Lecture 7 Architectural Design Methodology

- Arch Design Cycle
- Arch Design Method
- Domain-Specific S/W Arch

架构设计方法

S/W Arch Design Work

- S/W development process: SRS stage, S/W design, implementation, integration and test, delivery and sustaining engineering
- S/W design cycle: preliminary design (system design + architectural design), detailed design (module design), design patch work
- Arch model and design shall be completed prior to the coding and implementation stage, and used as guidelines for considerations on standard, development technology and implementation details.

S/W Development Life Cycle

1, Defining SRS Stage

- Tasks：defining system functions in SRS；
- Process：functional and non-functional requirements refining and reviewing
- Output：SRS

架构设计方法

2, System Analysis and Preliminary Design

- Tasks: system analysis and architectural design
- Process： class/component design, constraint defining, and mapping s/w units to h/w platform
- Output： preliminary design documents

3, Detailed Module Design

- component design, defining component's port, module algorithm and data structure
- Output: detailed design documents, API, etc.

架构设计方法

4, Code/Implementation

- Code SA module and realize functions
- Each component must cover certain portion of SRS
- Integrate the system

5, Delivery and Sustaining Engineering

- Test
- Delivery
- Engineering Support

架构设计方法

S/W Architecture Life Cycle

1, Non-formal modeling on SA

2, Analysis and documented design on SA

3, Refining and reviewing on SA design

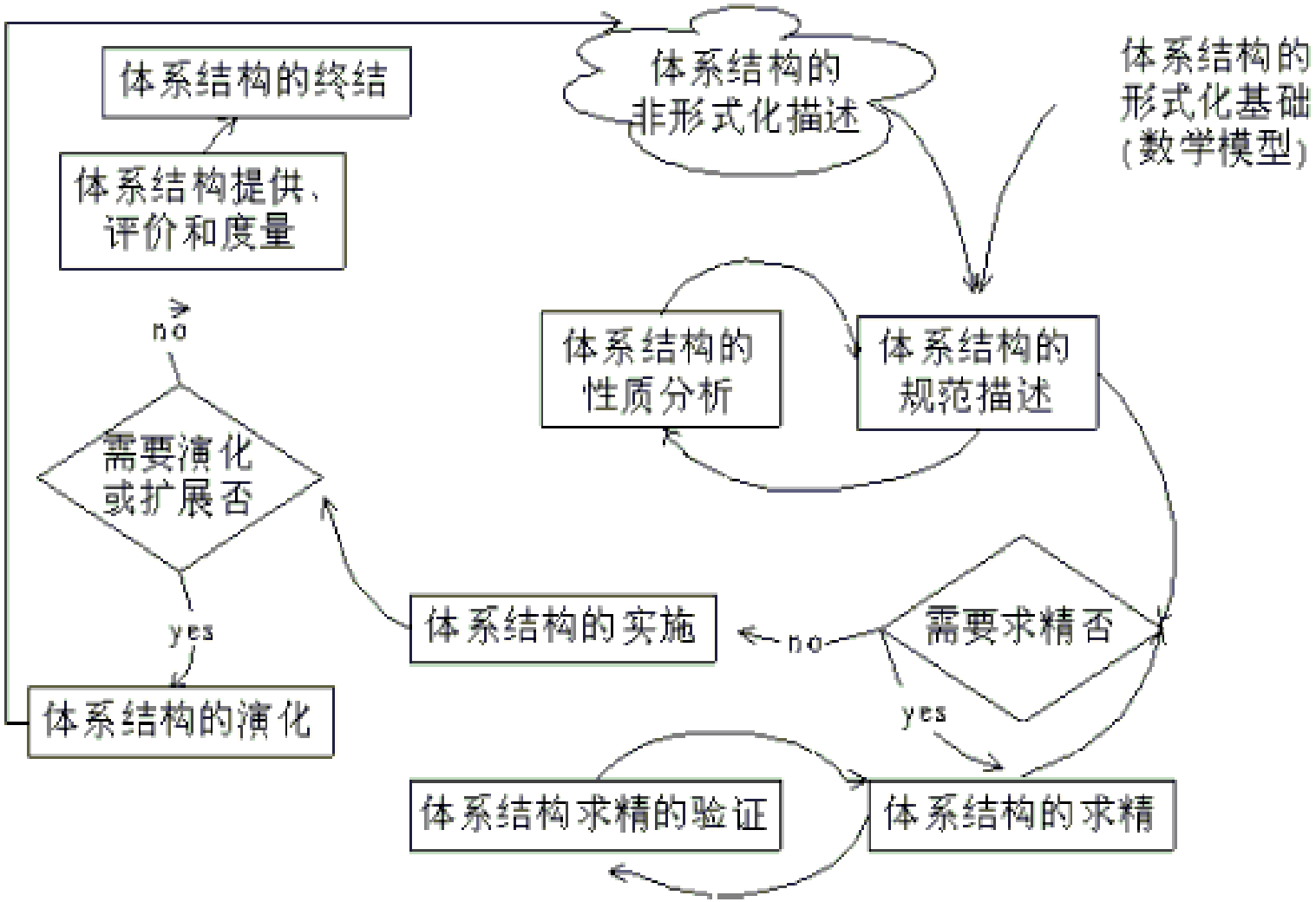
4, Implementation of SA design

5, Evolution and expansion of SA

6, Design measurement and improvement

7, End of SA life cycle

架构设计方法


体系结构的终结
体系结构的非形式化描述
体系结构的形式化基础（数学模型）
体系结构提供、评价和度量
no
体系结构的性质分析
体系结构的规范描述
需要演化或扩展否
yes
体系结构的实施
no
需要求精否
yes
体系结构的演化
体系结构求精的验证
体系结构的求精

架构设计方法

Things you need to know before designing

- Domain knowledge: business model/process, technical standards, core technology, etc.
- SRS understanding: deep understanding in functional and non-functional requirements
- Abstraction of system: overall structure, subsystems, external ports, etc.
- Architectural Model: arch style, key components, system constraints, etc.

SA Architectural Design Method

1, Component-based Architectural Design (CBAD)

- Build the arch model/description based on components
- Use the Object-Oriented Modeling Technology and Object-Oriented Analysis and Design method
- More focus on component/module design but not on overall performance of SA

架构设计方法

2, Use Case-driven Architectural Design (UCAD)

- Build the SA model based on Use Cases, i.e. from the scenarios that defines user's functional requirements
- Typically the UML is used as the designing language to work out a complete design
- Good to cover and trace SRS in design, but not appropriate for SA refining and improvement

架构设计方法

3, Domain-specific Software Architecture (DSSA)

- Start modeling SA from the domain model
- Domain can be modeled by various attributes: class/component, framework, semantics/regulations, etc.
- Problem domain analysis and output are not always enough to guide the SA design

架构设计方法

- DSSA (Domain Specific Software Architecture) starts from domain analysis and targets on a design that covers a core set of functions in the domain
- DSSA model is considered for an application domain, but the DSSA design provides a concrete SA solution in the domain

架构设计方法

4, Pattern-driven Architectural Design (PDAD)

A refined or advanced version of CBAD by emphasis on using arch and component patterns

- Emphasis on design reuse
- Better s/w performance
- Good for certain type s/w applications such as SDK, middleware, framework, etc.

架构设计方法

5, Model-driven Architecture (MDA)

A new SA design method that focuses on the association of arch elements with implementation to try to automate the development process

- Still a component-based design method
- Well defined component/module interface
- Use the CASE tools to generate code skeleton or framework API
- Greatly improve design and code quality

架构设计方法

SA Design Principles/Guidelines

- 抽象与封装 Abstraction and Encapsulation
- 信息隐藏 Information Hiding
- 模块化 Modularization
- 注意点分离 Separation of Concerns
- 内聚与解耦 Cohesion and Decoupling
- Open Standard and Open Architecture

SA Design Principles/Guidelines (cont' d)

- 充分性、完备性与原子性 Sufficiency, Completeness & Primitiveness
- 策略与实现的分离 Separation of Policy & Implementation
- 界面与实现的分离 Separation of Interface & Implementation
- 分治 Divide-and-Conquer
- 层次化 Layering

架构设计方法

Arch modeling language in development

- Requirement Description Language
- Component Description Language
- Architectural Modeling Language
- Framework Description Language
- Software Programming Language
- System Integration Language

DSSA Design Process

- Domain Definition
- Domain Engineering
- DSSA Design Method

DSSA Architectural Design

- Domain-driven Architecture is one of major SA types deployed in industry
- DSSA captures the backbone framework of a class of applications in one application domain
- DSSA presents the common features and behaviors of applications in one domain
- DSSA can be considered as a design blueprint for software development in a domain

DSSA Architecture Definition

A set of s/w components with a standard design that is used to build a class of applications with common features and functions in a particular application domain.

- Abstraction: problem domain and solution domain

- Mapping of problem domain to solution domain

- Design solution can be reused for a group of problems in a domain

DSSA Key Points

- DSSA development is not solely targeted on the requirements of one application, but on the framework that provides solution backbone for entire domain
  - Domain analysis, domain requirements, domain models, domain reference architecture and domain components
  - Solutions are not targeting on single application needs, but on common demands from whole domain

架构设计方法

- Two DSSA Processes: domain engineering and application engineering
  - Domain engineering captures the common needs from a class of applications in a domain and defines a core set of requirements for them
  - Application engineering develops a complete software product by deploying reused components based on the DSSA arch to meet the needs of a class of applications

DSSA's Domain Engineering

- Domain Concept
  - A class or a family of applications/systems， with a common capability or shared data set
  - Problem space introduced by a group of applications with similar requirements
  - A problem/task space, in which a class of similar applications can be developed to meet the needs for various customers in one domain

架构设计方法

- Domain Concept (cont'd)
  - In software engineering, it consists of problems, functions, solutions and knowledge space that can be used for a group of applications sharing common attributes
  - Vertical Domain: the business space covered by the group of applications
  - Horizontal Domain: the problem space divided by the functional units inside the applications

架构设计方法

- Domain Characters
  - The systems in a domain have similarity or common attributes in user requirements， shared data, and problem space
  - The solution to the problem set in the domain leads to new applications
  - The resources, assignments and its associations in a domain are presented by a base framework

架构设计方法

- Domain Engineering: the process to define and develop the domain-oriented reused software system, in which the DSSA is designed and constructed
- Three sections： domain analysis, domain design, domain implementation
- Domain Analysis
  - The process of domain definition, character abstraction and description, concept refinement, data extraction, function identification, framework development

架构设计方法

– Domain analysis methods

- Feature-oriented Domain Analysis(FODA)

  context analysis, domain modeling and arch modeling

- Organized domain model(ODM)，

  domain planning, domain modeling and reused resource set

- DSSA Domain Analysis

  develop a prototype application to be the base for applications in a domain

架构设计方法

– Domain analysis methods(cont'd)

- Feature-oriented Domain Analysis(FODA)

  context analysis, domain modeling and arch modeling

- Organized domain model(ODM)，

  domain planning, domain modeling and reused resource set

- DSSA Domain Analysis

  develop a prototype application to be the base for applications in a domain

架构设计方法

- Domain Design
  - goals are to construct the DSSA in a particular domain
  - framework: Generalization, standardization, and documentation
  - separate common features from applications to create DSSA framework
  - design the framework into multiple layers

架构设计方法

- Domain Implementation
  - goals are to develop reused s/w framework based on the domain model and DSSA
  - mapping of DSSA design features to s/w components
  - well-designed component interface
  - output: a DSSA framework containing domain model, DSSA design and reused functional components

DSSA Project Cases

- IBM/Aerospace/MIT/UCI had developed a DSSA application for aerospace electronic systems
- Teknowledge/Stanford/TRW developed an event-driven, object-oriented and concurrent DSSA design language RAPIDE
- USC/ISI/GMU developed an automatic code generator and an SDK for DSSA engineering

End of Lecture

Thanks!