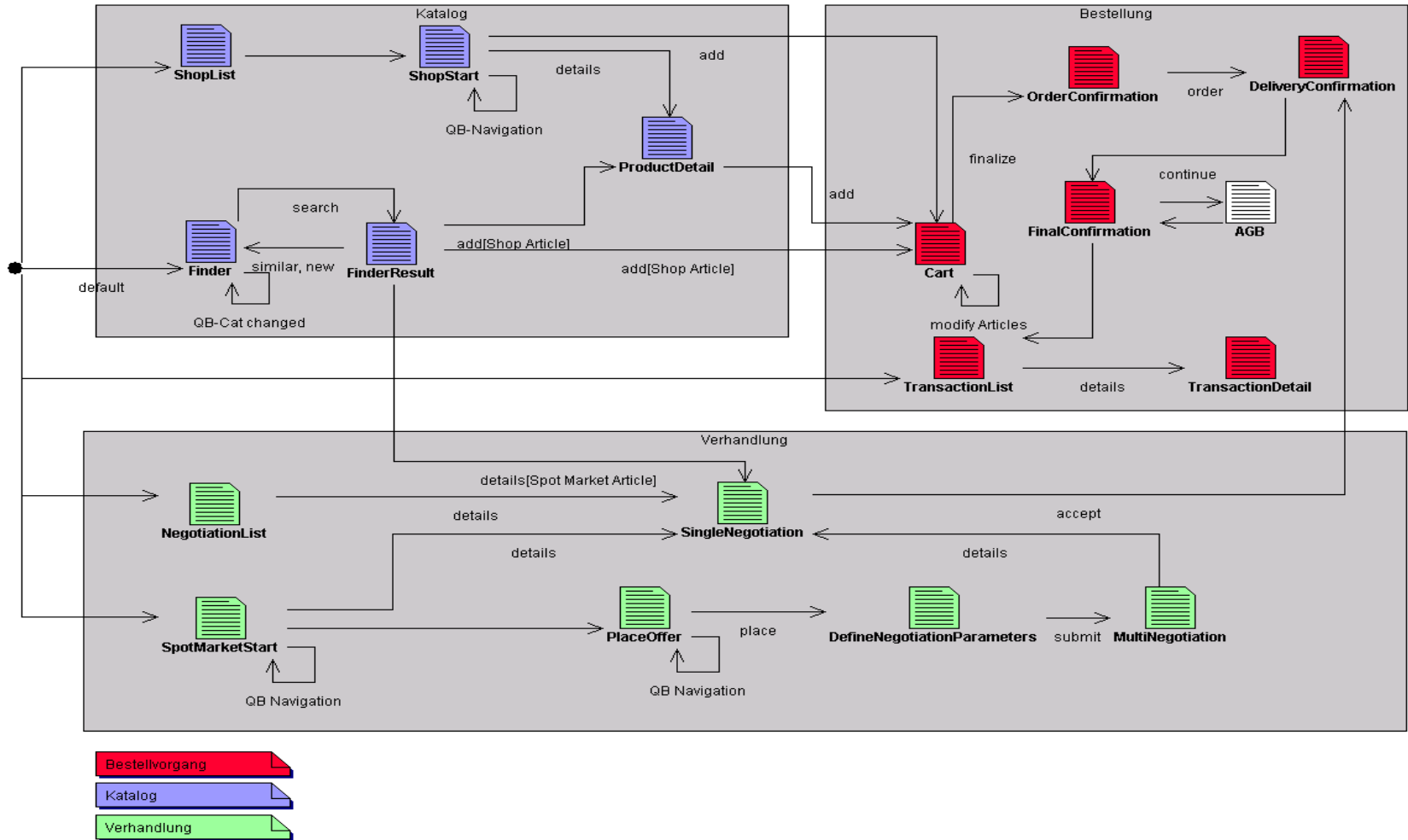
Notwendige Anpassungen in Together 2

x| RTF-Generierung

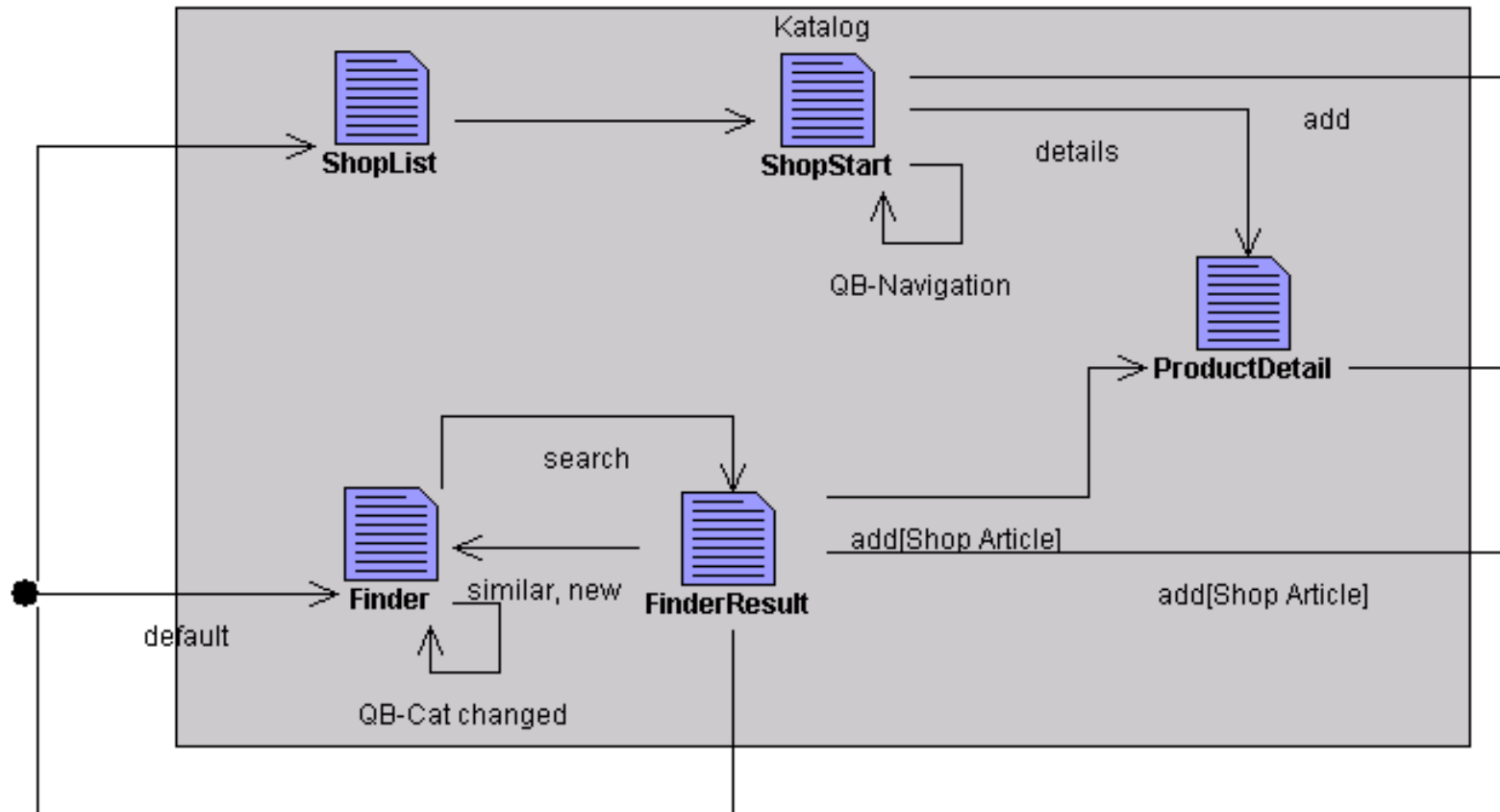
- Generierung einzelner Diagramme als RTF-Dokument
- Zusammenfassung der Einzeldokumente in einem Word-Master-Dokument

x| Erweiterung der Spezifikation durch nicht-generierte Texte und Grafiken

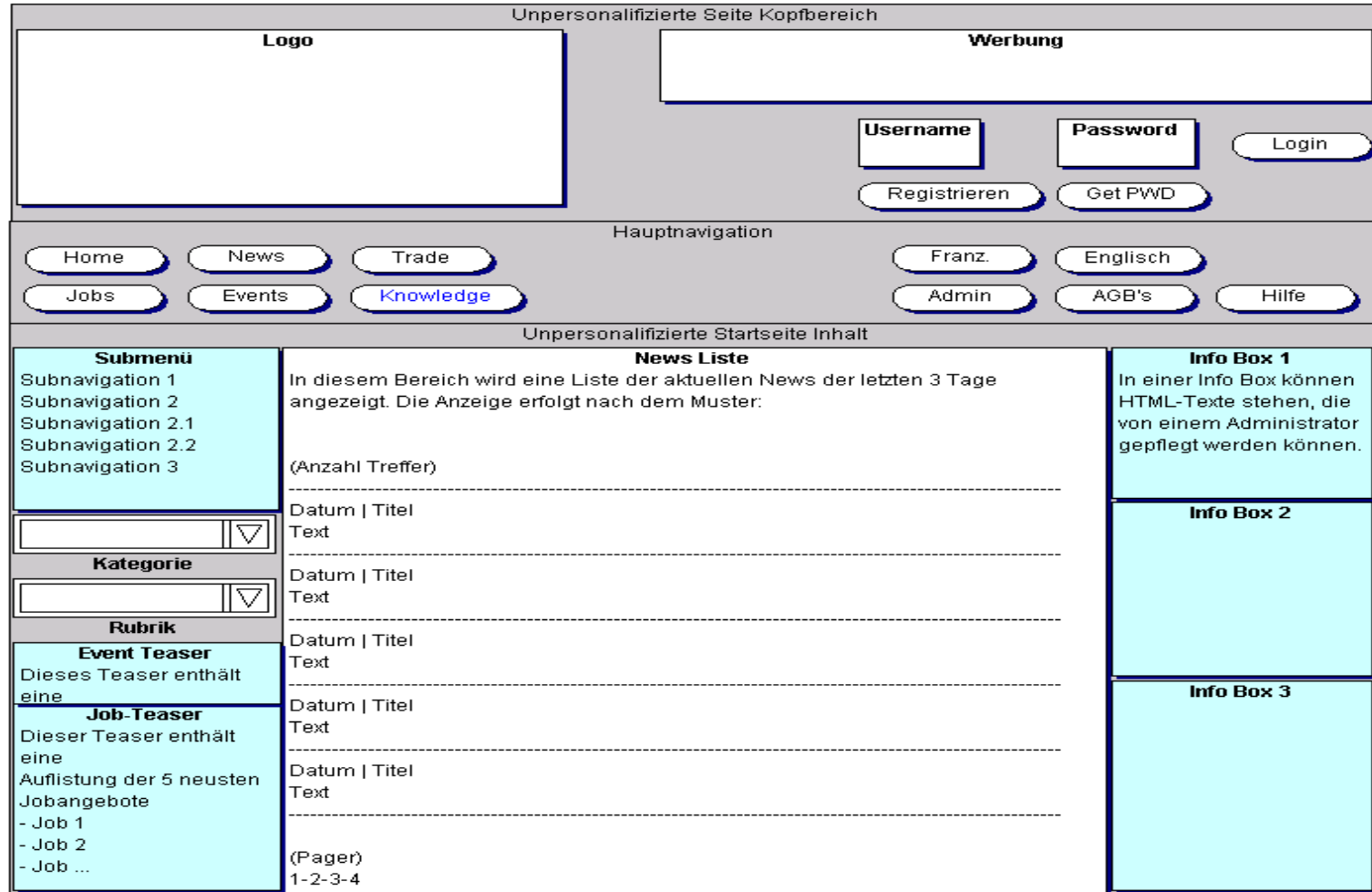
Beispiel für Screen-Flow-Diagramm



Screen-Flow-Diagramm Detail



Beispiel für einen Oberflächenentwurf



Herstellung eines Screen-Symbols

- x| Erstellung einer Zeichenklasse
- x| Änderungen in Konfiguration
 - | Verwendung der Klasse als ShapeType bei Auswahl des Stereotypes „Screen“ für „Objects“
 - | Ausblenden des Hintergrundrechtecks bei gegebenem Stereotype
- x| Erweiterung von „Objects“ mit dem Stereotype „Screen“

Design

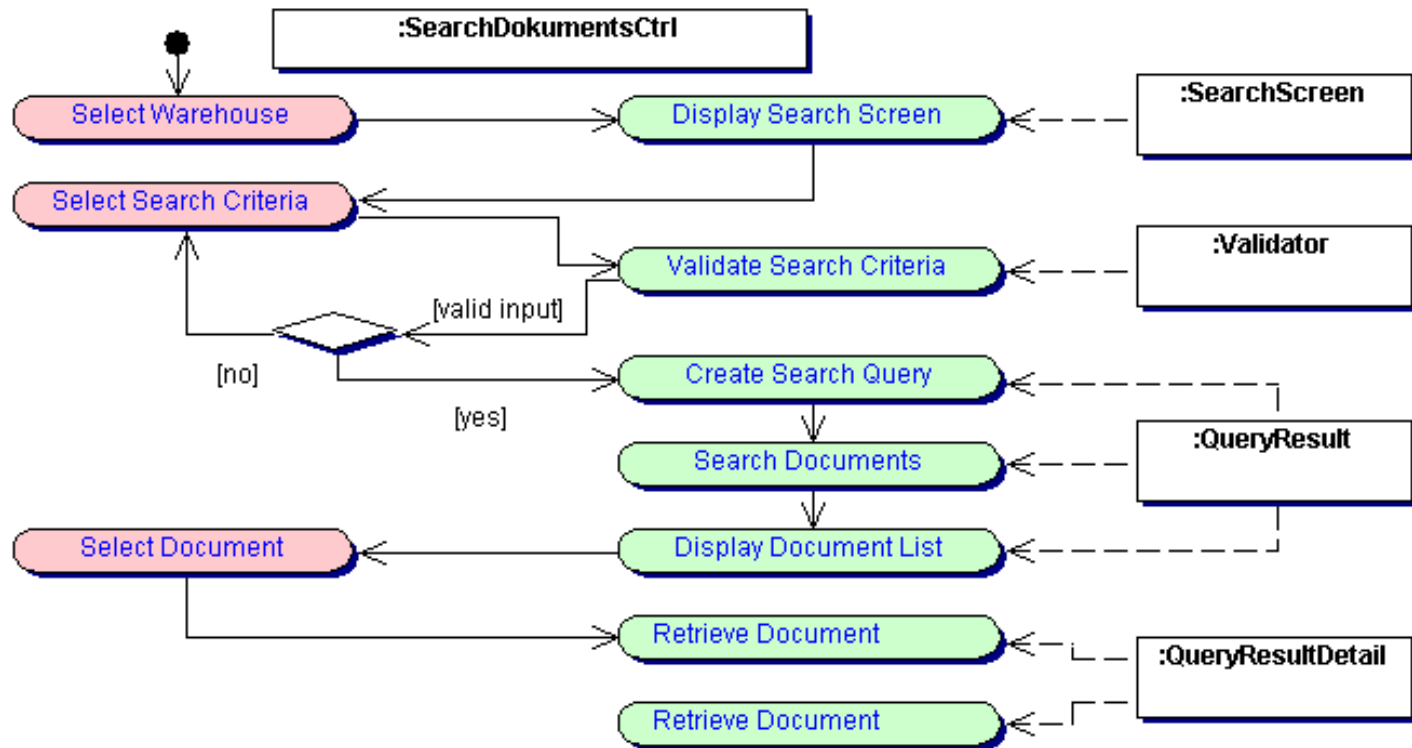
Für das Design wurde Together in folgender Art und Weise verwendet:

- x| Modellierung der grundlegenden Architektur mittels Deployment- und Component-Diagrammen

(Neuer Shape-Type „Datenbank“)

- x| Klassenfindung durch Erweiterung der User-Activity-Diagramme
- x| Entwurf des statischen Klassenmodells
- x| Keine weiteren Anpassungen notwendig

Beispiel für erweiterte User-Activities



Implementierung

In folgenden Bereichen wurde Together bei der Implementierung verwendet:

- x| Erstellung von Sequenzen

 - *Klassenfindung*

- x| Design des Klassenmodells

 - Erstellung eigener Templates für das verwendete Framework*

Notwendige Anpassungen in Together

- x| Erstellung eigener Templates für die Generierung von Standardklassen als Patterns
- x| Einbinden der Patterns in die Tool-Leiste des Klassendiagramms
- x| Neue Farbdefinitionen für Klassentypen anhand von „Stereotypes“

Erkannte Vorteile

- x| Layout der verwendeten Klassenmodelle weitaus strukturierter und klarer als in früheren Projekten
 - Bessere Kapselung der Komponenten*
 - Somit leichte Änderbarkeit der erstellten Komponenten (in bestimmten Grenzen)*
- x| Breiter Know-How-Transfer durch das gesamte Team
- x| Wecken des Interesses für „schönen“ Code
 - (Ein „MUSS“ für jedes erfolgreiche Projekt)*

Datenbankmodellierung

In folgenden Bereichen wurde Together bei der Datenbankmodellierung verwendet:

- xl Jeder Entwickler definiert sein eigenes logisches Datenmodell
- xl „Owner“ der Tabelle ist das aktuelle Modul
- xl Alle Linktabellen (für m-zu-n-Verknüpfungen) werden ausmodelliert
- xl Tabellen werden vom DB-Admin in einem Gesamtdiagramm zusammengefügt
- xl Pflege der DDL-Scripte durch den DB-Admin

Notwendige Anpassungen in Together

- x| Entfernen der Beschränkung für importierte Links wurde entfernt

Logische m-zu-n-Verknüpfungen wurden verboten

- x| Keine weiteren Anpassungen erforderlich

Resümee

- x| Insgesamt waren nur wenige Anpassungen in Together notwendig
- x| Akzeptanz bei Entwicklern war hoch
- x| Akzeptanz beim Kunden war hoch
- x| Projekt aufwand war nicht höher als ohne Together
- x| Wartung wird einfacher durch saubere Strukturierung der Applikation

Vielen Dank für Ihre Aufmerksamkeit

Andreas Lux

e x|x cellent solutions

| **Quality** is our business.