



Assoziation geschäftinterner Prozesse mit B2Bi Protokollen

Vortrag von Alexander Sack und René Grohmann



Gliederung

1. Motivation & Einleitung
2. Process Interconnection
3. Kooperative Prozesse
4. Generic Process Metamodel
5. Prozessabstraktion
6. Fazit
7. Literaturhinweise



Vorraussetzungen

- Mit Web Services ergeben sich neue Moeglichkeiten zum Anbieten von Diensten im Internet
- Der Web Service Standard bietet keine konkreten Loesungsvorschlaege fuer Implementation von Prozess Services.
- Interne Geschaeftsprozesse sind
 - Fein granular modelliert
 - Individuell
 - wichtiger Wettbewerbsfaktor / Differenzierungsmerkmal
 - intern ~ geheim/vertraulich
- Interne Prozesse und Services sind i.d.R. **nicht**:
 - Offen / einfach / transparent in der Nutzung
 - Generisch, dynamisch zu adaptieren
 - Fuer **B2Bi** Protokolle massgeschneidert.



Motivation

- Interne Geschäftsprozesse müssen mit B2Bi Protokollen assoziiert werden
- Forschungsbedarf hinsichtlich der kurzfristig/dynamischen Bildung und Bindung von kooperativen Multi-Enterprise Prozessen (MEPs)
- Um an Virtual Enterprises(VE) oder MEPs mitzuwirken, bedarf es automatisierter Möglichkeiten Partner zu finden und deren Prozesse in die eigene Wertschoepfungskette zu integrieren.
- Methoden der Anbindung von BP an MEP/VEP müssen sowohl Anforderungen an Vertraulichkeit als auch an Transparenz berücksichtigen
 - Grad an Transparenz sollte fuer unterschiedliche Geschäftspartner und Geschäftsvorfälle flexibel, zur Laufzeit (dynamisch) bestimmbar sein.
- Aspekte, wie generelle Dynamik und Nichtdeterminismus, werden von aktuellen Prozessmodellen nicht hinreichend berücksichtigt.



Aktuelle Forschungsansätze

- Process Interconnection (dynamisch)
 - Sichtbarkeits- und Transparenzmechanismen von Process Services
 - Kontrollierte Offenlegung von Prozesskontexten fuer MEPs
 - Process Service Matching zum (automatisierten) Auffinden eines passenden Prozess Services
- Gleichwertige Betrachtung von internen und B2Bi Prozessen unter dem Aspekt der kooperativen Prozesse
 - Beruecksichtigung der dynamischen / nicht-deterministischen Natur
 - Ein Object-Orientiertes Metamodell fuer "IE Cooperative Processes" based on Web-Services
- Prozessabstraktion zum Mapping auf B2Bi Protokolle
 - Grundlage fuer dynamische Adaption von Prozessen sowie Ihre statische/dynamische Erweiterbarkeit
 - Abbildung (Mapping) von internen Prozessen auf externe Prozesse



Gliederung

1. Motivation & Einleitung
2. **Process Interconnection**
3. Kooperative Prozesse
4. Generic Process Metamodel
5. Prozessabstraktion
6. Fazit
7. Literaturhinweise



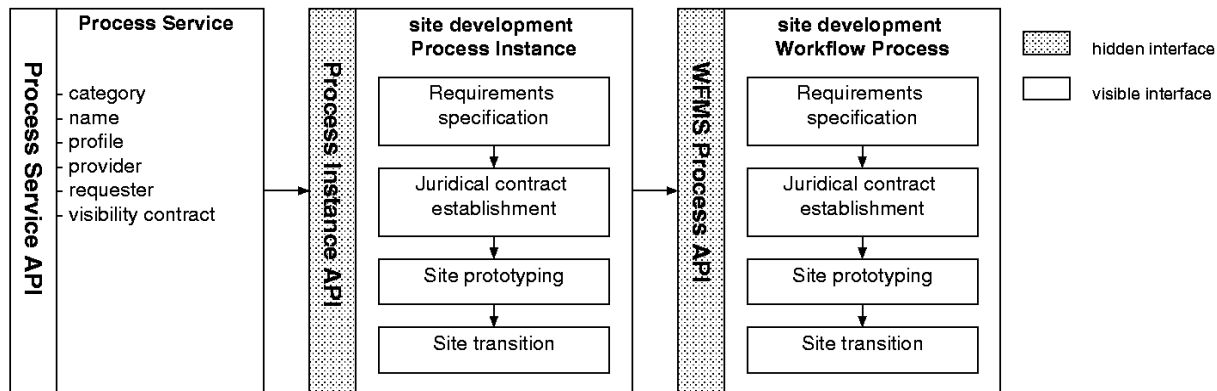
Process Interconnection

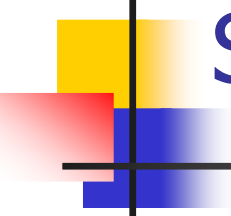
- Anforderungen
 - die Sichtbarkeit der Prozess Service (PS) APIs sollte flexibel
 - die Sichtbarkeit von PS-APIs muss zur Nutzungs-/Laufzeit eines Services aushandelbar sein
 - Sichtbarkeitsabstufungen
 - Feinere Unterteilung als in white-box / black-box Service APIs
 - Dienste sollten nach Ihrer Funktionalitaet und nach nichtfunktionalen Attributen "gematched" und gefunden werden koennen

Process Interconnection

- Ansatz:

- WFMS Prozess API in eine Process Instance API wrappen und Teile dieser in einem weiteren Schritt unter Beachtung von individuellen (dynamischen) Interconnection Contracts als Process Service API sichtbar machen



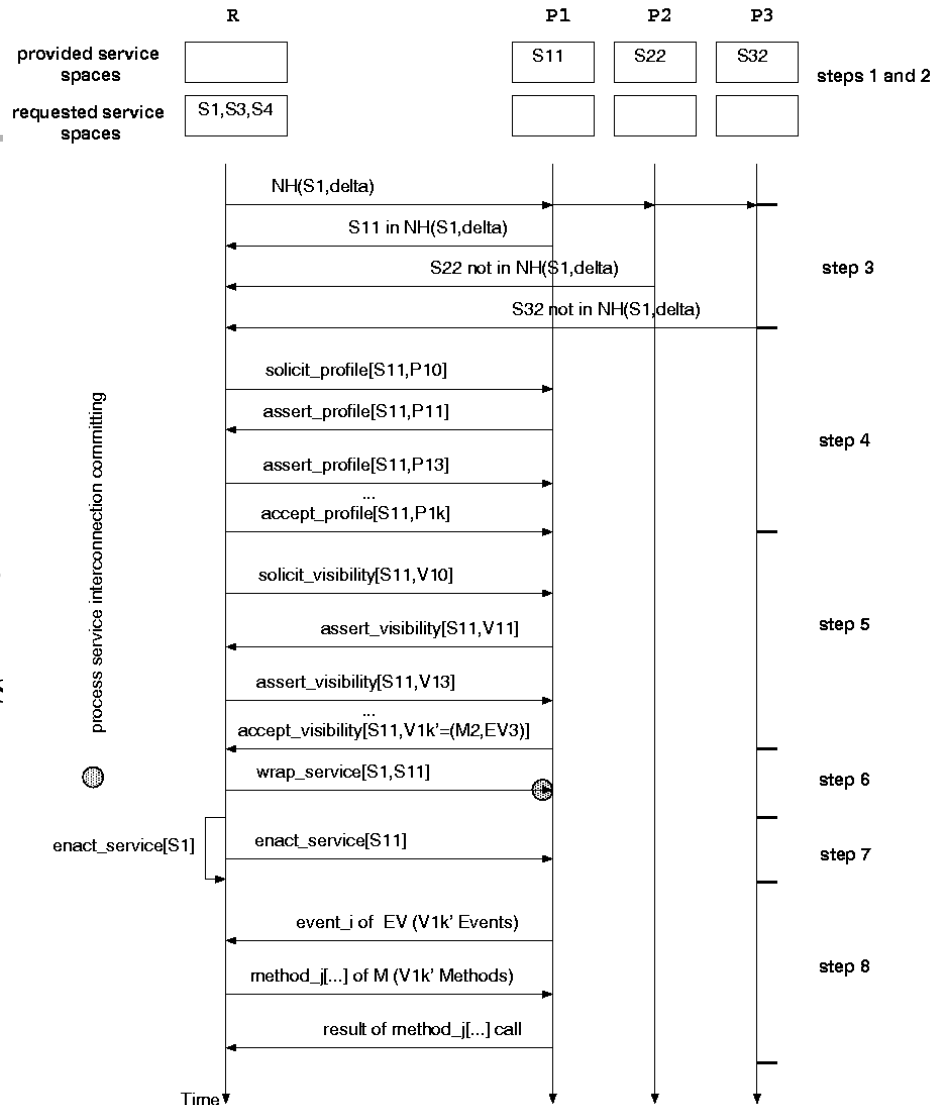


Process Service API Contracts & Service Matching

- Process Service API Contracts:
 - Sichtbarkeitsklassen fuer Methoden & Events der Process Instance API
 - totale Ordnungen (\leq)
 - Sichtbarkeitsvertraege (Visibility Contracts)
 - Menge aus Sichtbarkeitsklassen
 - Interconnection Contracts
 - Menge aus Sichtbarkeitsvertraegen, als genereller Interconnection Contract mit Partnern
- Service Matching
 - Profiling von Process Services APIs
 - Distance & neighbourhood Functions

Ein Process Interconnection Algorithmus

- Step 1 u. 2
 - Erzeugen, publizieren eines Dienstes
- Step 3
 - Finden eines Dienstes
- Step 4
 - Validieren eines Dienstes
- Step 5
 - Aushandeln des Contracts
- Step 6
 - Individuelle Service API erzeugen
- Step 7/8 Servicenutzung





Gliederung

1. Motivation & Einleitung
2. Process Interconnection
3. **Kooperative Prozesse**
4. Generic Process Metamodel
5. Prozessabstraktion
6. Fazit
7. Literaturhinweise



Kooperative Prozesse – Idee

- Interconnection, wie vorgestellt, löst nur einen Teil der vorhandenen Probleme
- Gleichgestellte Betrachtung der internen und externen Unternehmensprozesse als **kooperative Prozesse**
- **Inter-Enterprise** Prozesse sind Spezialfaelle von kooperativen Prozessen mit speziellen Anforderungen
 - Vertrauenswuerdigkeit
 - Transparenz



Kooperative Prozesse (Eigenschaften)

- Kooperative Prozesse in der Regel **dynamisch**
 - unerwartete Vorfälle (Events)
 - unerwartete Ausnahmen (Exceptions)
 - unerwartete Aufgaben
 - Unvorhersehbare Situationen
- Kooperative Prozesse sind **nichtdeterministisch**
 - unvorhersehbare Situationen
 - unvorhersehbare Prozesse



Kooperative Prozesse (Herausforderungen)

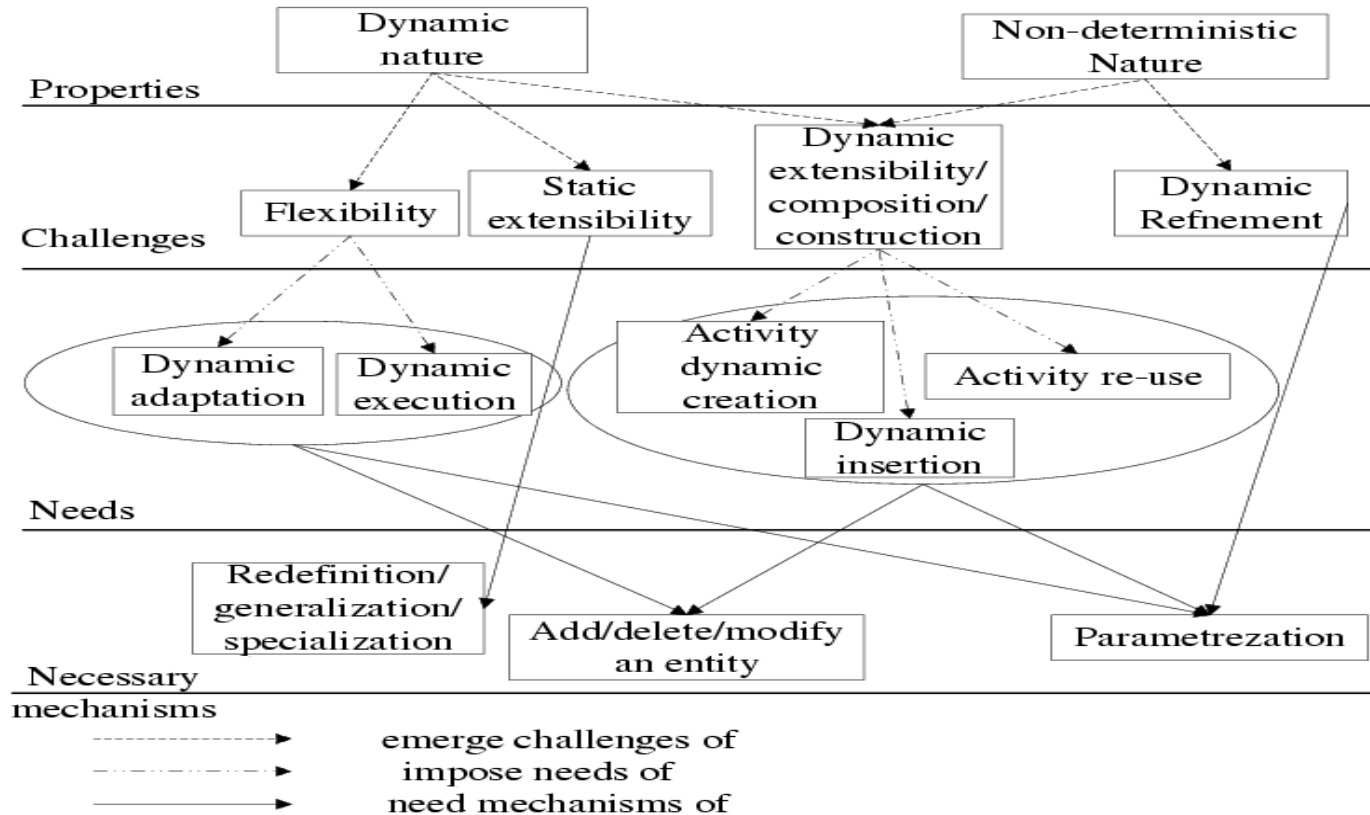
- **Flexibility: dynamische Flexibilität**
 - dynamische Anpassung an unerwartete bzw. Ausnahme Situationen
- **Extensibility: dynamische Erweiterbarkeit**
 - dynamisches Einfügen von Aktivitäten, in unerwarteten Situationen
 - Ergänzung der Prozessdefinition um unerwartete Aufgaben
- **Refinement: dynamische Verfeinerung**
 - Dynamische Vervollständigung der Definition einer Prozess Einheit (Entity)
- **Extensibility II: statische Erweiterbarkeit**
 - Erweiterung der kooperativen Prozessmodel Struktur
 - Weiterentwicklung neuer Entitäten (Aktivität, Data Flow, Control Flow)



Kooperative Prozesse (notwendige Mechanismen)

- Dynamische adaption
 - Faehigkeit zur Laufzeit and unvorhergesehene Exceptions oder Events anzupassen
- Dynamic execution
 - Faehigkeit das verhalten nicht behandelbare Exceptions oder Events von Aussen abzufragen, und es so zu beheben.
- Dynamic activity creation
 - Faehigkeit zur Laufzeit neue Aktivitaeten zu erzeugen
 - entweder um Ausnahmesituationen zu begegen
 - oder um die Processdefinition zu erweitern
- dynamic insertion
 - Faehigkeit zur Laufzeit neu erzeugte Aktivitaeten in einen Prozess aufzunehmen
- Activity re-use
 - Faehigkeit zur Laufzeit eine schon einmal benutzte Aktivitaet wiederzuverwenden

Kooperative Prozesse (Uebersicht)





Gliederung

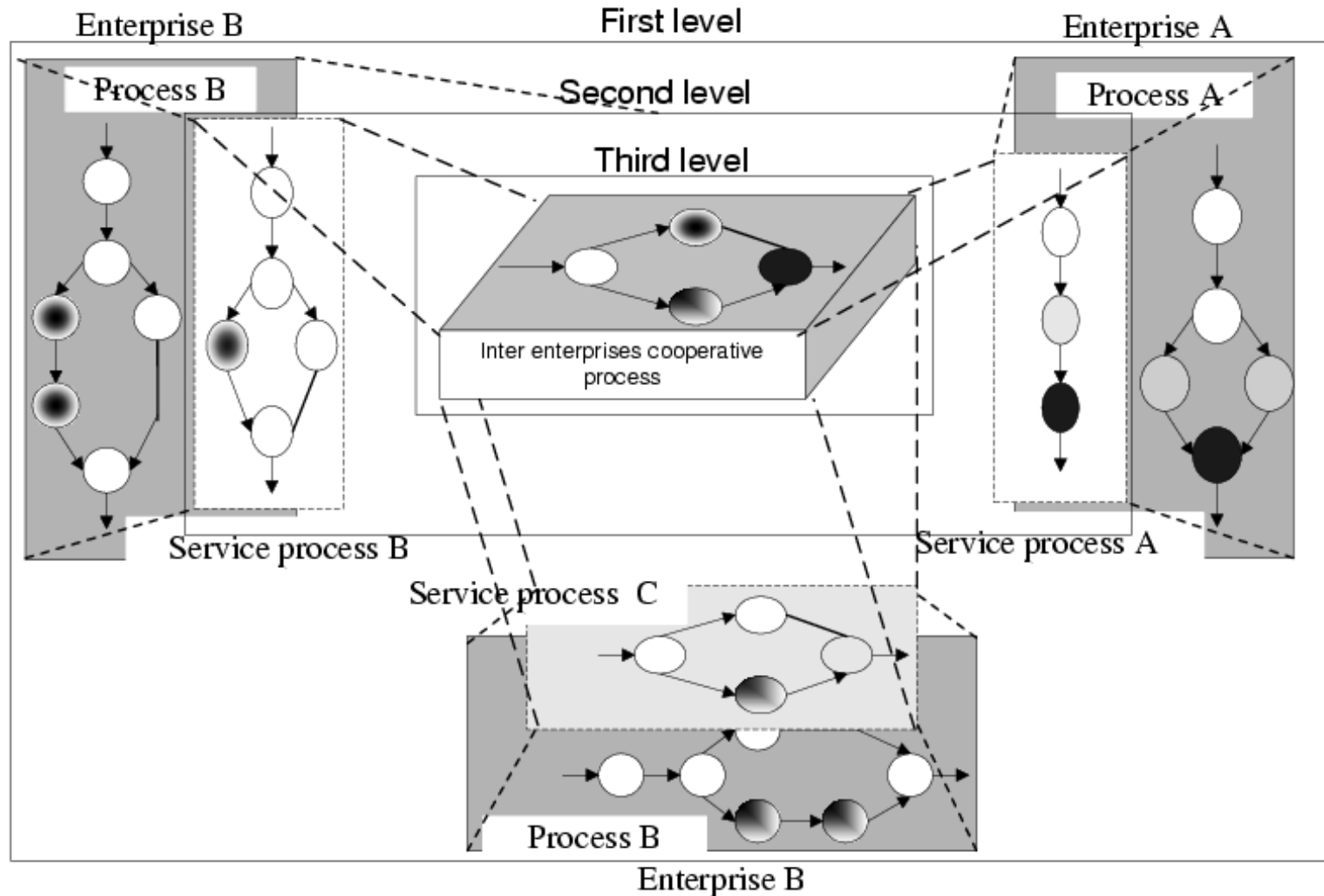
1. Motivation & Einleitung
2. Process Interconnection
3. Kooperative Prozesse
4. **Generic Process Metamodel**
5. Prozessabstraktion
6. Fazit
7. Literaturhinweise



Generic Process Metamodel nach Bhiri et al.

- Objektorientiertes Metamodell zur Beschreibung kooperativer Prozesse und zur Modellierung von kooperativen Inter-Enterprise Prozessen
- Erlaubt die Modellierung von kooperativen Prozessen auf 3 Ebenen
- Technische Umsetzung von Inter-Enterprise Prozessen soll mit Hilfe von Web Services realisiert werden
- Ist Nur ein Ansatz der als Basis für die Entwicklung eines komplette Kooperationssystems dienen kann

Einordnung des Metamodells in einen Gesamtansatz (Bhiri)





Motivation für das Generic Process Metamodel

- In vorhanden WfMS: Separate Definition der Aktivitäten und deren Control und Data flow
 - > Notwendigkeit eines zentralisierten Tools zur Interpretation und Dekodierung der verschiedenen Deklarationen und zur Ausführung des kooperativen Prozesses
- Zentralisierter Server verhindert Flexibilität und Skalierbarkeit von Workflow-Systemen und macht sie für die internetweite Nutzung unbrauchbar
- Erfüllung der bereits vorgestellten Anforderungen (Flexibilität, Dynamik,...)

Der Neue dezentralisierte Ansatz

- Denzentralisierung
 - Control und Data flows werden in den Aktivitäten gekapselt
 - Autonome, Anpassbare und Erweiterbare Entitäten die miteinander interagieren
 - Jede Entität bietet eine Menge von Operationen oder Services an

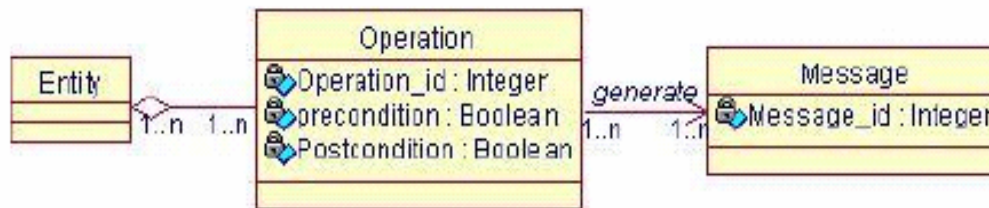
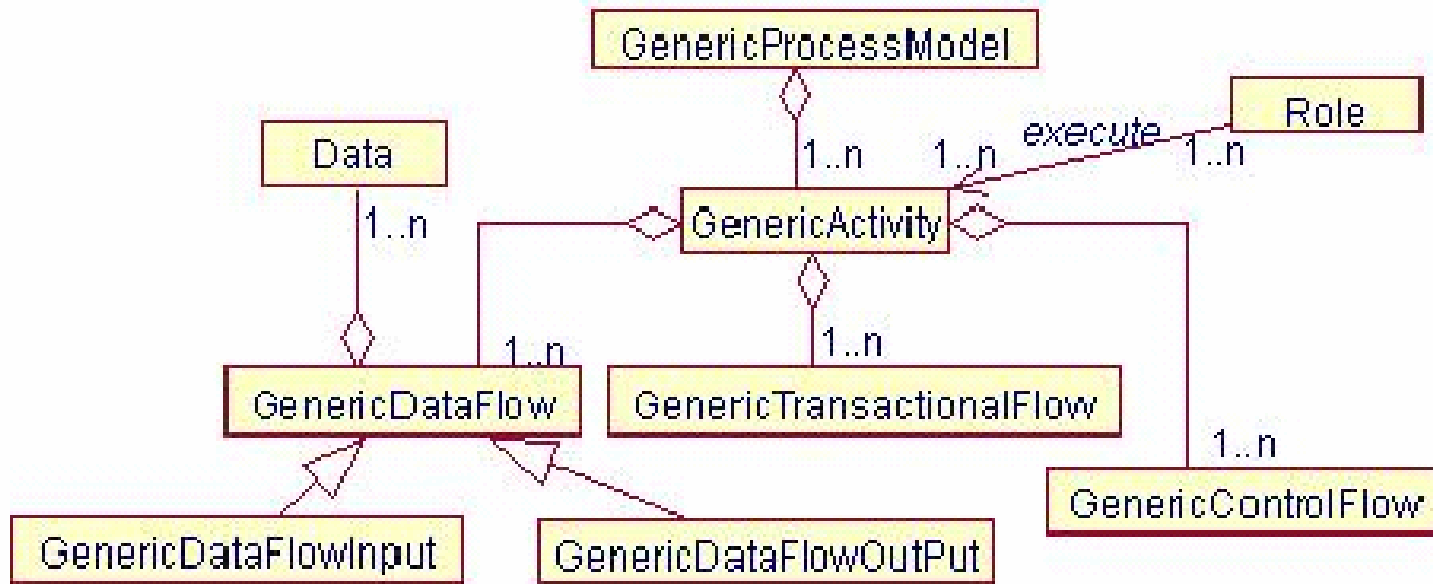


Abbildung zum Generic Process Metamodel





Definitionen des Generic Process Metamodel (1)

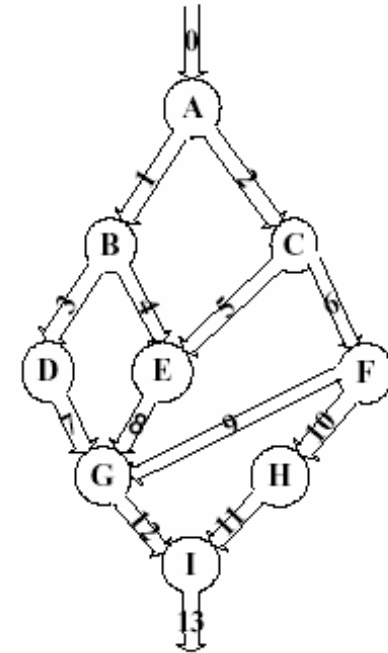
- Data
 - **D** ist die Menge aller Daten
 - Ein Datum d ist wie folgt definiert
 - $d = (A_d, M_d)$ (A_d : Attribute; M_d : Methoden)
- GenericDataFlow
 - Zu unterscheiden Input (gdf_i) und Output (gdf_o) data flow
 - $gdf_i = (A_{gdf_i}, M_{gdf_i}, D_{gdf_i})$ (gdf_o analog)
 - $D_{gdf_i} \cup D_{gdf_o} = \mathbf{D}$
 - **GDF** = **GDF_i** \cup **GDF_o** (Menge aller generic data flows)

Definitionen des Generic Metamodel (2)

- GenericControlFlow
 - $gcf = (A_{gcf} M_{gcf})$
 - **GCF** (Menge aller GenericControlFlows)
- GenericTransactionalFlow
 - $gtf = (A_{gtf} M_{gtf})$
 - **GTF** (Menge aller GenericTransactinalFlows)
- Role
 - $r = (A_r M_r)$
 - **R** (Menge aller Roles)

Definitionen des Generic Metamodel (3)

- GenericActivity
 - $ga = (A_{ga}, M_{ga}, GDF, GCF, GTF, R)$
mit $GDF \subset \mathbf{GDF}$
und $GCF \subset \mathbf{GCF}$
und $GTF \subset \mathbf{GTF}$
und $R \subset \mathbf{R}$
 - \mathbf{GA} (Menge aller GenericActivities)
- GenericProcessModel
 - $gpm = (A_{gpm}, M_{gpm}, GA)$
 - $GA \subset \mathbf{GA}$





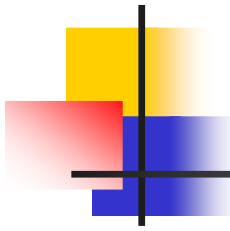
Mechanismen zur Erfüllung der bereits vorgestellten Anforderungen (1)

- Dynamic flexibility, Extensibility and Refinement
 - Durch die dynamische Modifikation von bestimmten Attributen
 - Durch das hinzufügen/entfernen von Subentitäten
 - Reusability durch die Autonomie der Entitäten
 - Parametrisierung von bestimmten Attributen
 - Parametrisierung, Hinzufügen/Entfernen/Modifikation von Objekten



Mechanismen zur Erfüllung der bereits vorgestellten Anforderungen (2)

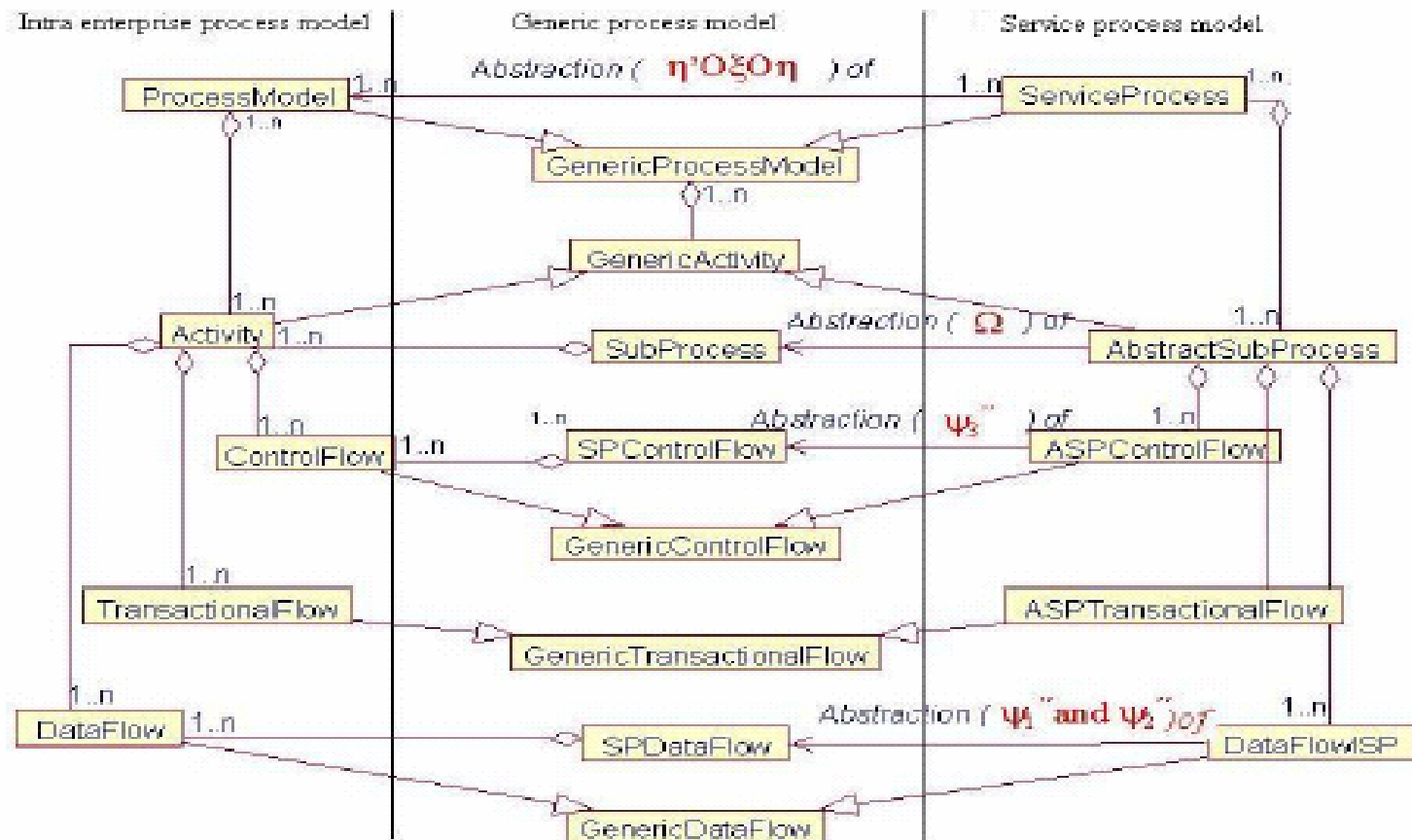
- Static extensibility
 - Hinzufügen neuer Typen um Veränderungen und Weiterentwicklungen des kooperativen Prozesses zu ermöglichen
 - Generalisierung/Redefinition/Spezialisierung
 - Mechanismus: Sub Classing



Intra-Enterprise vs. Service Process Metamodel

- Intra-Enterprise
 - Nur eine Spezialisierung des Generic Metamodel
 - Eine Aktivität: $a = (A_a M_a, DF=DF_i \cup DF_o, CF, TF, R)$
- Service Process
 - Abstraktion eines Intra-enterprise Process um Teile des Prozesses nach Außen hin zu öffnen
 - Problematik: Offenheit vs. Vertraulichkeit
 - Ebenfalls eine Spezialisierung des Generic Process Metamodel
 - Sammlung von abstrakten Subprozesse
 - AbstractSubProcess:
 $Asp = (A_{asp} M_{asp}, ASPDF=ASPDF_i \cup ASPDF_o, ASPCF, ASPTF, R)$

Beziehungen zwischen den Metamodellen





Gliederung

1. Motivation & Einleitung
2. Process Interconnection
3. Kooperative Prozesse
4. Generic Process Metamodel
5. **Prozessabstraktion**
6. Fazit
7. Literaturhinweise



Assoziation interner Geschäftsprozesse mit B2Bi Protokollen durch Prozessabstraktion

- Motivation
 - Aktuell kaum Abstraktionsmechanismen bei der Umsetzung aktueller interner Prozessimplementationen
 - fehlende Flexibilität
 - fehlende Dynamik
- Allgemeine Prinzipien
 - Nutzung von Abstraktionsfunktionen zur Abbildung interner Prozesse auf B2Bi Protokolle
 - Sichtbarkeits u. Strukturabstraktion
 - Late-Binding durch Polymorphie von Prozess Aktivitäten
 - Polymorphie durch Abstraktion durch Trennung von Aktivitätsschnittstelle und Aktivitätsimplementierung



Arten der Prozessabstraktion

- Strukturabstraktion
 - Vergrößerung: Zusammenfassen von Aktivitäten
 - Zerteilung des Prozess Modells in Subprozesse
 - Für Partner sind solche Subprozesse Black Boxes die einen Service repräsentieren. Sie benötigen einen Input und generieren dann einen Output. Sie interagieren mit anderen Subprozessen um den globalen Service Prozess zu bilden.
 - Activity state machine types (ASMT) & subtyping
- Sichtbarkeitsabstraktion
 - Kontrollierte Offenlegung von Prozesskontexten durch Visibility Contracts
 - Verstecken von Attributen und angebotenen Services
 - ASMT



Prozessabstraktion nach Bhiri

- Service Prozess durch Abstraktion von einem internen Geschäftsprozess
- Setzt auf dem Generic Process Metamodel auf
- Strukturabstraktion durch Konzepte wie
 - Partitionierung
 - Abstraktionsfunktionen zum mappen von Subprozessen auf abstrakte Subprozesse
- Sichtbarkeitsabstraktion durch
 - Abstraktionsfunktionen die sowohl Attribute, Methoden als auch die Objekte auf Teilmengen der Gesamtmengen abbilden

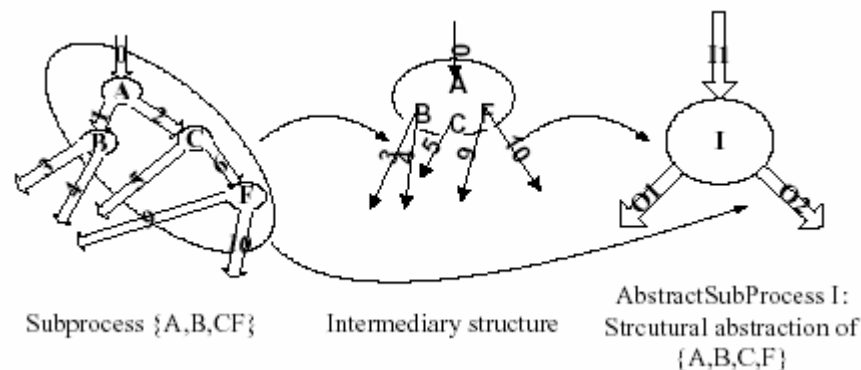


Prozess Partitionierung

- Def. Subprozess
 - Ein Prozess pm' ist ein Subprozess eines Prozesses pm wenn $A' \subseteq A$ (A = Menge der Aktivitäten). $S(pm)$ ist die Menge aller Subprozesse.
- Def. Disjunktion
 - Zwei Prozesse pm und pm' sind disjunktiv
 $A' \cap A = \{\}$
- Def. Prozess Partitionierung
 - $Pr(pm) = \{pm' \mid pm' \in S(pm) \text{ und } \forall pm'' \in Pr(pm), pm' \text{ und } pm'' \text{ sind disjunktiv}\}$
- Def. Komplette Prozess Partitionierung
 - $\bigcup_{pm' \in Pr(pm)} A(pm') = A(pm)$

Abstraktions Funktion: Subprozess -> abstrakten Subprozess

- Die Funktion Ω definiert einen AbstractSubProcess durch
 - Verstecken der internen Links
 - Gruppierung von Outputlinks die dasselbe Ziel (AbstractSubProcess) haben
 - Gruppierung von Inputlinks die von derselben Quelle (AbstractSubProcess) kommen



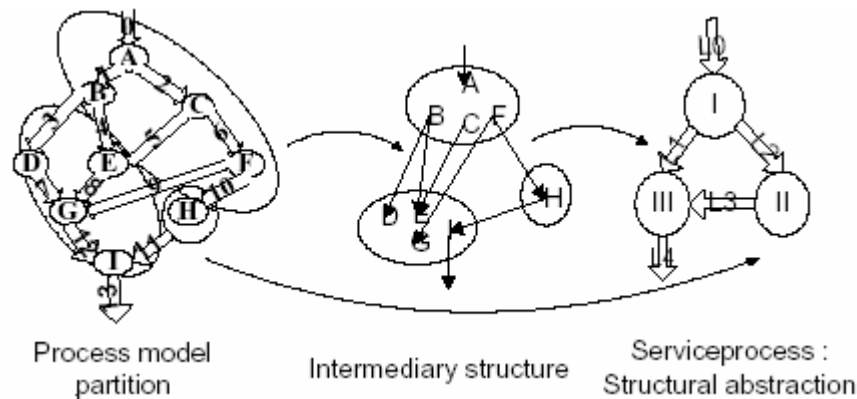
Prozess Strukturabstraktion

- Anwendung der Funktion Ω auf alle Subprozesse

- $\mathcal{E} : \mathbf{PR}(pm) \mapsto \mathbf{SP}$

$$pr(pm) \mapsto sp = (A_{sp}, M_{sp}, SUP)$$

$$SUP(sp) = \bigcup_{pm' \in pr(pm)} \Omega(pm')$$





Sichtbarkeitsabstraktion

- Verstecken von Attributen, Methoden und Services von einem Objekt

- $n: O \mapsto O$

$$o = (A_o, M_o, Ob) \mapsto o' = (A'_o, M'_o, Ob')$$

$$A_o \subseteq A'_o$$

$$M_o \subseteq M'_o \text{ und}$$

für alle $obj' \in Ob' \exists obj \in Ob$ das $obj' = n(obj)$



Abstraktionsfunktion

- Kombination von Sichtbarkeits- und Strukturabstraktion
- Def.: Abstraction function

$$Abstr: PR(pm) \mapsto SP$$

$$pr(pm) \mapsto sp$$

$\exists n \wedge n'$ (2 Sichtbarkeitsabstraktionsfunktionen)

$\wedge \varepsilon$ (eine Strukturabstraktion)

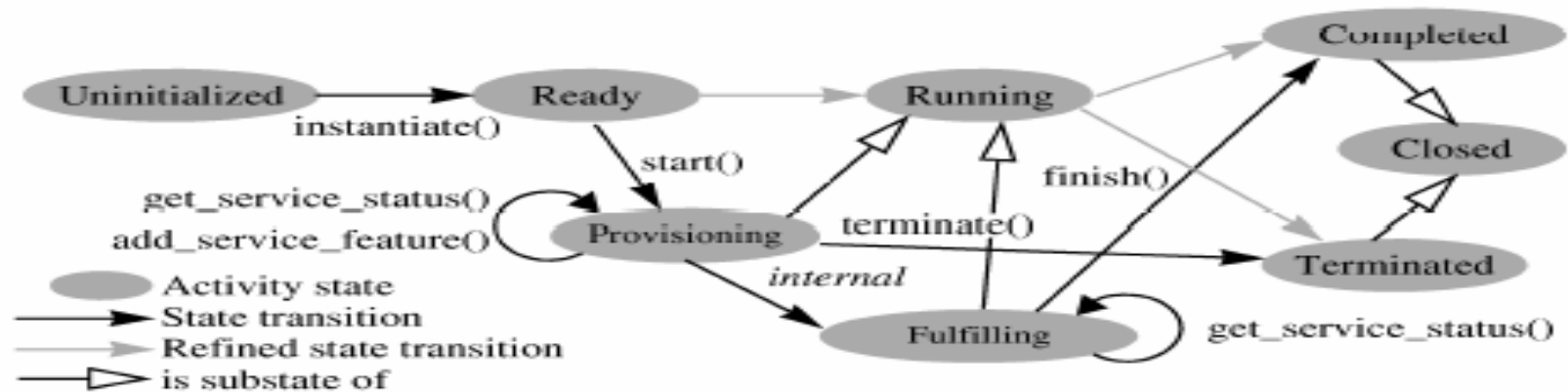
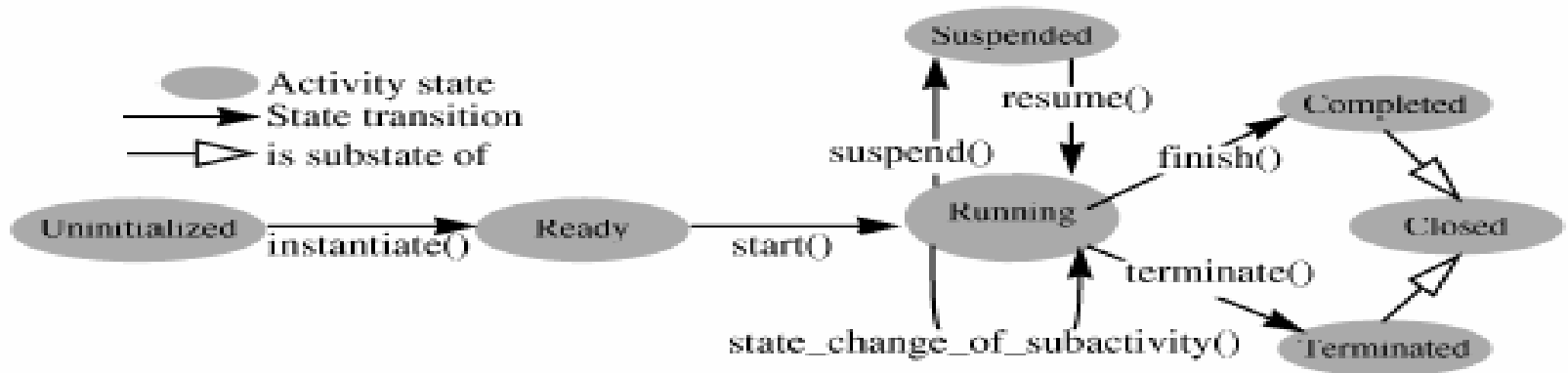
$$Abstr = (n' \circ \varepsilon \circ n).$$



Prozessabstraktion nach Schuster

- Abbildung interner Prozesse als Activity State Machines (ASM)
- Unterscheidung ASM & ASM Types (ASMT)
 - Trennung Implementation & Schnittstelle
- Strukturabstraktion durch Konzepte wie
 - Subtyping von ASM
 - Generalisierung von Activity State Machines
- Sichtbarkeitsabstraktion durch Konzepte wie
 - Einschränkungen der Transparenz durch Activity State Machine Types als Abstrakte Schnittstelle fuer ASM

Beispiel II





Polymorphic Process Model (PPM)

- Entkopplung von Aktivität Schnittstellen und Aktivitätsimplementation
- Prozesse sind modellierbar als Aktivitäten
 - ASM
 - Kompatibilität von Aktivitäten mit WfMC Standard
- Subtyping von Aktivitäten
- Polymorphie der Aktivitätstypen zur dynamischen Bindung an Aktivitätsvariablen
 - late binding
 - Dynamik



Gliederung

1. Motivation & Einleitung
2. Process Interconnection
3. Kooperative Prozesse
4. Generic Process Metamodel
5. Prozessabstraktion
6. **Fazit**
7. Literaturhinweise



Fazit

- Forschungsansätze
- Offene Themen:
 - Koordination der Prozesse (Koordinator-Auswahl)
 - Im Metamodell fehlt eine Beschreibungssprache für die Service Prozesse
 - Kompositionsfunktion (Zwischen Level 2 und 3)
- Keine praktischen Implementierungen vorhanden
- Lösungsansätze nur im Metabereich, bzw. für bestimmte Teilproblematiken.



Gliederung

1. Motivation & Einleitung
2. Process Interconnection
3. Kooperative Prozesse
4. Generic Process Metamodel
5. Prozessabstraktion
6. Fazit
7. Literaturhinweise



Literaturhinweise

- Bhiri, O. Perrin, W. Gaaloul, and C. Godart, "An Object Oriented Metamodel for Inter-enterprises Cooperative Processes based on Web Services," in *Proc. Modeling and Developing Process-Centric Virtual Enterprises with Web-Services, Special Session at the 7th World Conference on INTEGRATED DESIGN AND PROCESS TECHNOLOGY (IDPT 2003) "The Future of Software Engineering in the Networked World", June 16-20, 2003, Fragrant Hill Hotel -Beijing - China, 2003.*
- K. Baina, K. Benali, and C. Godart, "A Process Service Model for Dynamic Enterprise Process Interconnection," in *proc. CoopIS 2001, LNCS 2172*, C. Batini, Ed.: Springer, 2001, pp. 239–254.S.
- H. Schuster, D. Georgakopoulos, A. Cichocki, and D. Baker, "Modeling and Composing Service-Based and Reference Process-Based Multi-enterprise Processes," in *Proc CAiSE 2000, LNCS 1789*, B. Wangler and L. Bergman, Eds.: Springer, 2000, pp. 247-263.