

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação

Arquitetura de Software: Documentação

SSC-0527 – Engenharia de Software
Profa. Dra. Elisa Yumi Nakagawa
Tiago Volpato

Introdução

- Muitas vezes, arquiteturas de software são criadas e não são documentadas (e conseqüentemente, comunicadas) de forma efetiva, ou seja, desenvolvedores e outros com interesse no sistema (*stakeholders*) não têm acesso a uma representação adequada da arquitetura.

Introdução

- Perguntado sobre: “Como vocês documentam arquitetura de software??”
- Na prática, costuma-se ouvir:
 - O que mais precisamos além do diagrama de classes?
 - Desenhamos caixas e linhas (*boxes and lines*).
 - Não nos preocupamos com isso.
 - Usamos UML.
 - Como nós documentamos o quê??!!

Introdução

- Necessidade de documentação de arquitetura:
 - Define as demais atividades que deverão ser realizadas
 - É o primeiro artefato a agregar atributos de qualidade
 - É o melhor artefato nas primeiras fases do desenvolvimento
 - Elemento chave para posterior manutenção

Introdução

- Assim, a documentação de arquitetura de software torna-se o artefato principal em todas as fases do desenvolvimento em que a arquitetura é usada.
- *"Software architecture documentation speak for the architect, today, tomorrow and 20 years from now."*
[SEI]

Princípios de uma Boa Documentação

- Princípios para se criar uma boa documentação:
 - **Princípio 1:** Documentar sob o ponto de vista de quem irá utilizar a documentação
 - **Princípio 2:** Evitar repetições desnecessárias. Entretanto, redundância é aceitável ou desejável se:
 - Uma informação está definida em um local X da documentação e uma elaboração ou refinamento da mesma informação aparece num local Y. Neste caso, repetir a informação (ou parte dela) no local Y é comum.
 - Duas informações são mapeadas uma para outra. É difícil fazer isso sem repetir parte delas.
 - A repetição é conveniente por motivos práticos, quando trocar de página constantemente para entender o texto fica inviável (hyperlinks, quando disponíveis, por exemplo).

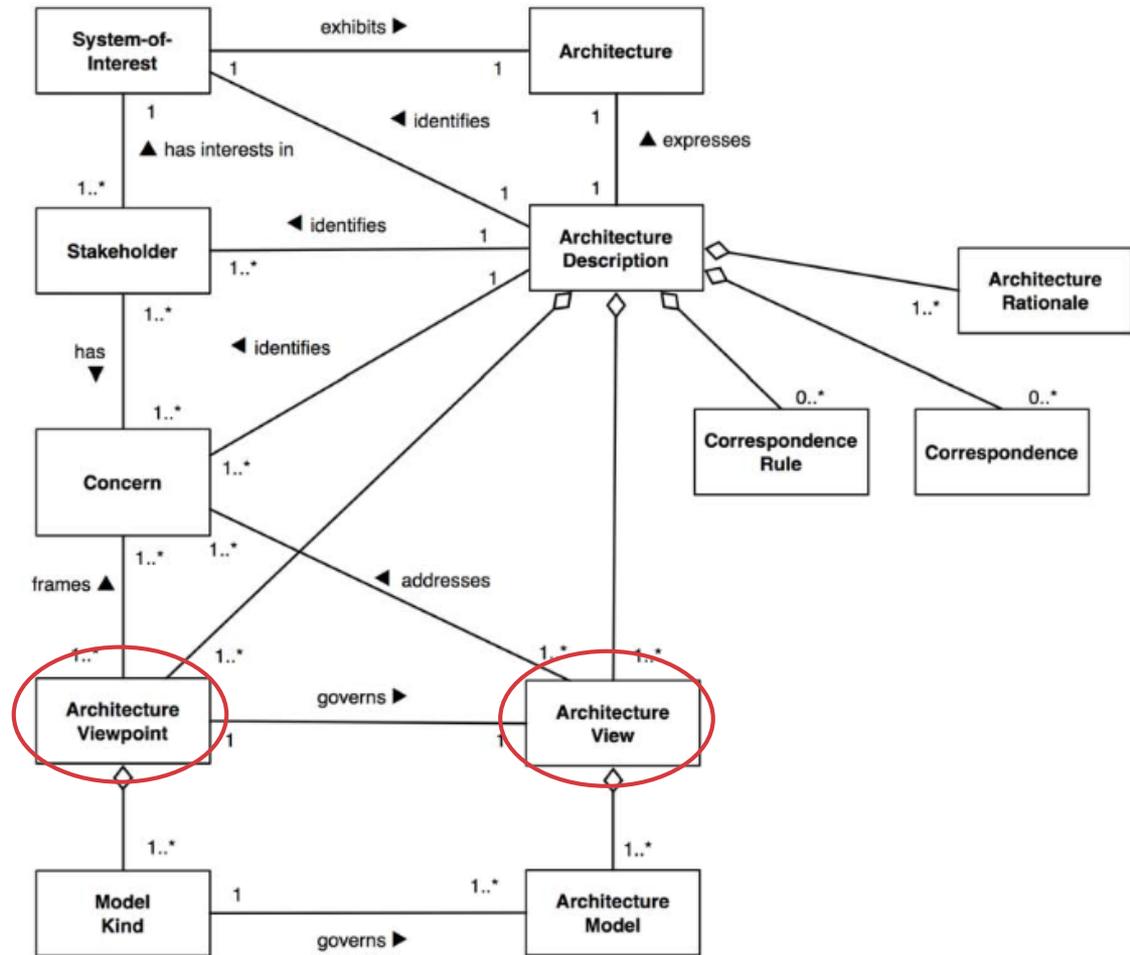
Princípios de uma Boa Documentação

- Princípios para se criar uma boa documentação:
 - **Princípio 3:** Evitar ambigüidade.
 - **Princípio 4:** Usar uma organização padrão para o documento a ser criado, isto é, um modelo ou *template*.
 - **Princípio 5:** Documentar a razão para as decisões tomadas
 - As mais importantes incluem aquelas que resultaram de uma discussão longa, ou que a mudança seria onerosa, ou aquelas que são cruciais para atingir requisitos chave.
 - Deve-se documentar tanto a razão para as decisões tomadas quanto alternativas rejeitadas que sejam importantes.

Princípios de uma Boa Documentação

- Princípios para se criar uma boa documentação:
 - **Princípio 6:** Manter a documentação atualizada
 - **Princípio 7:** Revisar a documentação criada.
 - Usuários da documentação são os melhores candidatos a revisores.

Modelo Conceitual ISSO/IEC/IEEE 42010



Modelo Conceitual ISSO/IEC/IEEE 42010

- Viewpoint
 - Artefato que estabelece as convenções (ou seja, tipos de modelo) para a construção, interpretação e uso de visões da arquitetura.
- View
 - Artefato expressando a arquitetura a partir da perspectiva de preocupações do sistema específico.

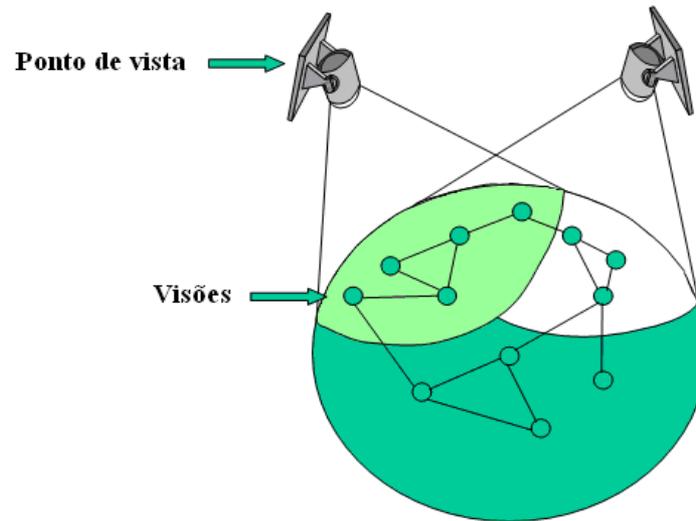
ADL (Architecture Description Language)

- Para a especificação de projetos arquiteturais, ADL têm sido propostas.
- Uma ADL é uma linguagem usada para representar a arquitetura de um sistema de software.
- Pode-se identificar uma diversidade de linguagens:
 - C2 [Medvidovic, 1996]
 - Unicon [Shaw, 1995]
 - Meta-H [Binn, 1993]
 - Rapide [Luckham, 1995]
 - Wright [Allen, 1997]
 - Darwin
 - ACME
 - SADL
 - Aesopa

Visões Arquiteturais

- A documentação de uma arquitetura consiste de múltiplas visões arquiteturais (*architectural views*)
- Visão arquitetural é uma abstração do sistema feita a partir de um conjunto de regras estabelecidas em um determinado ponto de vista (*viewpoint*).
- Ponto de vista é a perspectiva através da qual uma dada visão do sistema é construída.

Visões Arquiteturais



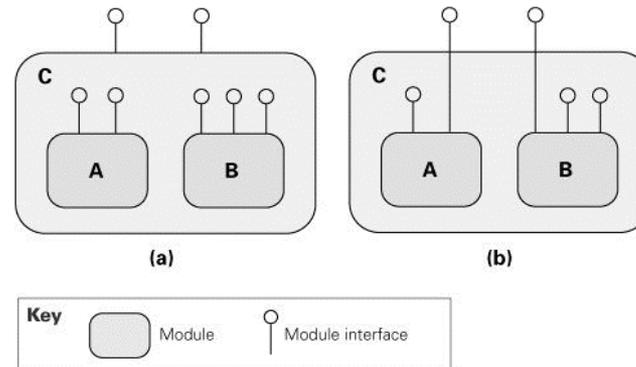
- Questões:
 - Quais visões arquiteturais são relevantes e para quê?
 - Quais notações são melhores para documentar cada visões arquiteturais?

Visões Arquiteturais

- Existem diversos conjuntos de visões arquiteturais propostos por diferentes autores.
- Um dos mais conhecidos é o 4+1 Views, proposto por Kruchten, 1995.
- Esse conjunto contém:
 - Visão de módulos
 - Visão em tempo de execução
 - Visão de implantação
 - Visão de implementação
 - Visão de dados
- OBS: Nem todas as visões são relevantes para todos os sistemas.

Visões Arquiteturais

- Visão de módulos
 - Essa apresenta a estrutura do sistema em termos de unidades de implementação.
- Qual técnica usar para representar essa visão?
 - Caixas e linhas, textos ou tabelas (notação informal)
 - Diagrama de classes da UML (notação semi-formal)

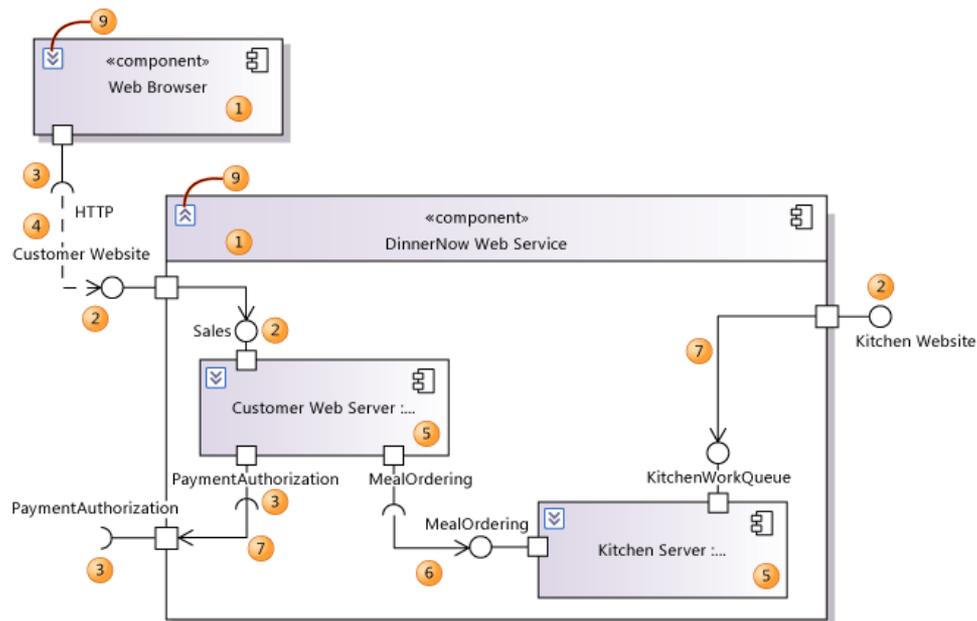


Visões Arquiteturais

- Visão em tempo de execução
 - Essa visão, também chamada de visão C&C (*Component & Connector*) mostra o sistema em tempo de execução.
 - Possibilita o entendimento do funcionamento do sistema e a análise das propriedades que se manifestam em tempo de execução, tais como o desempenho.

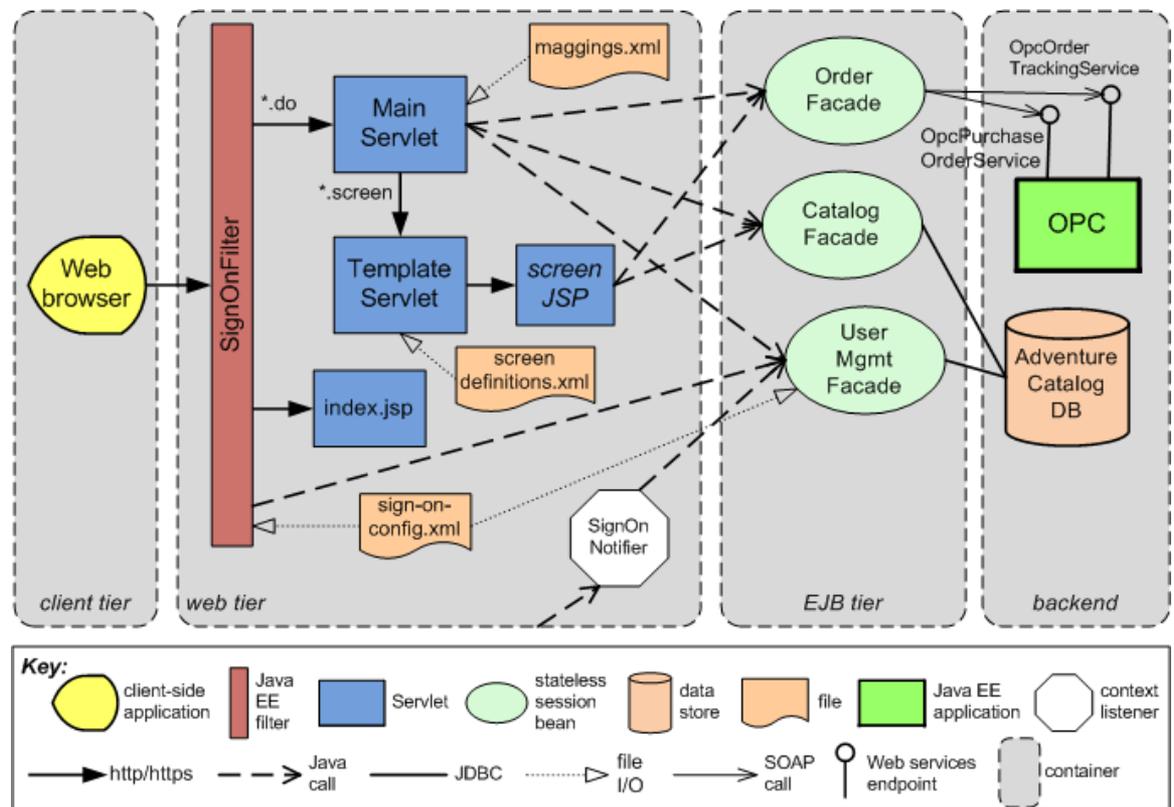
Visões Arquiteturais

- Visão em tempo de execução
 - Possibilita apresentar:
 - os grandes componentes e seus relacionamentos
 - as bases de dados, bem como aquelas que são compartilhadas
 - os elementos replicados
 - o fluxo de dados no sistema e
 - as partes do sistema que são executadas em paralelo.
- Qual técnica usar para representar essa visão?
 - Caixas e linhas (notação informal)
 - Diagrama de componentes da UML 2.0 (notação semi-formal)



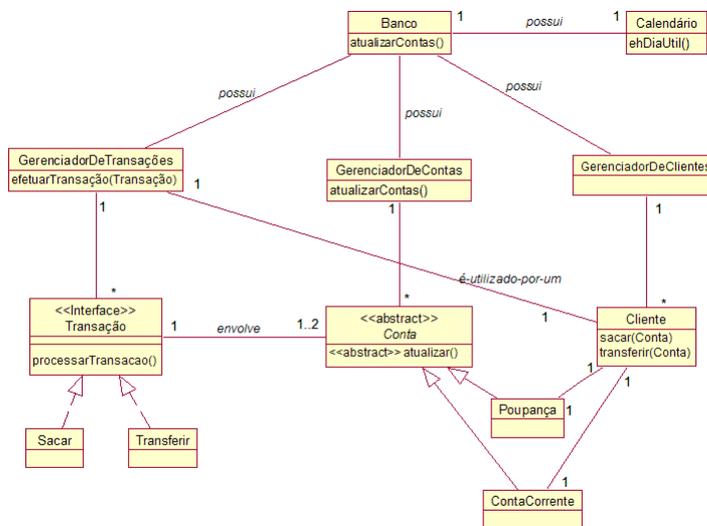
Visões Arquiteturais

- Visão de implantação
 - Mostra a estrutura de hardware (tipicamente uma rede) na qual o sistema é executado
- Qual técnica usar para representar essa visão?
 - Diagrama de redes (notação informal)
 - Diagrama de implantação (*deployment*) da UML 2.0 (notação semi-formal)



Visões Arquiteturais

- Visão de dados
 - É normalmente utilizada quando o sistema possui uma base de dados cuja estrutura precisa ser modelada
 - Esse modelo inicia como um modelo conceitual/lógico que vai sendo refinado até conter toda informação necessária para a criação da base de dados física.
- Qual técnica usar para representar essa visão?
 - técnicas da área de banco de dados, exemplo, MER.
 - Diagrama de classe da UML



Referências

- Bass, L., Clements, P., and Kazman, R. 2003. Software Architecture in Practice (2ed.). Addison-Wesley Longman Publishing Co.
- Gorton, I. 2011. Essential Software Architecture (2ed). Springer-Verlag New York, Inc.
- Kruchten, P. What do software architects really do? In: Journal of Systems and Software, v.81, p.2413-2416. 2008
- Hofmeister, C., Kruchten, P., Nord, R. L., Obbink, H., Ran, A. and America, P. A general model of software architecture design derived from five industrial approaches. In: Journal of Systems and Software, v.80, n.1, p. 106-126. 2007.
- Garland, J. and Anthony, R. 2003. Large-Scale Software Architecture: A Practical Guide Using UML. John Wiley & Sons, Inc., New York, NY, USA.Hofmeister
- ISO/IEC/IEEE 42010:2010 International Standard for Systems and Software Engineering -- Architectural description
- Malavolta, I.; Lago, P.; Muccini, H.; Pelliccione, P. and Tang, A. What Industry Needs from Architectural Languages: A Survey IEEE Transactions on Software Engineering, 2013, v. 39, n. 6, 869-891.
- Lago, P.; Malavolta, I.; Muccini, H.; Pelliccione, P. and Tang, A. The road ahead for architectural languages. IEEE Software, 2014, 32, 98-105.
- Medvidovic, N. and Taylor, R. N. A classification and comparison framework for software architecture description languages. In: IEEE Transactions on Software Engineering, 2000, v. 26, n.1, 70-93.
- Oquendo, F. pi-ADL: An Architecture Description Language based on the Higher Order Typed pi-Calculus for Specifying Dynamic and Mobile Software Architectures. In: ACM Software Engineering Notes, 2004, v. 29, n.3, 15-28.
- Clements, P.; Bachmann, F.; Bass, L.; Garlan, D.; Ivers, J.; Little, R.; Merson, P.; Nord, R.; and Stafford, J. Documenting Software Architectures: Views and Beyond. Addison-Wesley, 2011.
- Shaw, M. and Garlan, D. Characteristics of Higher-Level Languages for Software Architecture. Carnegie Mellon University, 1994. <http://www.sei.cmu.edu/reports/94tro23.pdf>