

# Pattern-based Analysis of the Control-flow Perspective of UML Activity Diagrams

Petia Wohed	UHP (SU/KTH)
Wil M.P. van der Aalst	TUE & QUT
Marlon Dumas	QUT
Arthur H.M. ter Hofstede	QUT
<b>Nick Russell</b>	<b>QUT</b>



# Background

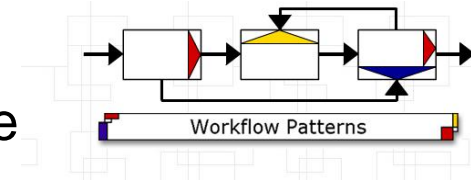
The upgrade of UML 1.4 to version 2.0

# Focus

UML 2.0 Activity Diagrams

# Analysis framework

Workflow patterns: the Control-flow perspective



# Contribution

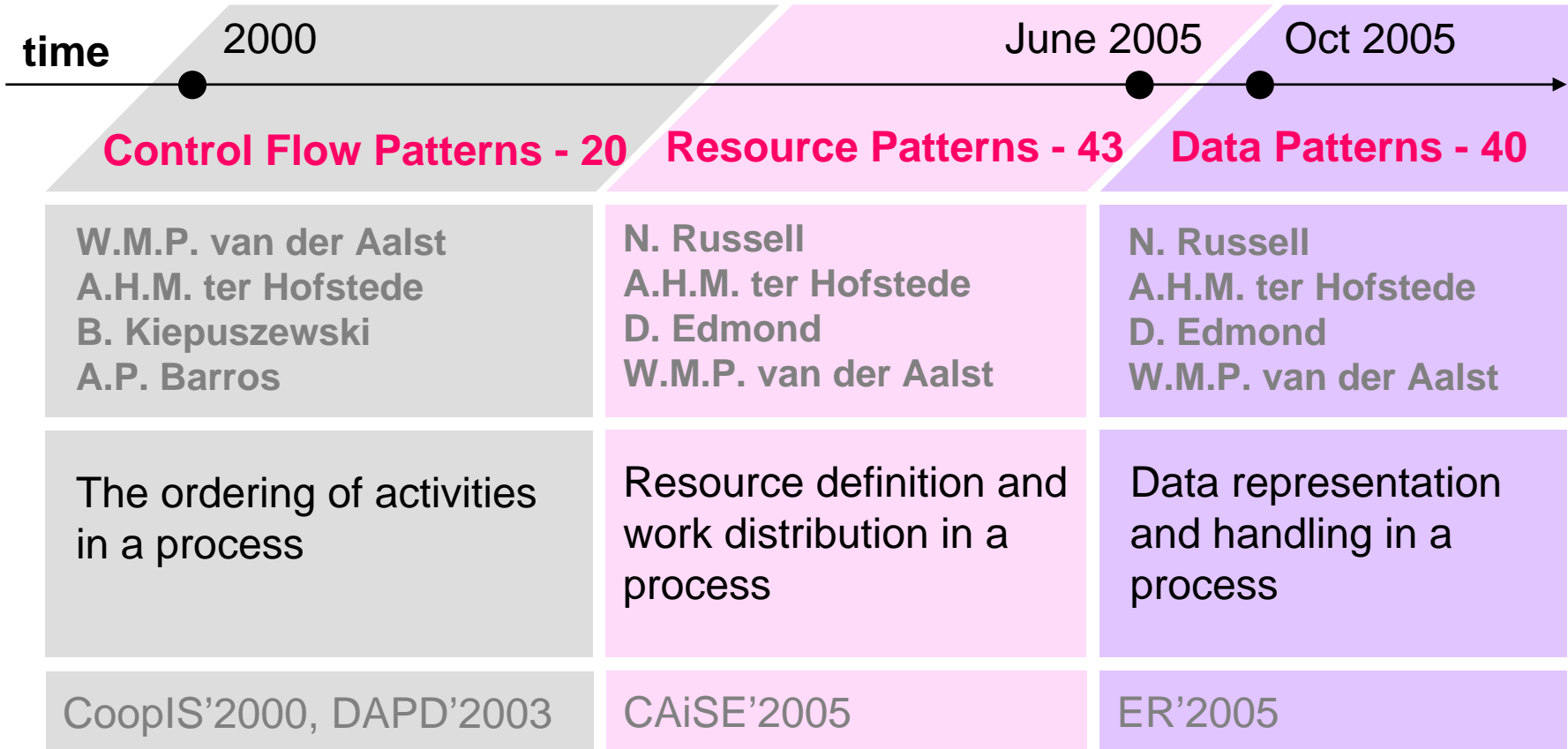
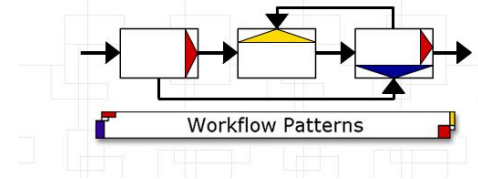
Identification of limitations of UML AD 2.0

Providing elements of reusable knowledge

Pointing out ambiguities in the specification

# The Workflow Patterns Framework

[www.workflowpatterns.com](http://www.workflowpatterns.com)

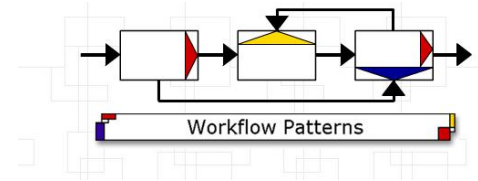


These perspectives follow S. Jablonski and C. Bussler's classification from:

Workflow Management: Modeling Concepts, Architecture, and Implementation. International Thomson Computer Press, 1996

# The Workflow Patterns Framework

[www.workflowpatterns.com](http://www.workflowpatterns.com)



time

2000

June 2005

Oct 2005

**Control Flow Patterns - 20**

**Resource Patterns - 43**

**Data Patterns - 40**

E  
v  
a  
l  
u  
a  
t  
i  
o  
n  
s

COSA  
FLOWer  
Eastman  
Meteor  
Mobile  
I-Flow  
Staffware  
InConcert  
Domino Workflow  
Visual Workflow  
Forte Conductor  
MQSeries/Workflow  
SAR R/3 Workflow  
Verve Workflow  
Changengine

Staffware  
WebSphere MQ  
FLOWer  
COSA  
iPlanet

Staffware  
MQSeries  
FLOWer  
COSA

XPDL, BPEL4WS, BPML, WSFL  
XLANG, WSCI, UML AD 1.4

BPEL4WS

XPDL, BPEL4WS

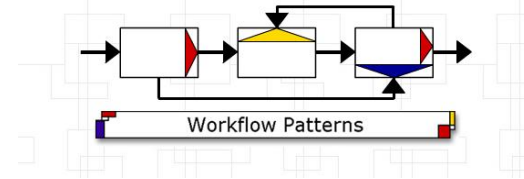
**Language Development: YAWL (Yet Another Workflow Language)**



a university for the **real** world<sup>®</sup>



# Impact of the Workflow Patterns



## Systems inspired or directly influenced by the patterns

FLOWer 3.0 of Pallas Athena

Bizagi of Vision Software

Staffware Process Suite

Pectra Technology Inc.'s tool

Life/A&H Claim System by InsuraPro

Ivolutia Orchestration

OpenWFE (an open source WFMS)

Zebra (an open source WFMS)

Alphaflow (an open source WFMS)

jBpm (a free workflow engine)

## Use of the workflow patterns in selecting a WFMS

the Dutch Employee Insurance Administration Office

the Dutch Justice Department

## Other

Pattern-based evaluations (e.g. ULTRAflow, OmniFlow, @enterprise, BPMN)

Citations (50+ academic papers)

Education (used in teaching at 10+ Universities)

Web site: 190,000+ hits

# Motivation of the Choice of Analysis Framework

- **The Workflow Pattern Framework is**
  - Well tested
  - Provides a sufficient level of granularity for a deep analysis
  - The most complete and powerful framework existing (to our knowledge) for evaluating the capabilities of a **process modelling language**
- **The Bunge-Wand-Weber Ontological Framework**
  - Broader scope, i.e. not specifically focusing on process modelling languages
  - Coarse-grained

# The Control-flow Patterns

## Basic Control-flow Patterns

- Sequence
- Parallel Split
- Synchronisation
- Exclusive Choice
- Simple Merge

## Advanced Synchronisation Patterns

- Multiple Choice
- Synchronising Merge
- Multiple Merge
- Discriminator

## Structural Patterns

- Arbitrary Cycles
- Implicit Termination

## Multiple Instances Patterns

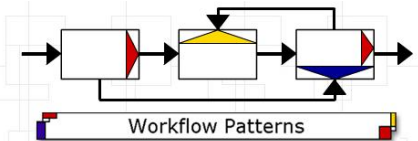
- MI without Synchronisation
- MI with a priory Design Time Knowledge
- MI with a priory Runtime Knowledge
- MI without a priory Runtime Knowledge

## State-based Patterns

- Deferred Choice
- Interleaved Parallel Routing
- Milestone

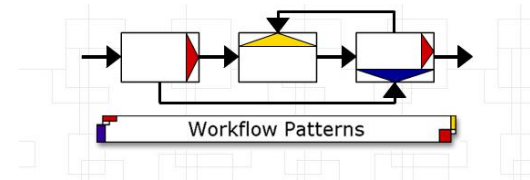
## Cancellation Patterns

- Cancel Activity
- Cancel Case



# WP 16 Deferred Choice

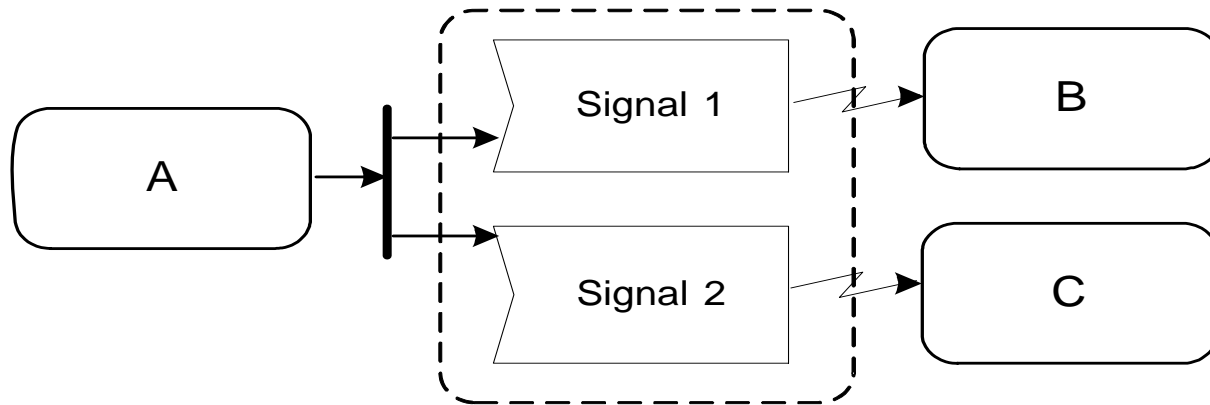
- Choice made by the environment not the system
- Essential in workflow context
- Not widely supported, though its importance seems to be increasingly recognised (e.g. BPEL)
- Naturally supported by notations that offer direct support for the notion of state, e.g. statecharts or Petri nets





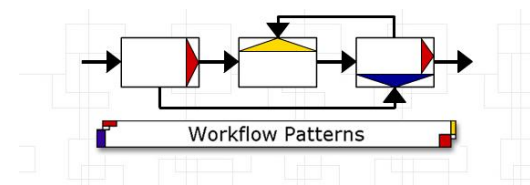
# WP 16 Deferred Choice, cont

## Solution in UML 2.0 AD



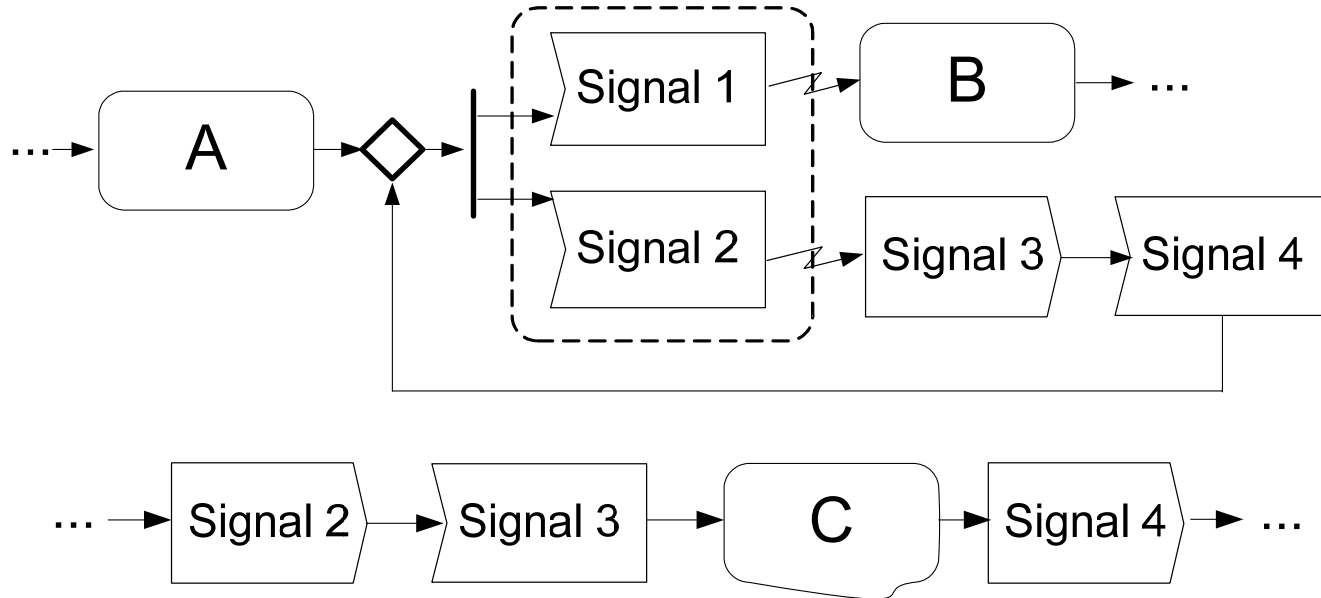
# WP 18 Milestone

- The ability to test whether a certain part of the process is in a certain state
- Not often supported
- Naturally supported by notations that offer direct support for the notion of state, e.g. statecharts or Petri nets



# WP 18 Milestone, cont

## Workaround in UML 2.0 AD



# Results

nr	Pattern	2.0	1.4	nr	Pattern	2.0	1.4
1	Sequence	+	+	11	Implicit Termination	+	-
2	Parallel Split	+	+	12	MI without Synchronisation	+	-
3	Synchronisation	+	+	13	MI with a priory Design time Knowledge	+	+
4	Exclusive Choice	+	+	14	MI with a priory Runtime Knowledge	+	+
5	Simple Merge	+	+	15	MI without a priory Runtime Knowledge	-	-
6	Multiple Choice	+	-	16	Deferred Choice	+	+
7	Synchronising Merge	-	-	17	Interleaved Parallel Routing	-	-
8	Multiple Merge	+	-	18	Milestone	-	-
9	Discriminator	+	-	19	Cancel Activity	+	+
10	Arbitrary Cycles	+	-	20	Cancel Case	+	+

# Recommendations

- Difficulties in supporting State-based patterns
  - Provide the notion of Place (as it exists in Petri nets)
- No support for MI without a priory runtime knowledge
  - Expand the ExpansionRegion notion (e.g., along the lines of the “multiple instance” tasks in YAWL)
- No support for Synchronizing Merge
  - Introduce an OR-join construct (as in YAWL)

# Conclusions

- Detailed analysis of control flow perspective of UML AD 2.0
- Identified shortcomings and provided related recommendations
- **Caveat:** UML not formally defined
- **Note:** The resource and data perspectives of UML have also been subjected to pattern-based analysis

# Thanks

- Questions?