Title: Distributed_Applications (09.07.2013)
Date: Tue Jul 09 14:31:28 CEST 2013
Duration: 90:53 min
Pages: 19
3. Step: code generation

AutoFocus is a platform to specify distributed systems developed by the group of Prof. Brey, TU München based on formal methods of systems engineering integrates hierarchical description techniques allows distributed and platform independent development project advanced to AutoFocus supporting the following functionality:
- Design modeling views and editors, such as system structure diagram, state transition diagram message sequence charts interactive simulation environment, code generation, consistency maintenance support.

MDA Concept

consists of 3 steps:
- development of platform independent models (PIMs)
- mapping to platform dependent models (PDMs)
- transformation between models (PIM → PSM, PSM → code)

1. Step: development of PIM
2. Step: mapping to PSM
3. Step: code generation

Service-Oriented Modeling

Ideas and proposals emerged to transfer the service approach to the design and modeling of software systems. 

Definition: Service-oriented modeling (SOM) is the discipline of modeling business and systems, for the purpose of designing and specifying service-oriented business systems within SOA.
- Create models that provide a comprehensive view of the analysis, design, and architecture of all software components in an organization.
- Envision the coherence of services in an interoperable computing environment.

Definition: The service-oriented modeling framework (SOMF) is a service-oriented development life cycle methodology that provides practices, disciplines and a universal language to provide tactical and strategic solutions to enterprise problems.
- Service Evolution
- Life Cycle Structure
- Life Cycle Modeling
- SOM Framework
- Other SOA Design Methodologies
Service Evolution

SOM advocates the transformation of a service through 4 states.

1. Conceptual Service
2. Analysis Service
3. Design Service
4. Solution Service

Life Cycle Structure

Identifies the elements for service development and operations. It consists of 4 major components.

- **Timeline**: defines the life span of a service.
- **Events**: 2 types of events during the service life span:
  - Predicted and scheduled events, e.g., milestone, planning stage or deployment stage.
  - Unpredicted events, e.g., stock market crash, trading volume exceeds capacity of trading service.
- **Seasons**: services live through 2 major life cycle seasons:
  - Design-time season: services are conceptualized, analyzed, designed, constructed and tested.
  - Run-time season: services are managed, monitored, and controlled to ensure proper performance.

**Disciplines**: identify modeling and non-modeling best practices and standards to be pursued throughout the service life cycle.

- **Continuous discipline**: e.g., service-oriented conceptualization, business integration or construction.
- **Continuous discipline**: e.g., service portfolio management, service governance.

Life Cycle Modeling

The following core processes can be identified in which business and IT personnel must be engaged to produce design and solution artifacts.

- **Conceptual modeling**: identify driving concepts behind future solution services.
- **Discovery & analysis modeling**: discover and analyze services for granularity, reusability, interoperability, loose-coupling, and identify consolidation opportunities for the existing software assets.
- **Business integration modeling**: identify service integration and alignment opportunities with business.

SOM Framework

Modeling components and disciplines are integrated into a SOM framework.
Other SOA Design Methodologies

A brief overview of some other SOA design methodologies:
- Creating Service-Oriented Architectures (CSOA) by Barry & Associates
  - Focus is on technical aspects
    - Consist of the 5 phases:
      - Experiment with Web services
      - Adapt existing systems to use Web Services
      - Remove inter-system dependencies
      - Establish internal SOA
      - Incorporate external services
- Service-Oriented Transformation of Legacy Systems (SOTLS) by Nadhas
  - Focus is on technical aspects
  - Consist of the phases:
    - Planning, analysis, service design, service construction, service test, service deployment/execution and service management/monitoring

Design of distributed applications

Issues
- Steps in the design of distributed applications
- Design - Development environment
- Service-Oriented Modeling

Introduction

When a group of programs has the task to build a distributed application, in addition to distributed code management, there is also the need for distributed file services.

Definitions
- Motivation for replicated files
- Two consistency types
- Replica placement
Introduction

When a group of programmers has the task to build a distributed application, in addition to distributed code management there is also the need for distributed file services.

Definitions

Motivation for replicated files
Two consistency types
Replica placement

Layers of a distributed file service

The functions of a distributed file service are usually arranged in a hierarchical way.

Layer semantics

Update of replicated files

Basically, there are two types of approaches for multiple update control: the optimistic and the pessimistic approach.

Optimistic concurrency control
Pessimistic concurrency control
Voting schemes

Voting schemes provide pessimistic concurrency control.

Introduction

Voting schemes are algorithms for maintaining mutual consistency of replicas even in situations of computer crashes and network partitions.

Let us assume there exist $\mathcal{R}$ replicas of file $d$. Let $\omega_i$ be the weight of the vote of computer $i$; $K$ be the set of all computers considered.

Let the sum of all weights be $\text{SUM} = \sum_{i \in K} \omega_i$.

Definitions

Multiple-reader, single-writer strategy
Voting scheme variants