DLI Projekt: Kontakte

Vorlesung Aktuelle Themen der Dienstleistungsinformatik

Markus Marzotko, Thorben Seeland, Dominic Wirkner

Prof. Dr. Bernhard Steffen, Dipl.-Inf. Markus Doedt

Technische Universität Dortmund

22. Januar 2013

1 Einführung

2 Das Projekt

3 SAP Connector

4 SIB-Programmierung

5 jABC-Modell

1 Einführung

2 Das Projekt

3 SAP Connector

4 SIB-Programmierung

5 jABC-Modell

Test

Den Kontakt einer Gruppe hinzufügen

```
// Gruppe setzen
String groupURL = null;
switch (contactInfoCopy.getType()) {
case CUSTOMER:
groupURL = customerGroupURL;
contact.addGroupMembershipInfo(new
    GroupMembershipInfo(false, groupURL));
```

Was ist Compliance?

Definition 1

„Die Binsenweisheit, dass Unternehmen Gesetze einhalten müssen, heißt nun Compliance.“

Uwe H. Schneider, ZIP 2003, S. 645f

Definition 2

Bei „Compliance“ geht es um die „Erfüllung“, „Entsprechung“ bzw. „Konformität“ mit staatlichen Gesetzen sowie mit Regeln und Spezifikationen, mit Grundsätzen (ethische und moralische) und Verfahren sowie mit Standards (z.B. ISO) und Konventionen, die klar definiert worden sind. Die Erfüllung der Compliance kann sowohl auf Zwang (z.B. durch Gesetze) als auch auf Freiwilligkeit (z.B. Einhaltung von Standards) beruhen.

laut Compliance-Magazin.de

Was ist Compliance?

Definition 1

„Die Binsenweisheit, dass Unternehmen Gesetze einhalten müssen, heißt nun Compliance.“

Uwe H. Schneider, ZIP 2003, S. 645f

Definition 2

Bei „Compliance“ geht es um die „Erfüllung“, „Entsprechung“ bzw. „Konformität“ mit staatlichen Gesetzen sowie mit Regeln und Spezifikationen, mit Grundsätzen (ethische und moralische) und Verfahren sowie mit Standards (z.B. ISO) und Konventionen, die klar definiert worden sind. Die Erfüllung der Compliance kann sowohl auf Zwang (z.B. durch Gesetze) als auch auf Freiwilligkeit (z.B. Einhaltung von Standards) beruhen.

laut Compliance-Magazin.de

Was ist Compliance?

Definition 1

„Die Binsenweisheit, dass Unternehmen Gesetze einhalten müssen, heißt nun Compliance.“

Uwe H. Schneider, ZIP 2003, S. 645f

Definition 2

Bei „Compliance“ geht es um die „Erfüllung“, „Entsprechung“ bzw. „Konformität“ mit staatlichen Gesetzen sowie mit Regeln und Spezifikationen, mit Grundsätzen (ethische und moralische) und Verfahren sowie mit Standards (z.B. ISO) und Konventionen, die klar definiert worden sind. Die Erfüllung der Compliance kann sowohl auf Zwang (z.B. durch Gesetze) als auch auf Freiwilligkeit (z.B. Einhaltung von Standards) beruhen.

laut Compliance-Magazin.de

1 Einführung

2 Das Projekt

3 SAP Connector

4 SIB-Programmierung

5 jABC-Modell

Orchestrierung von Webservices

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren ⇝ Karteileichen
- **erste Überlegungen:**

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren ⇝ Karteileichen
- **erste Überlegungen:**

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- erste Überlegungen:

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- erste Überlegungen:

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- erste Überlegungen:

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- erste Überlegungen:

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- erste Überlegungen:

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- erste Überlegungen:

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- erste Überlegungen:

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- **erste Überlegungen:**
  - drei Teile: Google, SAP, SIBs
  - GUI Elemente für Dateingabe

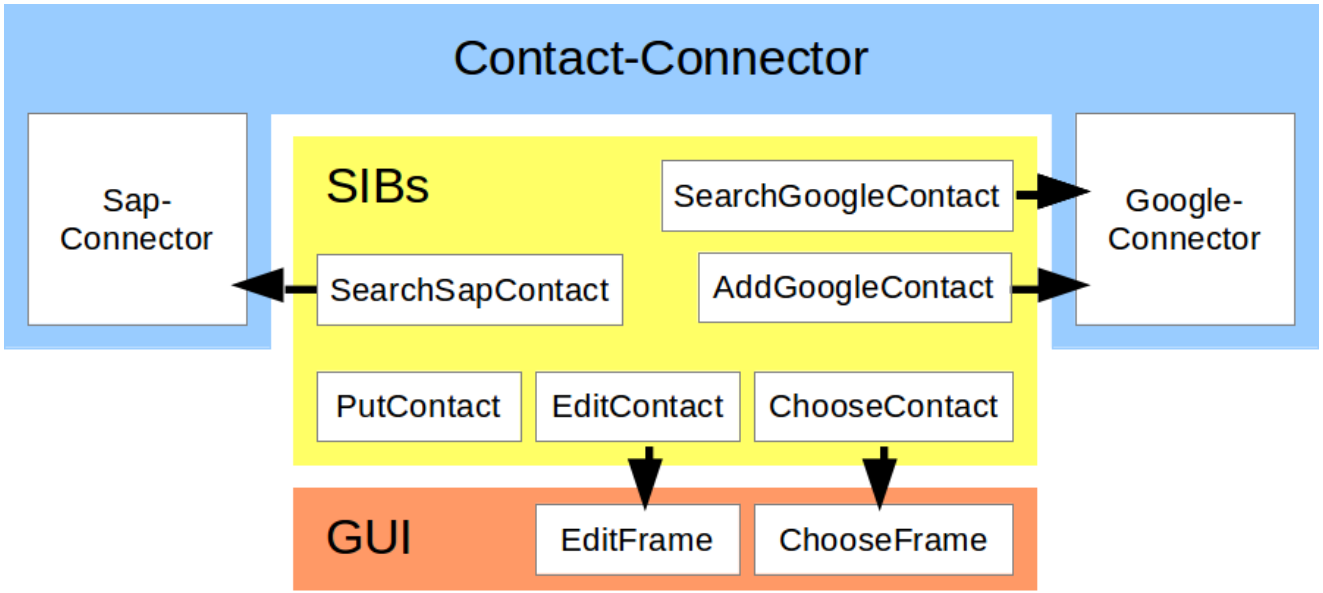
Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- **erste Überlegungen:**
  - drei Teile: Google, SAP, SIBs
  - GUI Elemente für Dateingabe

Die Aufgabe

- **Datenmigration:** Kontakte von SAP zu Google
- **Strategie:** nicht einfach kopieren → Karteileichen
  - Kontakte nur bei Bedarf übertragen!
  - zunächst bei Google anfragen
  - erst dann Migration von SAP
  - sonst manuelle Eingabe
- **erste Überlegungen:**
  - drei Teile: Google, SAP, SIBs
  - GUI Elemente für Dateingabe

Projekt-Komponenten



Eingesetzte Tools

- **Standard:** automatisierte Tests (JUnit), Versionsverwaltung (git)
- Apache Maven: Build-Management-Tool

Eingesetzte Tools

- **Standard:** automatisierte Tests (JUnit), Versionsverwaltung (git)
- Apache Maven: Build-Management-Tool

Eingesetzte Tools

- **Standard:** automatisierte Tests (JUnit), Versionsverwaltung (git)
- **Apache Maven:** Build-Management-Tool
  - unterstützt Software-Lebenszyklus → automatische Ausführung der Tests
  - einfaches Einbinden von Abhängigkeiten (externe Pakete)
  - verpackt alle Komponenten zu einer JAR-Datei → jABC-Projekt

Eingesetzte Tools

- **Standard:** automatisierte Tests (JUnit), Versionsverwaltung (git)
- **Apache Maven:** Build-Management-Tool
  - unterstützt Software-Lebenszyklus → automatische Ausführung der Tests
  - einfaches Einbinden von Abhängigkeiten (externe Pakete)
  - verpackt alle Komponenten zu einer JAR-Datei → jABC-Projekt

Eingesetzte Tools

- **Standard:** automatisierte Tests (JUnit), Versionsverwaltung (git)
- **Apache Maven:** Build-Management-Tool
  - unterstützt Software-Lebenszyklus → automatische Ausführung der Tests
  - einfaches Einbinden von Abhängigkeiten (externe Pakete)
  - verpackt alle Komponenten zu einer JAR-Datei → jABC-Projekt

Eingesetzte Tools

- **Standard:** automatisierte Tests (JUnit), Versionsverwaltung (git)
- **Apache Maven:** Build-Management-Tool
  - unterstützt Software-Lebenszyklus → automatische Ausführung der Tests
  - einfaches Einbinden von Abhängigkeiten (externe Pakete)
  - verpackt alle Komponenten zu einer JAR-Datei → jABC-Projekt

1 Einführung

2 Das Projekt

3 SAP Connector

4 SIB-Programmierung

5 jABC-Modell

Aufgabe

- Erhalte Kontaktobjekt mit Angaben zu Typ, Vorname, Nachname, Firma..
- Filtere aus Datenbank entsprechende Datensätze
- Gib aufbereitete Liste aller zutreffenden Kontakte zurück

Verwendete WSDLs

Lieferant

- Find Supplier by Name and Address
- Read Supplier Basic Data

Mitarbeiter

- Find Employee by Elements
- Find Employee Address by Employee

Kunde

- Find Customer by Elements

Programmablauf

- Art des Filterobjekts überprüfen und entsprechenden Webservice aufrufen
- IDs auslesen und anderen Webservice für alle IDs (einzeln) aufrufen
- Rückgabeobjekte auslesen und Daten geordnet zurückgeben

→ Bei Kunde reicht ein Webserviceaufruf

Lieferant, Kunde und Mitarbeiter: 3 unterschiedliche Welten

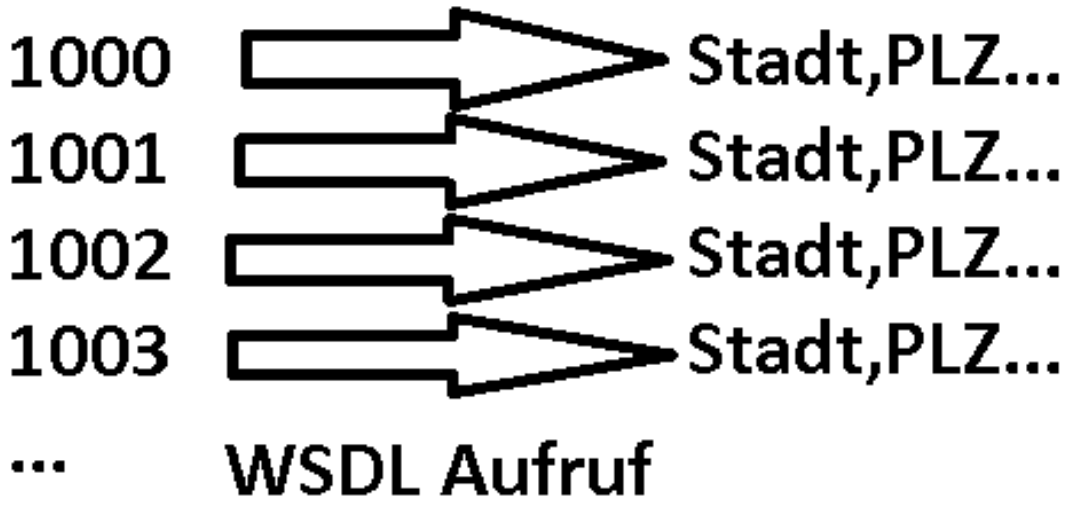
- Lieferant, Kunde und Mitarbeiter verwenden völlig unterschiedliche Klassen
- lediglich die Übergabe von Passwort und Username ist gleich
- Employee kommt aus dem SAP Human Ressources (HR) Bereich

→ Selbst einfachste Zuweisungen verkommen hier zur Akkordarbeit

Problematik bei Lieferant und Kunde

- Webservices geben hier nur Liste von IDs und Namen zurück
- Für Adressinformationen weiterer Aufruf mit anderem Webservice nötig
- **PROBLEM:** Aufruf geschieht für jede ID einzeln

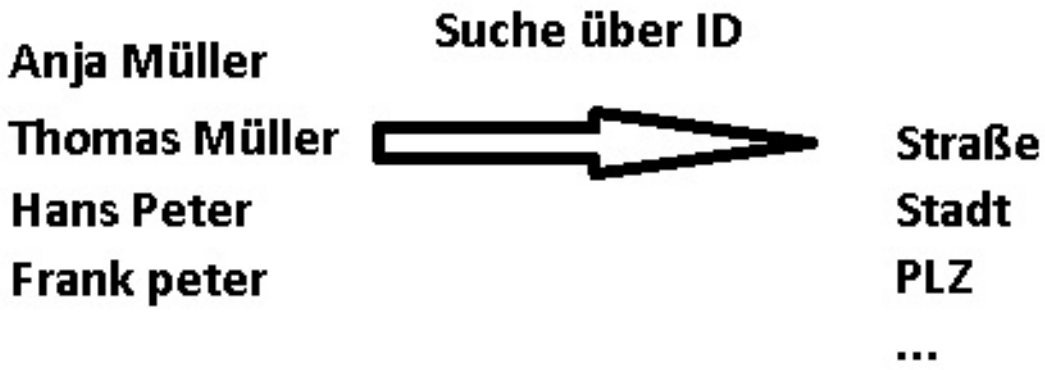
Problemdarstellung



Lösung

- GUI so erstellen, dass zunächst nur Name/Firma angezeigt werden
- Name/Firma sind bereits nach erstem Webservice Aufruf vorhanden
- Erst nach Klick auf Namen werden Adressdaten via Webservice angefordert

Lösung


Suche über ID
Anja Müller
Thomas Müller
Hans Peter
Frank peter
Straße
Stadt
PLZ
...

1 Einführung

2 Das Projekt

3 SAP Connector

4 SIB-Programmierung

5 jABC-Modell

SIB-Programmierung

Vorüberlegungen

- **3 Sorten von SIBs:** Google, SAP, GUI
  - **Google-SIBs:** Kontakt suchen und hinzufügen
  - **SAP-SIB:** Kontakt suchen
  - **GUI-SIBs:**

Vorüberlegungen

- **3 Sorten von SIBs:** Google, SAP, GUI
  - **Google-SIBs:** Kontakt suchen und hinzufügen
  - **SAP-SIB:** Kontakt suchen
  - **GUI-SIBs:**

Vorüberlegungen

- **3 Sorten von SIBs:** Google, SAP, GUI
- **Google-SIBs:** Kontakt suchen und hinzufügen
- **SAP-SIB:** Kontakt suchen
- **GUI-SIBs:**

Vorüberlegungen

- **3 Sorten von SIBs:** Google, SAP, GUI
- **Google-SIBs:** Kontakt suchen und hinzufügen
- **SAP-SIB:** Kontakt suchen
- **GUI-SIBs:**

Vorüberlegungen

- **3 Sorten von SIBs:** Google, SAP, GUI
- **Google-SIBs:** Kontakt suchen und hinzufügen
- **SAP-SIB:** Kontakt suchen
- **GUI-SIBs:**

Vorüberlegungen

- **3 Sorten von SIBs:** Google, SAP, GUI
- **Google-SIBs:** Kontakt suchen und hinzufügen
- **SAP-SIB:** Kontakt suchen
- **GUI-SIBs:**
  - Eingabe von Kontakt-Attributen
  - Auswahl aus einer Liste von Kontakten

Vorüberlegungen

- **3 Sorten von SIBs:** Google, SAP, GUI
- **Google-SIBs:** Kontakt suchen und hinzufügen
- **SAP-SIB:** Kontakt suchen
- **GUI-SIBs:**
  - Eingabe von Kontakt-Attributen
  - Auswahl aus einer Liste von Kontakten

Vorüberlegungen

- **3 Sorten von SIBs:** Google, SAP, GUI
- **Google-SIBs:** Kontakt suchen und hinzufügen
- **SAP-SIB:** Kontakt suchen
- **GUI-SIBs:**
  - Eingabe von Kontakt-Attributen
  - Auswahl aus einer Liste von Kontakten

Google + SAP: suche Kontakt

- **Input:** eine Instanz der Klasse `Contact`
  - dient als Filter für die Suche
- **Output:** Liste von `Contact`-Objekten
- **Branches:**
  - **found:** mehr als 0 Ergebnisse
  - **not found:** keine Ergebnisse
  - **error:** es wurde eine *Exception* geworfen

Google + SAP: suche Kontakt

- **Input:** eine Instanz der Klasse `Contact`
  - dient als Filter für die Suche
- **Output:** Liste von `Contact`-Objekten
- **Branches:**
  - **found:** mehr als 0 Ergebnisse
  - **not found:** keine Ergebnisse
  - **error:** es wurde eine *Exception* geworfen

Google + SAP: suche Kontakt

- **Input:** eine Instanz der Klasse `Contact`
  - dient als Filter für die Suche
- **Output:** Liste von `Contact`-Objekten
- **Branches:**
  - **found:** mehr als 0 Ergebnisse
  - **not found:** keine Ergebnisse
  - **error:** es wurde eine *Exception* geworfen

Google + SAP: suche Kontakt

- **Input:** eine Instanz der Klasse `Contact`
  - dient als Filter für die Suche
- **Output:** Liste von `Contact`-Objekten
- **Branches:**
  - **found:** mehr als 0 Ergebnisse
  - **not found:** keine Ergebnisse
  - **error:** es wurde eine *Exception* geworfen

Google + SAP: suche Kontakt

- **Input:** eine Instanz der Klasse `Contact`
  - dient als Filter für die Suche
- **Output:** Liste von `Contact`-Objekten
- **Branches:**
  - **found:** mehr als 0 Ergebnisse
  - **not found:** keine Ergebnisse
  - **error:** es wurde eine *Exception* geworfen

Google + SAP: suche Kontakt

- **Input:** eine Instanz der Klasse `Contact`
  - dient als Filter für die Suche
- **Output:** Liste von `Contact`-Objekten
- **Branches:**
  - **found:** mehr als 0 Ergebnisse
  - **not found:** keine Ergebnisse
  - **error:** es wurde eine *Exception* geworfen

Google + SAP: suche Kontakt

- **Input:** eine Instanz der Klasse `Contact`
  - dient als Filter für die Suche
- **Output:** Liste von `Contact`-Objekten
- **Branches:**
  - **found:** mehr als 0 Ergebnisse
  - **not found:** keine Ergebnisse
  - **error:** es wurde eine *Exception* geworfen

Google: Kontakt hinzufügen

- **Input:** eine Instanz der Klasse `Contact`
- **Output:** keiner
- **Branches:**
  - **default:** Kontakt erfolgreich hinzugefügt
  - **error:** es wurde eine *Exception* geworfen

Google: Kontakt hinzufügen

- **Input:** eine Instanz der Klasse `Contact`
- **Output:** keiner
- **Branches:**
  - **default:** Kontakt erfolgreich hinzugefügt
  - **error:** es wurde eine *Exception* geworfen

Google: Kontakt hinzufügen

- **Input:** eine Instanz der Klasse `Contact`
- **Output:** keiner
- **Branches:**
  - **default:** Kontakt erfolgreich hinzugefügt
  - **error:** es wurde eine *Exception* geworfen

Google: Kontakt hinzufügen

- **Input:** eine Instanz der Klasse `Contact`
- **Output:** keiner
- **Branches:**
  - **default:** Kontakt erfolgreich hinzugefügt
  - **error:** es wurde eine *Exception* geworfen

Google: Kontakt hinzufügen

- **Input:** eine Instanz der Klasse `Contact`
- **Output:** keiner
- **Branches:**
  - **default:** Kontakt erfolgreich hinzugefügt
  - **error:** es wurde eine *Exception* geworfen

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
- **Branches:**

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
- **Branches:**

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
- **Branches:**

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
- **Branches:**

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
  - wenn Button ÖK"geklickt wurde
- **Branches:**

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
  - wenn Button ÖK"geklickt wurde
- **Branches:**

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
  - wenn Button ÖK"geklickt wurde
- **Branches:**

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button ÖK"
  - **cancel:** Eingabe abgebrochen mit Button "CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button ÖK"
  - **cancel:** Eingabe abgebrochen mit Button "CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button ÖK"
  - **cancel:** Eingabe abgebrochen mit Button "CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-Daten eingeben

- **Input:** eine Instanz der Klasse `Contact`
  - Formular wird entsprechend befüllt
  - zudem Parameter für Fenstertitel und Validierung
- **Output:** geänderte(!) Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button ÖK"
  - **cancel:** Eingabe abgebrochen mit Button "CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-auswählen

- **Input:** Liste von `Contact`-Objekten
  - zudem Parameter für Fenstertitel
- **Output:** eine Instanz der Klasse `Contact`
  - wenn Button „OK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button „OK"
  - **cancel:** Eingabe abgebrochen mit Button „CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-auswählen

- **Input:** Liste von `Contact`-Objekten
  - zudem Parameter für Fenstertitel
- **Output:** eine Instanz der Klasse `Contact`
  - wenn Button „OK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button „OK"
  - **cancel:** Eingabe abgebrochen mit Button „CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-auswählen

- **Input:** Liste von `Contact`-Objekten
  - zudem Parameter für Fenstertitel
- **Output:** eine Instanz der Klasse `Contact`
  - wenn Button „OK" geklickt wurde
- **Branches:**
  - ok: Eingabe bestätigt mit Button „OK"
  - cancel: Eingabe abgebrochen mit Button „CANCEL"
  - error: es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-auswählen

- **Input:** Liste von `Contact`-Objekten
  - zudem Parameter für Fenstertitel
- **Output:** eine Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - ok: Eingabe bestätigt mit Button ÖK"
  - cancel: Eingabe abgebrochen mit Button "CANCEL"
  - error: es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-auswählen

- **Input:** Liste von `Contact`-Objekten
  - zudem Parameter für Fenstertitel
- **Output:** eine Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button ÖK"
  - **cancel:** Eingabe abgebrochen mit Button "CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-auswählen

- **Input:** Liste von `Contact`-Objekten
  - zudem Parameter für Fenstertitel
- **Output:** eine Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button ÖK"
  - **cancel:** Eingabe abgebrochen mit Button "CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-auswählen

- **Input:** Liste von `Contact`-Objekten
  - zudem Parameter für Fenstertitel
- **Output:** eine Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button ÖK"
  - **cancel:** Eingabe abgebrochen mit Button "CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

GUI: Kontakt-auswählen

- **Input:** Liste von `Contact`-Objekten
  - zudem Parameter für Fenstertitel
- **Output:** eine Instanz der Klasse `Contact`
  - wenn Button ÖK" geklickt wurde
- **Branches:**
  - **ok:** Eingabe bestätigt mit Button ÖK"
  - **cancel:** Eingabe abgebrochen mit Button "CANCEL"
  - **error:** es wurde eine *Exception* geworfen, oder UNKNOWN

Besonderheit der GUI

- **warten auf Eingabe:** wie `trace()`-Methode anhalten?
  - Swing-Frame läuft in einem eigenem Thread!
- **Lösung hier:** mittels `synchronized`-Block in Frame und SIB

```
public String trace(ExecutionEnvironment env) {

...
synchronized (frame) {
frame.wait();
...
```

- **Vorgehen:**

Besonderheit der GUI

- **warten auf Eingabe:** wie `trace()`-Methode anhalten?
  - Swing-Frame läuft in einem eigenem Thread!
- **Lösung hier:** mittels `synchronized`-Block in Frame und SIB

```
public String trace(ExecutionEnvironment env) {

...
synchronized (frame) {
frame.wait();
...
```

- **Vorgehen:**

Besonderheit der GUI

- **warten auf Eingabe:** wie `trace()`-Methode anhalten?
  - Swing-Frame läuft in einem eigenem Thread!
- **Lösung hier:** mittels `synchronized`-Block in Frame und SIB

```
public String trace(ExecutionEnvironment env) {

...
synchronized (frame) {
frame.wait();
...
```

- **Vorgehen:**

Besonderheit der GUI

- **warten auf Eingabe:** wie `trace()`-Methode anhalten?
  - Swing-Frame läuft in einem eigenem Thread!
- **Lösung hier:** mittels `synchronized`)-Block in Frame und SIB

```
public String trace(ExecutionEnvironment env) {

...
synchronized (frame) {
frame.wait();
...
```

- **Vorgehen:**
  - `trace()` erstellt den Frame und wartet auf ein `notify()`
  - `notify()` wird vom Action-Listener der Buttons aufgerufen
  - im Anschluss kann `trace()` die Eingabe vom Frame erfragen

Besonderheit der GUI

- **warten auf Eingabe:** wie `trace()`-Methode anhalten?
  - Swing-Frame läuft in einem eigenem Thread!
- **Lösung hier:** mittels `synchronized`)-Block in Frame und SIB

```
public String trace(ExecutionEnvironment env) {

...
synchronized (frame) {
frame.wait();
...
```

- **Vorgehen:**
  - `trace()` erstellt den Frame und wartet auf ein `notify()`
  - `notify()` wird vom Action-Listener der Buttons aufgerufen
  - im Anschluss kann `trace()` die Eingabe vom Frame erfragen

Besonderheit der GUI

- **warten auf Eingabe:** wie `trace()`-Methode anhalten?
  - Swing-Frame läuft in einem eigenem Thread!
- **Lösung hier:** mittels `synchronized`)-Block in Frame und SIB

```
public String trace(ExecutionEnvironment env) {

...
synchronized (frame) {
frame.wait();
...
```

- **Vorgehen:**
  - `trace()` erstellt den Frame und wartet auf ein `notify()`
  - `notify()` wird vom Action-Listener der Buttons aufgerufen
  - im Anschluss kann `trace()` die Eingabe vom Frame erfragen

Besonderheit der GUI

- **warten auf Eingabe:** wie `trace()`-Methode anhalten?
  - Swing-Frame läuft in einem eigenem Thread!
- **Lösung hier:** mittels `synchronized`)-Block in Frame und SIB

```
public String trace(ExecutionEnvironment env) {

...
synchronized (frame) {
frame.wait();
...
```

- **Vorgehen:**
  - `trace()` erstellt den Frame und wartet auf ein `notify()`
  - `notify()` wird vom Action-Listener der Buttons aufgerufen
  - im Anschluss kann `trace()` die Eingabe vom Frame erfragen

Besonderheit der GUI

- **warten auf Eingabe:** wie `trace()`-Methode anhalten?
  - Swing-Frame läuft in einem eigenem Thread!
- **Lösung hier:** mittels `synchronized`)-Block in Frame und SIB

```
public String trace(ExecutionEnvironment env) {

...
synchronized (frame) {
frame.wait();
...
```

- **Vorgehen:**
  - `trace()` erstellt den Frame und wartet auf ein `notify()`
  - `notify()` wird vom Action-Listener der Buttons aufgerufen
  - im Anschluss kann `trace()` die Eingabe vom Frame erfragen

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
- **Lösung hier:** mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

- **Effekt:**

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
  - Lösung hier: mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

  - Effekt:

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
- Lösung hier: mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

- Effekt:

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
- Lösung hier: mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

- Effekt:

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
- **Lösung hier:** mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

- **Effekt:**
  - verstellt Optionen der aktiven JVM
  - auch nach Ausführung des Modells weiterhin wirksam
  - sollte in realem Szenario vermieden werden

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
- **Lösung hier:** mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

- **Effekt:**
  - verstellt Optionen der aktiven JVM
  - auch nach Ausführung des Modells weiterhin wirksam
  - sollte in realem Szenario vermieden werden

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
- **Lösung hier:** mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

- **Effekt:**
  - verstellt Optionen der aktiven JVM
  - auch nach Ausführung des Modells weiterhin wirksam
  - sollte in realem Szenario vermieden werden

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
- **Lösung hier:** mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

- **Effekt:**
  - verstellt Optionen der aktiven JVM
  - auch nach Ausführung des Modells weiterhin wirksam
  - sollte in realem Szenario vermieden werden

Problem mit jABC

- **eigener Java-Code im jABC:** Unterschiede zum „normalen“ JRE
  - Code läuft ausserhalb von jABC...
  - ABER: Ausführung im Modell wirft Exceptions?
- **Lösung hier:** mittels „bad Practice“

```
...
System.setProperty("javax.xml.parsers.SAXParserFactory",
    ...);
System.setProperty("javax.xml.parsers.SAXParser", ...);
System.setProperty("oracle.xml.parser.v2.SAXParser",
    ...);
...
```

- **Effekt:**
  - verstellt Optionen der aktiven JVM
  - auch nach Ausführung des Modells weiterhin wirksam
  - sollte in realem Szenario vermieden werden

1 Einführung

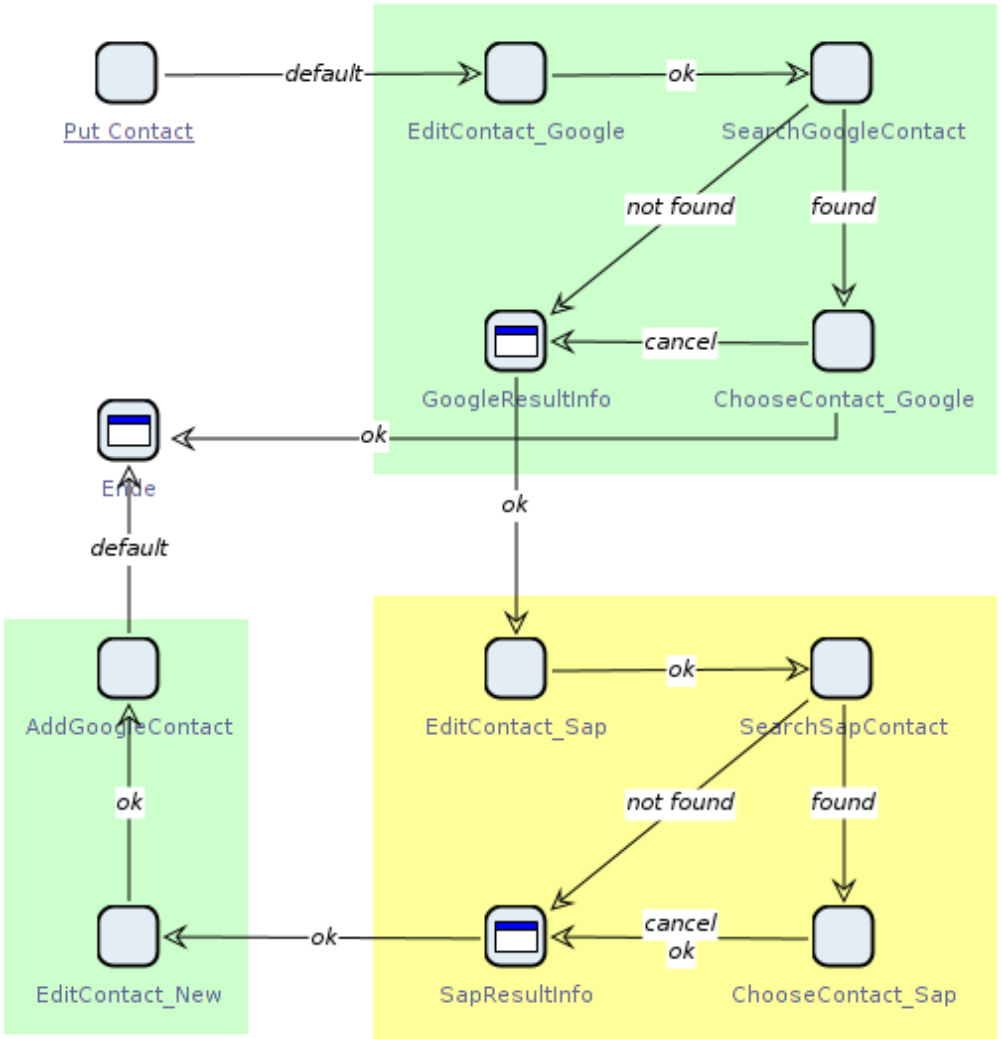
2 Das Projekt

3 SAP Connector

4 SIB-Programmierung

5 jABC-Modell

Modell und Anwendung



Vielen Dank für Ihre Aufmerksamkeit!

1. Einführung
2. Das Projekt
3. SAP Connector
4. SIB-Programmierung
5. jABC-Modell