

## Arquitetura de Software III – Análise Arquitetural / ADLs

Tópicos Avançados de  
Programação Orientada a Objetos  
MAC 413 / 5715

Prof. Fabio  
IME / USP

24/09/02

1

## Análise Arquitetural

- ◆ Às vezes recebemos uma arquitetura pronta e precisamos avaliá-la/analisa-la.
- ◆ Podemos utilizar a nossa experiência.
- ◆ Ou podemos utilizar um processo mais formal. Exemplos propostos por Kazman:
  - SAAM (*Software Architecture Analysis Method*)
  - ATAM (*Architecture Tradeoff Analysis Method*)

24/09/02

2

## SAAM – Kazman 1994 (*Software Architecture Analysis Method*)

- ◆ Método em 5 fases para análise de arquiteturas para sistemas de médio porte.
- 1. Desenvolvimento de cenários
- 2. Descrição da(s) arquitetura(s) candidata(s)
  - ◆ componentes ativos (transformam dados)
  - ◆ componentes passivos (armazenam dados)
  - ◆ conexões de controle
  - ◆ conexões de dados
- 3. Avaliação de cada cenário em cada arq.:
  - ◆ suporte direto
  - ◆ suporte indireto (exige mudanças na arq.)
  - ◆ não oferece suporte

24/09/02

3

## SAAM (*Software Architecture Analysis Method*)

- 4. Determinação da interação entre cenários
  - ◆ mudanças arquiteturais exigidas por um cenário afetam outros cenários
  - ◆ permite avaliar o quão desacoplado e coeso são as componentes do sistema.
- 5. Avaliação global das arquiteturas
  - ◆ monta-se tabela onde as linhas são os cenários e as colunas são as arquiteturas candidatas
  - ◆ em cada entrada
    - ◆ + indica que tudo bem
    - ◆ - indica que a arq. exige mudanças p/ atender ao cenário

24/09/02

4

## ATAM - Kazman 1999 (*Architecture Tradeoff Analysis Method*)

- ◆ Um processo mais sofisticado a ser utilizado por uma empresa ou instituição para decidir qual arquitetura adotar para um sistema de médio ou grande porte. Normalmente dura 3 dias.
- ◆ 1a Etapa – Apresentação
  1. apresentação do método ATAM
  2. apresentação das metas da empresa/inst.
  3. apresentação da arquitetura básica
    - ◆ concentrando-se em como as metas da empresa são atendidas.

24/09/02

5

## ATAM - Kazman 1999 (*Architecture Tradeoff Analysis Method*)

- 2a Etapa – Investigação e Análise
  4. identificação de alternativas arquiteturais.
  5. identificação dos requisitos não-funcionais.
  6. análise das alternativas arquiteturais
    - em relação aos requisitos não-funcionais.
- 3a Etapa – Teste
  7. análise, discussão e determinação de prioridade de cenários.
  8. análise das alternativas arquiteturais em relação aos cenários.
    - também análise dos riscos

24/09/02

6

## ATAM - Kazman 1999 (Architecture Tradeoff Analysis Method)

### 4a Etapa – Relato

9. Apresentação dos resultados

- relatório contendo
  - ◆ cenários de uso
  - ◆ avaliação de alternativas arquiteturais
  - ◆ riscos
  - ◆ forças
  - ◆ plano de ação e estratégias a serem adotadas pela instituição

24/09/02 7

## Exemplos de aplicação do SAAM

- ◆ Exemplos no livro do Antonio Mendes:
  - Capítulo 8: Avaliação Arquitetural de um Sistema de Informações
    - ◆ WWW ou Harvest
  - Capítulo 9: Avaliação Arquitetural de um Sistema de Telecomunicações
    - ◆ central telefônica

24/09/02 8

## Arquiteturas de Domínio Específico

- ◆ Ao contrário de arquiteturas genéricas, arquiteturas de domínio específico podem ser mais especializadas e, portanto, mais eficientes para casos específicos.
- ◆ Segundo [Taylor 95], uma DSSA contém:
  - um modelo do domínio
  - requisitos de referência
  - uma arquitetura de referência
  - a infra-estrutura ou ambiente de suporte
  - um processo de desenvolvimento

24/09/02 9

## Arquiteturas de Domínio Específico Arquitetura de Referência

Descreve todos os sistemas em um domínio. Inclui:

- Estilo arquitetural (ou combinação deles)
- Componentes e conectores típicos
- Interface com outros domínios
  - ◆ (se é necessário usar serviços de outros domínios).
- ◆ Exemplo: *Common ORB Architecture*

24/09/02 10

## Arquiteturas de Domínio Específico Exemplos

- ◆ Arquiteturas em Sistemas de Informação
  - objetivo: armazenamento, recuperação e processamento de dados.
  - casos típicos: sistemas bancários, sistemas de compra/venda, sistemas de bibliotecas, etc.
  - Tipos:
    - ◆ arquitetura monolítica (ex. agenda no Palm)
    - ◆ arquitetura cliente/servidor de 2 camadas
    - ◆ arquitetura cliente/servidor de múltiplas camadas
    - ◆ pode ser baseado na Web ou não

24/09/02 11

## Arquiteturas de Domínio Específico Exemplos

- ◆ Arquiteturas em Sistemas de Tempo Real
  - relógio de tempo real, escalonador de tempo real, controle de tarefas periódicas e não periódicas
  - processamento de interrupções em tempo real
  - gerenciador de recursos específico
- ◆ Arquiteturas em Sistemas Inteligentes Adaptativos
  - Percepção concorrente
  - Sensibilidade a prioridades
  - Sensibilidade a condições limites de tempo
  - Controle geral do comportamento
  - Raciocínio (regras, rede neural, etc.)
  - Ação

24/09/02 12

## Linguagens de Descrição de Arquiteturas (*Architecture Description Languages ou ADLs*)

- ◆ Module Interconnection Languages (MIL)
  - proposto em 1975 por DeRemer e Kron.
  - trabalhava com o código-fonte de módulos, compilando a aplicação com a arquitetura descrita através de MIL.
  - o compilador fazia verificação de tipos entre os módulos.
  - exigia que o código-fonte estivesse disponível e que todos os módulos estivessem na mesma linguagem.
  - gerava uma aplicação centralizada.

24/09/02

13

## Linguagens de Descrição de Arquiteturas (*Architecture Description Languages ou ADLs*)

- ◆ Configuration Languages
  - Kramer & Magee, Imperial College, Inglaterra.
  - pais das ADLs e da configuração dinâmica.
  - Conic 89, Rex 92, Darwin 94.
  - Uma especificação Darwin lista todos os componentes que fazer parte de uma aplicação distribuída e define as ligações entre elas.
  - O sistema carrega dinamicamente todas as componentes em diferentes máquinas e estabelece as conexões (Regis ou CORBA).

24/09/02

14

## Linguagens de Descrição de Arquiteturas (*Architecture Description Languages ou ADLs*)

- ◆ ADLs (finalmente)
  - UniCon (Shaw 95-96) introduziu o conceito de *Conectores Arquiteturais*.
  - Assim, uma definição UniCon é composta de:
    - ◆ descrição das componentes (interfaces utilizadas e exportadas)
    - ◆ descrição dos conectores (pipes, E/S por arquivos e chamadas a funções locais ou remotas)
    - ◆ script de configuração da aplicação [distribuída]
  - ◆ ADLs podem ser usadas não só para instanciar aplicações mas também para verificar matematicamente a consistência e propriedades de aplicações.

24/09/02

15

## Bibliografia

- ◆ Sobre Análise Arquitetural e Arquiteturas de Domínios Específicos veja
  - *Arquitetura de Software – Desenvolvimento orientado para arquitetura*. Antonio Mendes. Editora Campus, 2002.
- ◆ Sobre Linguagens de Descrição de Arquiteturas veja

<http://choices.cs.uiuc.edu/~kon/thesis/node15.html#SECTION00155300000000000000>

24/09/02

16