

Introdução ao Catalysis

Tópicos Avançados de Engenharia de Software

João Bosco
jbapf@cin.ufpe.br

Roteiro

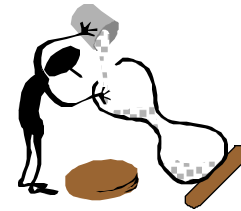


- Dificuldades
- Motivação
- Componentes
- Desenvolvimento Baseado em Componentes (DBC)
- Catalysis

[Dificuldades]



- “Don’t read it all in one night! If you think this is a bit long for a Preface, wait until you see the rest of the book...”
- “Note - This is the 1st half of the Catalysis lecture. Based on progress reviewing the lecture material, 3 to 4 classes may be required”



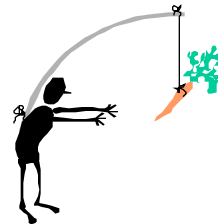
jbaptf@cin.ufpe.br

3

[Motivação]



Por que estudar o Catalysis, ou Desenvolvimento Baseado em Componentes (DBC)?



jbaptf@cin.ufpe.br

4

[Motivação] Requisitos de Softwares



- Integridade
 - Bugs vs Tempo
 - Softwares embutidos
- Desenvolvimento em times
 - Separação das equipes
 - Dependências, convenções arquiteturais e regras de desenvolvimento explícitas
- Flexibilidade
 - Negócios precisam inovar
 - Diferentes versões do mesmo sistema

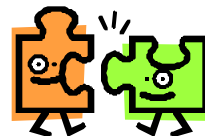
jbapt@cin.ufpe.br

5

[Motivação] Software Flexível



- Desenvolver uma vez, utilizar várias
- Reuso != copiar-e-colar
- Partes Reusáveis:
 - Não podem ser modificadas pelos projetistas
 - Montam um *kit* coerente
 - **Componentes**



jbapt@cin.ufpe.br

6

Exemplos de Partes Reusáveis



- Código compilado [fonte]
- Casos de testes
- Modelos de projetos: frameworks e padrões
- Interface de usuário
- Planos, estratégias e regras arquiteturais

jbaptf@cin.ufpe.br

7

Componentes



*“Componentes reutilizáveis são artefatos **autocontidos**, claramente identificáveis, que descrevem ou realizam uma **função específica** e têm **interfaces** claras em conformidade com um dado modelo de arquitetura de software, **documentação** apropriada e um grau de **reutilização** definido.”*



Sametinger [1997]

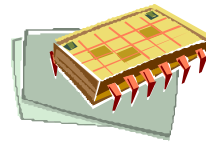
jbaptf@cin.ufpe.br

8

Componentes



- Aspectos de um componente
 - Descreve ou realiza uma função específica
 - Está em conformidade com, e provê um conjunto de interfaces definidas
 - Tem disponível uma documentação adequada
- Cases: Componentes de Hardware e Automóveis



jbaptf@cin.ufpe.br

9

Onde estão os Componentes?



- Pequena Escala:
 - Componentes 'arraste-e-solte' para *forms*, como os que vêm no Visual Studio .NET ou JBuilder
- Média Escala:
 - Programas com componentes auto-contidos, como o 'pipe' do Unix, ou chamadas OLE/COM numa mesma aplicação.
- Grande Escala:
 - Componentes distribuídos em máquinas diferentes, acessíveis através de DCOM, CORBA ou WebServices

jbaptf@cin.ufpe.br

10

Desafios para o DBC



- Tecnologias vs. Metodologias
- Se faz necessário:
 - Técnicas e procedimentos repetíveis e claros
 - Arquiteturas padrão e bem definidas
 - Comunicação sem ambigüidade

Métodos para DBC



- É necessária uma abordagem que considere a reutilização em todas as etapas do desenvolvimento
- Deve ser considerado em todos os níveis, não somente código executável
- Desenvolvimento de Componentes
- Desenvolvimento com Componentes

Métodos para DBC

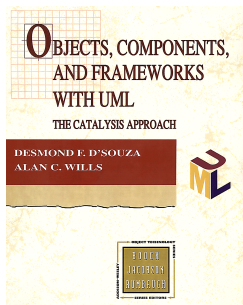


- Abordagem relativamente nova
- Alguns métodos são extensões de métodos já existentes, em particular, métodos de desenvolvimento Orientado a Objetos
- Métodos
 - Catalysis
 - {RUP}
 - Select Perspective
 - UML Components
 - Abordagem de DBCD [Almeida, 2003]

jbapt@cin.ufpe.br

13

Catalysis



“... Software development continues to be, as always, a difficult and fascinating mixture of art, science, black magic, engineering, and hype... Catalysis defines an approach that covers all of them...”

[D'Souza, 1999]

jbapt@cin.ufpe.br

14

[Catalysis] Definição



- [Aurélio, ~1500 d.c.]

Catálise: s. f. **Quím** - Fenômeno que incentiva uma reação química ou a alteração de sua velocidade pela adição de uma substância (catalisador), que aparece inalterada quimicamente no fim da reação.

- [D'Souza, 1999]

"Método sistemático para partir e separar software de maneira gerenciável."



jbapt@cin.ufpe.br

15

[Catalysis] Origem



- Desenvolvido na Universidade de Brighton, Inglaterra, por:
 - Desmond D'Souza – Trabalha no laboratório de computação da Platinum's ICON
 - Alan Wills – Consultor da TriReme
- Método de DBC completo, cobrindo todas as fases, desde a especificação dos componentes até sua implementação para reutilização [D'Souza, 1999]

jbapt@cin.ufpe.br

16

[Catalysis] Intenções



- Busca solucionar problemas comuns no DBC, através de:
 - Rastreamento de requisitos
 - Semântica para verificação de consistência
 - Níveis de granularidade e refinamento
 - Distinção entre modelos do domínio, modelos do sistema e arquitetura
 - Comunicação (dicionário)

jbaptf@cin.ufpe.br

17

[Catalysis] Princípios Básicos



- **Abstraction**
- **Precision**
- **Pluggable Parts**

Abstração

Foco nos **aspectos essenciais**, adiando os outros. Criação de descrições para organização dos requisitos e arquitetura.

Precisão

Expõe lacunas e **inconsistências** cedo. Torna os **modelos abstratos** mais precisos, e evita confusões.

Partes Conectáveis

Todo trabalho está em adaptar e compor as partes.

jbaptf@cin.ufpe.br

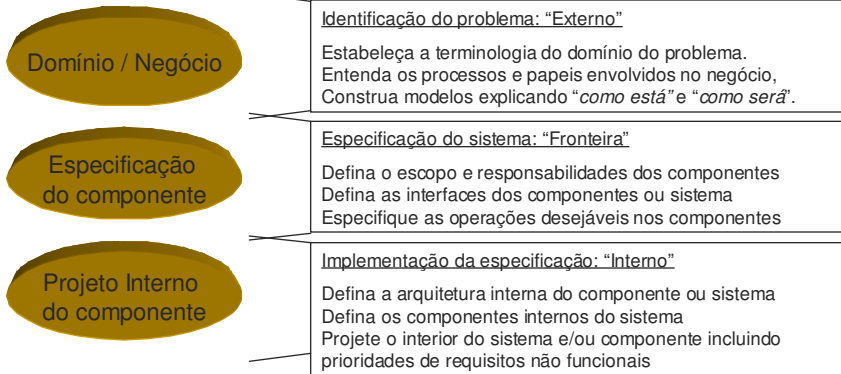
18

[Catalysis] Níveis de Modelagem



Nível e/ou Escopo

Objetivos



jbapf@cin.ufpe.br

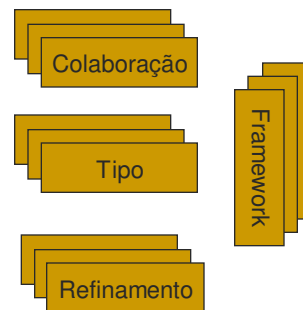
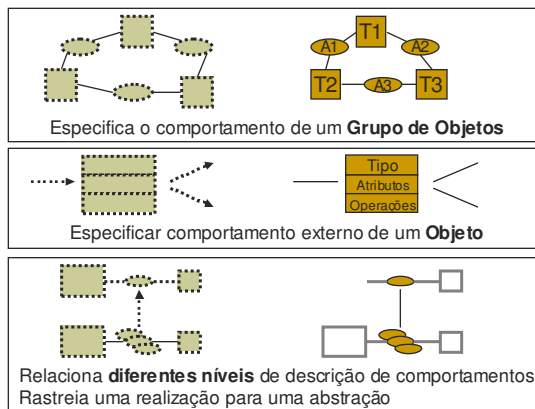
19

[Catalysis] Framework de Construção



Propósito

Modelo de Construção



jbapf@cin.ufpe.br

20

[Referências]



- [Almeida, 2003] Almeida, E.S. **Uma Abordagem para o Desenvolvimento de Software Baseado em Componentes Distribuídos**, Dissertação de Mestrado, Universidade Federal de São Carlos, 2003.
- [Aurélio, 1500]
- [D'Souza, 1999] D'Souza, D., F., Wills, C., A. **Objects, Components, and Frameworks with UML – The Catalysis Approach**. Addison-Wesley, 1999.
- [Sametinger, 1997] Sametinger, J. **Software Engineering with Reusable Components**. Springer-Verlag, 1997.

jbapt@cin.ufpe.br

21

[Perguntas?]



jbapt@cin.ufpe.br

22