Software Systems Architecture

Xiaobin Xu

doublexb@163.com

Module 1: Software Systems Architecture basics

Content:

- History of Software Development
- Definition of Software Architecture
- Some Related Concepts of Software Systems Architecture
- Where do architectures come from?
- Values of Software Architecture
- Current Research & Practicing in Software Systems Architecture

History of Software Development

- ***Assemble Language***
- ***Small size of program***

1970’ ▸

- ***Advanced Language***
- ***Structure-Oriented Theory***
- ***Dataflow/Control flow Design Methods***

1980’ ▸

- ***Application Development Library: Class/Functions Library***
- ***Object-Oriented Theory***
- ***Object Modeling & Design Technology***

1995 ▸

- ***Application Development Framework: J2EE, .NET***
- ***Component Technology: COM/DCOM, CORBA …***
- ***Object Modeling & Design Standardization: UML***

Future ▸

- ***Model-Driven Development: MDA***
- ***……***

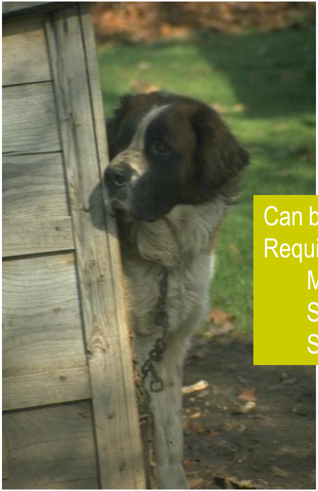
History of Software Development

- Review of the history of the software:
  - **The size and the complexity of software is becoming *larger and more complex*.**
    - The application areas of software: science computing, manufacturing, commerce, education and amusement.
  - **The abstraction level of software is becoming *more high*.**
    - Machine Language—>Assemble Language—>Advanced Language—> Application Framework
    - Structure-Oriented Programming —> Object-Oriented Programming—> Aspect-Oriented Programming

History of Software Development

- Results of the development of software:
  - **Good architecture design has always been a major factor in determining the success of a software system .**
  - **The architecture and designing is more important than the data structure and the program algorithm.**

The Definition of SA



Can be built by one person

Requires

- Minimal modeling
- Simple process
- Simple tools

Architecting a house

Built most efficiently and timely by a team

Requires

- Modeling
- Well-defined process
- Power tools

Architecting a high rise

Differences

- Scale
- Process
- Cost
- Schedule
- Skills and development teams
- Materials and technologies
- Stakeholders
- Risks

Definition of SA

- The software architecture of a program or computing system is the structure or structures of the system, which comprise **software elements**, the externally **visible properties** of those elements, and the **relationships** among them.

  ----《Software Architecture in Practice》, Addison-Wesley 1997

- Architecture is the organizational structure of a system. An architecture can be recursively decomposed into **parts** that interact through **interfaces, relationships** that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems.

  ----UML 1.3

Definition of SA

- Software architecture is the **fundamental organization** of a system, embodied in its **components**, their **relationships** to each other and the environment, and the **principles** governing its design and evolution

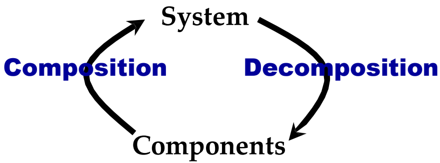
----- IEEE 1471-2000

Definition of SA

- Software Architecture
  - **Software Elements**: functions, Interfaces, programs, class modules, layers, subsystem, clients/servers etc.
  - **Visible Properties**: provided services, performance characteristics, fault handling, shared resource usage, and so on
  - **Relations**: composition mechanism and style of these elements
- An architecture is the result of a set of business and technical decisions.

Definition of SSA

- **A Software Architecture Include:**
  - the constituent elements **— Component**
  - the interaction rules/mechanism **— Connector**
- ***So, it can be defined briefly* as**:
  - the *components comprised in the system, and the relationships or interaction mechanisms of those components.*
  - *Software Architecture Design = Decomposition + Composition*



Definition of SA

- **Decomposition/Composition**
  - reducing the complexity of software design and construction.
  - controlling the risks of software development
  - improving the efficiency of organization and management

But, we must consider

- How do we break the system down into pieces?
- Do we have all the necessary pieces?
- Do the pieces fit together?

Definition of SA

- Hundreds of definitions on CMU web page: http://www.sei.cmu.edu/architecture/definitions.html

Related Concepts of SA

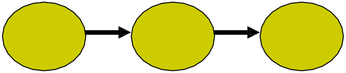
- **Component**
  - A logical and functional unit of the system.
- **Note:**
  - **A component may be divided into more little unit of components.**
  - **A component serves certain responsibilities.**
  - **The component is an abstract and conceptual word, it'll be different specific objects (for example, modules, subsystems, layers, packages, classes etc.) in different scenarios.**

Related Concepts of SA

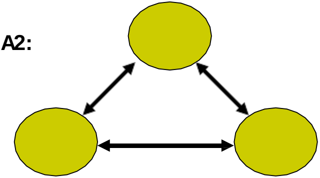
- **Connector**

The interaction rules or mechanisms among components.

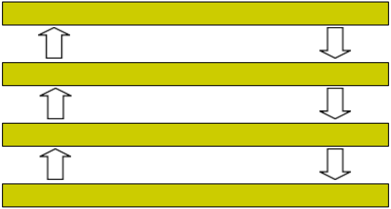
SA1:



SA2:



SA3:



Related Concepts of SA

- **Functional Property of the SA**

  the characters of the SA that meets the functional requirements.

- **Non-functional Property of the SA**

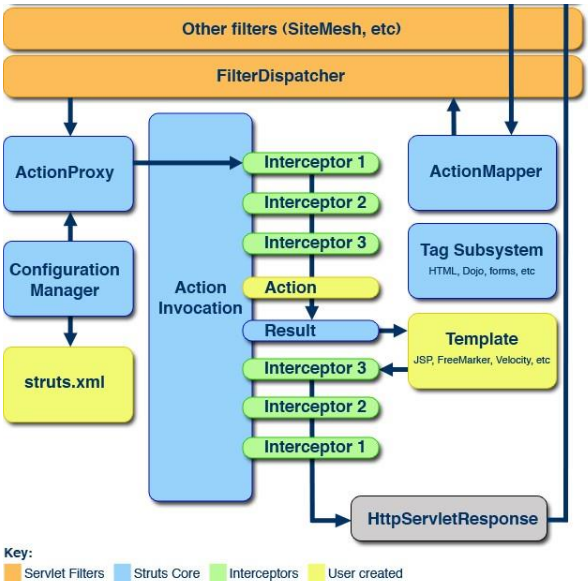
  the characters of the SA that meets the non-functional requirements. For example,

  - performance
  - portability
  - flexibility/extensibility
  - reliability/security
  - …

Related Concepts of SA

- **Framework**
  - A framework is a reusable application infrastructure for a specified problems.
  - some necessary basic components for the specified problems
  - interaction mechanism and constraints among components
  - a context or environment for the applications developed based on the framework
- Commonly, a framework mainly presents a class library. For example: .NET Framework, JavaEE Framework etc.

Architecture of Struts 2



Comments for SA

- Architecture is at a high-enough level of abstraction that the system can be viewed as a whole.
- At the architectural level, all implementation details are hidden.
- The architecture must support the functionality required of the system.
- The architecture must conform to the system qualities (also known as non-functional requirements) : performance, security and reliability, flexibility or extensibility.

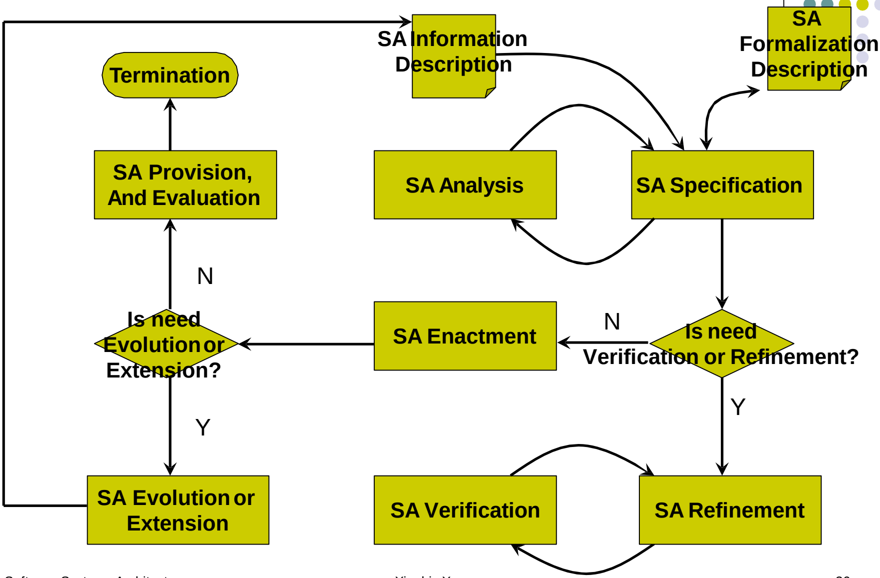
Where Do Architectures Come From?

- **Architectures are influenced by system stakeholders.**
- **Architectures are influenced by the developing organization.**
- **Architectures are influenced by the background and experience of the architects.**
- **Architectures are influenced by the technical environment.**

Architecture Business Cycle

- Activities of architecture business cycle:
  - Creating the business case for the system
  - Understanding the requirements
  - Creating or selecting the architecture
  - Documenting and communicating the architecture
  - Analyzing or evaluating the architecture
  - Implementing the system based on the architecture
  - Ensuring that the implementation conforms to the architecture

The lifetime model of SA


SA Information Description
SA Formalization Description
Termination
SA Provision, And Evaluation
SA Analysis
SA Specification
N
Is need Evolution or Extension?
SA Enactment
N
Is need Verification or Refinement?
Y
Y
SA Evolution or Extension
SA Verification
SA Refinement

Why Is Software Architecture Important?

- **Architecture is the vehicle for stakeholder communication.**
- **Architecture manifests the earliest set of design decisions.**
  - ***The Architecture Defines Constraints on Implementation***
  - ***The Architecture Dictates Organizational Structure***
  - ***The Architecture Inhibits or Enables a System's Quality Attributes***
  - ***Predicting System Qualities by Studying the Architecture***
  - ***The Architecture Makes It Easier to Reason about and Manage Change***

Why Is Software Architecture Important?

  - ***The Architecture Helps in Evolutionary Prototyping***
  - ***The Architecture Enables More Accurate Cost and Schedule Estimates***
- **Architecture as a transferable, re-usable model.**
  - ***Software Product Lines Share a Common Architecture***
  - ***Systems Can Be Built Using Large, Externally Developed Elements***

Values of Architecture

- **Architecture serves both technical and organizational purposes:**
- **Organization side:**
  - ***Communicating inside organization*, *and between customers and vendors***
  - ***Providing the high-level information of systems costs and risks evaluating***
  - ***Work allocation and project schedule***

Values of Architecture

- **Technical side:**
  - ***meet system requirements and objectives***
  - ***Specify the constraints of detailed design, construction and testing phrase***
  - ***enable flexible distribution/partitioning of the system***
  - ***reduce cost of maintenance and evolution***
  - ***increase reuse and integrate with legacy and third party software***

Who focus on Software Architecture?

- An architectural view is a simplified description (an abstraction) of a system from a particular perspective or vantage point, covering particular concerns, and omitting entities that are not relevant to this perspective

About Kruchten and this paper

- Philippe Kruchten
  - Over 16 years of experience as the leader of RUP development team in Rational corp. (now owned by IBM)
  - Valuable experiences in industry (Telecom, Air traffic control system) which he used them for confirmation of his model
- The “4+1 view model” paper:
  - 60 citations according to ACM portal site

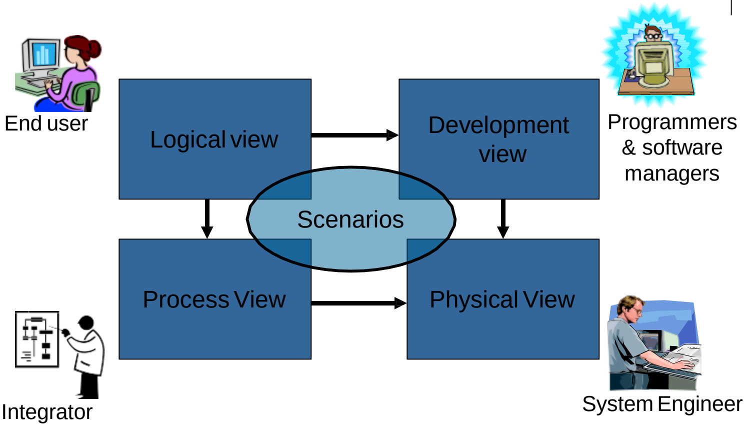
Problem

- Arch. documents over-emphasize an aspect of development (i.e. team organization) or do not address the concerns of all stakeholders
- Various stakeholders of software system: end-user, developers, system engineers, project managers
- Software engineers struggled to represent more on one blueprint, and so arch. documents contain complex diagrams

Solution

- Using several concurrent *views* or *perspectives*, with different notations each one addressing one specific set for concerns
- “4+1” view model presented to address large and challenging architectures

4+1 View Model of Architecture



Characteristics of a Good Architecture

- Resilient
- Simple
- Approachable
- Clear separation of concerns
- Balanced distribution of responsibilities
- Balances economic and technology constraints

Research of SA

- **Formalization Research**

  refers to *how to describe the SA in specific rules*

  how to view and present the SA

  ADL--- *Architecture Description Language*

- **Verification & Evaluation Research**

  refers to *how to verify and evaluate the SA whether it meets the constraints of functional and non-functional requirement.*

  ATAM---Architecture Tradeoff Analysis Method

Assignment1

- Select a research hotspot related with Software Architecture.
- Tasks: Do extensive reading & Write a paper or report.