ARKTIS/Makros - flexible, erweiterbare WfMS

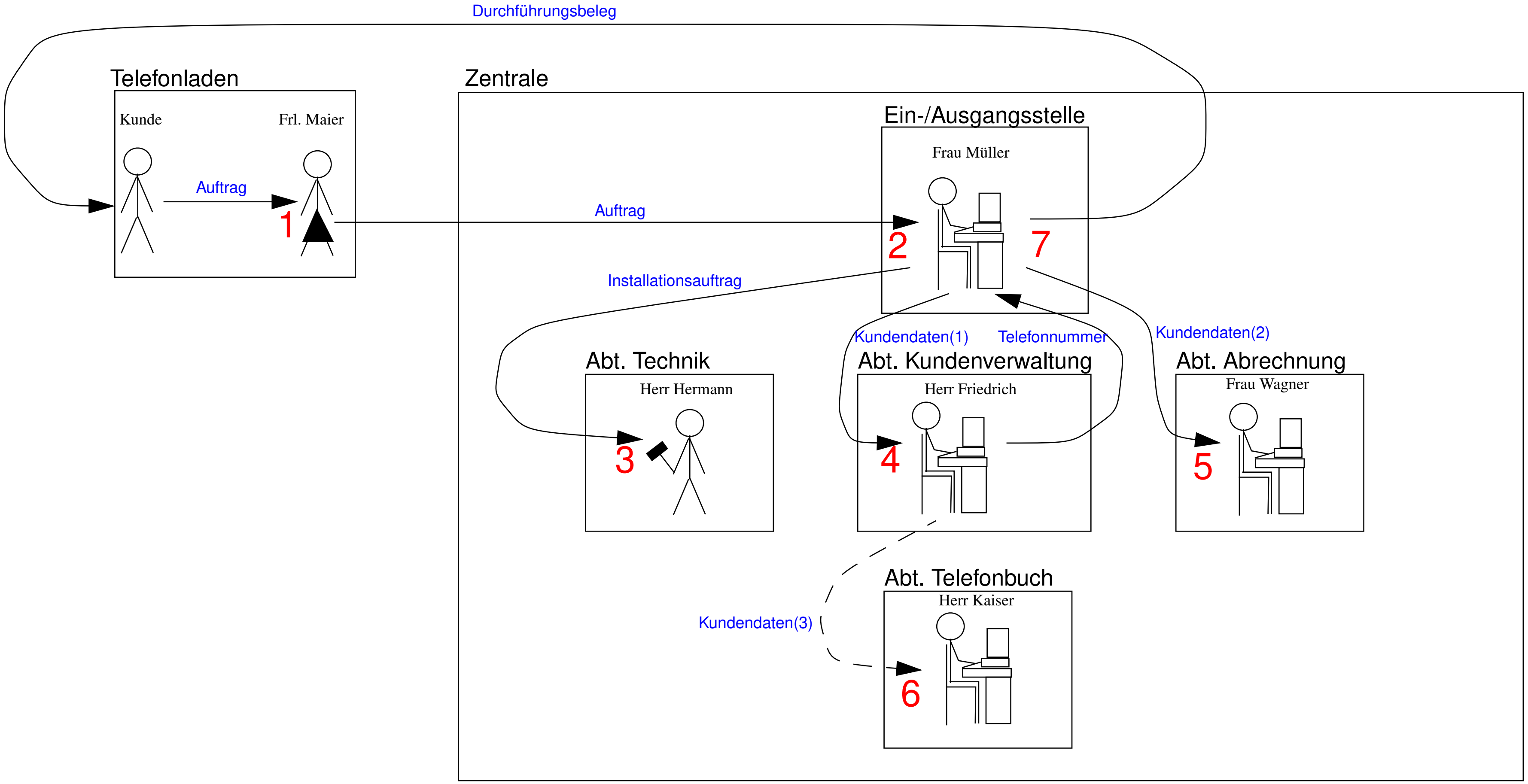
Übersicht

1. Workflows - ein Beispiel
2. Prozeßmodellierung - Ansätze
3. Makrokomponente - die Idee
4. Makros - Motivation und Einsatz
5. Makrokomponente - die Bestandteile
6. Implementierung - Beispiel Prozeßkontrollnetze


ARKTIS
Makros

Workflows

<u>Beispiel:</u> Telefonanschluß



Workflows

WAS ?

- Kunden betreuen (1)
- eingehende Aufträge bearbeiten (2)
- Installation durchführen (3)
- Telefonnummer zuteilen / Kundendaten aufnehmen (4)
- Abrechnungsdaten erfassen (5)
- Kundendaten für Telefonbuch aufnehmen (6)
- Kundenmitteilung erstellen (7)

WER ?

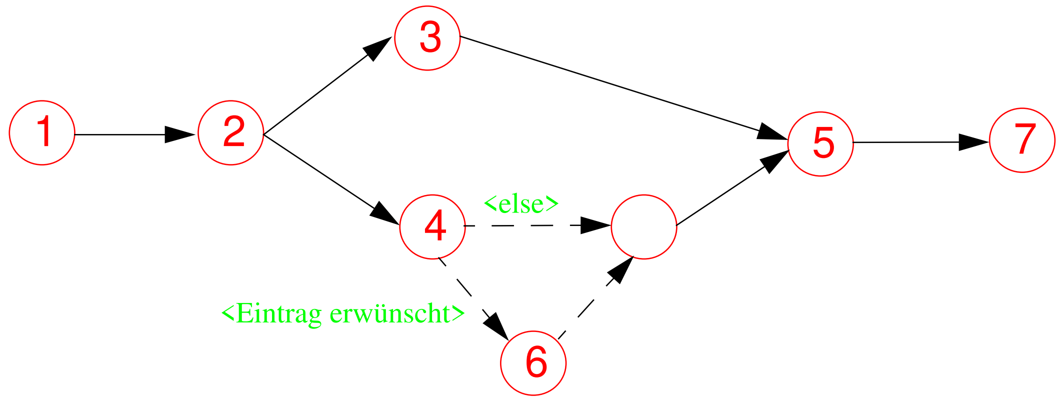
- Kundenbetreuerin (Frl. Maier) (1)
- Beschäftigter der Ein-/Ausgangsstelle (Frau Müller) (2)
- Techniker (Herr Hermann) (3)
- Sachbearbeiter “Neukunden” (Herr Friedrich) (4)
- Sachbearbeiter “Abrechnung” (Frau Wagner) (5)
- Sachbearbeiter “Telefonbuch” (Herr Kaiser) (6)
- Beschäftigter der Ein-/Ausgangsstelle (Frau Müller) (7)

WOMIT ?

- (1) → (2) Auftrag
- (2) → (3) Installationsauftrag
- (2) → (4) Kundendaten1
- (2) → (5) Kundendaten2
- (4) → (6) Kundendaten3

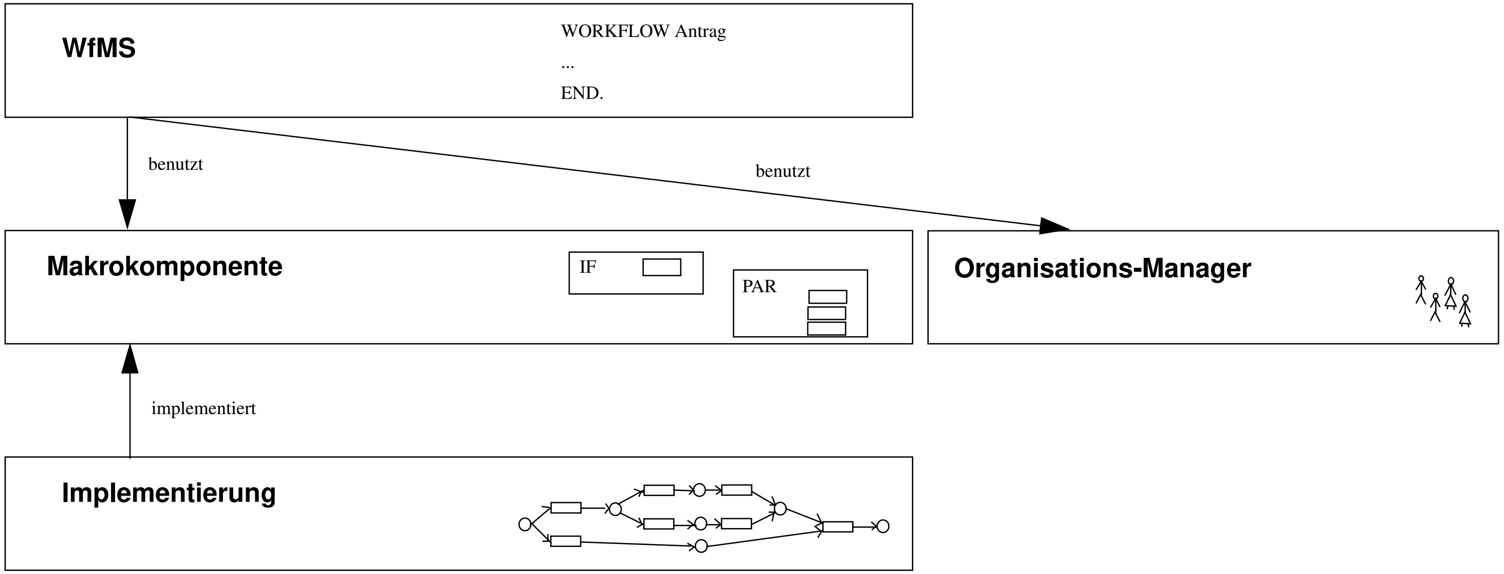
Workflows

☛ WANN ?



- Graph beschreibt zeitliche Abhängigkeiten zwischen einzelnen Tätigkeiten
- Parallelausführung verschiedener Tätigkeiten möglich
- optionale Ausführung von Tätigkeiten möglich

Schichtenmodell



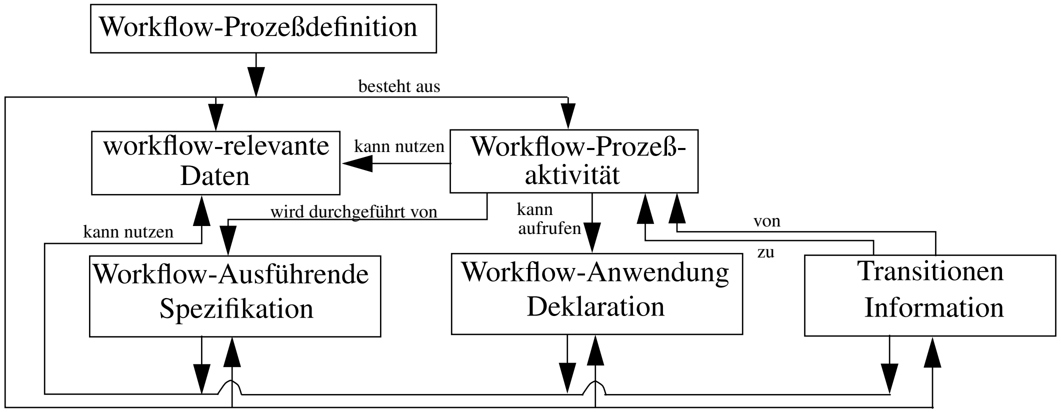
- Bei heutigen WfMS: Komponenten weitestgehend fest, nur eingeschränkte Anpassung möglich
- Makrokomponente ermöglicht das Erstellen und nachträgliche Anpassen des WfMS

Metamodell der WfMC

Workflow Management Coalition (WfMC)

- Zusammenschluß von über 130 Mitgliedern
- besteht seit 1993
- Ziel: Entwickeln von Standards für Workflow Management-Produkte
- prominente Mitglieder: IBM, IDS Scheer, Fujitsu Software Corporation, Microsoft Corporation, SAP AG

Metamodell



Aspektorientierte Modellierung

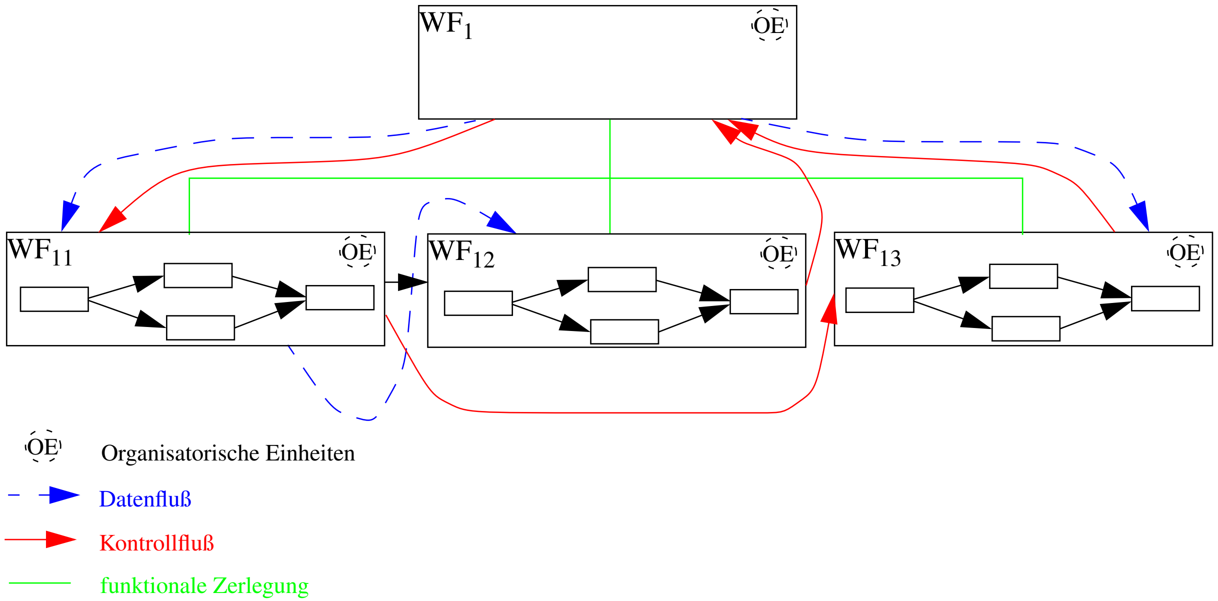
Aspekte (Jablonski)

- Betrachtung des Prozesses aus verschiedenen Perspektiven

• funktional	was wird ausgeführt?
• verhaltensbezogen	wann wird ausgeführt?
• informationsbezogen	was wird benötigt?
• organisatorisch	wer führt aus?

Aspektorientierte Modellierung (2)

Beziehung zwischen Aspekten

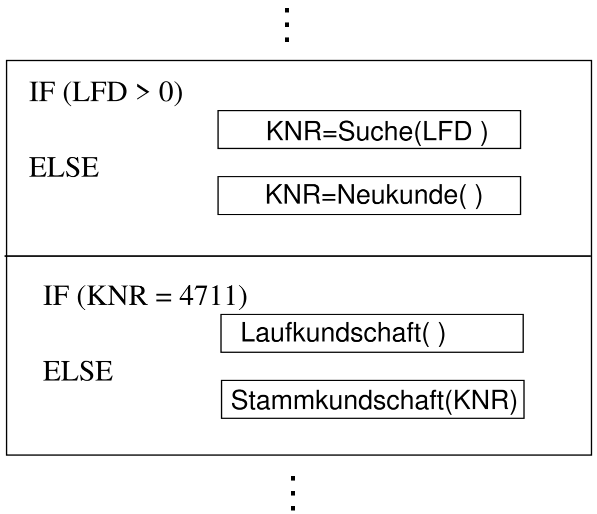


- funktionaler Aspekt gibt grundlegende Struktur vor (Zerlegung in Sub-Workflows)
- Daten- und Kontrollfluß weitestgehend unabhängig voneinander modellierbar
- elementare Sub-Workflows erfordern Zusammenführen der Aspekte

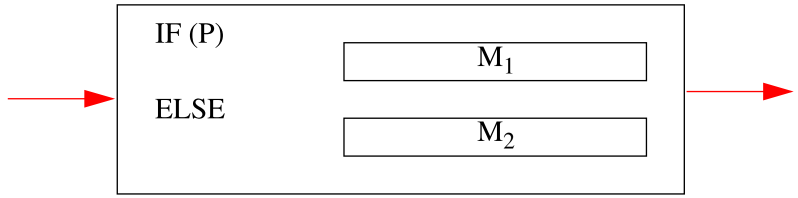
Makros

- Strukturen innerhalb von Workflows immer wiederkehrend
- Definieren eines “Gerüsts”
- Instanziieren mit geeigneten Parametern

Workflow:



Makrorumpf *IF* :



Makros:

Makros (2)

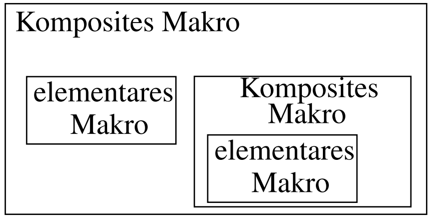
Begriffe

- **Makrorümpfe:** Workflow-Bausteine, die durch Angabe geeigneter Parameter zu Makros instanziiert werden können.
- **Makros:** instanziierte Makrorümpfe.
- **Elementare Makros:** Makros, die keine anderen Makros als Submakros benutzen.
- **Komposite Makros:** Makros, die andere Makros als Submakros benutzen.
- **Grundmakros:** Makros, zu denen eine direkte Implementierung oder Implementierungsvorschrift existiert.
- **Abgeleitete Makros:** Makros, die aus Grundmakros und / oder abgeleiteten Makros aufgebaut werden können.

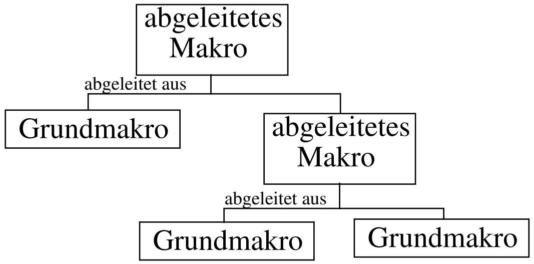
Makros(3)

Beziehungsdiagramme

komposit / elementar



Grundmakros / Abgeleitete Makros



Elementare Makros sind Grundmakros, aber
Grundmakros müssen nicht elementar sein

Interfaces

Klassifikation von Makros:

- verschiedene Informationen für Deklaration / Instanziierung von Makros notwendig
  - Eingangstransitionen
  - Ausgangstransitionen
  - Typ der einsetzbaren Makros
  - Rückgabetyp der einsetzbaren Funktionen
  - Deklaration von Parametern einfachen Typs
  - Deklaration von Kollektionen

- Zuordnen von *Interfaces*
  - Interface bestimmt Makrotyp
  - Makro kann von mehreren Interfaces abgeleitet werden

Grundmakros

- Grundstock verwendbarer Makros mit zugehöriger Implementierung
- Flexibel erweiterbar

Kontrollfluß

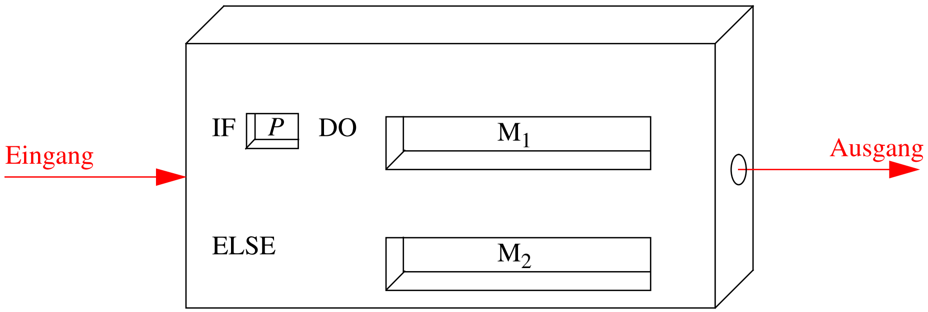
- leeres Makro (“DoNothing”)
- Funktionsmakro (z. B. “INC”)
- Sequenz (“SEQ”)
- Serialisierung (“SER”)
- Parallelität (“PAR”)
- bedingte Verzweigung (“IF”)
- Iteration (“REPEAT”)

Datenfluß

- Verschieben (“MOVE”)
- Kopieren (“COPY”)
- Extrahieren (“EXTRACT”)
- Verschmelzen (“MELT”)

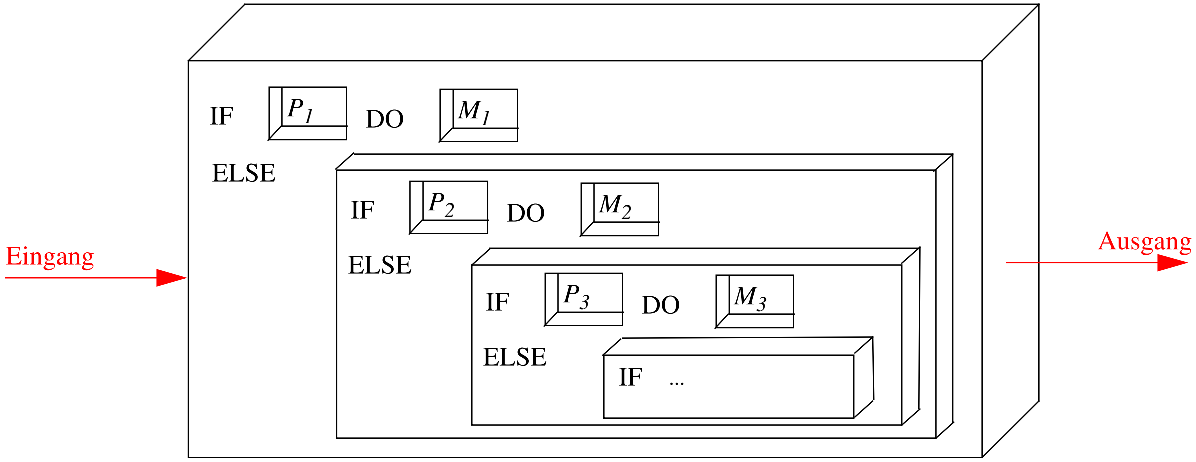
Grundmakros (2)

```
INTERFACE   CondInterface;
IN          Eingang[CF, *];
OUT         Ausgang[CF, *];
FUNCTION    p OF BOOLEAN;
MACRO       M1 IS MacroInterface;
MACRO       M2 IS MacroInterface;
END.

MACRO IF IS CondInterface;
End.
```

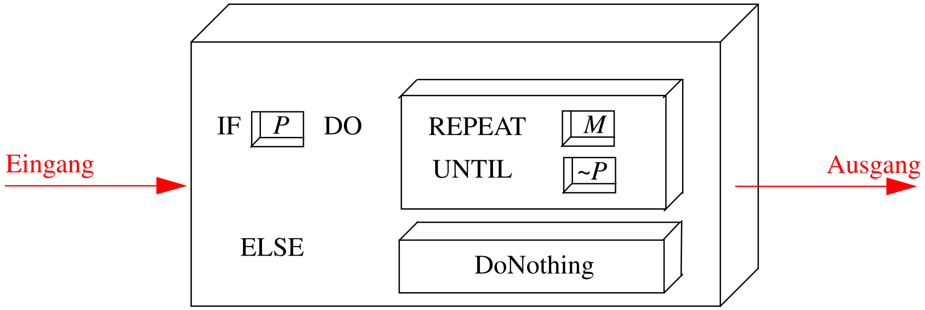


Abgeleitete Makros

SWITCH



WHILE



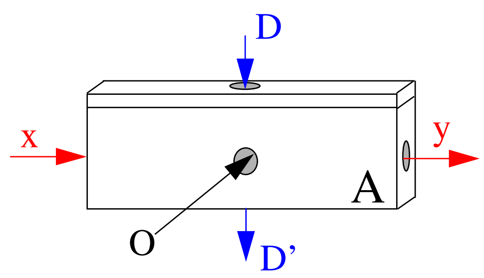
Abgeleitete Makros (2)

```
INTERFACE NEW DynamicParmInterface;
IN          Eingang  [CF, DOMAIN Token1];
OUT         Ausgang  [CF, DOMAIN Token2];
COLLECTION  Zweige   (FUNC p, MACRO m IS MacroInterface);
END;

MACRO       Switch IS DynamicParmInterface;
BODY        #IF (#EMPTY(Zweige) = #FALSE)
                #MACRO_USE IF( #FIRST(Zweige).p,
                               #FIRST(Zweige).m,
                               #MACRO_USE Switch (#REST(Zweige))
                                   GIVING Eingang
                                   TAKING Ausgang);
            #ELSE #EXEC Fehler;
            #END;
END.
```

```
INTERFACE NEW IterationInterface;
IN          Eingang  [CF, DOMAIN *];
OUT         Ausgang  [CF, DOMAIN *];
FUNCTION    p OF BOOLEAN;
MACRO       M IS MacroInterface;
END;

MACRO While IS IterationInterface;
BODY:           #MACRO_USE IF(p, #MACRO_USE Repeat(~p, m), #EXEC DO_NOTHING);
END.
```

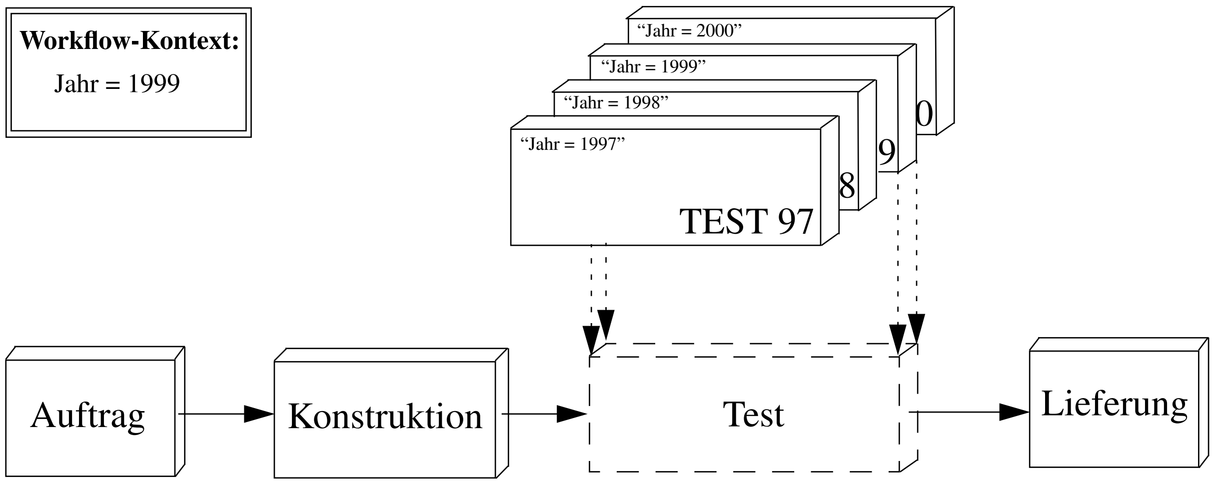
Hybridmakros

- Makros beschreiben *entweder* Kontrollfluß *oder* Datenfluß
- Aktivitäten sind in Kontrollfluß *und* Datenfluß eingebettet
- zusätzlich organisatorische Informationen notwendig
- Spezielle Makros für Kreuzungspunkte erforderlich
  - Hybridmakros



Adaptive Workflows

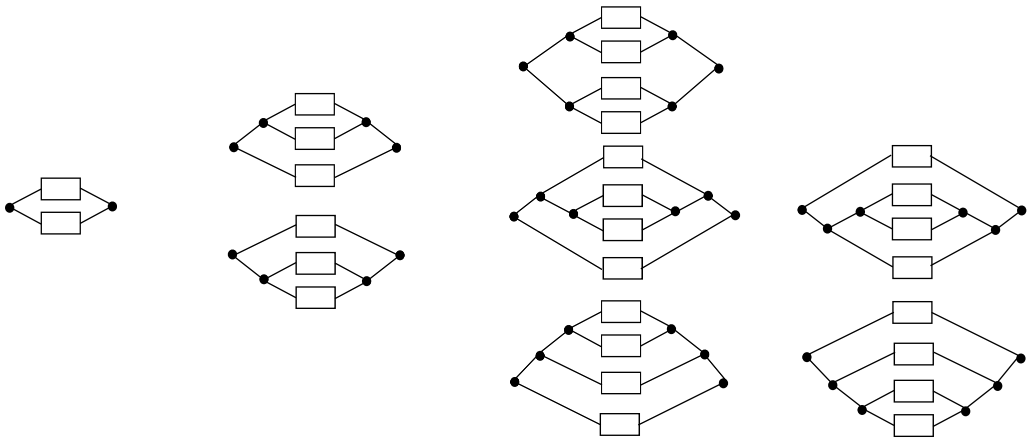
- Vollständige Definitition eines Workflows im voraus nicht immer möglich
- Möglichkeit der dynamischen Anpassung zur Laufzeit → *Polymorphe Makros*
- Definition eines Stellvertreters und mehrerer Auswahlmakros
- Definition eines Kontextes



Variable Teilmakrozahl

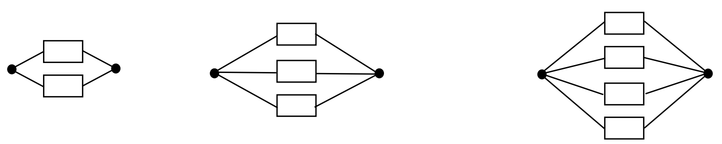
- Komposite Makros mit beliebig vielen Teilmakros (SER, PAR)
- Übergabe als *Kollektion*
- Verschiedene Strategien zur Umwandlung
  - Rekursives Auflösen
  - Algorithmisches Expandieren

Rekursion



a) 2 makros b) 3 makros c) 4 makros

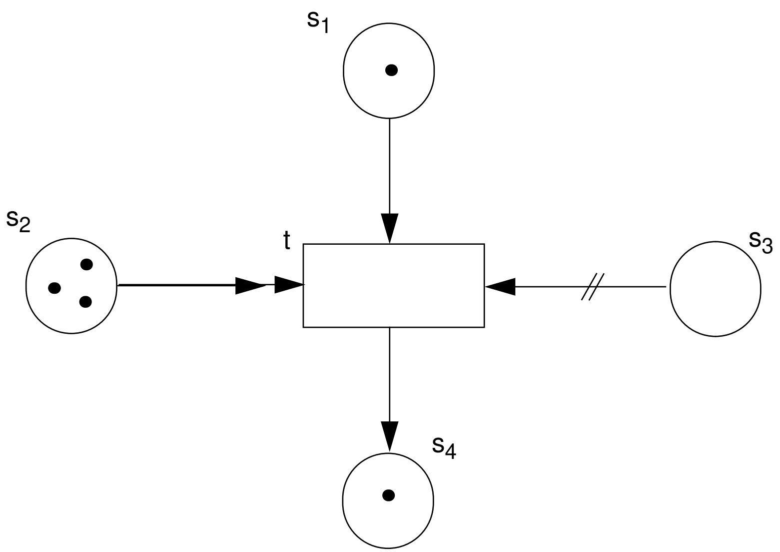
Expansion

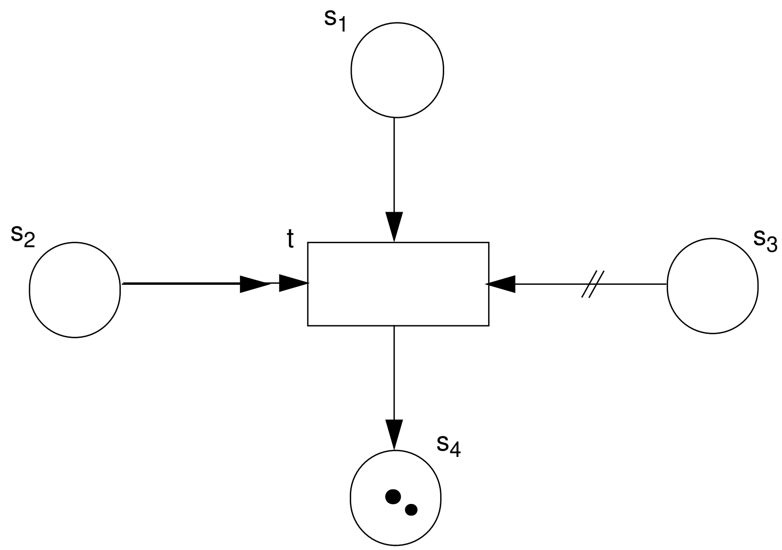


a) 2 makros b) 3 makros c) 4 makros

- Ausführung der Zweige synchronisiert, einmalig oder exklusiv

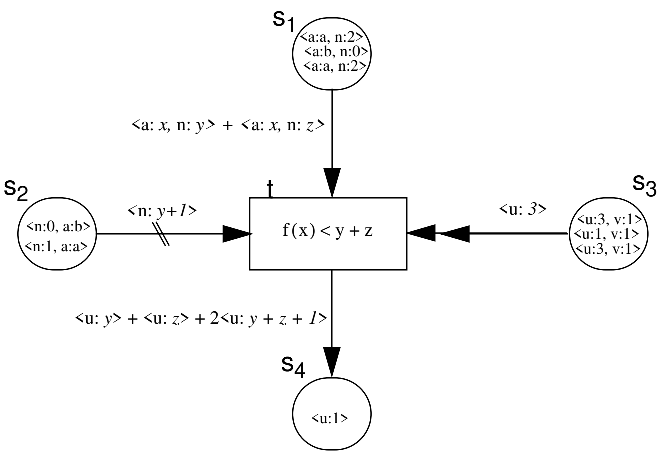
Petri-Netz





Prozeßkontrollnetz

Petri-Netze erweitern



Vorspann:

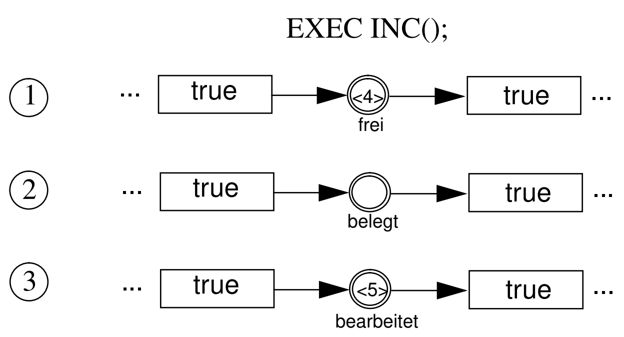
f: $\mathrm{CHAR} \rightarrow \mathrm{NAT}_0$ mit f(a) = 1; f(b) = f(c) = ... = f(z) = 0

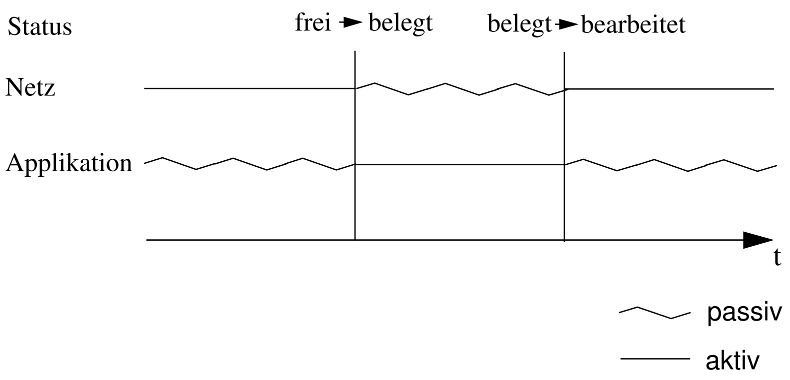
Definitionsbereiche:

D_{s1} = a: CHAR x n: NAT_0
D_{s2} = n: NAT_0 x a: CHAR
D_{s3} = u: NAT_0 x v: NAT_0
D_{s4} = u: NAT_0

Magische Stelle

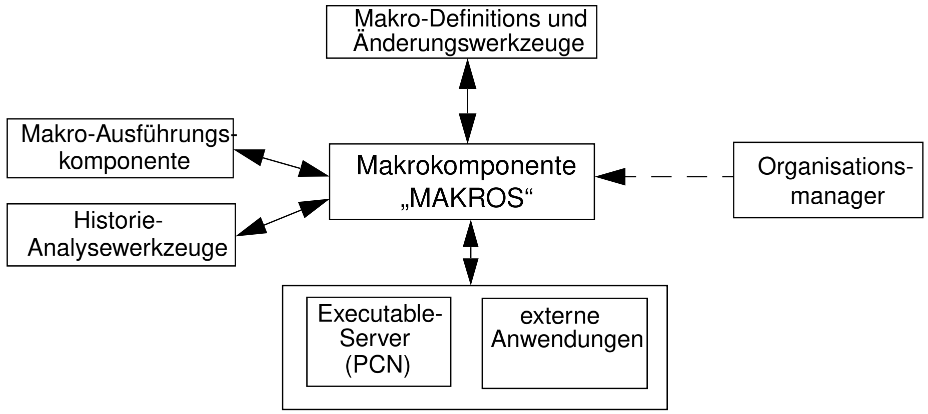
- Kommunikation mit “Außenwelt” ermöglichen
- Netz gibt Kontrolle vorübergehend ab
- Veränderung der Marken auf “magische” Art und Weise → *“Magische Stelle”*
- Zustands-Flag signalisiert aktuelle Verarbeitungsphase



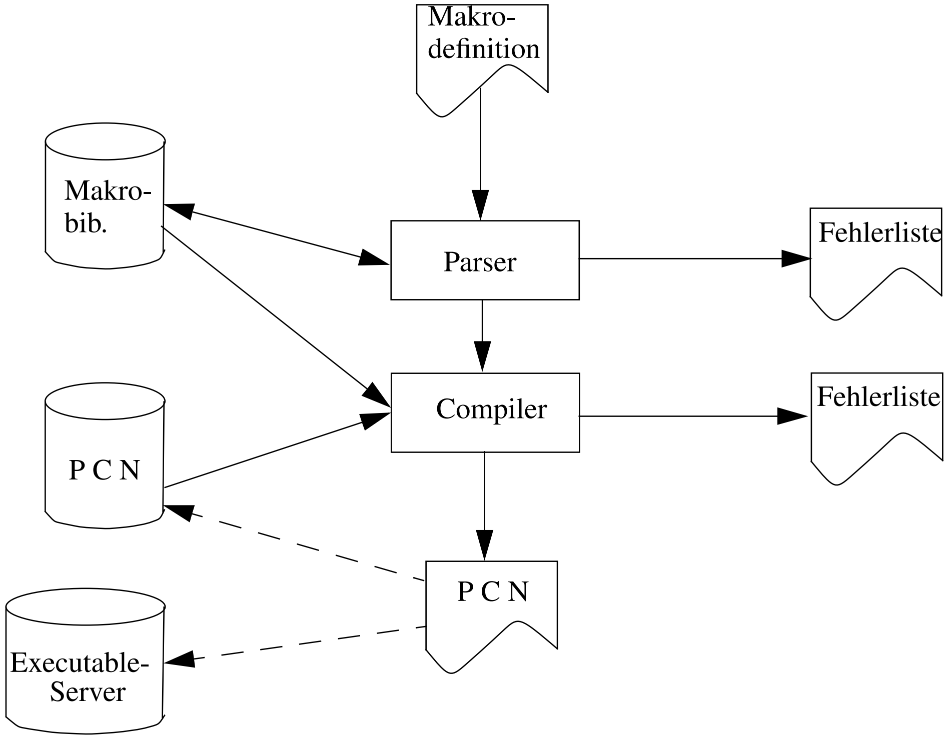


Die Makrokomponente

Makrobearbeitung • Neudefinieren von Makros • Ändern von Makros	**Makroverwaltung** • Bereitstellen bisheriger Definitionen • Verbindung zu externen Anwendungen
Ausführungskomponente • Ausführen von Makros	**Analysewerkzeuge** • Auswerten von Abläufen



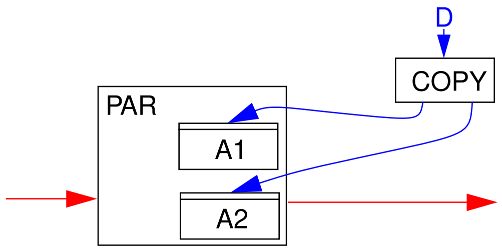
Makro-Neudefinition



- Trennung von Parser und Compiler
  - Bei Wechsel der Implementierung können Makrodefinitionen weiterverwendet werden
  - Grundmakros müssen neu implementiert werden

Workflow-Definition

- Definition eines Workflows auf WfMS-Ebene
- Hybridmakros in Makrokomponente nicht voll einsetzbar

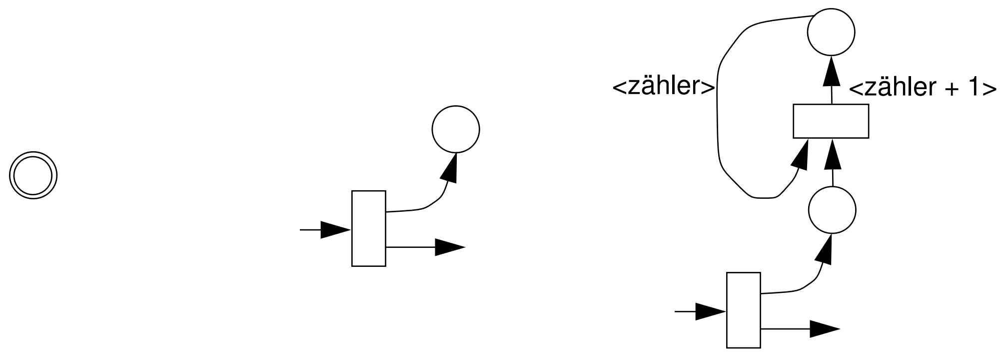



```
WORKFLOW WF1;
    m1 = A1();// A1 und A2 sind Makros vom Typ Applikation
    m2 = A2();
CF:...
    #MACRO_USE PAR((„k=1“, m1), („k=2“, m2));
    ...
DF:...
    #EXEC COPY()
    TAKING Eingabe FROM D
    GIVING Ausgabe1 TO m1.Eingabe,
           Ausgabe2 TO m2.Eingabe;
...
```

gleiche Instanz!

Ablaufanalyse

- Implementierungsschicht stellt Überwachungsmechanismen zur Verfügung
- Ausgezeichnete Stellen im Netz zur Überwachung / Analyse eines Laufs

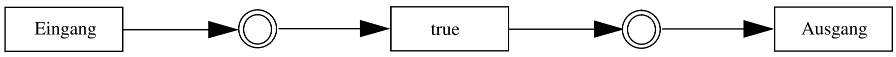


a) Magische Stelle (aktiv) b) passive Statusstelle c) Statusstelle mit Zähler

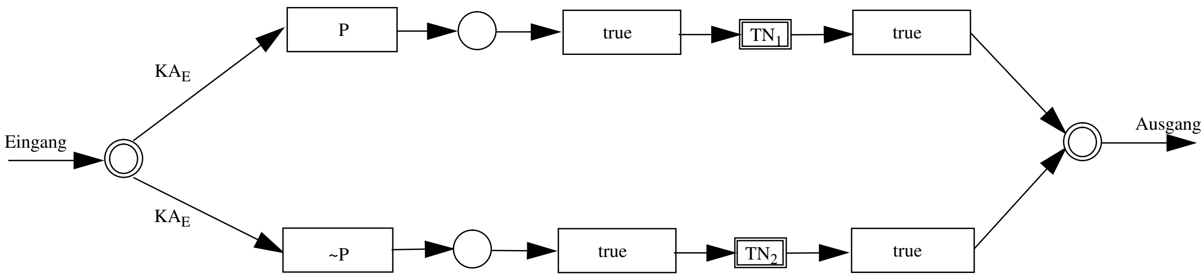
- Magische Stellen melden “Passieren” an Monitor (Überwachung)
- Statusstellen dienen der Erreichbarkeitsanalyse (Fehlersuche)
- Zähler dienen der Lastanalyse (Optimierung)

Implementierung

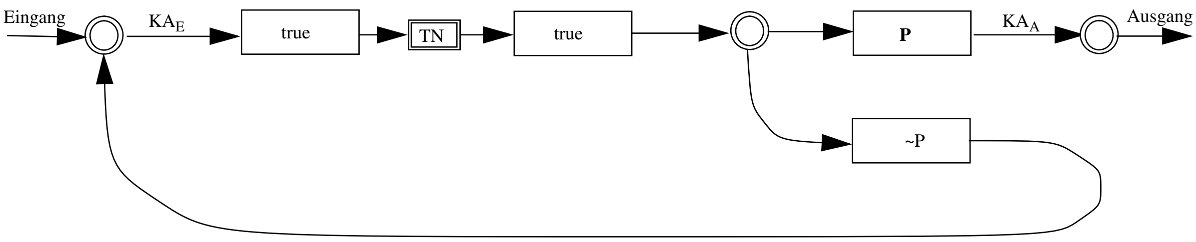
"DoNothing"



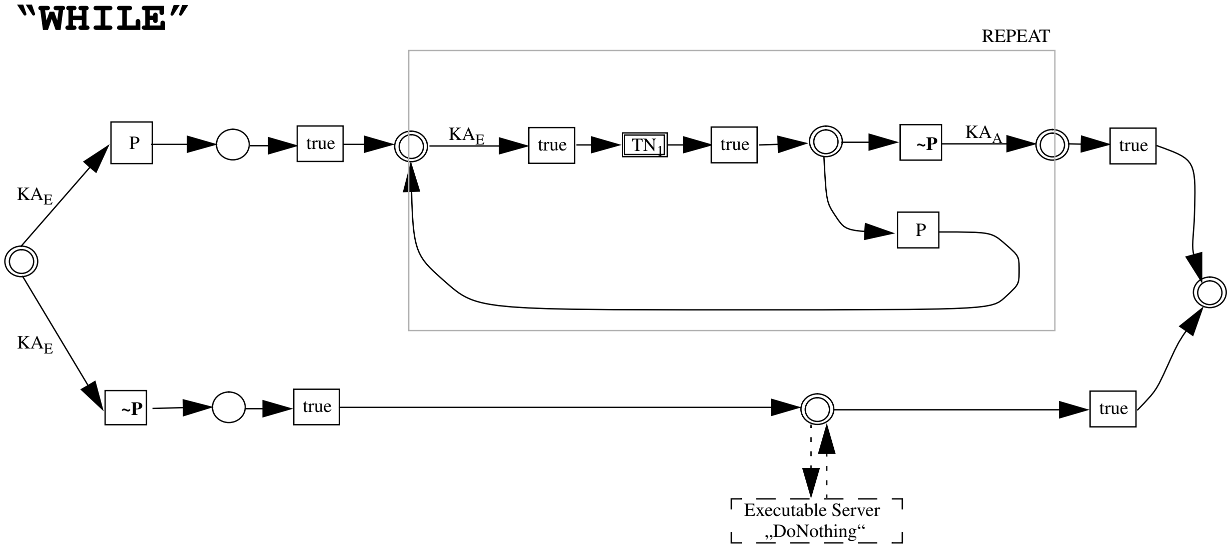
"IF"



"REPEAT"



"WHILE"



Zusammenfassung

- Workflows
- Modelle zur Prozeßmodellierung - WfMC / Jablonski
- Idee der Makrokomponente
- Makros
- Aufbau der Makrokomponente
- Implementierungsapekte

FAZIT

☛ Einführen der Makroschicht bringt:

- einfache Anpaßbarkeit / Erweiterbarkeit des WfMS
- weitestgehende Trennung der Modellierungsaspekte
- Austauschbarkeit der Definitionselemente (Makros)
- Unterstützung von Modularität und Wiederverwendbarkeit
- Polymorphe Makros
- Rekursion / Expansion für Makros mit beliebiger Parameterzahl
- einfachen Wechsel der Implementierung
- ...

Ausblick

Ziele:

- komplettes WfMS gemäß WfMC-Anforderungen
- Makrobeschreibungen in neutralem Austauschformat (XML)
- Bereitstellen einer nutzbaren Implementierungsschicht (PCN)