

Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Coordenação de Pós-Graduação em Informática - COPIN

PROJETO DE SOFTWARE ORIENTADO A OBJETO

RESUMO - Architectural Blueprints - The 4+1 View Model of Software
Architecture
Junho de 2008

João Arthur Brunet Monteiro
Mestrado em Ciências da Computação - CEEI/UFCG
Campina Grande, Junho de 2008

1 Resumo

A descrição da arquitetura é uma atividade de elevada importância para o processo de concepção do software. No artigo *Architectural Blueprints - The 4+1 View Model of Software Architecture* [1] é apresentado um modelo para descrição de arquitetura baseado em múltiplas visões, além das notações para cada uma dessas visões. A abordagem apresentada pelo autor visa eliminar alguns problemas que surgem na descrição da arquitetura quando se leva em consideração poucas visões do sistema. Por englobar vários aspectos do software, a descrição da arquitetura deve levar em consideração estas diferentes visões do sistema para que a qualidade de uma não seja prejudicada em detrimento da outra. O uso de múltiplas visões permite a separação de “*concerns*” relacionados aos vários aspectos presentes na arquitetura.

O autor apresenta quatro visões distintas:

- Visão lógica: Decomposição dos serviços que o sistema deve prover em um conjunto de abstrações (classes, objetos etc). Essas abstrações e seus relacionamentos são mostrados através de diagramas de classes.
- Visão de processos: Leva em consideração requisitos não funcionais do sistema, tais como, controle de concorrência, tolerância a falhas, comunicação distribuída entre os componentes etc. Esta visão é responsável pela representação de *threads*, comunicação entre as abstrações implementadas na visões lógica, processos, e execução de tarefas.
- Visão física: Esta visão leva em consideração requisitos não funcionais do sistema (disponibilidade, tolerância a falhas, escalabilidade e desempenho) sob a ótica dos recursos físicos disponíveis.
- Visão de desenvolvimento: Foca na organização dos módulos do software no processo de desenvolvimento. Esta visão de desenvolvimento do sistema é representada por módulos e diagramas de subsistemas mostrando as interfaces de entrada e saída desses componentes. Esta visão ajuda na organização das entidades e sua interação na ótica do processo de desenvolvimento.

Ao agrupamento dessas visões dá-se o nome de Cenários - casos de uso que descrevem a interação entre objetos e processos, através de diagramas de objetos e diagramas de interação de objetos. Segundo o autor, o uso de Cenários juntamente com as visões anteriores é redundante (por isso o “+1”), no entanto, o autor justifica que esta perspectiva é útil para descobrir elementos da arquitetura e para validação e ilustração dos papéis depois do término do design da arquitetura.

O autor também destaca que as diferentes visões não são necessariamente independentes. Elementos de uma visão podem estar conectados a elementos de outras visões seguindo determinadas regras e heurísticas. Além disso, o uso da abordagem é flexível no que diz respeito a omissão do uso de algumas visões caso o sistema não necessite de tal artefato.

O uso dessa abordagem realmente pode tornar a atividade de descrever a arquitetura mais simples e viável. Acredito nesta afirmativa pelo fato de que diferentes usuários do sistema de diferentes domínios não buscam pela mesma informação ao olhar para a descrição da arquitetura. Se esta descrição é feita levando em consideração somente uma visão, buscar por um elemento na arquitetura pode ser demasiadamente complicado. Explorar a pluralidade dos “concerns” dos usuários facilita o entendimento da arquitetura como um todo, além de tornar a sua descrição mais flexível.

Por fim, acredito que quando a descrição da arquitetura de um sistema aborda diferentes visões, essa descrição se torna mais completa, visto que engloba diferentes domínios do sistema, algo impraticável de ser feito com apenas uma visão.

Referências

- [1] P. Kruchten. The 4+ 1 View Model of architecture. *Software, IEEE*, 12(6):42–50, 1995.