Customizing UML for Component Design

Marc Born

GMD Fokus

born@fokus.gmd.de

Olaf Kath

Humboldt-Universität zu Berlin

kath@informatik.hu-berlin.de

www.dot-profile.de

Motivation

- Definition of a design method for component-based distributed (telecommunication-) systems
  - System realized by interworking of autonomous distributed entities
  - Interactions and behavior of entities supported by different middleware platforms
  - Support for all relevant concepts of application domain within design method, e.g.
    - different interaction kinds, e.g. continuous media interaction
    - design support for non-functional object interaction aspects, e.g. secure or transactional binding

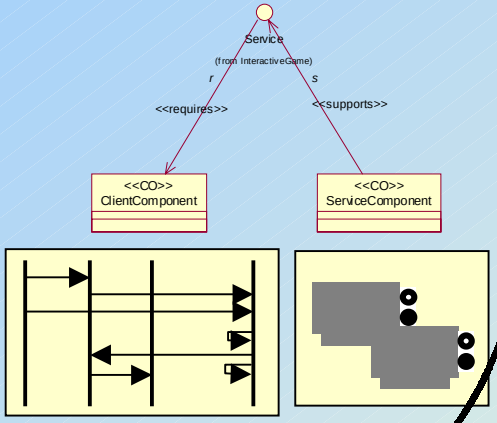
Motivation & Goals (ctnd.)

- Provision of tool support for design method
  - (graphical) notation support
  - Generation of middleware platform specific code, e.g. interface definitions and implementation code
  - Simulation of systems behavior prior to implementation (whereever possible)
- Enabling design reuse
  - Enable import of IDL 2.3.1 specs
  - Allow for migration of other existing designs (e.g. TINA specifications, Z.130) to emerging platforms

General Approach

Concept Space
System entities
and relations

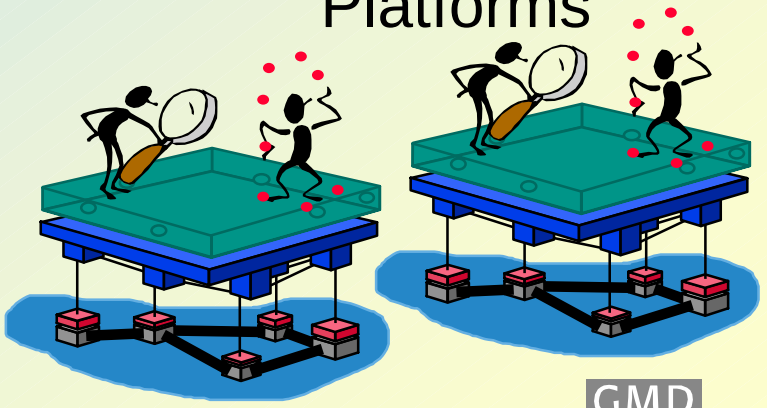
Representation of Concepts



Notations

Mapping of Concepts (Code Generation)

Middleware Platforms



HUMBOLDT-UNIVERSITÄT ZU BERLIN

GMD FOKUS

Concept Space Definitions

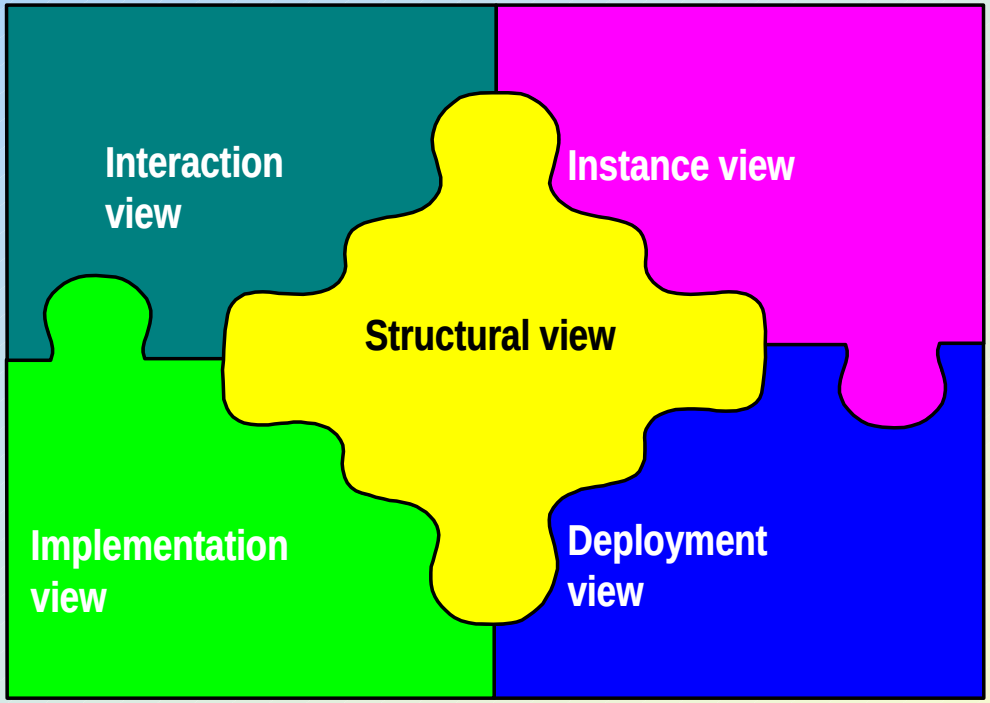
- Foundation: Discerning evaluation of current approaches
  - RM-ODP, X.901 - 904
  - Rational Unified Process
  - TINA Computational Modeling Concepts
  - CORBA Component Model
  - Enterprise Java Beans
  - ITU ODL, Z.130

Concept Space Definitions

- Basic concepts
  - Structural aspects
    - Computational object, interfaces, types, interaction kinds
  - Configuration issues
    - Single ports (like CCM), multiple ports (to aquire references dynamically)
  - Interaction rules
    - Binding and binding rules
  - Implementation structure
    - Artifacts, implementation elements
  - Deployment aspects
    - Component, assembly, deployment „constraints

Concept Space Definitions

- Basic concepts
- Structured into views



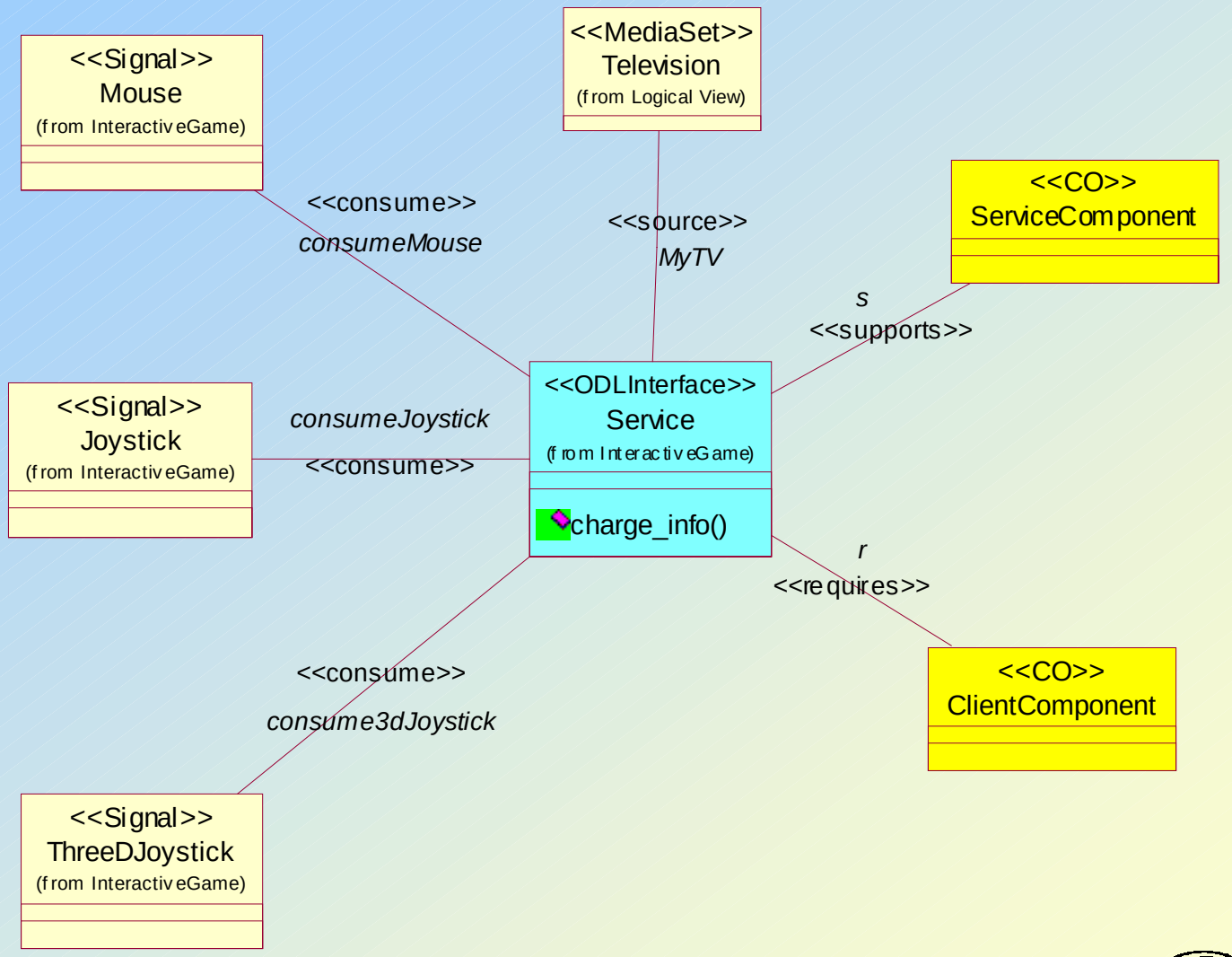
Notation

- More then one concrete notation possible
- Candidate Unified Modeling Language:
  - Object-oriented approach
  - Graphical notation
  - Metamodel based
  - Built in extension mechanism
  - Wide spread + existing tool support
- Approach: Definition of UML Profile
  - Extension of metamodel for defined concepts
  - Introduction of stereotypes, tagged values, constraints
  - Realization of profile with existing UML tool

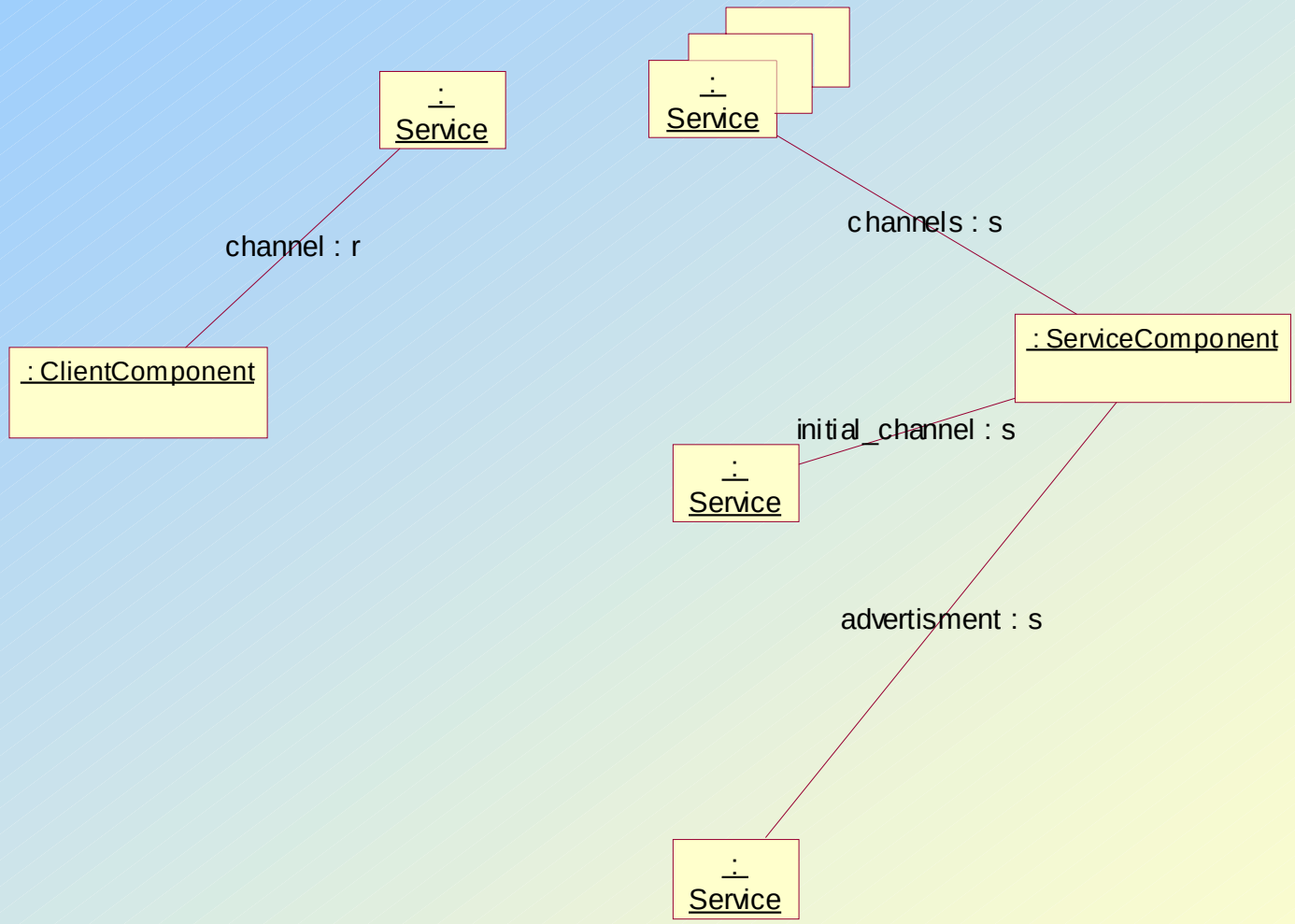
Example

- Hypothetical Task: Devlop an *Interactive TV Service* component
  - provide a number of combined audio/video channels to clients
  - shall be able to receive input from its clients in form of joystick and mouse events.

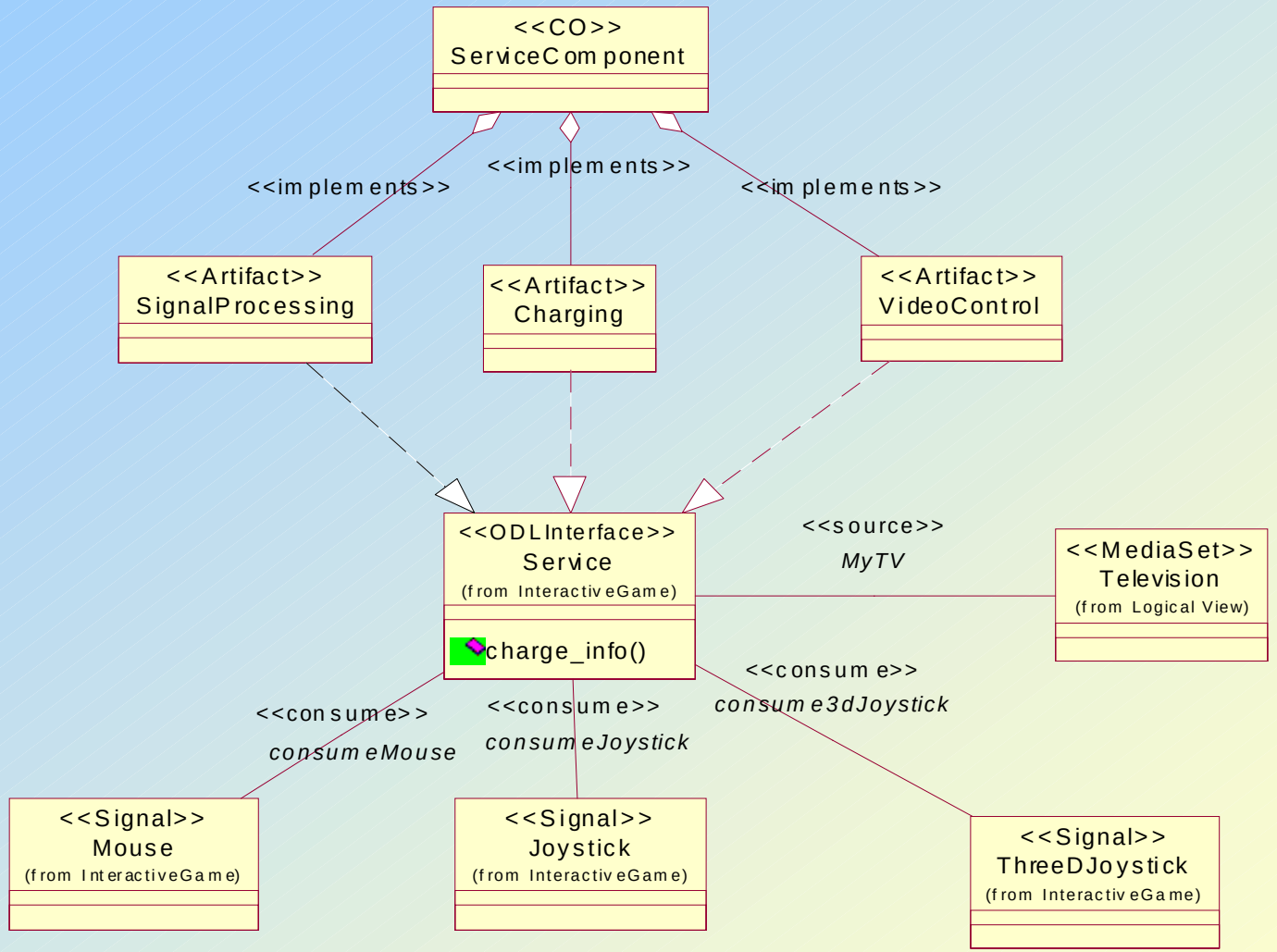
Structural Specifications



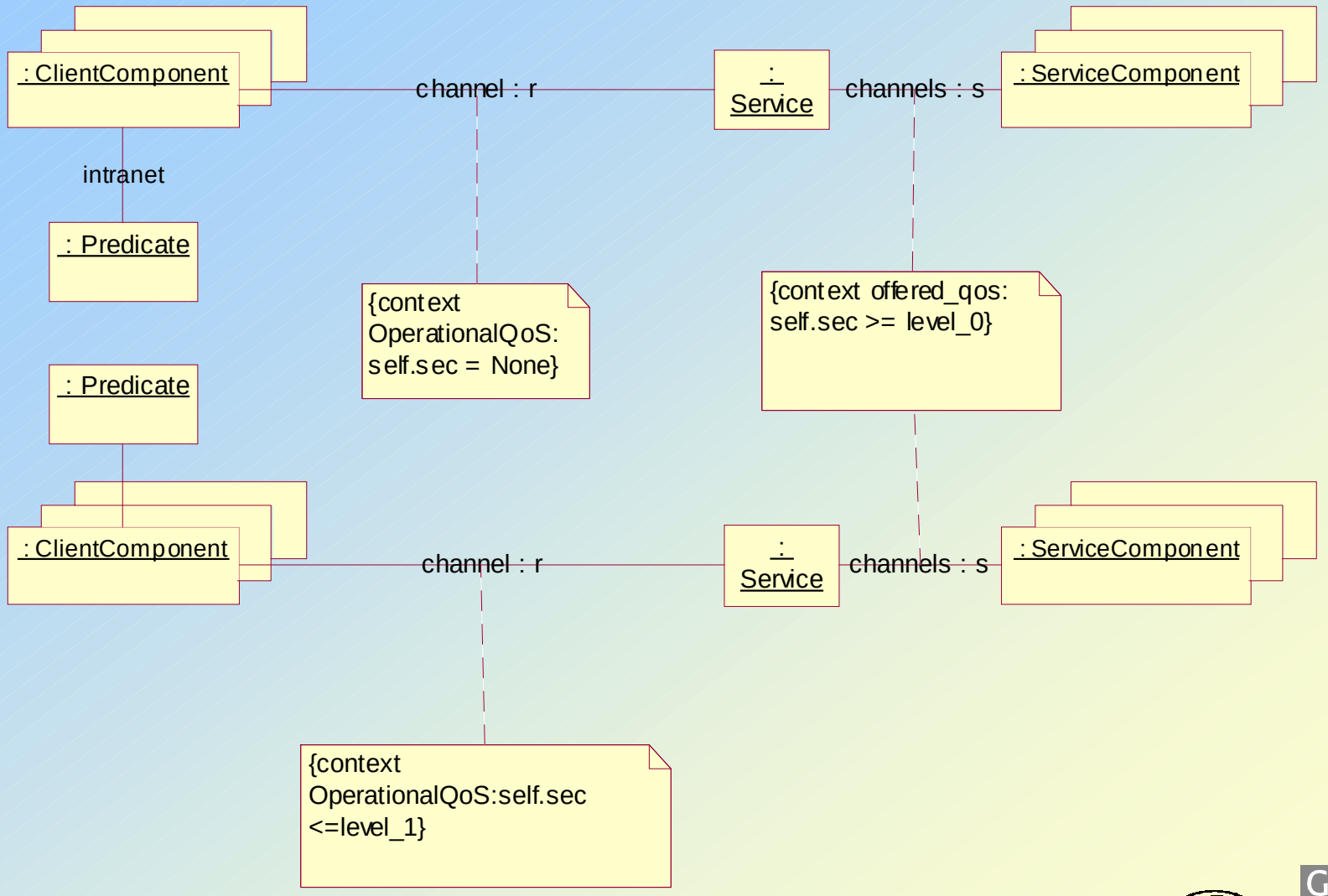
Configuration Specification



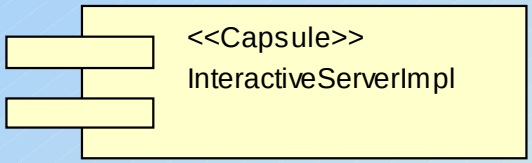
Implementation Specification

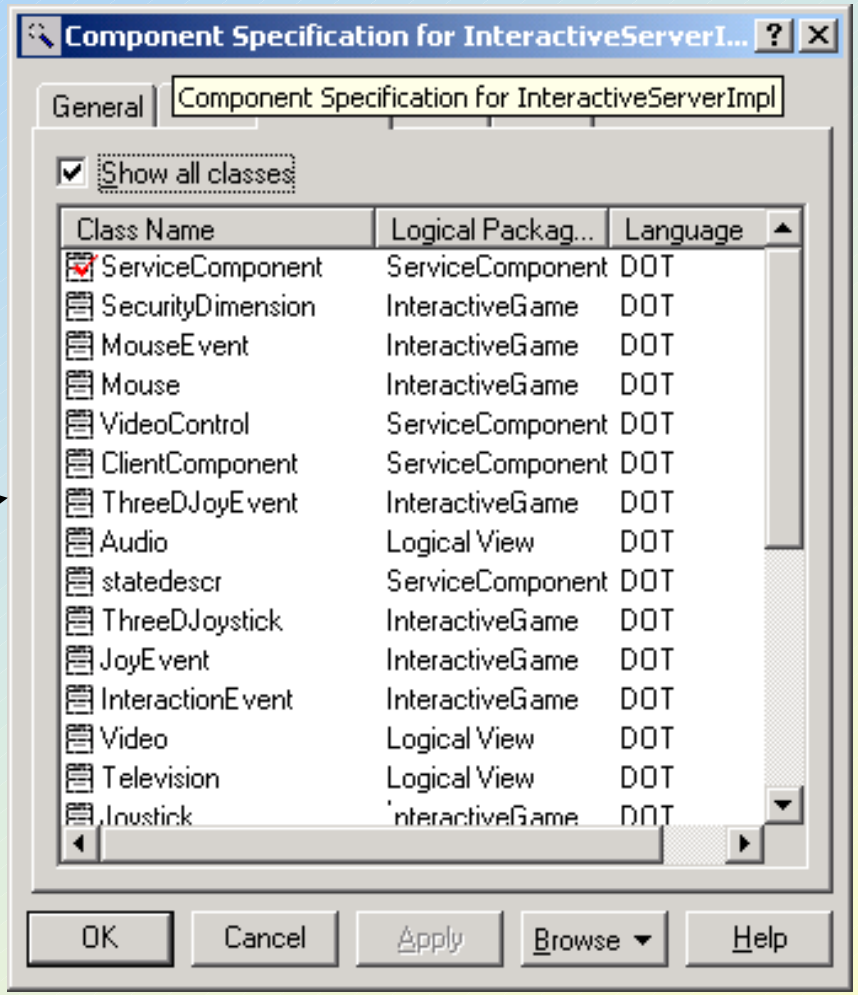


Interaction Rules

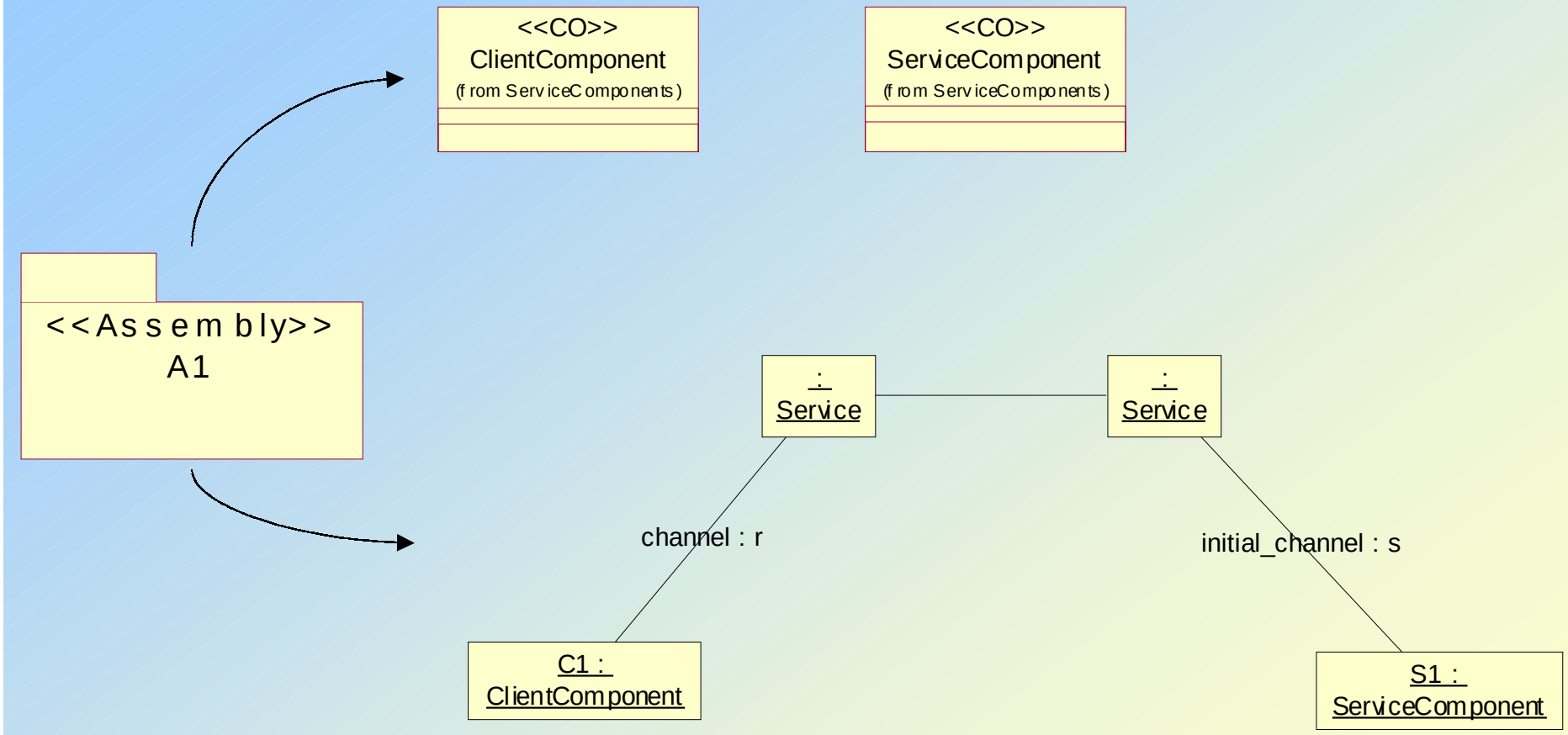


Deployment Specification (Components)

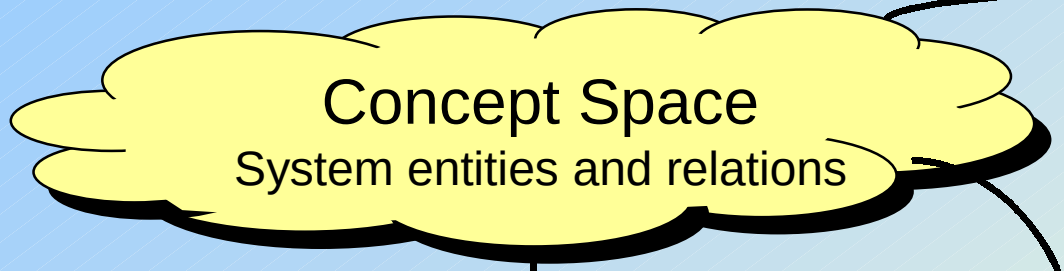




Deployment Specification (Assemblies)



Code Generation Aspects



SDL 2000 Generation
- •Package for interface part
  - Data definition
  - Signal definition
  - Interface definition
- •Package for implementation
  - Agent definition
  - Gate definition

CORBA 2.3 platform + continuous media extensions
- •Similar to CORBA Component Model approach
- •IDL generation for operational part
- •Implied IDL for signal/continuous media communication
- •C++ classes for implementation support
- •Makefiles for compilation
- •Interceptor code for binding rules and QoS negotiation
- •Deployment descriptors for automated deployment

Planned Environments
- • CORBA Components Platform
- • Quality Aware Middleware (QAM, KPN Research)
- • ...

Demonstration

- Realization of UML profile + code generators with Rational ROSE
  - Minimal dependency on UML tool
  - Concept space and code generators realized as separate libraries
- Approach
  - Model Structure, Binding Rules and Ports, Implementation
  - Generation of platform specific IDL, C++ implementation code and makefile as well as SDL code for simulation
  - Embed Service in TINA Platform Environment through import of relevant platform specification