

Ringvorlesung **Schwerpunkte der Informatik**

Modellbasierter Software-Entwurf am Beispiel von Geschäftsprozessen

15. Mai 2003

W. Reisig, A. Martens



1. Modellbasierter Software- Entwurf

Nur 5% der (industriellen) Softwareentwickler benutzen eine
Modellierungssprache!

die benutzen zu 80% UML

die „*Unified Modelling Language*“

ein Industrie-Standard,

ein riesiger Sprung gegenüber dem üblichen ad-hoc-codieren,

dennoch:

es gibt ein Leben neben und nach UML!

... insbesondere für spezielle Anwendungsbereiche

Grady Booch *Mit-Entwickler von UML:*



kann eine Person bauen
minimaler Entwurf
einfacher Prozeß
einfache Werkzeuge

ein Team ist nötig

Entwurf (durch Architekt)
Analyse (durch Statiker)
Genehmigung
Realisierungsprozeß
aufwändige Werkzeuge

3

Was soll man modellieren?

Michael Jackson, 70er Jahre(!):

nicht nur *what must be built* „the machine“
sondern auch *what is given* „the problem domain“
nur so werden kausale Zusammenhänge sichtbar
... und die machen ein System aus!

4

ein Zitat von Grady Booch, 12/2002

... the following technologies are on my watch list
for the next 3-5 years,

in terms of ... building better software faster:

- platforms
- component-based development and web services
- model-driven development MDD
- reusable assets/architectural patterns
- agents
- aspect-oriented programming

5

Grady Booch: *Wo MDD helfen kann:*

- Semantische Lücke zwischen Anwendungen und Plattformen schließen
- verschiedene Sichten auf ein System verweben
- legacy integrieren, vorh. Software wiederverwenden
- elevate models as to a first class citizenship
- models: a peer of traditional text languages (and potentially a master)

6

Grady Booch: *Was UML unterstützt:*

- visualisieren
- informal spezifizieren
- tentativ konstruieren
- dokumentieren

7

Grady Booch: *Was UML nicht unterstützt:*

Umgang mit verifizierten Modellen:

- verifizieren
- analysieren
- vertikal abstrahieren/verfeinern
- horizontal zerlegen/komponieren

8

Grady Booch: *Zukunft von UML*

Precision is especially important for the next phase in the growth of the UML,
namely as a language for Model-Driven Development

... dort trifft man alte Bekannte:

Petri-Netze

Prozeßalgebren

Abstract-State Machines

Temporale Logik

Z

Larch

...

9

2. Thema der heutigen Vorlesung: *Geschäftsprozesse*

- modellbasiert entwerfen
- im Hinblick auf Web Services

betrifft 2 der Technologien aus Brooch's Liste:

- component-based development and web services
- model-driven development MDD

10

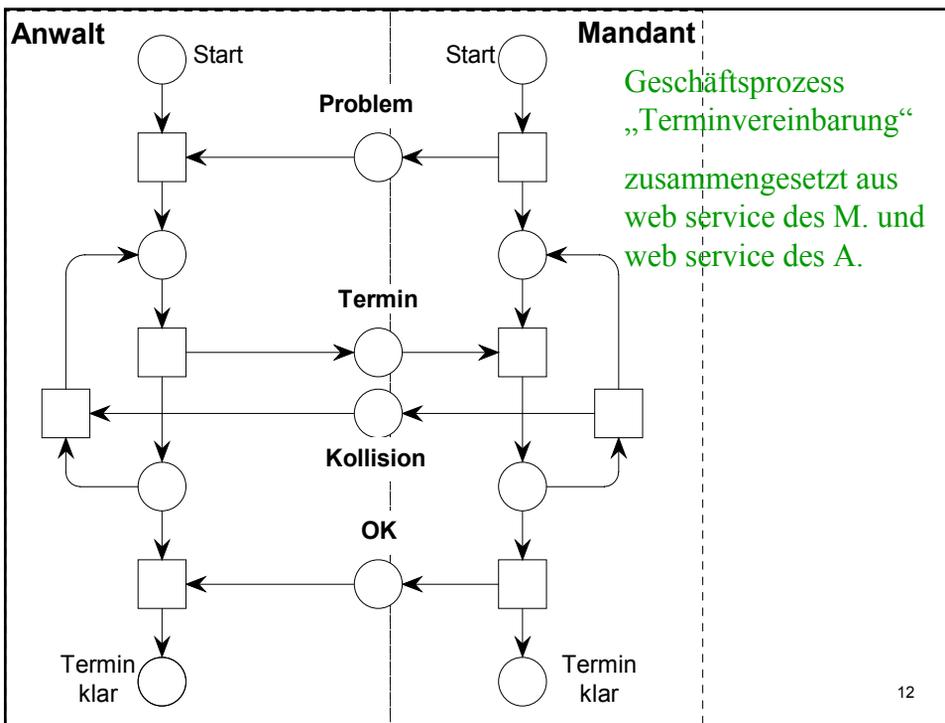
Beispiele für Geschäftsprozesse

Firmenfeier

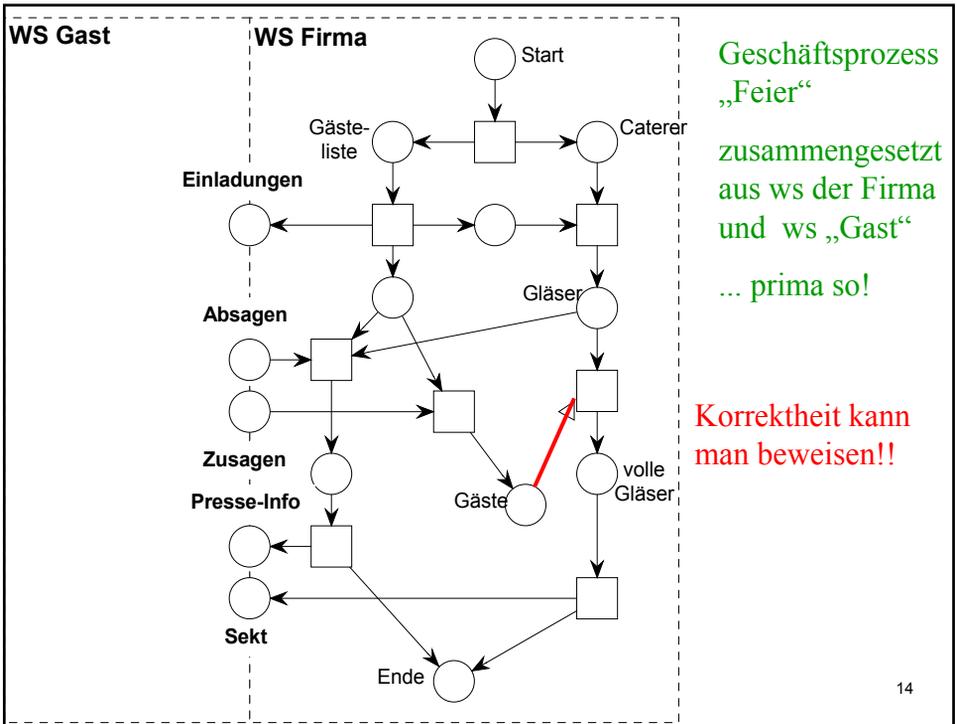
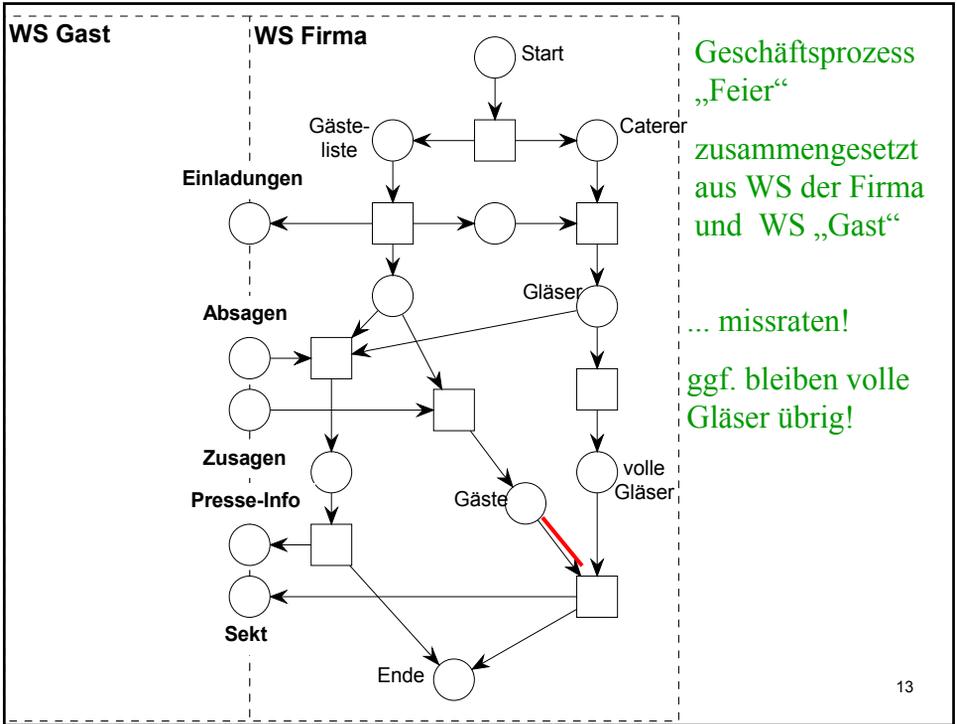
Terminplanung

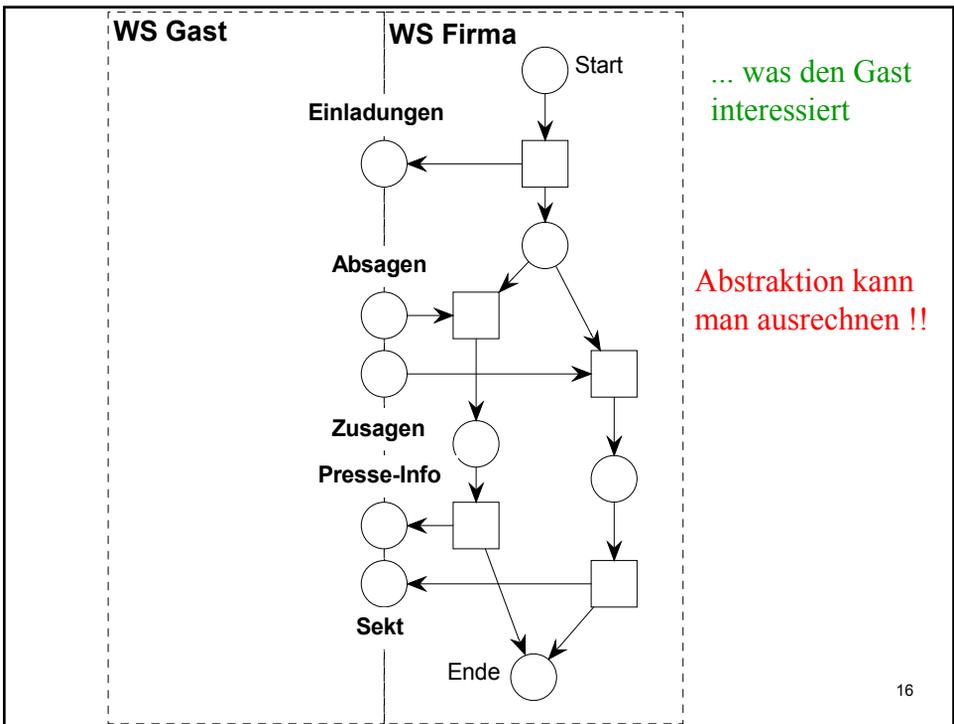
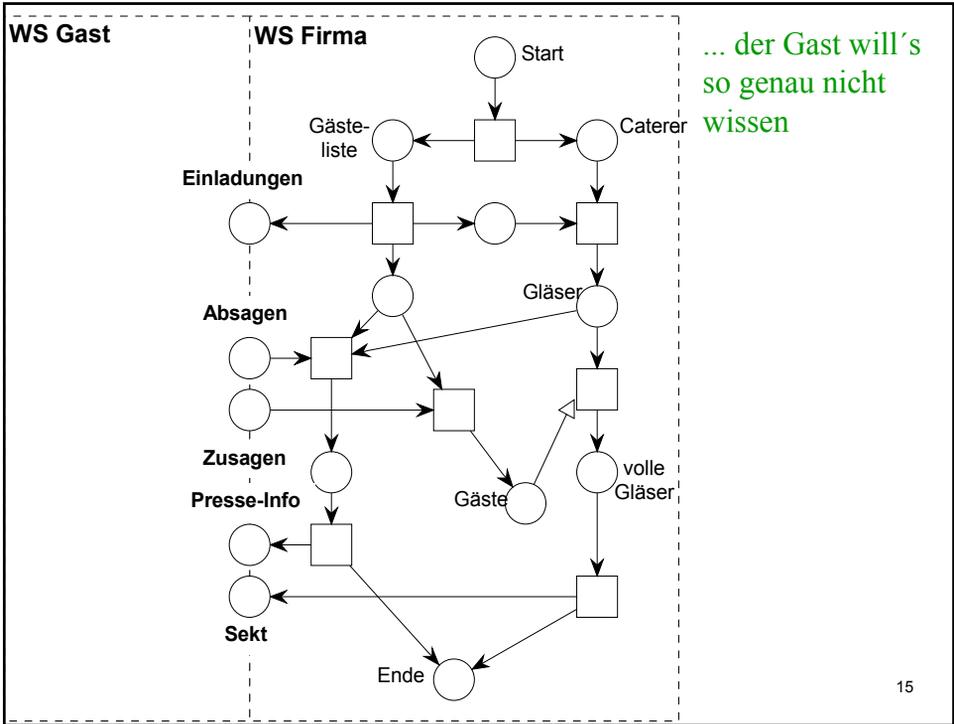
Reisebuchung

11



12





Was ist ein Geschäftsprozeß?

Ein Geschäftsprozess ist eine Folge von *Aktivitäten*. Sie

- stehen in logischem Zusammenhang,
- sind inhaltlich abgeschlossen.

Sie werden ausgeführt

- durch Menschen und/oder Maschinen
- mit Hilfe von Ressourcen
- und eingehenden Informationen
- auf ein Unternehmensziel hin.

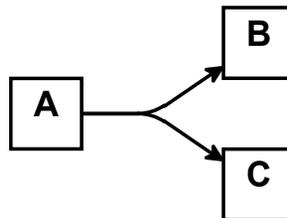
17

Wozu formal modellieren?

Beispiel:

ein eingehender Brief wird im Sekretariat registriert (Aktion A).

Danach wird er er von Herrn Meier (Aktion B) oder Frau Müller (Aktion C) bearbeitet.



kurz: Erst A, dann B oder C

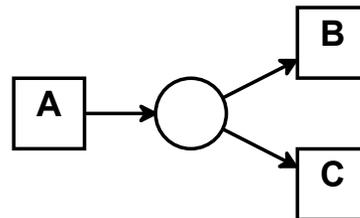
das ist reichlich unklar!

18

Variante I: nichtdeterminiert

Entscheidung ist offen
Konflikt zwischen B und C
zwei Verhalten möglich

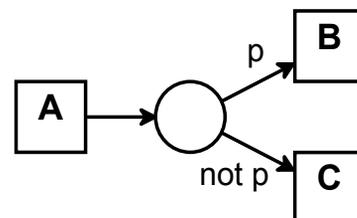
Allgemeinstes Verhalten
... noch genauer zu klären!



19

Variante II: determiniert

Kriterium p entscheidet
Aufgaben zwischen B und C geteilt
genau ein Verhalten



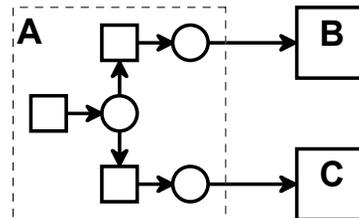
20

Variante III: interne Entscheidung

A entscheidet

Konflikt innerhalb von A

B und C haben keinen Einfluß

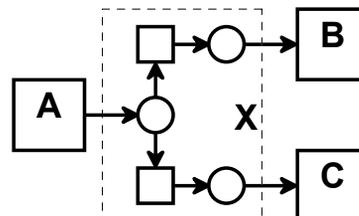


21

Variante IV: externe Entscheidung

eine zusätzliche Komponente X entscheidet.

A, B und C haben keinen Einfluß



22

3. Verteilte Geschäftsprozesse und ihre Implementierung

sie sind besonders wichtig

(d.h. interessant / konzeptuell und technisch anspruchsvoll / zukunftssträftig):

- Mehrere Organisationen kooperieren
- Das verlangt spezielle Modellierungsmethoden und neue Implementierungstechniken

25

Anforderungen an die Modellierung

- Verteilte Organisationen
- Dezentrale Kontrollstrukturen
- Unabhängige Organisationen
- Lokale Autonomie der Modellierung
- Konkurrierende Organisationen
- Wahrung von Geschäftsgeheimnissen
- Zeitlich begrenzte Allianzen
- Flexibilität

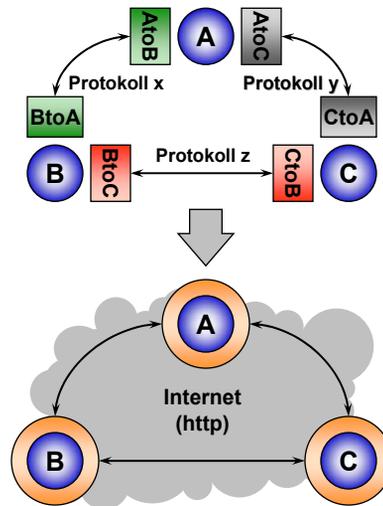
Implementierung: mit Web-Services

Ein *Web Service* ist ein Anwendungsmodul im Web.

- inhaltlich abgeschlossen,
- ggf. selbst komponiert
- selbsterklärend

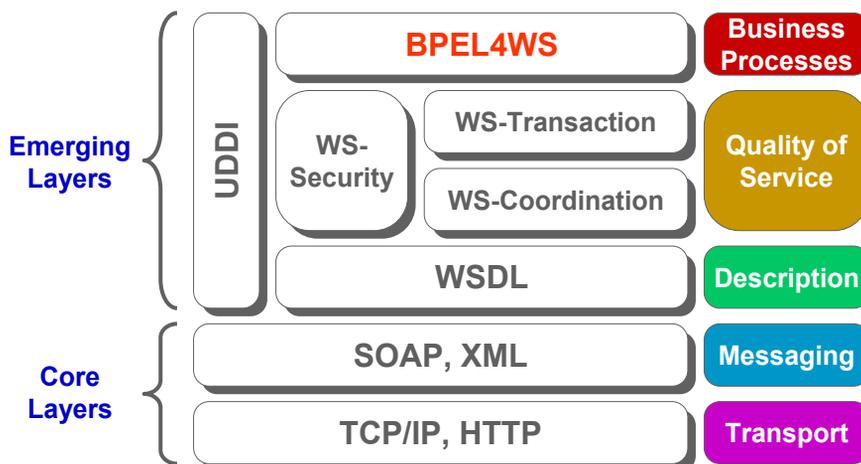
Ein Web-Service

- kapselt die Funktionalität einer Anwendung,
- verfügt über eine standardisierte Schnittstelle,
- ist unabhängig von Hardware und Software-Plattformen.



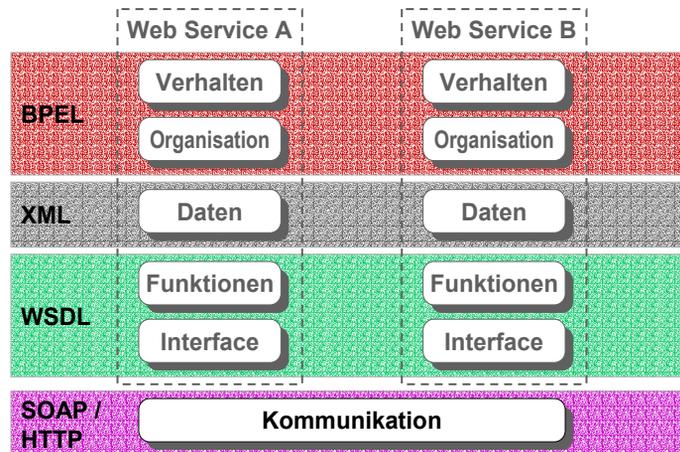
27

Web Service Technology Stack



28

Verteilte GP mit Web Services



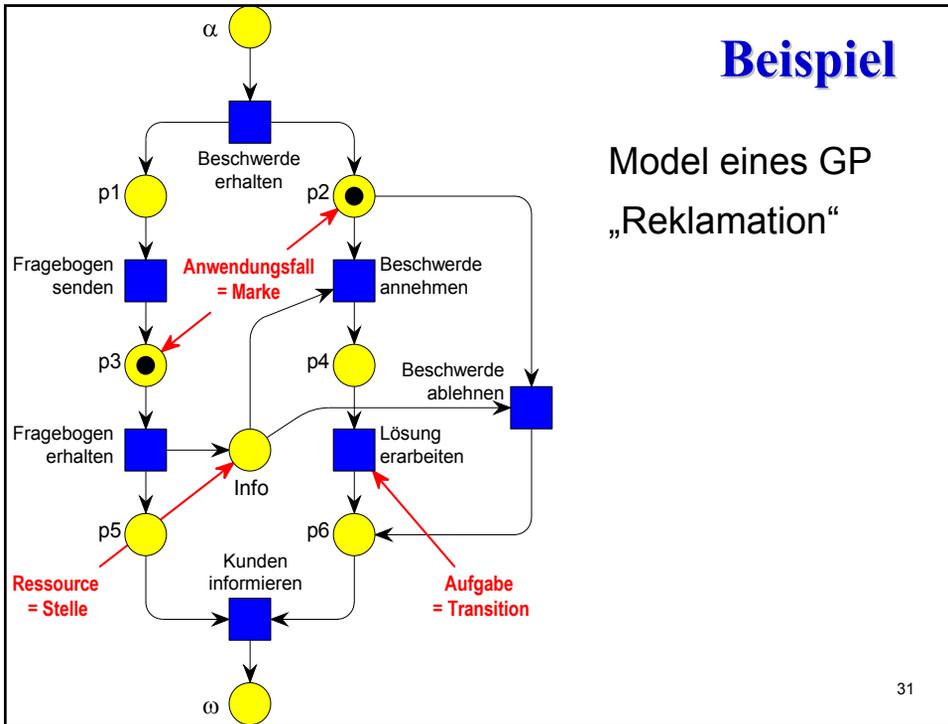
29

4. Modulare Geschäftsprozesse

30

Beispiel

Model eines GP
„Reklamation“



Reklamation mit Interface

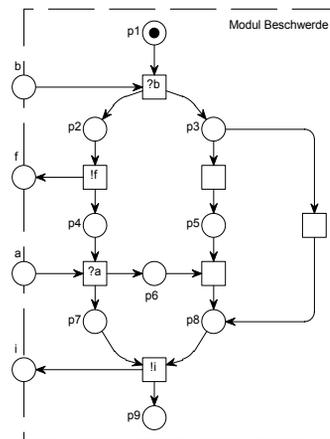
Workflow-Modul:

Workflow-Netz

- Petrinetz
- definierte Anfangs- / Endstelle
- stark zusammenhängend

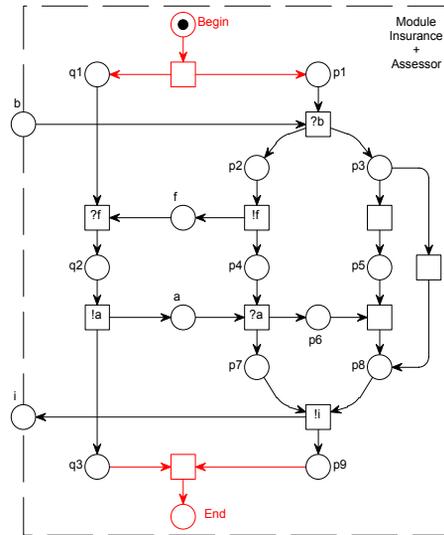
Interface

- Input-Kanäle (z.B. *a*)
- Output-Kanäle (z.B. *f*)



Komposition zweier Modulen

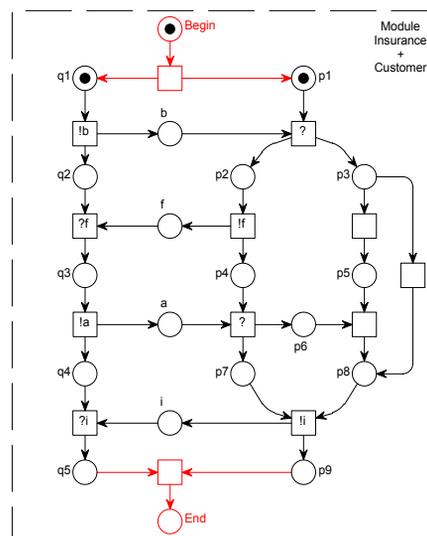
- **Gegeben** zwei Module:
 - Insurance
 - Assessor
 - Syntaktisch **Kompatibilität**:
 - Disjunkte Prozesse
 - Passende Schnittstellen
 - **Komposition** :
 - Fusion der Schnittstellen
 - Export offener Schnittstellen
 - Initialisierung / Terminierung
- Neuer (komponierter) **Web Service**



33

vollständig passende Modulen

- **Gegeben** zwei Module:
 - Insurance
 - Customer
 - Syntaktisch **Kompatibilität**:
 - vollständig passende Schnittstellen
 - **Komposition**:
 - Jedes Modul ist eine **Umgebung** für das andere
- Neuer (verteilter) **Geschäftsprozess**



34

Wichtige Fragestellungen

- Ist ein gegebener Modul überhaupt „bedienbar“ ?
- ggf. nicht „vollständig bedienbar“
- sind zwei Module äquivalent?
- ist ein Modul Implementierung eines anderen?
- gibt es eine „abstrakteste Variante“ bzgl. gegebener Eigenschaften

alle diese Fragen sind konstruktiv lösbar!

... anstehende Dissertation von Axel Martens

... das sind *prinzipiell* neue Fragestellungen

35

Worüber wir sprechen können müssen

klassisch:

Gegenstände:

- Zeichenketten
- Zahlen
- Textfelder

Operationen:

- rechnen
- vergleichen
- suchen
- sortieren

Geschäftsprozeß:

Gegenstände:

- Prozeß
- Partner
- Kunde
- Frage/Antwort
- Sitzung
- Partner-Profil
- Dialog
- Transaktion

Operationen:

- weiterreichen
- vermitteln
- abonnieren
- Rechnung stellen
- authentifizieren
- Ressourcen reservieren
- protokollieren
- aktualisieren

Programmiersprachen

Spezifikationssprachen

36

Wie es weitergeht

am Lehrstuhl: Projekt zu BPEL4WS

- Semantik mit PN und ASM
- Verifikationstechniken
- visual BPEL
- Analysewerkzeuge
- Referenzmodelle
- Tutorial

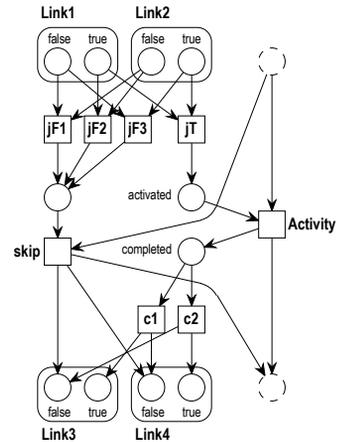
37

vielen Dank für Ihr Interesse

38

BPEL

```
<flow>
  <link name="Link1"/>
  <link name="Link2"/>
  <link name="Link3"/>
  <link name="Link4"/>
  ...
  <empty name="Activity"
    joinCondition="Link1 AND Link2"
    suppressJoinFailure="no">
    <target linkName="Link1"/>
    <target linkName="Link2"/>
    <source linkName="Link3"
      transitionCondition="OK" />
    <source linkName="Link4"
      transitionCondition="not OK" />
  </empty>
</flow>
```



39

Analyse von Modelle

ToDo !

40

Dynamik der Modelle

ToDo !

41

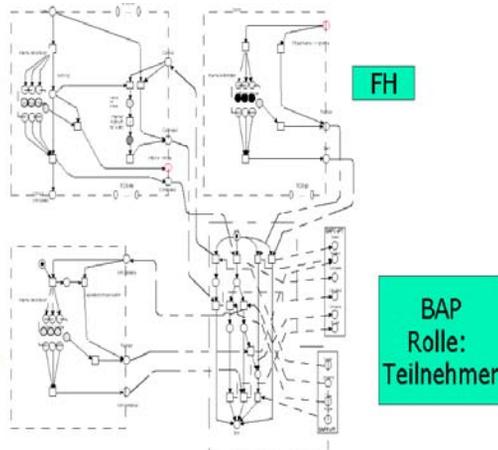
Ausblick

- Einführendes Beispiel ✓
- Geschäftsprozesse ✓
- Verteilte Geschäftsprozesse ✓
- Modulare Geschäftsprozesse ✓
- Geschäftsprozesse mit BPEL4WS ✓
- Ausblick ⇄
 - Erweiterbarkeit
 - Werkzeugunterstützung
 - GRID-Technologien

42

Erweiterbarkeit

- BPEL-Box = saubere Trennung von
 - ♦ Prozessaktivität, **PMA**
 - ♦ Fehlerbehandlung / Kompensation
 - ♦ Transaktionsprotokoll
- Austauschbarkeit einzelner Komponenten **GH**



43

Werkzeugunterstützung

- Handhabung
 - ♦ Modularer Prototyp auf JAVA-Basis, d.h. erweiterbar & integrierbar
 - Entwicklung zum Plugin für Eclipse **ToDo !**

44

Business-GRID

- Zukunftsorientiert
 - Word Wide Web => Great Global GRID
 - Kompatibilität & Vergleichbarkeit notwendig für Business-GRID **ToDo!**

45

Wo kommen Web Services vor?

„Eine Versicherung ist ein Web Service mit Außendienst“

Das Finanzamt sollte Web Services betreiben

Die Fa. SAP hat ihr ERP (Enterprise Resource Planning) System R3 auf Web Services umgestellt

IBM ist Vorreiter bei der Middleware für WS

46

Interessante Fragen an einen Web Service

Ist er bedienbar?

Hat er eine angemessene abstrakte Darstellung?

Passt er zu einem gegebenen anderen WS?

Simuliert er einen gegebenen anderen WS?

... wenigstes in einer bestimmten Hinsicht?

Unser Beitrag:

Verfahren, um solche Fragen zu beantworten

... gemacht hat es vor allem Herr Martens

47

Spannende Fragen

wissenschaftlich interessant

wirtschaftlich wichtig

auf weitere Zusammenarbeit!

... bei BPEL4WS

Business Process Execution Language for Web Services

vielen Dank

48

Einführendes Beispiel

- Struktur
 - ♦ Agenten, Module, Netze
 - Geschäftsprozess
- Verhalten
 - ♦ Zustände, Übergänge, Abläufe
 - intuitiver Begriff der Korrektheit
- Analyse
 - ♦ Konflikt, Konfusion, Verklemmung
 - Bedarf an methodischer Unterstützung

ToDo !

49

Gliederung

- Einführendes Beispiel ✓
- Geschäftsprozesse ⇐
 - ♦ Aspekte von Geschäftsprozessen
 - ♦ Modellierung von Geschäftsprozessen
 - ♦ Vernünftige Prozessmodelle
- Verteilte Geschäftsprozesse
- Modulare Geschäftsprozesse
- Geschäftsprozesse mit BPEL4WS
- Ausblick

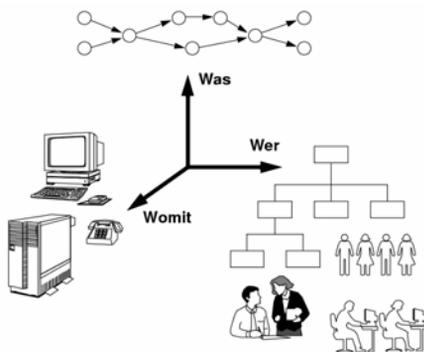
50

allgemein: Was ist ein Geschäftsprozeß?

- Ein Geschäftsprozess
 - ist ein Vorgang in Wirtschaftseinheiten,
 - kann funktions-, hierarchie- und standort-übergreifend ablaufen.
- Ein Geschäftsprozess zeichnet sich aus durch:
 - einen definierten Anfang/ definiertes Ende,
 - erforderliche Eingaben (z.B. Kundenwünsche),
 - produzierte Ergebnisse (z.B. Befriedigung der Kundenwünsche).

51

Was ist Geschäftsprozesses?



Ein Geschäftsprozess ist eine Folge von **Aktivitäten**, die:

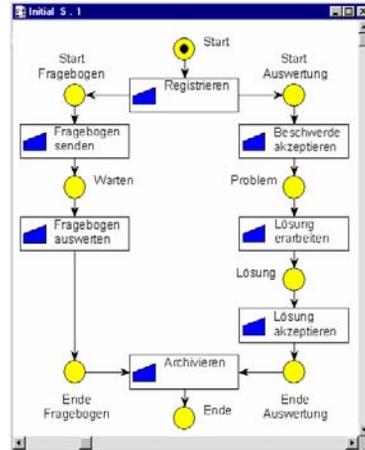
- im logischen Zusammenhang stehen,
- inhaltlich abgeschlossen sind,
- mit Hilfe von Ressourcen
- und eingehenden Informationen
- durch Menschen und/oder Maschinen
- auf ein Unternehmensziel hin ausgeführt werden.

52

Vernünftige Prozessmodelle

Ein Workflow-Netz heißt **sound** (vernünftig, gesund), wenn:

- jeder angefangene Prozess zu einem Ende kommen kann,
 - das Netz am Ende aufgeräumt ist
 - und jede Transition erreicht werden kann
- + Effektive Analyse-Verfahren
+ Syntaktische Richtlinien



53

Wichtige Eigenschaften von Prozeßmodellen

Sicherheitseigenschaften:

Kein Objekt

- wird vergessen, bleibt unberücksichtigt,
- wird im falschen Kontext betrachtet.

Jede Aufgabe

- ist relevant,
- ist durchführbar

Lebendigkeitseigenschaften:

Jede angefangene Aufgabe

- kann erfolgreich gelöst oder kompensiert werden.
- Jeder angefangene Prozess
- kann beendet werden.

54

Modellbasierter Software-Entwurf für Web-basierte Geschäftsprozesse

Für Vortrag bei IBM

Seit Jahrzehnten wird aus der Wissenschaft heraus vorgeschlagen, den Entwurf von Software systematisch auf Modellen aufzubauen. Tatsächlich wächst der Anteil modellbasierter Software seit einigen Jahren.

Modelle von Software zur Ausführung Web-basierter Geschäfts-prozesse können Fehler und ungünstige Entwurfsentscheidungen aufzeigen und wichtige Eigenschaften garantieren, bevor die Software selbst codiert wird. Wir zeigen, wie man solche Modelle erstellt und wie man wichtige Konzepte von Geschäfts-prozessen, insbesondere Bedienbarkeit, Komponierbarkeit und Äquivalenz, in solchen Modellen formuliert und verifiziert. Schließlich zeigen wir, wie solche Modelle in der von IBM und Microsoft gemeinsam vorgeschlagenen Sprache BPEL4WS (Business Process Execution Language for Web Services) implementiert werden.

55

Literatur

Grady Booch: *Growing the UML*, Software and Systems Modeling Vol 1 Nr 2 pp157-160 (2002)

Michael Jackson: *Some Basic Tenets of Description*, Software and Systems Modeling Vol 1 Nr 1 pp 5 - 9 (2002)

56