

# A Theory of Service Behavior



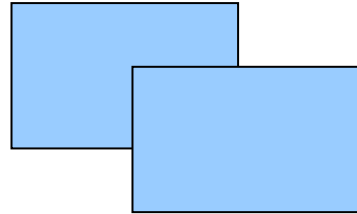
Karsten Wolf



UNIVERSITÄT ROSTOCK

# Our Theory

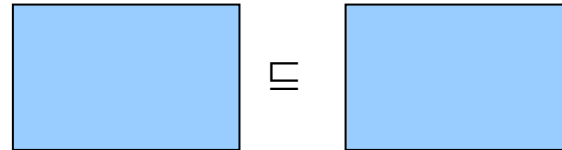
- Objects



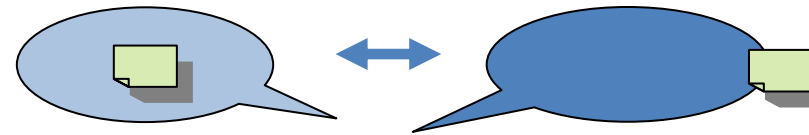
- Operations



- Relations



- Interfaces to other aspects

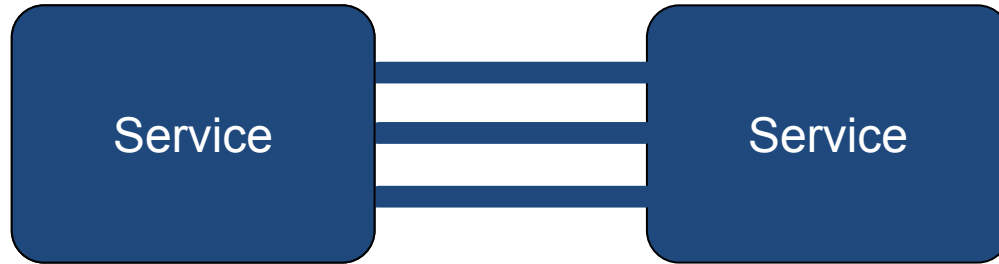


- Problems

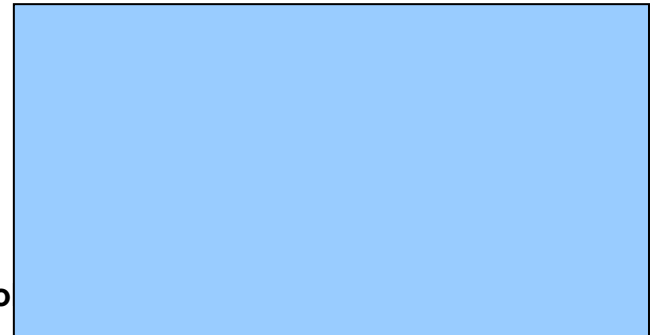
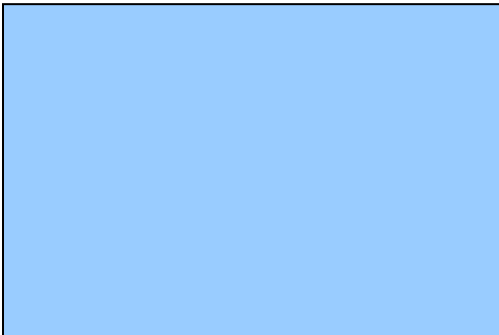
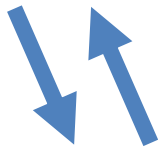


# Objects

BPEL  
BPMN  
UML



BPMN 2.0  
WS-CDL  
Let's Dance  
(BPEL4Chor)



read more: [www.service-techno](http://www.service-techno)

# Operations

- Simple

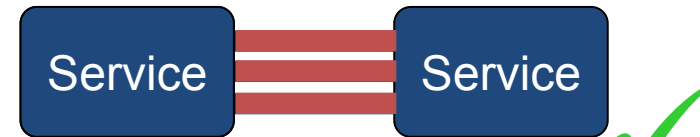
- Composition



- Mirror,



- Extraction of *realized* choreography



- Model checking completed systems



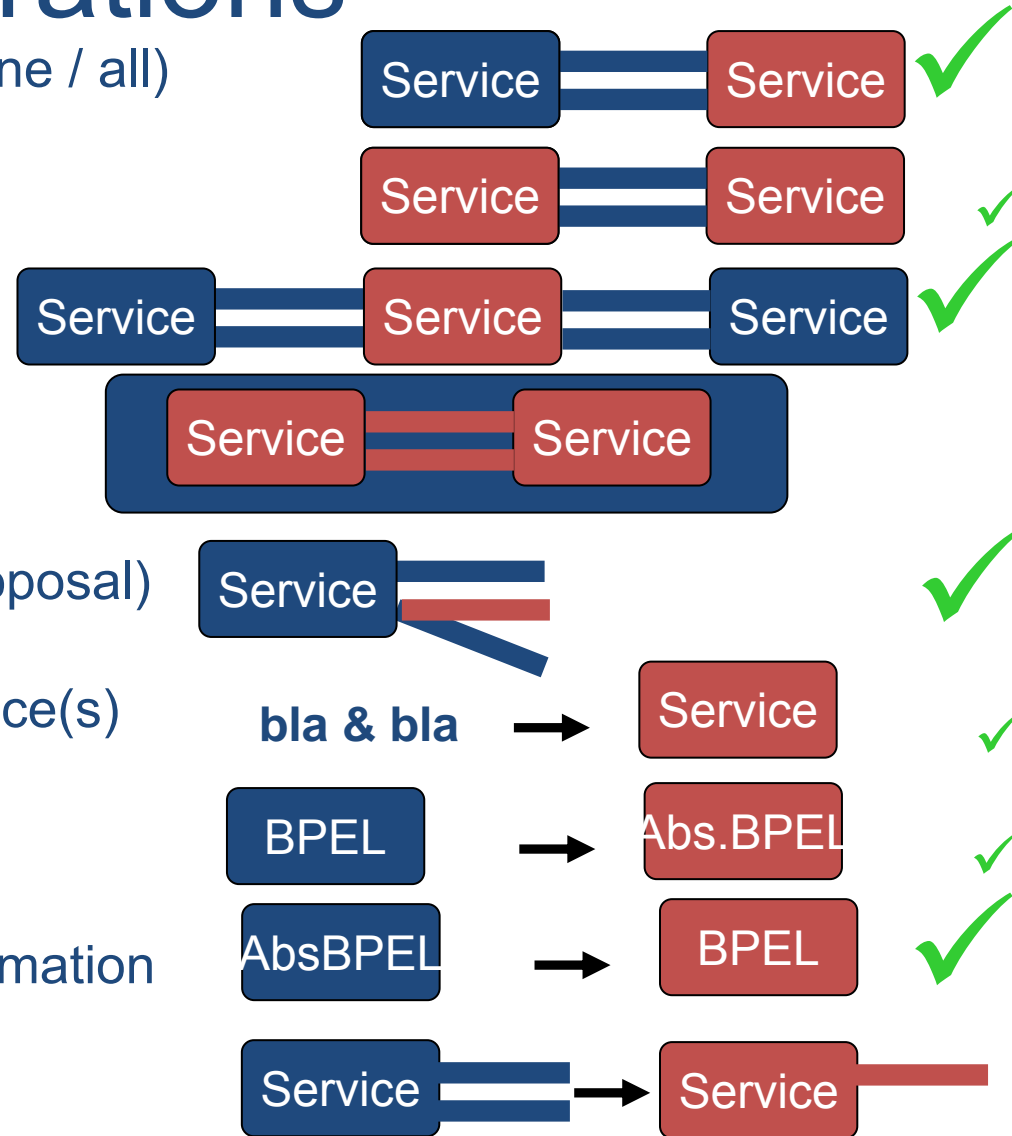
- Soundness of component



# Operations

## Behavioral

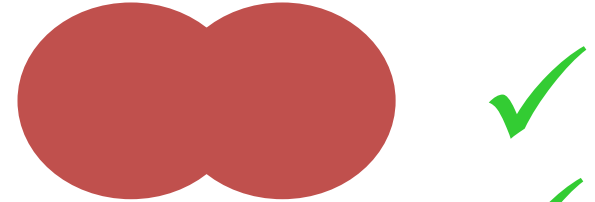
- Service → compatible partner (one / all)
- Choreography → realizing peers
- Services → adapter
- Decomposition
- Service → corrected service (proposal)
- Declarative specification → service(s)
- Service → Public View
- Compatibility preserving transformation
- Projection



# Operations

- Set

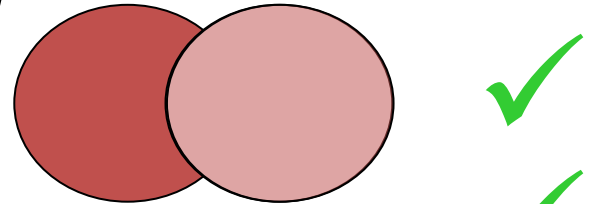
- union



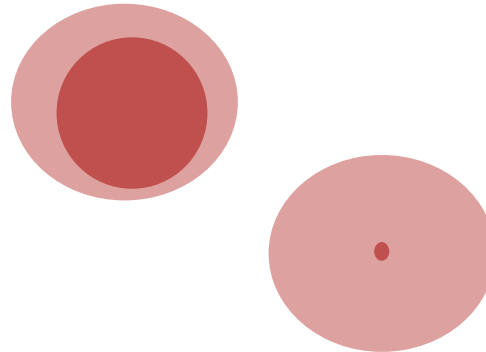
- intersection



- complement

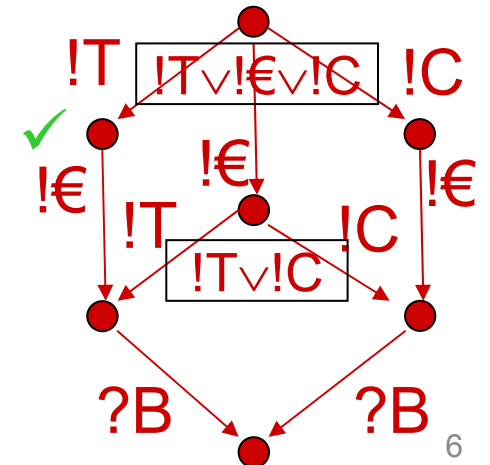


- inclusion check



- emptiness check

- on finite representations of sets



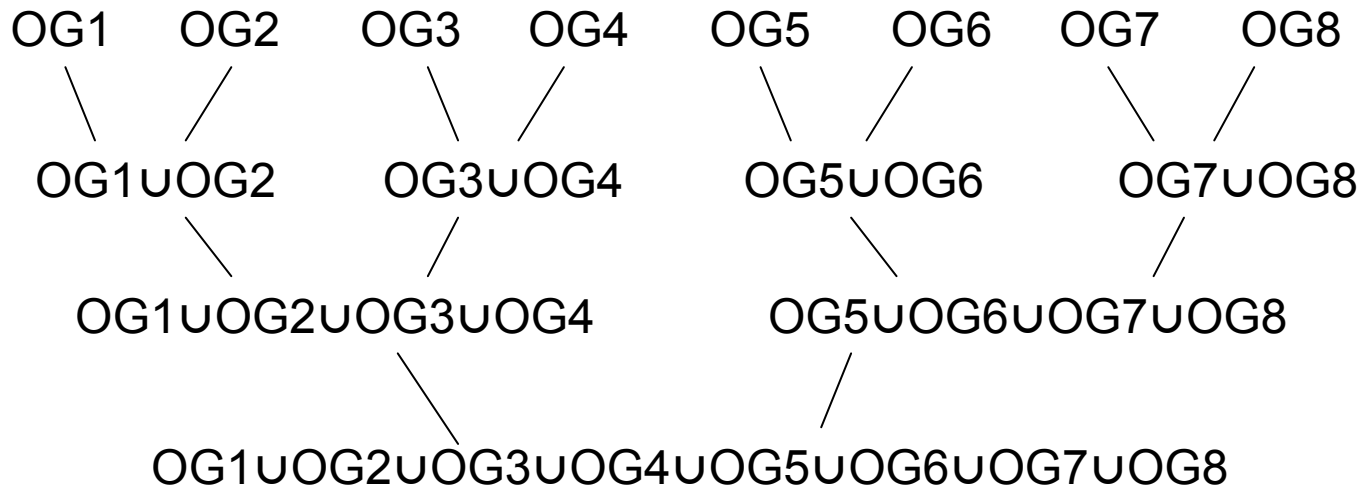
# Example: Substitutability

- Can S be exchanged with S'?
    - $\text{CompPartners}(S) \sqsubseteq \text{CompPartners}(S')$
    - $\text{CompPartners}(S) \cap \overline{\text{CompPartners}(S')} = \emptyset$
- Counterexamples!

# Example: Behavioral Querying

- Express properties as annotated automata:
  - “a precedes b”
  - “c does not occur”
  - “d may occur”
  - “e will occur”
- Set algebra implements boolean connectives
  - conjunction – intersection
  - disjunction – union
  - negation – complement

# Example: Behavioral Discovery



log n membership checks

instead of n

# Relations

- What do we want?
  - Every compatible partner of  $S$  is compatible to  $S'$

$\rightarrow \approx$  „should testing equivalence“



- similar notion for choreographies

# Interface to Semantics

- So far: Messages named a,b,c,d, ... ... no relation
- But what if
  - a = empty form, b = filled form
  - a = X's password

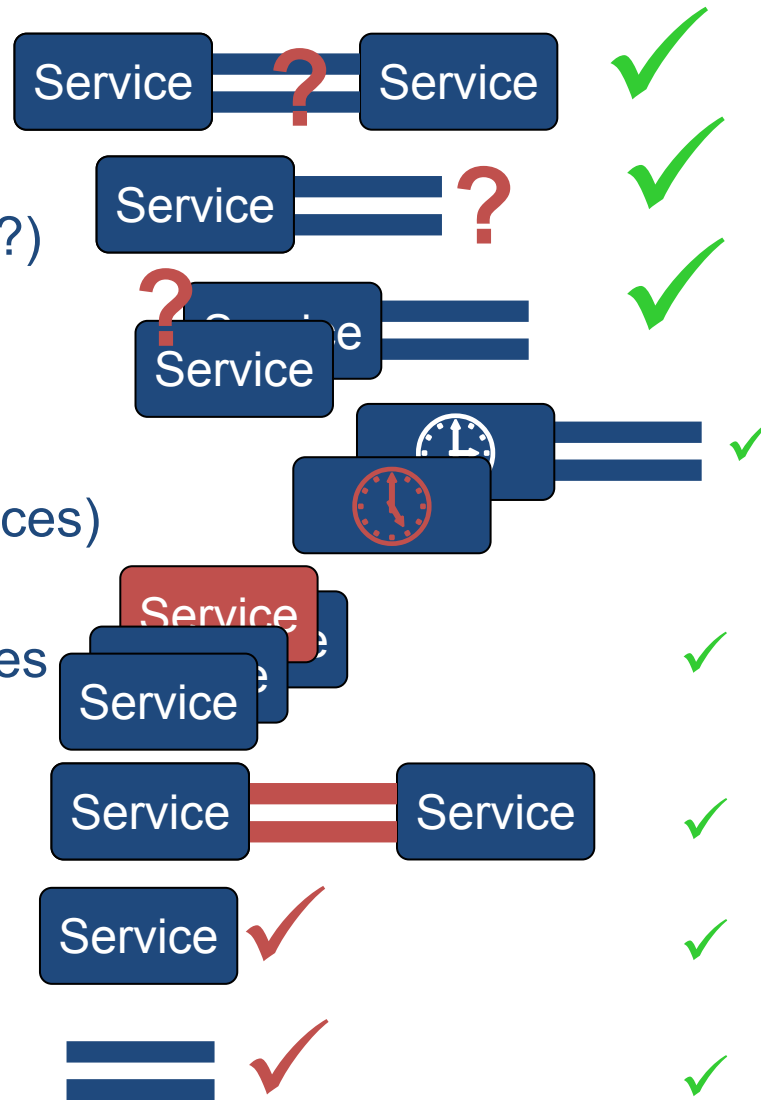
# Activities

Activity	Specification	Example pro	Example con
create	$\rightarrow a$	simple ack, owned information	password, foreign information
transform	$a \rightarrow b$	meter $\rightarrow$ feet zip code $\rightarrow$ city name	meter $\rightarrow$ SFR
delete	$a \rightarrow$	electronic message	real item legally relevant document
split	$a \rightarrow b c d$ ( $a \rightarrow a b c d$ )	compound to elementary	
merge	$a b c \rightarrow d$ ( $a b c \rightarrow a b c d$ )	elementary to compound	
recombine	$a b c \rightarrow d e f$		
copy	$a \rightarrow a a$	electronic message	single use key real item



# Problems covered:

- compatibility verification
- controllability (is there a compatible partner?)
- substitutability (can I exchange a service?)
- instance migration (same for running instances)
- navigation/querying in behavioral repositories
- service composition, adaptation
- validation, correction, test
- realizability, desynchronizability



# Conclusion

**Yes we can!**