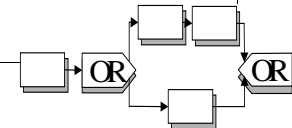
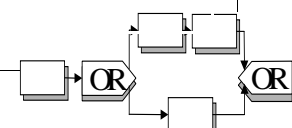
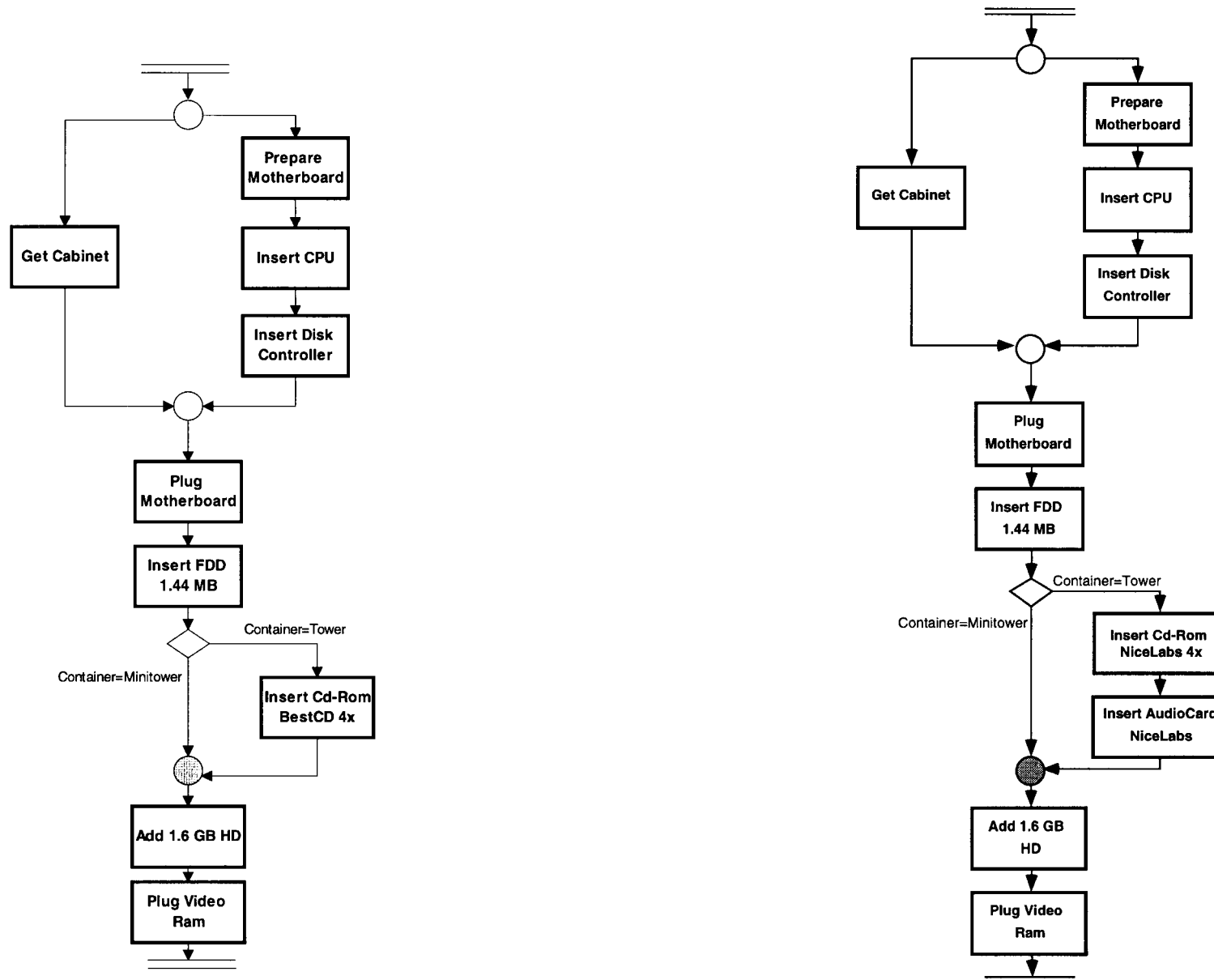


Schema-Evolution

- Änderung des Schemas (d.h. der Workflow-Definition) aufgrund geänderter Randbedingungen, z.B. bei
 - neuen Organisationsstrukturen
 - neuen Verfahrensregeln (z.B. zusätzliche Kreditprüfung bei allen Krediten > 100.000 DM)
- Änderungen des Schemas
 - Manuell
 - Automatisch (→ Planungsmethoden der Künstlichen Intelligenz)
- Problematik: Behandlung bereits aktiver, evtl. langlebiger Workflow-Instanzen
 - Abbruch?
 - Weiterführung nach altem Schema?
 - Migration zu neuem Schema?
- Literatur (Auswahl)
 - Rinderle, St. Dadam, P.: *Schemaevolution in Workflow-Management-Systemen*, Informatik Spektrum 26(1):17-19 (2003).
 - Rinderle, S.; Reichert, M.; Dadam, P.: *Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-Instanzen bei der Evolution von Workflow-Schemata*, Informatik - Forschung und Entwicklung 17(4):177-197 (2002).
 - Alfred Fent, Herbert Reiter, Burkhard Freitag: *Design for Change: Evolving Workflow Specifications in ULTRA-flow*. CAiSE 2002: 516-534.



Beispiel (Casati et al.)



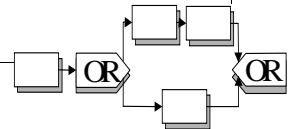
Migrations-Strategien (WIDE, Casati et al.)

■ Drei verschiedene Migrations-Strategien

- Koexistenz (“Concurrent to Completion“)
- Migration zu finalem Schema
- Migration zu Ad-Hoc-Schema

■ Koexistenz

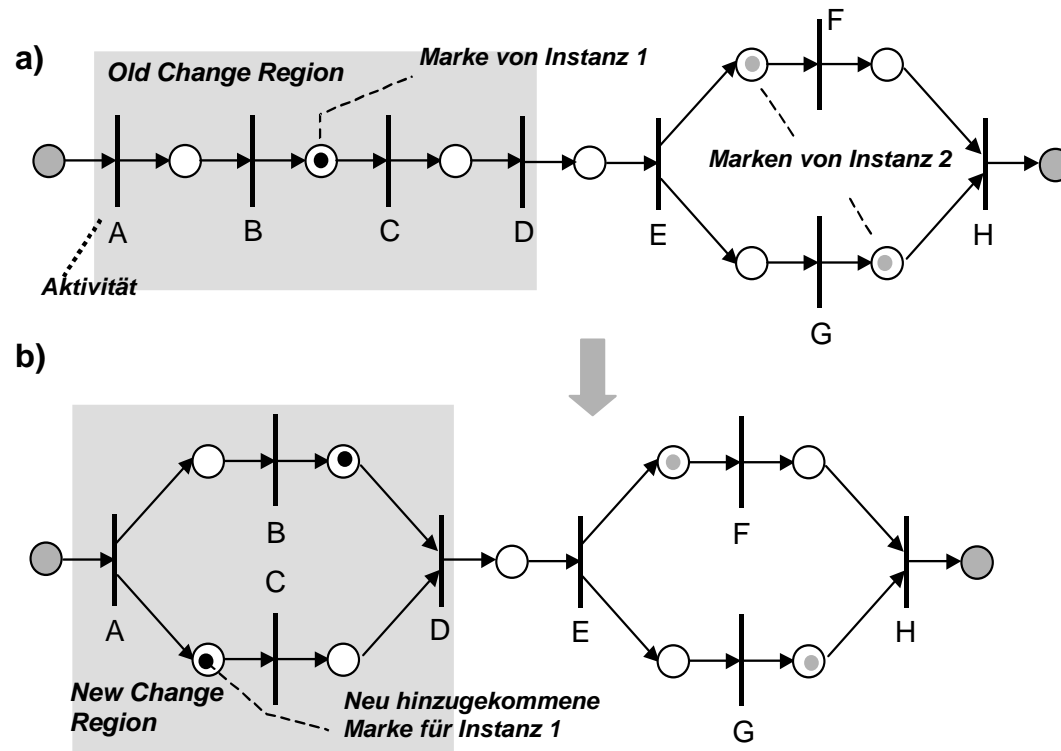
- Bereits in Ausführung befindliche Workflow-Instanzen werden nach altem Schema zu Ende geführt
- Neue Instanzen benutzen ausschließlich neues Schema
- In bestimmten Anwendungsbereichen sinnvoll
 - Verwaltung, Administration (bei neuen Verordnungen und Gesetzen)
- Vorteil: Einfach zu realisieren
- Nachteil: Uneinheitliche Behandlung von Fällen (z.B. Kunden, Patienten)



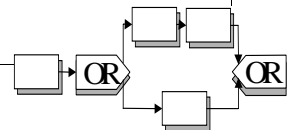
Koexistenz-Strategie bei Petrinetz-basierten WfMS

(Ellis et al.)

- Beispiel: Bisher sequentiell ausgeführte Aktivitäten (B und C) sollen in Zukunft parallel bearbeitet werden



- Neue Instanzen laufen automatisch auf Basis der neuen Struktur



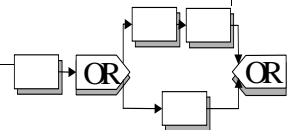
Migrations-Strategien (2)

■ Migration zu finalem Schema

- Alle bereits in Ausführung befindliche Workflow-Instanzen werden bzgl. des noch abzuarbeitenden Kontroll- und Datenflusses an *neues* Schema angepaßt
- Vorteil: Einheitlichere Behandlung von Fällen
- Nachteil: Bereits ausgeführter Workflow-Teil kann inkonsistent zu neuem Schema sein, z.B.:
 - Neuer Teil enthält zusätzliche Aktivitäten, für die bereits ausgeführter Teil des alten Schemas keinen adäquaten Datenfluß liefert
 - Neuer Teil sieht Behandlungsschritt vor, für die bereits ausgeführter Teil des alten Schemas nicht die notwendigen diagnostischen Aktivitäten enthält

■ Möglicher Ausweg: Konditionales partielles Rollback

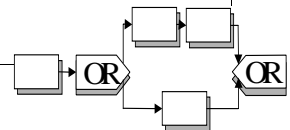
- Definition: Eine Workflow-Instanz I heißt konform zu Schema S , wenn alle bisher durchgeführten Schritte den Vorgaben von S entsprechen
- Grundprinzip: Migration von I zu finalem Schema S_{new} ist unproblematisch, wenn I (gestartet auf der Basis von S_{old}) zum Zeitpunkt der Schema-Evolution noch konform zu S_{new} ist
- Falls I nicht (mehr) konform zu S_{new} : Rollback zu letztem Zustand, der konform zu S_{new} ist; Wiederaulauf
- Problem: Arbeitsverlust, Wiederholung von Aktivitäten



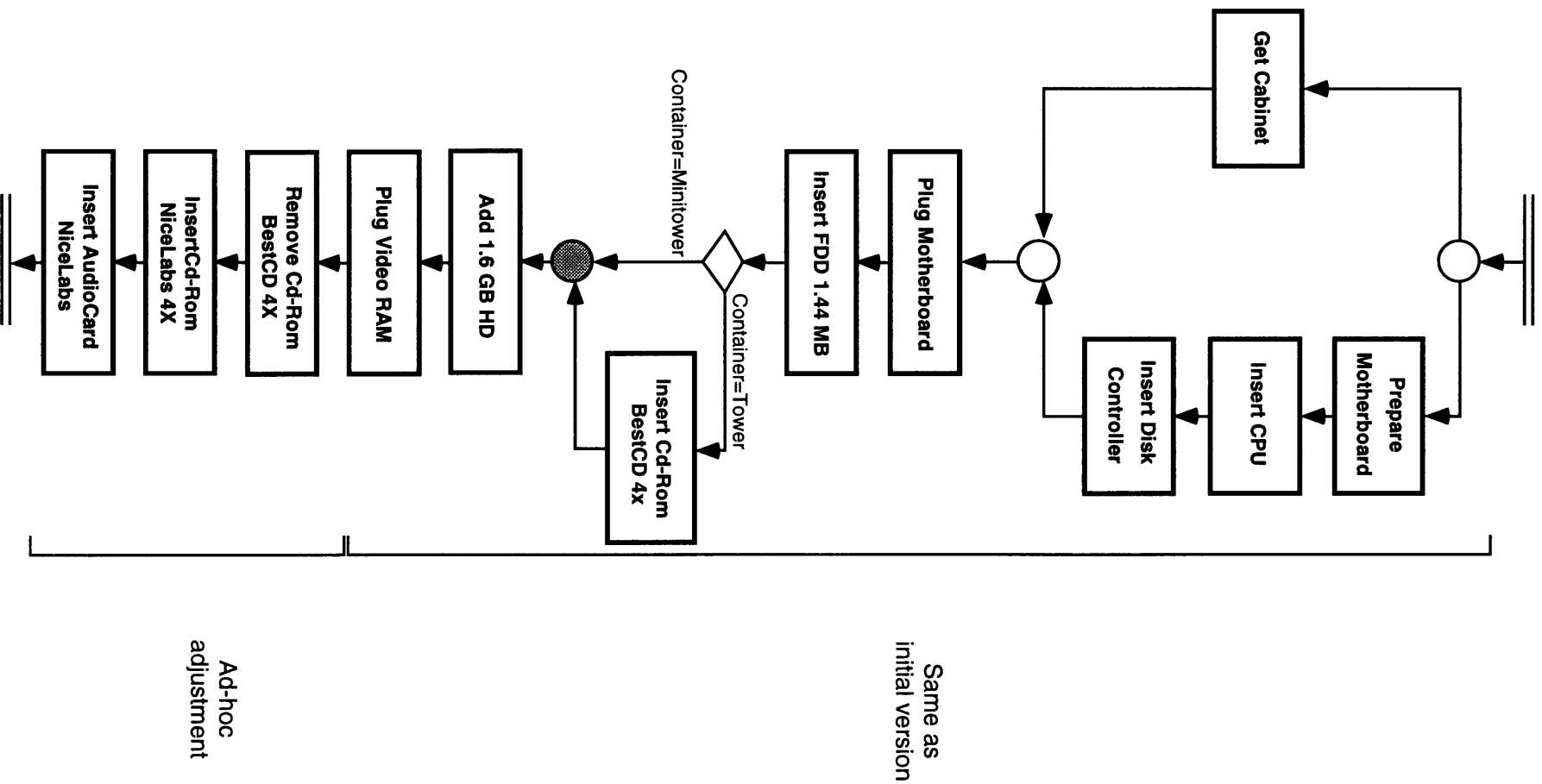
Migrations-Strategien (3)

■ Migration zu Ad-Hoc-Schema

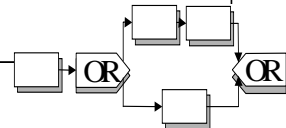
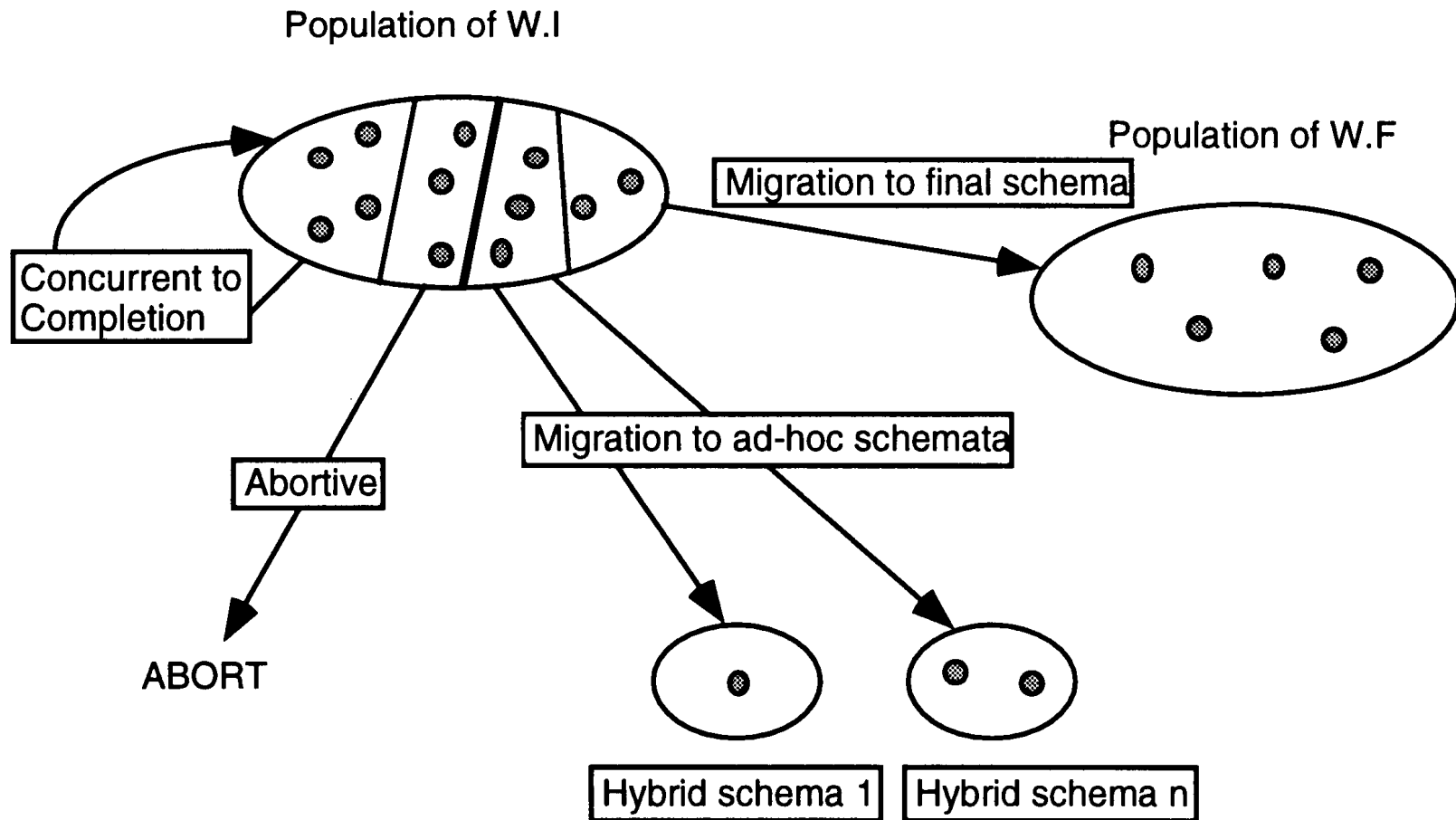
- Alle bereits in Ausführung befindliche Workflow-Instanzen werden bzgl. des noch abzuarbeitenden Kontroll- und Datenflusses an instanzspezifische Ad-Hoc-Version des neuen Schemas angepaßt
- Vorwärtsgerichtete Behandlung möglicher Inkonsistenzen zwischen altem und neuem Schema
- Beispiele:
 - Einfügen zusätzlicher Datenflüsse, um Aktivitäten des neuen Schemas zu versorgen
 - Einfügen von Aktivitäten, die bereits durchgeführte, aber in neuem Schema nicht mehr vorgesehene Aktivitäten kompensieren
 - Einfügen vorbereitender Aktivitäten, die im bereits ausgeführten Teil des alten Schema nicht vorgesehen waren
- Vorteil: Fallspezifische Migration, Vermeidung von Inkonsistenzen
- Nachteil:
 - Verlagerung der Migrationsproblematik auf Ad-Hoc-Adaptation
 - Wer entscheidet (automatisch), welche Adaptationen notwendig sind?



Ad-Hoc-Migration: Beispiel



Migrations-Strategien: Zusammenfassung



Flexibles Workflow-Management: Zusammenfassung

- Workflow-Systeme müssen flexibel auf veränderte Situationen reagieren können, u.a. auf Veränderungen bzgl.
 - Ressourcensituation
 - zeitlichen Vorgaben (z.B. Veränderung bzgl. Deadlines)
 - Menge der notwendigen und sinnvollen Aktivitäten (Kontrollfluß)
 - Datenfluß
- Prinzipielle Ansätze
 - Vormodellierung
 - Late Modeling/Late Binding
 - Ad-hoc-Adaptation
 - Schema-Evolution
- Problematik: Bislang kaum Integration der Ansätze in kommerzielle Produkte

