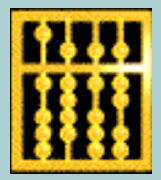
Design and Practical Guideline to a CORBA-based-Communication Framework

DWARF: Distributed Wearable Augmented Reality Framework

Studienarbeit

Lothar Richter

Chair for Applied Software Engineering

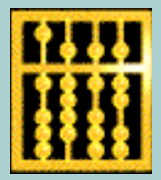


Summary

- Development of a short and easy understandable practical introduction into CORBA and DWARF
- Up to now no really simple AND short introduction was available yet
- Students are given a jump-start to gain first practical experience with DWARF
- The tutorial is structured according to didactical considerations to reach optimal learning success

Overview

- Motivation
- Requirements Analysis:
  - Goal, Scope
- System Design:
  - Selection and structure of presented topics
  - Training Process
- Object Design:
  - Definition of chapter contents and example design
- Conclusions



Motivation

- Idea originated from experiences made during TRAMP
- specific problems encountered:
  - ➢ no former experience with CORBA technology
  - ➢ direct switch from architecture to source code level
  - ➢ no structured and simple but yet sufficient user documention of DWARF available

→ most of the time spent to learn how to use the middleware

Requirements of a Tutorial

A tutorial should reach a destinct goal:

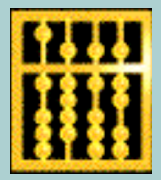
Enable users to acquire the knowledge necessary for productive use of Dwarf in reasonable time.

Students with now relevant practical knowledge should be enabled to use DWARF within one week

Requirements (cont.)

An instructive tutorial is characterized by several items:

- Carefully choosen scope:
  - Intended users
  - Which knowledge is expected from the user
  - What is to be covered
  - What is not to be covered



Requirements (cont.)

An instructive tutorial is characterized by several items:

- Careful choosen scope:
  - Intended users

    Students after Vordiplom
  - Which knowledge is expected with the user
  - What is to be covered
  - What is not to be covered

Requirements (cont.)

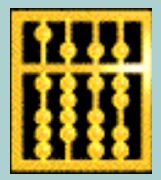
An instructive tutorial is characterized by several items:

- Careful choosen scope:
  - Intended users

    Students after Vordiplom
  - Which knowledge is expected with the user

    basic programming experience

    object oriented paradigm



Requirements (cont.)

- What is to be covered
- What is not to be covered

Requirements (cont.)

- What is to be covered

  - Theory: CORBA Basics
    - Event Communication
    - Principles behind DWARF
    - Services types offered by DWARF-Interfaces
  - Practical: Handling CORBA Objects
    - How to use the Notification Service
    - Creating DWARF Services

Requirements (cont.)

- What is not to be covered

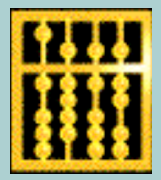
Requirements (cont.)

- What is not to be covered

  Paradigm of object orientation and its advantages

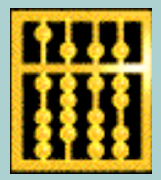
  CORBA technology and CORBA services into depth

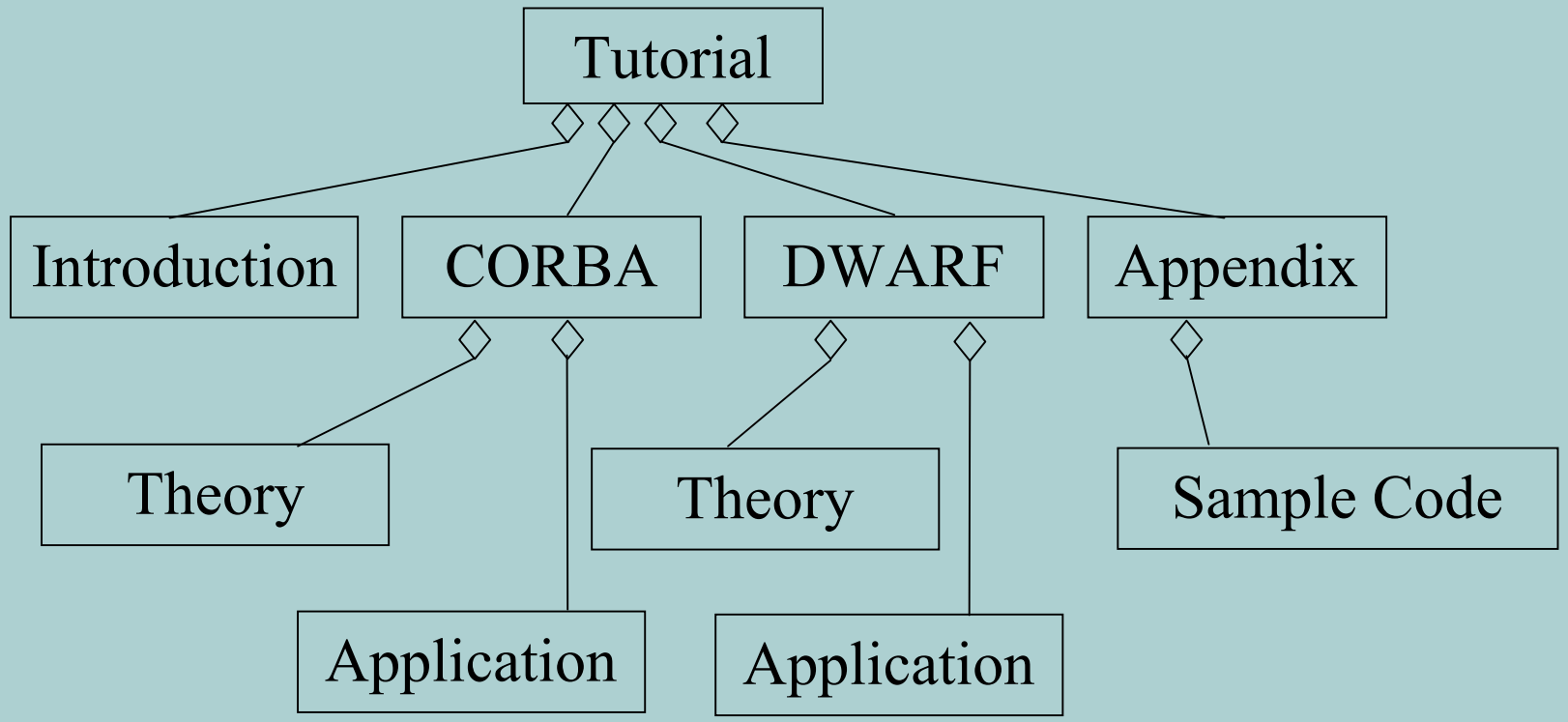
  Design and development of distributed systems

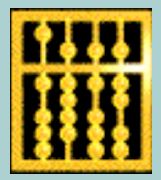


System Design – Tutorial Design

- Define the scope items
  - Identify the solution domain concepts,
  - Sort them according to complexity and priority
  - Select the concepts necessary to reach the goal
- Define audience and expected knowledge
  - Adopt presentation level to the audiences knowledge level
  - Select appropriate programming examples
- Define an appropriate tutorial structure



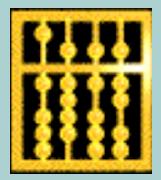

Tutorial
Introduction
CORBA
DWARF
Appendix
Theory
Application
Theory
Application
Sample Code



Training Process I (SD)

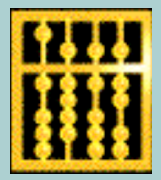
Consist of three iterations:

- Introduce the basic CORBA concepts in theory
- Improve understanding through practical experiences:
  - Analyze code examples
  - Modify code examples and explore behaviour



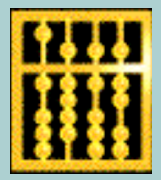
Training Process II (SD cont.)

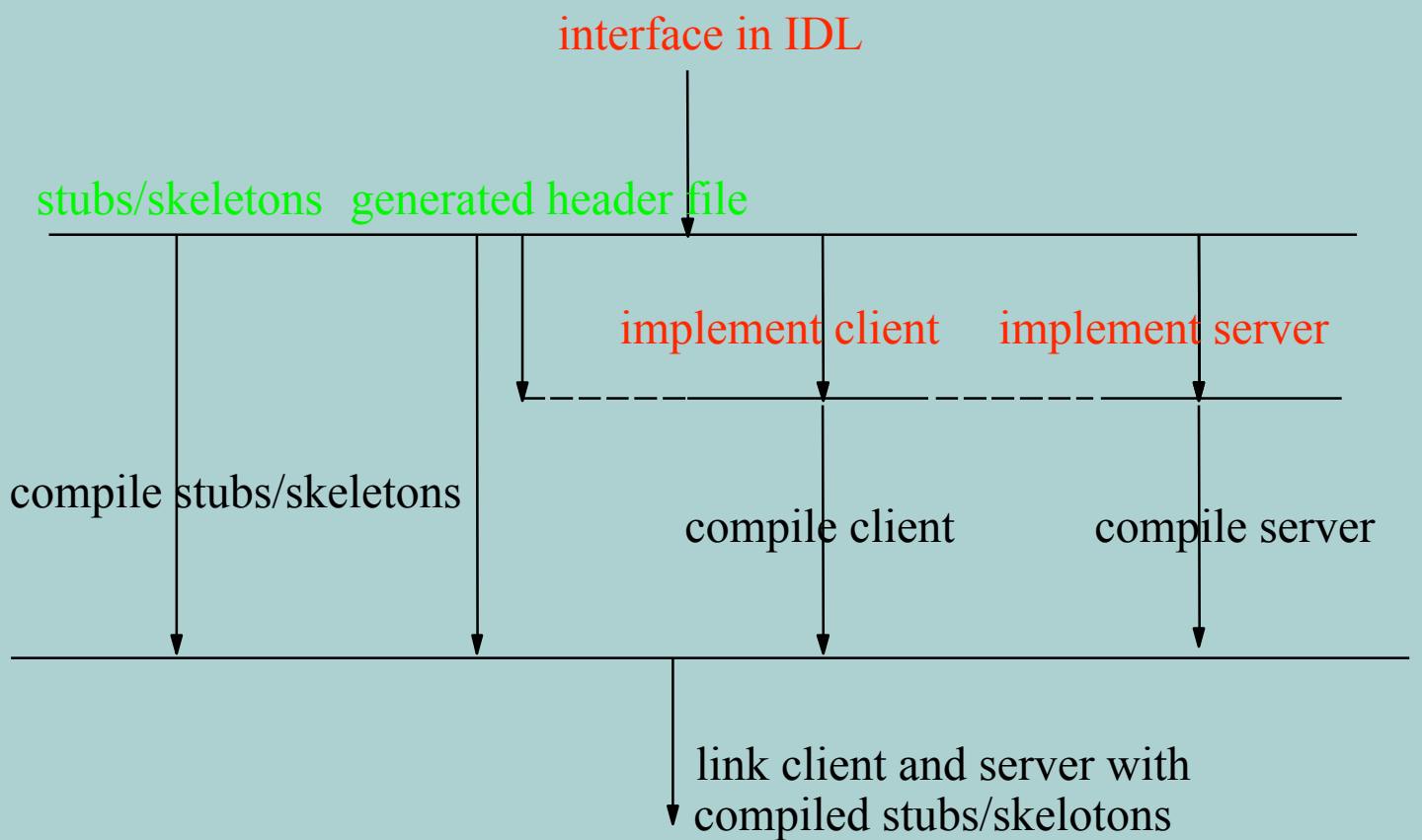
- Introduce CORBA Notification service concepts in theory
- Introduce advanced concepts where examples become more complex
- Gain practical experience



Training Process III (SD cont.)

- Introduce DWARF concepts in terms of CORBA Services
- Use gathered experience to demonstrate the use of the various DWARF interfaces to create a simple DWARF service





Generation of a CORBA System as
UML Activity Diagram

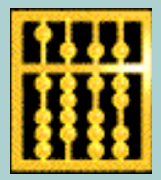
Object Design – Lesson Design

Content Items:

- Identified fundamental CORBA ideas
- Advanced topics with CORBA services
- Introduction of the DWARF service interfaces

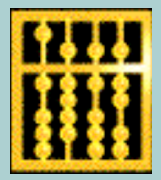
Basic Corba Ideas (OD)

- CORBA objects and CORBA interfaces
- Interrelationship between interface declaration, compilation into stubs and skeletons
- Object adapters and derivation of servant objects on behalf of server processes
- CORBA object references are different from Java or C++ references – they are connector objects



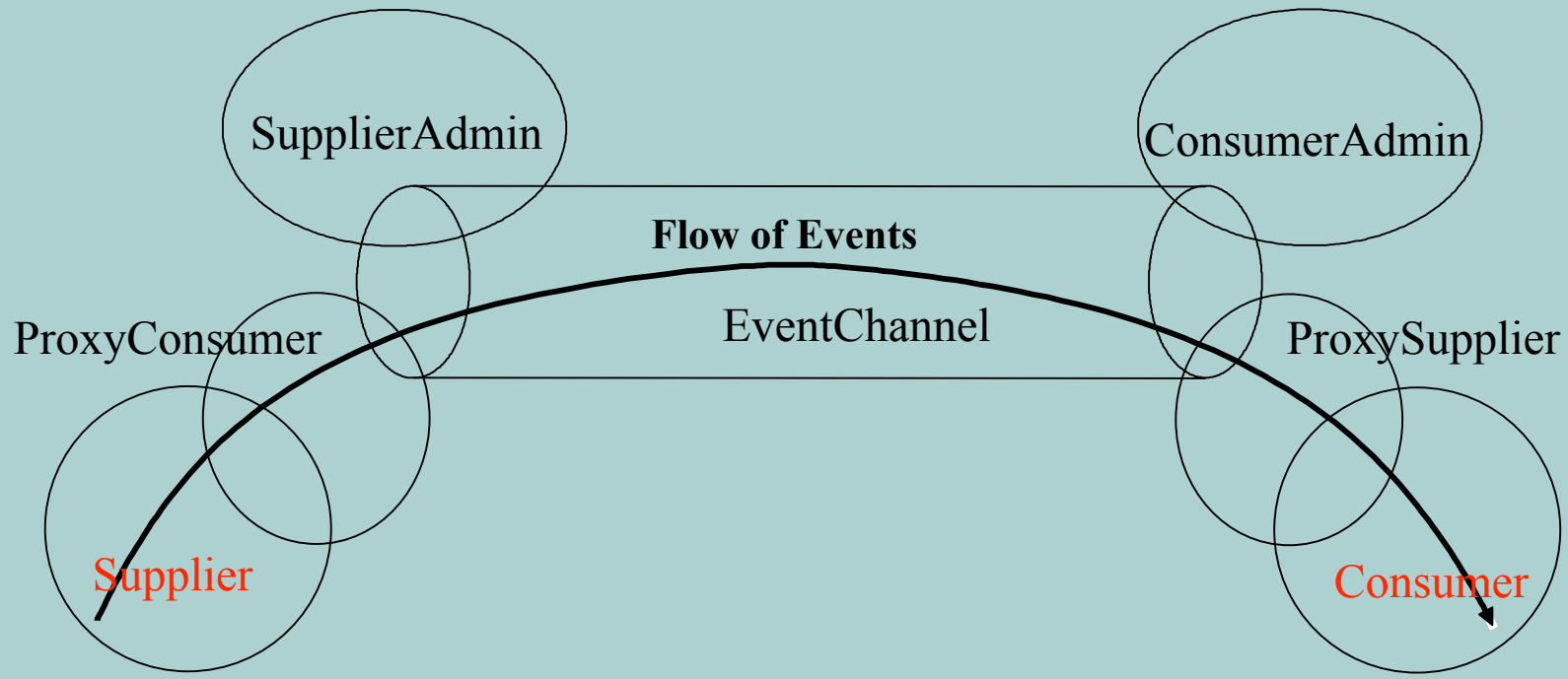
Basic CORBA Ideas (OD cont.)

- Parameter passing and possible pitfalls
- Inheritance of interface
- Changing the roles: A client can act as server can act as client can act as server .... (welcome to multithreading)



Advance Topics: CORBA Notification Service (OD)

- Introducing asynchronous communication using events
- Components which are involved in the communication line and provided by the service libraries
- How to adress these and where fit my components in ?
- Getting in contact with the specification documents and get an idea how to handle CORBA services with a complex interface structure



This is not UML !

Code Snippets

```
// eventChannel.hh

#ifndef MY_EVENTCHANNEL
#define MY_EVENTCHANNEL

class EventChannel{
private:
  CorbaInit* myOrbConnection; // holds usefule object pointers
  CosNotifyChannelAdmin::EventChannelFactory_var myChannelFactory;
  CosNotifyChannelAdmin::EventChannel_var myEventChannel;
  void createMyChannel();
public:
  EventChannel(CorbaInit* intialRef);
  EventChannel(CorbaInit* intialRef,
                  CosNotifyChannelAdmin::EventChannelFactory_var& channelFactoryToUse);
  EventChannel(CorbaInit* intialRef, const char* channelFactRefFile);

  CosNotifyChannelAdmin::EventChannel_var getEventChannel();
};

#endif

// end eventChannel.hh
```

Code from supplierDriver.cc

```
...
int main(int argc, char* argv[])
{
....

  // creating the proxy consumer object
  // specific type is set via parameters and subsequent narrowing
  CosNotifyChannelAdmin::ProxyID pxID; //  ID's used to distinguish objects
  CosNotifyChannelAdmin::ProxyConsumer_var tmp_consIOR = sadminIOR->
  obtain_notification_push_consumer(CosNotifyChannelAdmin::STRUCTURED_EVENT, pxID);
    // check whether IOR was of the assumed type
    if (CORBA::is_nil(tmp_consIOR)){
      cerr << "no proxy interface available" << endl;
      throw 0;
    }else{
      cout << "obtain the proxy consumer IOR" << endl;
    }

....
```

Introduction to DWARF-Middleware (OD)

- Understand DWARF as a special kind of CORBA service:

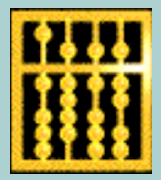
  A service specialized on run-time reconfiguration of distributed systems consisting of CORBA objects

- Explain the principle function of the middleware
- Introduce the different components (interfaces)

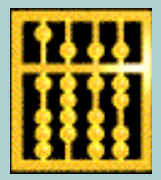
  which allow users to influence the specific behaviour

  of their DWARF services
- Figure out the interaction of these interfaces within

  the middleware

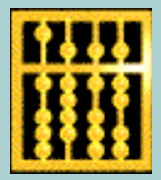


Phase	Impl. by the service	Impl. by the service manager
Description Phase	ServiceDescription AbilityDescription NeedDescription	SMgrDescribeServices SMgrRegisterServices
Negotiation Phase	SvcSelection SvcSession	AsdSelection
Connection Phase	SvcProt... SvcProtPush... SvcStartup	



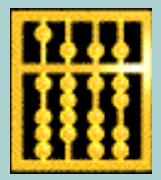
Use of Dwarf-Middleware (OD)

- The use of the DWARF middleware is demonstrated with the development of a simple example service
- It is shown how to describe the DWARF-interface for a service which exports the time
- Based on this the implementation of a specific servant objects is shown and the necessary driver code to access the middleware and register with the service manager
- A requesting service is also generated to demonstrate all the interactions



Conclusions

- High complexity of CORBA and DWARF middleware impose a high demand of training time for beginners
- The presented work alleviate this problem by reducing the training time
- This is done by the structured presentation of well selected theoretical concepts in combination with practical exercises



Thank You For Your Attention