Verteidigung zum großen Beleg über das Thema:

Muster bei der Kombination von Objektdiensten mit Anwendungssystemen

Andreas Mucha
Lehrstuhl Softwaretechnologie
Institut für Software- und Multimediatechnik
Technische Universität Dresden

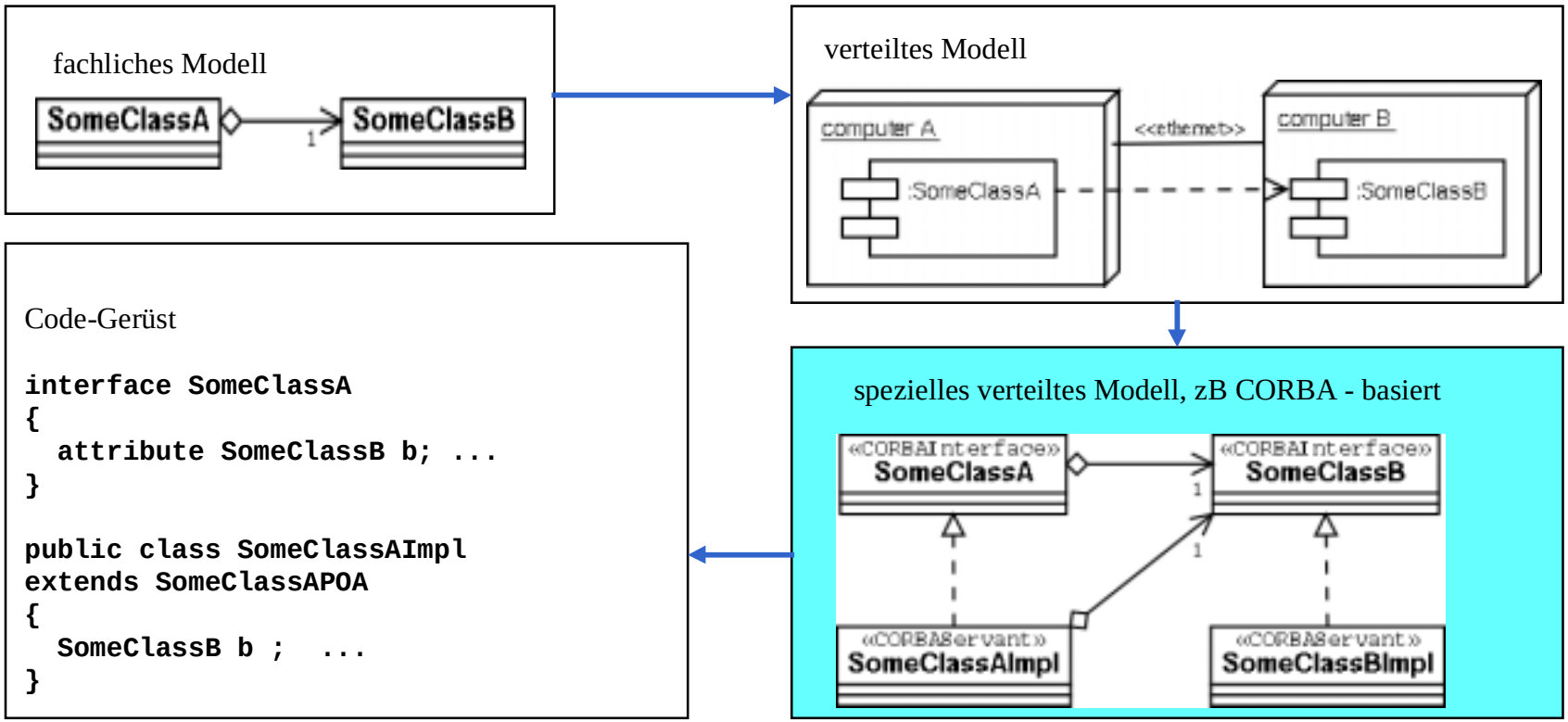
August 2001

Inhalt

- Kontext und Zielstellung der Arbeit
- Verknüpfungsmuster und Rollen von Dienstobjekten
- Vorgehensweise
- Ergebnisse der Arbeit
- Ausblick

Kontext der Arbeit

Idee eines Modellierungskonzeptes für verteilte Anwendungen

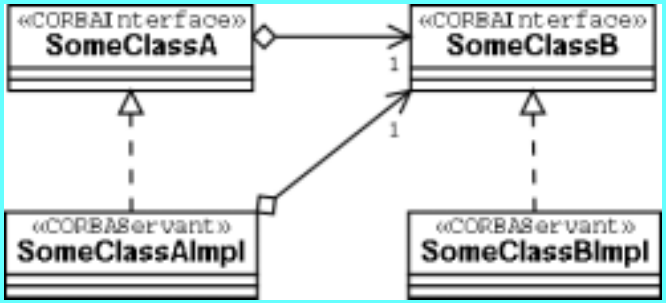


Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

Kontext der Arbeit

Idee eines Modellierungskonzeptes für verteilte Anwendungen

spezielles verteiltes Modell, CORBA - basiert



- ***CORBA-Dienste / Objekte***
- CORBA-Facilities
- Modellierung in UML
- Codegenerierung aus dem UML - Modell

- Verknüpfung von CORBA-Diensten bzw. CORBA-Objekten mit Anwendungssystemen,
- Modellierung in UML

↓ Arbeit von Mike Fischer

typische Verwendung von Dienstobjekten, Verknüpfungsmuster

Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

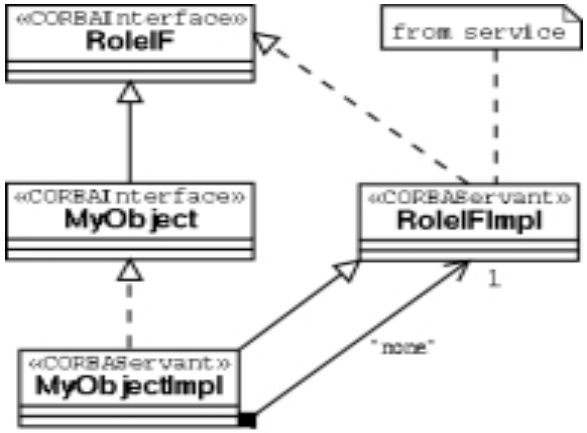
Zielstellung der Arbeit

- Überprüfung und, falls erforderlich, Erweiterung der vorgeschlagenen Verknüpfungsmuster anhand einer Beispielanwendung
- gegebenenfalls neue Muster identifizieren und dokumentieren
- typische Rollen für Dienstobjekte identifizieren
- typische Verknüpfungsmuster für Dienstobjekte identifizieren

Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

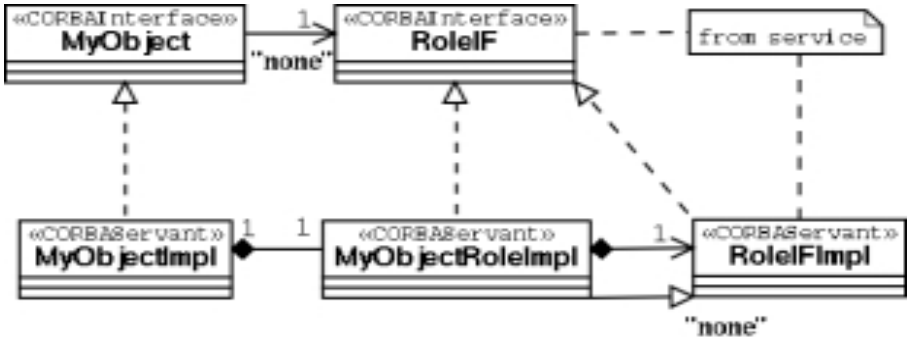
Verknüpfungsmuster

generalization




```
serviceRoleRealization = generalization

serviceRoleRealizationProperties =
[ useNoneImpl |
  useImplByGeneralization |
  useImplByDelegation ]
```

delegation

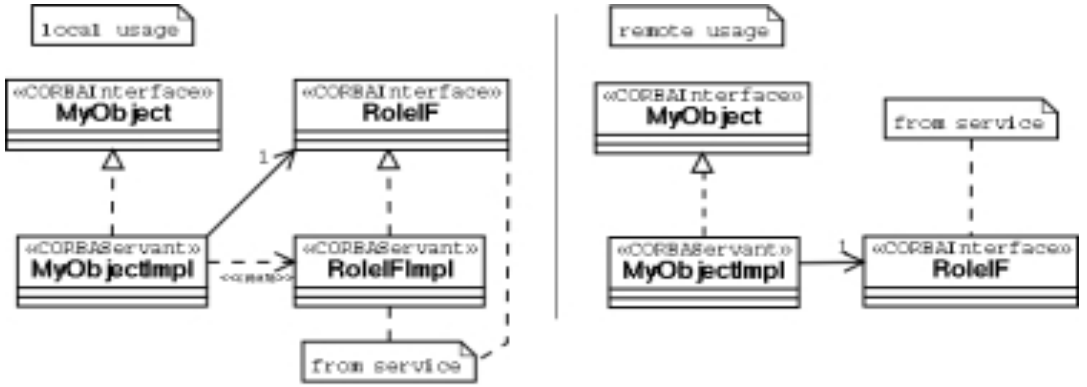



```
serviceRoleRealization = delegation

serviceRoleRealizationProperties =
[
  [ useNoneImpl |
    useImplByGeneralization |
    useImplByDelegation]
  +
  [ hideDelegation | exposeDelegation ]
]
```


Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

Verknüpfungsmuster

usage



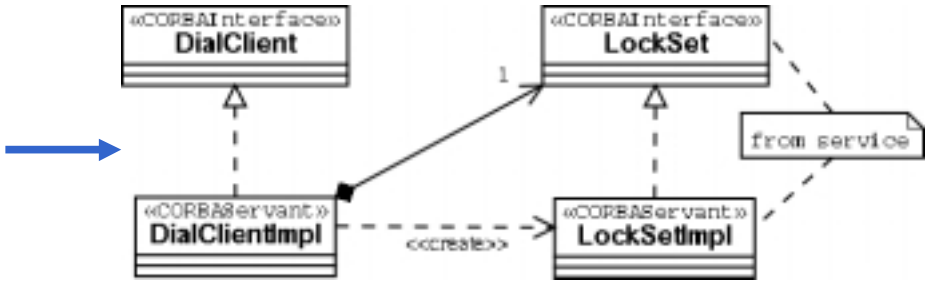

```
serviceRoleRealization = usage

serviceRoleRealizationProperties =
[ localUsage | remoteUsage ]
```

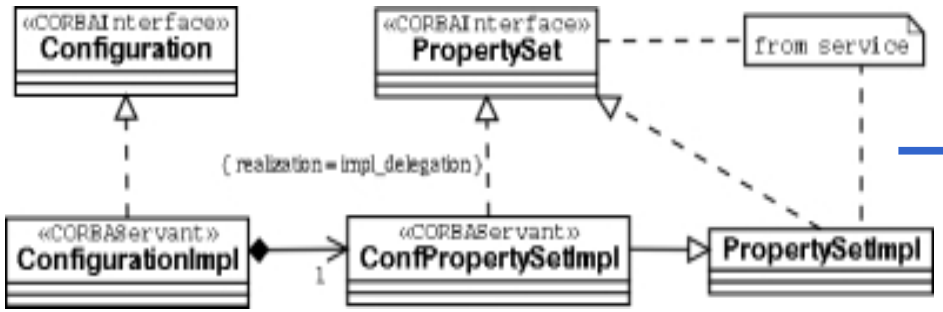
Rollen und Verknüpfungsmuster von Dienstobjekten

Rolle	**Interface**	**Verknüpfungs-muster**	**Einschränkungen für Eigenschaften**	**weitere benötigte Dienstobjekte**
Name-Context	Name-ContextExt . . .	usage	-	NameComponent

Vorgehensweise

```
public class DialClientImpl implements
DialClientOperations
{
    ...
    private LockSet m_Lock = null ;
    ...
    void method(...)
    {
        LockSetFactory fac = null ;
        ...
        m_Lock = fac.create();
        ...
```






```
Public class ConfigurationImpl extends
ConfigurationPOA
{
    private ConfPropertySetImpl m_Prop = null;
    ....
    Public ConfigurationImpl(...)
    {
        m_Prop = new ConfPropSetImpl(...) ;

public class ConfPropertySetImpl extends
PropertySetImpl
{
    ...
```

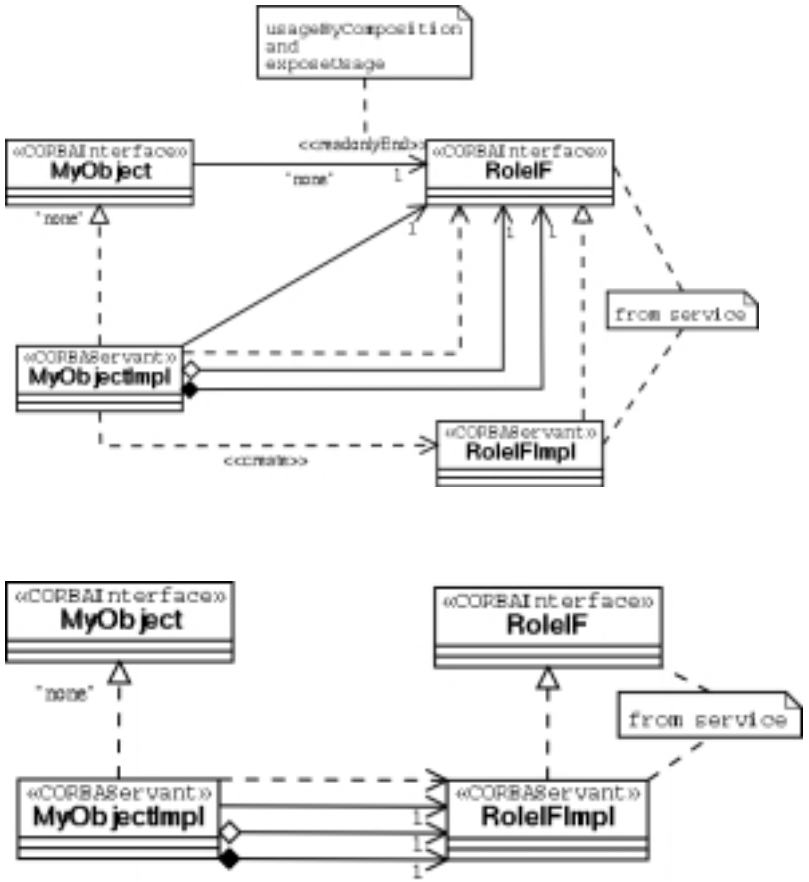

Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

Ergebnisse der Arbeit

Erweiterungen der Verknüpfungsmuster
usage

```
serviceRoleRealizationProperties =
[
   [ remoteUsage | localUsage ]
   +
   [ hideUsage | exposeUsage ]
   +
   [ usageByDependency | usageByAssociation |
     usageByAggregation | usageByComposition ]
   +
   [ corbaServant | noneCorbaServant ]
   +
   [ useInterface | useServant ]
]
```

Einführung des Stereotyp `<<readonlyEnd>>`

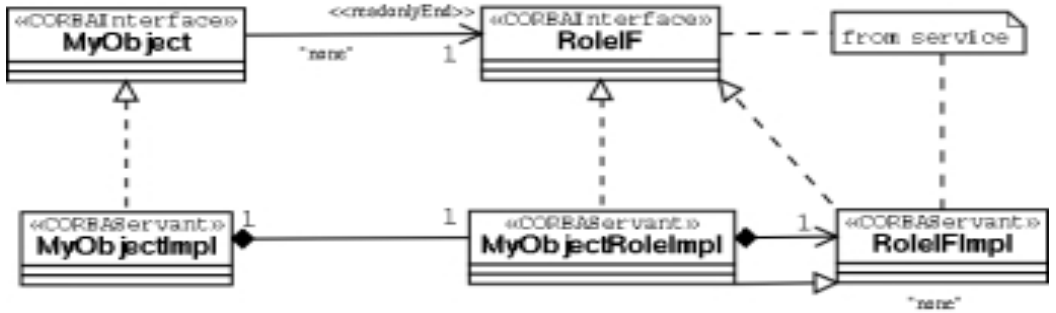


Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

Ergebnisse der Arbeit

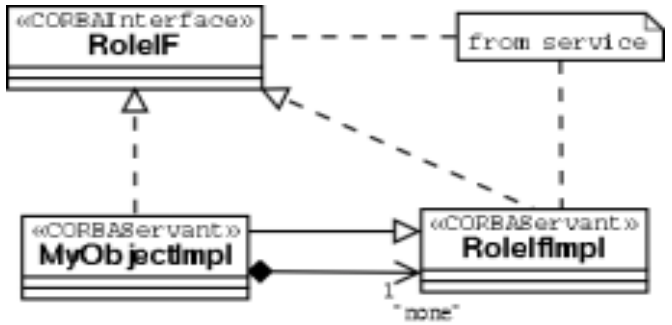
Erweiterungen der Verknüpfungsmuster
delegation

Einführung des Stereotyp `<<readonlyEnd>>`



Das *realization*-Verknüpfungsmuster

```
serviceRoleRealization = realization

serviceRoleRealizationProperties =
[ useNoneImpl |
  useImplByGeneralization |
  useImplByDelegation ]
```



Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

Ergebnisse der Arbeit

Auszug aus der Rollentabelle

-Verknüpfungsmuster *usage* tritt am häufigsten auf

-2 Klassen :

schwach koppelnde Verknüpfungsmuster: *usage*

stark koppelnde Verknüpfungsmuster: *realization, delegation, generalization*

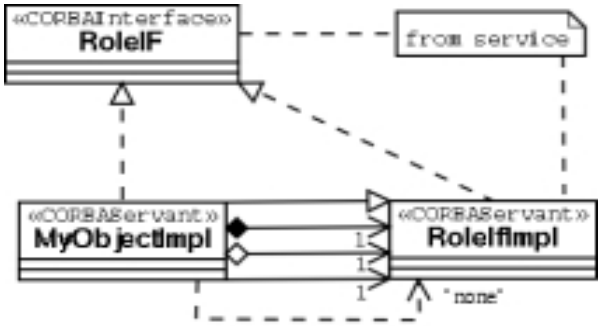
Rolle	**Interface**	**Verknüpfungs-muster**	**Einschränkungen für Eigenschaften**	**weitere benötigte Dienstobjekte**
Collection	SortedSet	**usage** generalization delegation realization	-	Iterator
Collection-Factory	SortedSet-Factory . . .	**usage** generalization delegation realization	-	Operations
Operations	Operations . . .	**realization** generaliaztion	useNoneImpl	-
TimeService	TimeService	**usage**	-	-
UTO	UTO	**usage**	-	-
Lock	LockSet	**usage**	-	-
LockFactory	LockSetFactory	**usage**	-	-
Relationship	Relationship	usage **generalization** delegation realization	-	NamedRole Relationship
Relationship-Factory	Relationship-Factory	usage **generalization** delegation realization	-	NamedRole NamedRoleType Relationship

Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

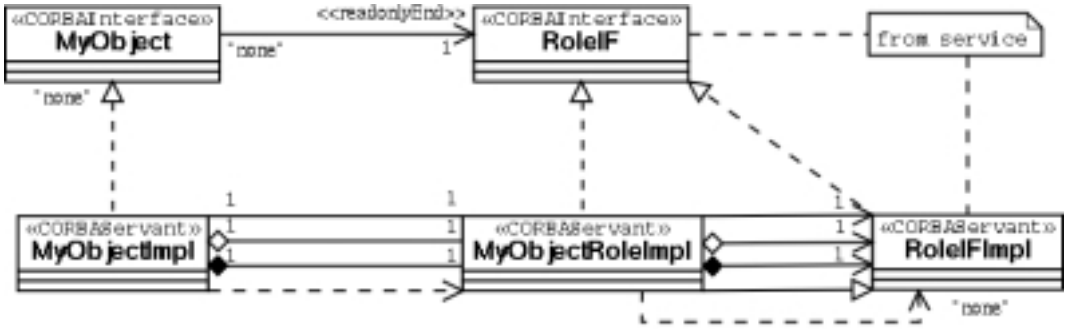
Ergebnisse der Arbeit

Mögliche Erweiterungen der Verknüpfungsmuster, ausgehend vom *usage* - Verknüpfungsmuster

Erweiterungen des
realization-Verknüpfungsmusters



Erweiterungen des
delegation-Verknüpfungsmusters

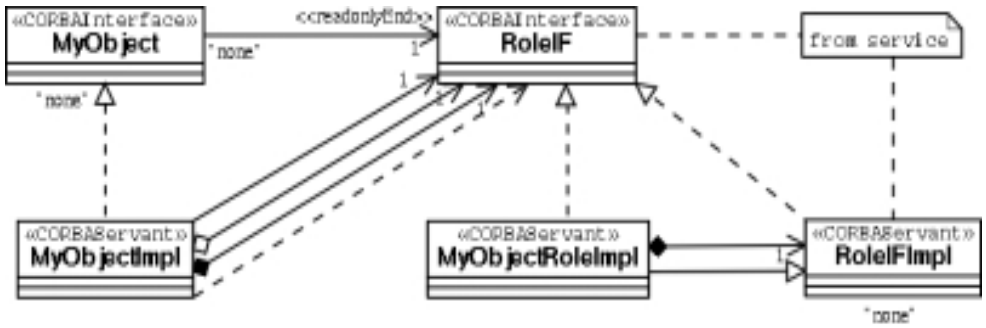


Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

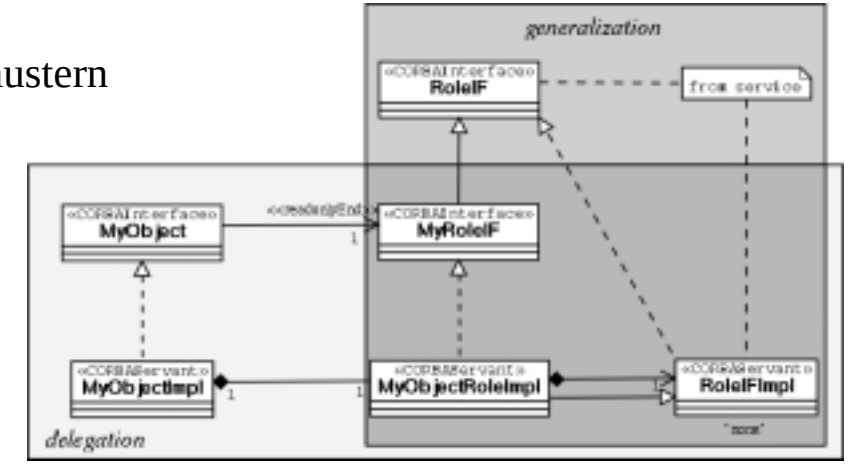
Ergebnisse der Arbeit

Mögliche Erweiterungen der Verknüpfungsmuster ausgehend vom *usage* - Verknüpfungsmuster

Erweiterungen des
delegation-Verknüpfungsmusters



Kombinationen von Verknüpfungsmustern



Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik

Ausblick

- Erweiterungen der Verknüpfungsmuster sinnvoll bzw. relevant ?
- Formulierung der Einschränkungen für Wertebelegung der Tags
- andere Multiplizitäten möglich bzw. sinnvoll ?
- Rollen überprüfen, gegebenenfalls erweitern ,
  weitere Dienste untersuchen
- typische Interaktionen von Dienstobjekten mit Anwendungssystemem ?
- Interaktionen als Sequenzdiagramme → Interaktionsmuster ?

- Transformation von Modellen
- Codegenerierung
- RMI , DCOM ... ?

Andreas Mucha, Technische Universität Dresden, Lehrstuhl Softwaretechnologie,
Institut für Software - und Multimediatechnik