



Intelligent Networks and Management of Distributed  
Systems

# QoS Aggregation for Web Service Composition using Workflow Patterns

Gregor Goldmann, Gero Mühl, Michael C. Jaeger

Berlin University of Technology, Germany





# Overview

- > Basics
- > Application
- > Problem Case
- > Solution





# Composition of Web Services

- > Goal is to arrange individual Web services to form a more complex one (or to realise a workflow or part of a workflow)
- > Mediated vs. brokered vs. peer-2-peer approach
  - > In a workflow scenario, focus is on the mediated composition
- > Different approaches already exist
  - > composition frameworks [hulletal2003], [florescouetal2002]
  - > composition languages (BPEL4WS, WSCI, WSFL, XLANG, ...)
  - > formalisation with Petri nets [martens2004]



# Motivation

- > State of the art
  - > technical challenges of service integration solved
  - > services and components are usually still manually integrated into a workflow
- > Vision for the future
  - > automated integration of services
  - > using semantic information and non-functional attributes
  - > making implicit design information explicit to offer highest degree of automation possible



# Approach

- > Increase degree of automation of ...
  - > ... service selection
  - > ... service integration
- > by using techniques from the Semantic Web and Quality-of-Service-(QoS)-aware middleware to ...
  - > ... describe the service characteristics with semantic networks
  - > ... describe the QoS



# Contribution

- > An algorithm for aggregation of attributes in a composition
  - > a composition of Web services consists of individual services
  - > aggregation of attributes is desired to determine the attributes of the whole composition



# Aggregation of Properties

July 2004



# Classification of QoS Dimensions

- > Classifications exist (e.g. [FrolundKoistinen2002]), but the fact that:
  - > QoS dimensions can have different directions,
  - > the dimensions might be discrete,
  - > might refer to statistical definitions (e.g. percentile),
  - > and are just non-numeric ...
- > ... makes classifications of QoS categories hard to apply for this application
- > Aggregation rules for an abstract set of composition pattern for each QoS dimension.



# Aggregation in a Composition

- > Creation of an abstract composition model
  - > composition of Web services is evolving, changes likely
  - > different composition languages are in competition
- > Classification of structural elements rather than classification of different QoS dimensions
  - > identification of structural elements of a composition is desired (such as a sequence, a loop, etc.)
  - > combinations of the structural elements must be determined (such as a parallel split with a discriminating join)



# The Composition Patterns

- > Many patterns are theoretically possible, but are all relevant?
- > Fortunately, comprehensive work exists already:  
'Workflow Patterns' [aalstetal2003]
  - > identification of common structural and functional characteristics of workflow languages and management systems
  - > strong relevance for the languages to define a composition of Web services [aalst2003]

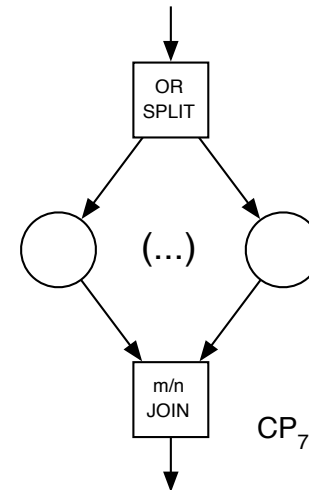
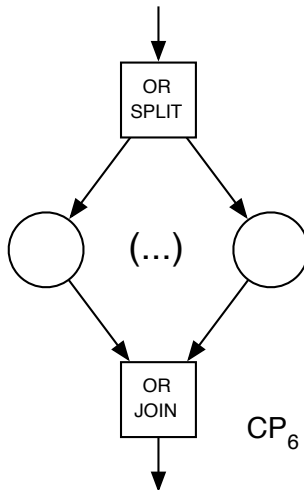
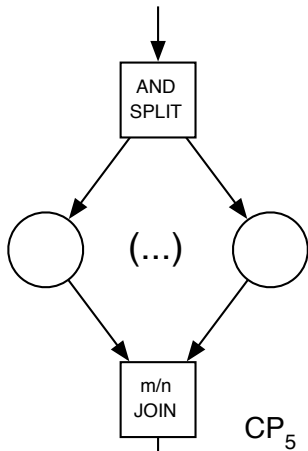
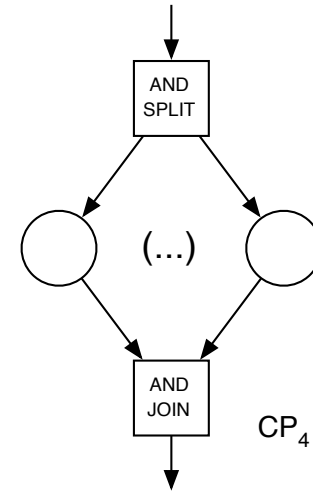
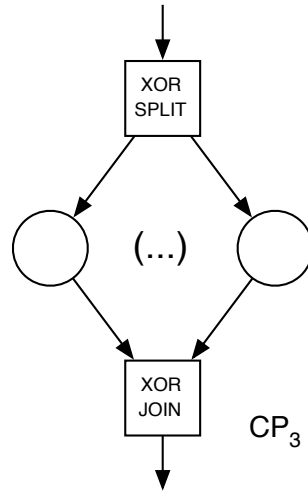
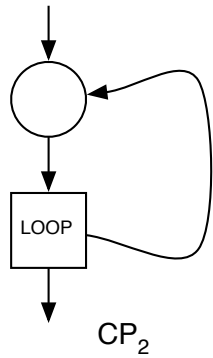
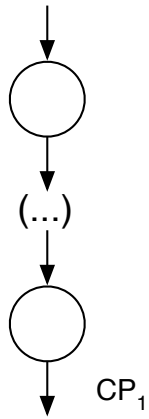


## Sorting out ...

- > Relevant are only patterns with constructs that address structural characterisations of workflow
- > Dropping the not relevant patterns
  - > e.g. implicit termination, milestone, cancellation patterns
- > Elimination of redundancies in the aggregation case
  - > e.g. sequence and unordered sequence
  - > e.g. multiple instances



# The Resulting Patterns



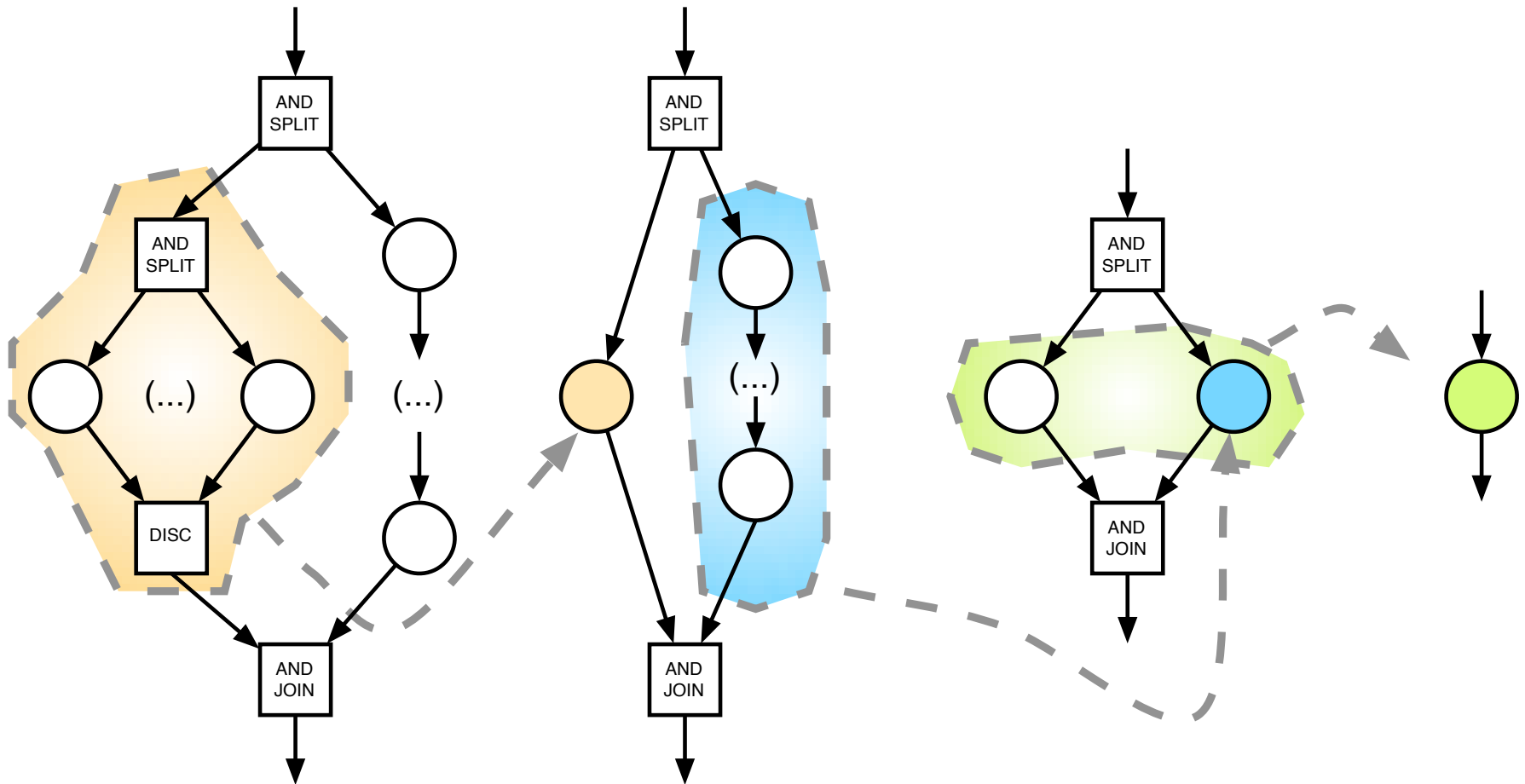


# Assumptions

- > Aggregation of QoS dimensions uses some assumptions:
  - > **trust**: the values are correct
  - > **independence**: individual services execute independently from each other
  - > **uniformity**: all values are referring to the same definition
  - > **equipartition**: execution of services in conditional cases has simple distribution



# An Example





# Example Aggregation Rules

#	Comp. Pattern	Max. Execution Time	Min. Execution Time
Sequential Composition Patterns			
1	Sequence	$x_a = \sum_{i=1}^n x_i$	$x_a = \sum_{i=1}^n x_i$
2	Loop	$x_a = kx$	$x_a = kx$
Parallel Composition Patterns			
3	XOR-XOR	$x_a = \max\{x_1, \dots, x_n\}$	$x_a = \min\{x_1, \dots, x_n\}$
4	AND-AND	$x_a = \max\{x_1, \dots, x_n\}$	$x_a = \max\{x_1, \dots, x_n\}$
5	AND-DISC	$x_a = \max\{x_1, \dots, x_n\}$	$x_a = \min\{x_1, \dots, x_n\}$
6	OR-OR	$x_a = \max(\mathcal{P}(\mathbb{T}))$	$x_a = \min(\mathcal{P}(\mathbb{T}))$
7	OR-DISC	$x_a = \max(\mathcal{P}(\mathbb{T}))$	$x_a = \min(\mathcal{P}(\mathbb{T}))$



## Related Work

- > Aggregation of properties is addressed in different other fields:
  - > bandwidth in computer networks
  - > execution time in software components for real time applications [PuschnerSchedl1997]
  - > QoS aggregation in workflows [Cardoso2002]
- > related work does not focus on Web service composition, no pattern oriented approach



## Conclusions & Outlook

- > Contributions are the composition patterns and the aggregation rules approach for particular QoS dimensions
- > Our work has demonstrated an approach to aggregate service properties in compositions
- > Many future research opportunities
  - > modelling dependencies
  - > addressing trust issues for the values
  - > using aggregation during run-time for controlling
  - > using formalisation (e.g. by using Petri nets [martens2004]) to verify formal characteristics of the composition



# Thank You.

Please do not hesitate to ask and comment ...

Michael C. Jäger  
mailto:mcj@ivs.tu-berlin.de  
FG Intelligente Netze und Management verteilter  
Systeme  
Technische Universität Berlin

Einsteinufer 17, Sek EN-6  
10587 Berlin, Germany  
<http://ivs.tu-berlin.de/>



# Selected References

W.M.P. van der Aalst, A.H.M. ter Hofstede, A. Kiepuszewski, and A.P. Barros. Advanced workflow patterns. In 7th International Conference on Cooperative Information Systems (CoopIS 2000), volume 1901 of Lecture Notes in Computer Science, pages 18–29. Springer-Verlag, Berlin, 2000.

W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow patterns. *Distributed and Parallel Databases* 14(3), pages 5–51, 2003.

Jorge Cardoso. Quality of Service and Semantic Composition of Workflows. PhD thesis, Department of Computer Science, University of Georgia, Athens, GA (USA), 2002.

Svend Frolund and Jari Koistinen. Quality of service specification in distributed object systems design. *Distributed Systems Engineering Journal*, 5(4), December 1998. <http://www.w3.org/2001/sw/Activity>, 2003.

Daniela Florescu, Andreas Gruenhausen, and Donald Kossman. XL: an XML Programming Language for Web Service Specification and Composition. In *Proceedings of the 11th international conference on World Wide Web*, pages 65–76. ACM Press, May 2002.

Richard Hull, Michael Benedikt, Vassilis Christophides, and Jianwan Su. E-services: A look behind the curtain. In *Proceedings of the 22nd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2003)*, San Diego, USA, June 2003. ACM Press.

King-Shan Lui; Nahrstedt, K.; Shigang Chen. Routing with topology aggregation in delay-bandwidth sensitive networks *Networking, IEEE/ACM Transactions on*, Vol.12, Iss.1, Feb. 2004  
Pages: 17- 29

Peter Puschner and Anton Schedl. Computing maximum task execution times - a graph-based approach. *Journal of Real-Time Systems*, 13(1):67–91, July 1997.