

UTILIZANDO ESTILOS ARQUITETURAIS NO PROJETO DE UM SISTEMA DE COMPRAS PELA INTERNET

Rodrigo Carlo Gurgel Martiniano¹
Bartira Paraguaçu Falcão Dantas Rocha²

Resumo

Este trabalho apresenta Estilos Arquiteturais, sub-área da Arquitetura de Software, que tem como objetivo o desenvolvimento de sistemas de software baseados em padrões de arquiteturas pré-definidas, possibilitando o reúso de configurações de componentes pertencentes a um sistema que tenha características particulares com outros sistemas. As principais características e uma classificação de Estilos Arquiteturais, assim como a apresentação de um estudo de caso usando estilos heterogêneos de um sistema de compras pela Internet, são expostas neste trabalho.

Palavras-chave: *Estilos Arquiteturais; Arquitetura de Software; Componentes de Software.*

1 INTRODUÇÃO

Ao longo das últimas décadas, o software teve grande evolução, tornando-se parte essencial nos sistemas de computação. Contudo, ao mesmo tempo em que os sistemas computacionais tornam-se onipresentes em inúmeras aplicações, o tamanho e a complexidade do software têm crescido tanto que as técnicas utilizadas já não são suficientes para lidar com projetos de software com o nível de detalhes inerentes ao sistema. O crescimento em escala dos sistemas de software traz fatores de preocupação para seus desenvolvedores, como desempenho, qualidade, confiabilidade e manutenção. A engenharia de software apresenta diversas técnicas para construção de sistemas computacionais de uma forma segura e clara.

Segundo Mendes (2002), a Arquitetura de Software procura analisar as propriedades dos softwares no nível de subsistema ou módulo, atuando como uma estrutura a fim de atender os requisitos de sistemas, além de ser utilizada

¹ Discente do Curso de Bacharelado em Sistemas de Informação (FARN). E-mail: ro_gurgel@ig.com.br

² Mestre em Sistemas e Computação (UFRN), professora dos cursos de Bacharelado em Sistemas de Informação e Licenciatura em Computação (FARN). E-mail: bartira@farn.br

como aspecto técnico para projetos de sistemas, bem como suporte na estimação de custos e na gerência do processo. Também serve de base para análise da consistência fornecendo suporte ao reúso.

Um dos principais aspectos do projeto arquitetural é o uso dos padrões de organização de sistemas (padrões de arquitetura de software) ou *estilos arquiteturais*. Tais padrões buscam desenvolver projetos seguindo tipos definidos de arquiteturas que mais se adequam às características de um sistema específico.

2 ARQUITETURA DE SOFTWARE

Segundo Mendes (2002) Arquitetura de Software é o estudo da organização global dos sistemas de software bem como do relacionamento entre subsistemas e componentes. Desde suas origens, quando descrições qualitativas de organização de sistemas eram consideradas úteis, a arquitetura de software tem amadurecido ao longo da última década, buscando englobar e exportar notações, ferramentas e técnicas de análise.

Várias definições são dadas para arquitetura de software, sendo que a maioria delas faz uso dos conceitos de componentes e conexões. Uma Arquitetura de Software descreve os elementos com os quais um sistema é construído, assim como a forma como esses componentes irão interagir - uso de padrões (GARLAN; SHAW, 1994).

A Arquitetura de Software é resultado de uma série de influências exercidas pelas diversas pessoas envolvidas na concepção e utilização do sistema. Essas diversas visões, e muitas vezes divergentes, podem gerar uma série de conceitos ambíguos ou até mesmo contraditórios. Arquitetos devem identificar e ativamente comprometer os diversos envolvidos para solicitar suas necessidades e expectativas, além de gerenciar expectativas e negociar prioridades através de diplomacia, negociação e comunicação.

A criação de uma arquitetura envolve diversas atividades:

- 1 Entender os requisitos do sistema (Uso de técnicas como cenários, casos de uso, máquinas de estados finitos, criação e protótipos, etc)
- 2 Criar ou selecionar a arquitetura (Uso de estilos e Linguagens de Descrição de Arquiteturas)
- 3 Representar e comunicar a arquitetura
- 4 Analisar e avaliar a arquitetura

- 5 Implementar baseando-se na arquitetura desenvolvida
- 6 Fazer manutenção

3 ESTILOS ARQUITETURAIS

Ao caracterizar as arquiteturas de software que são utilizadas em sistemas, identificamos seus componentes, mecanismos de interação e propriedades, podendo dessa forma classificá-las. Como resultado, isso nos possibilita o reúso desse conhecimento. No decorrer da década de 1990, houve uma crescente preocupação em compreender a organização dos sistemas de software, culminando com o surgimento da área de arquitetura de software. Dessa forma, famílias de arquiteturas começaram a ser identificadas que correspondem a classes de arquiteturas que possuíam aspectos e componentes peculiares a elas, bem como as formas de combiná-los. Isso caracteriza um *estilo arquitetural* (MENDES, 2002).

Estilos Arquiteturais caracterizam uma arquitetura de software de acordo com os componentes, mecanismos de interação e propriedades, isto é, são classes de arquiteturas semelhantes que possuem aspectos e componentes peculiares e elas, bem como as formas de combiná-las (MENDES, 2002).

Um estilo arquitetural permite que um profissional (engenheiro, projetista ou arquiteto de software) determine a classe à qual pertence a organização de um sistema. Características dos componentes (subsistemas) e conectores dos sistemas, topologia da arquitetura, restrições semânticas e mecanismos de interação entre os componentes ajudam a identificar o estilo que retrata a arquitetura de software do sistema (MENDES, 2002).

Para Bass et al. (1998), um estilo arquitetural de software é de alguma forma análogo a estilos arquitetônicos em edifícios, como gótico ou grego. Ele consiste de algumas características chaves e regras para combinar essas características de forma que a integridade arquitetural seja preservada. Um estilo de software determina:

- Um conjunto de tipos de componentes (por exemplo, repositório de dados) que executam alguma função em tempo de execução.
- Um layout topológico destes componentes que indicam o relacionamento entre eles em tempo de execução.
- Um conjunto de restrições semânticas (por exemplo, em um repositório de dados não é permitido mudar os dados armazenados nele).

- Um conjunto de conectores (por exemplo, chamada de procedimento remoto) que mediam a comunicação, coordenação ou cooperação entre componentes.

Mary Shaw e David Garlan catalogaram um conjunto de estilos arquiteturais por eles observados em um conjunto de sistemas complexos existentes (BASS et al. 1998). Alguns desses estilos que foram catalogados são apresentados a seguir.

3.1 Arquiteturas Centradas em Dados

Arquiteturas Centradas em Dados têm como objetivo obter a integração de dados, os quais poderão ser acessados e atualizados por componentes do sistema. Um cliente é executado como uma linha independente de controle. Os dados compartilhados que todo cliente acessa podem ser um repositório passivo (como um arquivo) ou um repositório ativo (como um quadro-negro – *blackboard*). A principal característica desse tipo de estilo é que um repositório de dados centralizado serve de comunicação entre vários clientes.

3.2 Arquiteturas de Fluxo de Dados

Arquiteturas de Fluxo de Dados têm o objetivo de alcançar as qualidades de reuso e modificabilidade. O estilo fluxo de dados é caracterizado por visualizar o sistema como uma série de transformações em pedaços sucessivos de entradas de dados. Dados entram no sistema e então fluem pelos componentes, um de cada vez até que eles são direcionados a algum destino final (saída ou depósito de dados). O estilo tem dois subtipos, seqüencial e pipe-and-filter.

3.3 Arquiteturas de Máquinas Virtuais

Arquiteturas de Máquina Virtuais têm o objetivo de alcançar a qualidade de portabilidade. Máquinas virtuais são estilos de software que simulam alguma funcionalidade não nativa ao software e/ou hardware no qual são implementados. Isso pode ser útil de vários modos: pode permitir a simulação (e teste) de plataformas que ainda não tenham sido construídas (como um hardware novo) e pode simular modelos críticos (como simuladores de vôo) que seriam muito complexos, caros, ou perigosos para testar diretamente em um sistema real (BASS et al. 1998).

3.4 Arquiteturas de Chamada-e-Resposta

Arquiteturas de Chamada e Resposta têm o objetivo de alcançar as qualidades de modificabilidade e escalabilidade. Arquiteturas de chamadas e

respostas têm sido o estilo arquitetural dominante em grandes sistemas de software durante os últimos 30 anos. Assim como esse estilo, um número de sub-estilos com características semelhantes têm surgido. Um exemplo desse tipo de arquitetura é o Orientado a Objetos.

3.5 Arquiteturas de Componentes Independentes

Arquiteturas de Componentes Independentes consistem em vários processos ou objetos independentes que se comunicam por mensagens. Todas estas arquiteturas têm o objetivo de alcançar modificabilidade através da decomposição da computação em vários pedaços. Eles enviam dados um ao outro, mas tipicamente não controlam um ao outro diretamente. As mensagens podem ser passadas a participantes nomeados ou usar o paradigma de publicar/subsubescrever, entre participantes não nomeados.

3.6 Estilos Heterogêneos

Para Bass et al. (1998), sistemas são raramente construídos a partir de um único estilo, sendo tais sistemas considerados heterogêneos. Há três tipos de heterogeneidade que são mostradas a seguir:

- *Localmente heterogêneo*: significa que sua estrutura em tempo de execução irá revelar padrões de diferentes estilos.
- *Hierarquicamente heterogêneo*: significa que um componente de um estilo, quando decomposto, é estruturado de acordo com as regras de um estilo diferente.
- *Simultaneamente heterogêneo*: significa que qualquer um de vários estilos pode definir o sistema.

Esta última forma de heterogeneidade reconhece que estilos não dividem arquitetura de software em categorias distintas. O estilo Centrado em Dados pode ser composto de clientes independentes, como se fosse uma arquitetura de componentes independentes. As camadas em um Sistema em Camadas pode incluir objetos ou componentes independentes ou até mesmo subrotinas em um sistema de programas principal e subrotinas. Os componentes em um sistema pipe-and-filters normalmente são implementados como processos que operam independentemente, enquanto esperam uma determinada entrada em sua porta; novamente, isso é semelhante a sistemas de componentes independentes cuja ordem de execução é predeterminada (BASS et al., 1998).

3.7 Partes de um estilo: Componentes e Conectores

Os tipos permitidos de componentes e conectores são formas importantes de distinção entre estilos. Componentes e conectores são os blocos primários de construção de arquiteturas. Nesta categoria, um componente é uma unidade de software que executa alguma função em tempo de execução. Exemplos incluem programas, objetos, processos, e filtros. Um conector é um mecanismo que media a comunicação, coordenação, ou cooperação entre componentes. Implementações de conectores podem ser feitas em cima de muitos componentes, freqüentemente eles não correspondem a elementos do sistema executando. Exemplos incluem representações compartilhadas, chamadas de procedimento remoto, protocolos de passagem de mensagens e fluxos de dados (Bass et al.1998).

Selecionar os tipos de partes constituintes não identifica exclusivamente o estilo, existindo outros tipos de características a serem consideradas.

3.8 Usando Estilos em Projetos de Sistemas

Qual estilo você deveria escolher para projetar um sistema? A resposta (é claro) depende das características do sistema a ser projetado. Se você pedir para um arquiteto que lhe fale sobre a arquitetura de um sistema, a resposta será expressa em termos de soluções arquiteturais. Se o sistema tem que ser super confiável, isso poderia ser problemático, pois provavelmente se ouviria falar primeiro dos aspectos de tolerância a falhas e redundância da arquitetura, ou talvez o sistema tenha que ter alto desempenho, o que ocasionaria outras respostas do arquiteto. Se o arquiteto soubesse que o sistema ia durar por muito tempo, crescendo e sendo modificado, você pensaria sobre objetos em camadas e tipos de dados abstratos usados para encapsular mudanças. Se o sistema também tivesse que ser seguro poderia utilizar criptografia (BASS et al., 1998).

Um estilo pode servir como a descrição primária de sistemas em uma área onde há muitas incertezas. Outros estilos podem se bem aplicados e podem ser frutíferos. Mas uma regra boa é “primeiro as primeiras coisas”. Começar com a estrutura arquitetural que provê uma maior melhora nas qualidades (inclusive funcionalidade) que se acredita serem mais problemáticas. A partir dessa estrutura arquitetural, considere um estilo apropriado que enderece essas qualidades. Nesse ponto, outras estruturas e estilos podem entrar em conjunto para ajudar a endereçar assuntos secundários (BASS et al., 1998).

4 DEFININDO UMA ARQUITETURA PARA UM SISTEMA DE COMPRAS PELA INTERNET

O sistema a ser modelado trata-se de um sistema de comércio eletrônico via WEB, onde foram analisadas características específicas do sistema em questão. O sistema envolve a compra de determinados produtos por parte de um cliente, assim como a administração destes produtos. Fazendo uma análise dos componentes que fazem parte do sistema, identificamos os componentes *Cliente*, *Admin* (Administrador do Sistema), *Produtos*, *Usuário* e *Compras*. O relacionamento existente no sistema ocorre entre *Admin* e *Produto*, *Cliente* e *Compras*, *Compras* e *Produtos* e entre *Compras* e *Usuário*. O estilo identificado para modelar o sistema é heterogêneo, tendo como sub-estilo principal o estilo de arquitetura Cliente-Servidor. Os sub-estilos Orientado a Objeto (Chamada e Resposta) também é usado.

As figuras FIGURA 1, FIGURA 2, e FIGURA 3, apresentam os componentes e o relacionamento entre eles.

O estudo de caso tem por finalidade mostrar a importância da Arquitetura de Software, em específico o uso de Estilos de Arquitetura de Software, selecionados a partir das características do sistema e das características já definidas por algumas arquiteturas e estilos.

Na FIGURA 1 o estilo utilizado é o cliente/servidor, onde os componentes ADMIN e CLIENTE são vistos como os clientes no estilo e o SISTEMA é visto como o servidor de serviços. Podemos observar que o relacionamento entre ADMIN-SISTEMA envolve fluxo de controle e fluxo de dados, enquanto o relacionamento entre CLIENTE-SISTEMA só envolve fluxo de dados. O componente SISTEMA é visto como uma nova arquitetura, isto é, não foi modelado a partir de nenhum dos estilos apresentados, sendo composto pelos componentes PRODUTOS, USUÁRIO e COMPRAS. O relacionamento entre eles é apresentado nas figuras FIGURA 2 e FIGURA 3.

Na FIGURA 2 podemos observar o relacionamento entre o componente ADMIN e PRODUTOS, que faz parte de SISTEMA. O componente ADMIN representa o administrador do sistema que tem como principais atividades incluir, excluir e alterar produtos do sistema. O componente PRODUTO é visto como um objeto que possui os métodos *Incluir*, *Excluir*, *Alterar* e *Localizar*. O estilo utilizado é o Orientado a Objetos.

Na FIGURA 3 é representado o relacionamento de CLIENTE com o SISTEMA. O estilo utilizado mais uma vez foi o estilo Orientado a Objetos. Esse

relacionamento representa as ações que ocorrem no momento em que o cliente acessa o site de compras e pode:

- *Realizar a busca por um determinado produto:* envolve o método *Busca* do objeto COMPRAS com o método *Localizar* do objeto PRODUTO. Cada produto buscado pode ser colocado em um carrinho que pode ser representado por uma estrutura de dados dentro do componente COMPRAS (não é apresentado na arquitetura pois não é uma característica arquitetural, e sim implementacional)
- *Comprar os produtos selecionados:* envolve a identificação do cliente (método *Identificar* do objeto COMPRAS que pode *autenticar* ou *cadastrar* novo usuário – métodos do objeto USUÁRIO); após ter sido devidamente identificado, o cliente pode *finalizar* ou *alterar* – métodos do objeto COMPRAS – a lista de compras feita.

Os métodos citados, assim como as características internas dos objetos, não são apresentados na arquitetura pois são descritos na fase de implementação.

FIGURA 1 – Visão Geral da Arquitetura – *Cliente/Servidor*

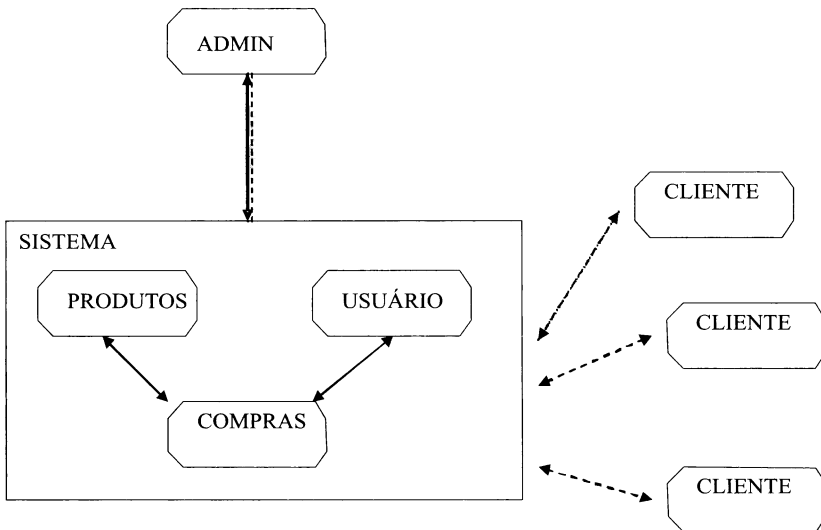


FIGURA 2 – Arquitetura de Chamada e Resposta – *Orientado a Objetos*

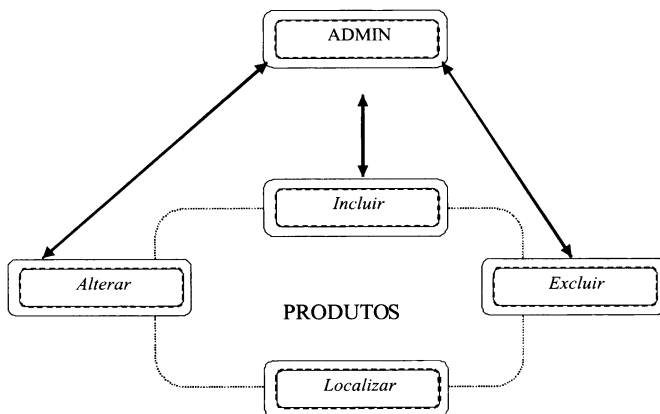
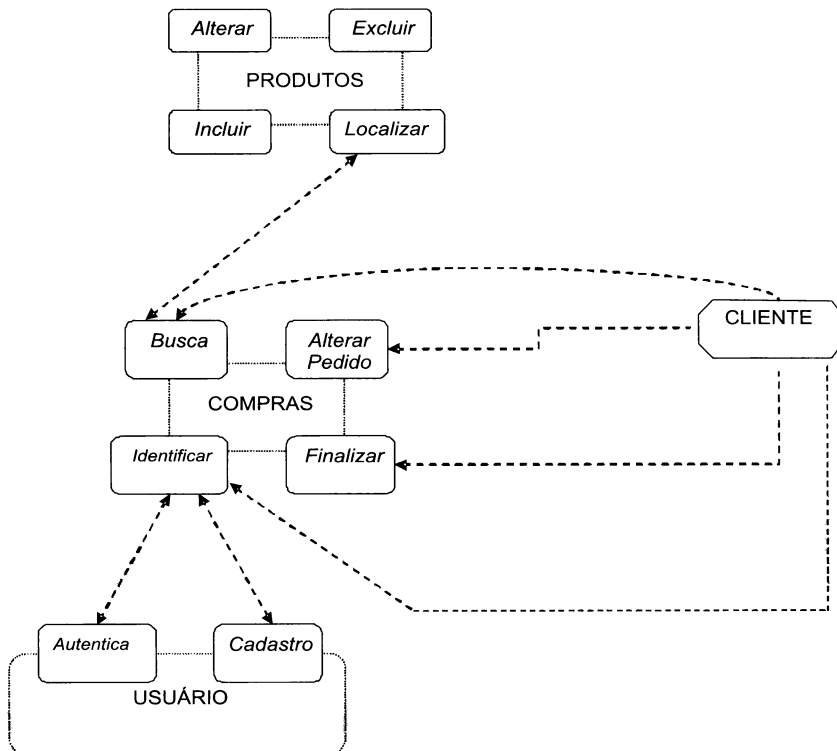


FIGURA 3 – Arquitetura de Chamada e Resposta – *Orientado a Objetos*



5 CONCLUSÃO

A Arquitetura de Software permite a concepção de sistemas a partir de seus componentes e conexões, abstração essa que permite o projetista dedicar-se à estrutura do sistema sem se preocupar com detalhes de implementação. A idéia de estilos de arquitetura de software pré-definidos viabiliza e agiliza a concepção do sistema como um todo. A escolha desses estilos é feita de acordo com características específicas do sistema que está sendo projetado. Vários estilos são propostos e a partir deles é possível escolher qual o mais adequado em cada sistema em particular.

O estudo de caso de um sistema de comércio eletrônico permite a descrição de seus componentes e conectores, onde a identificação das características desses elementos é de fundamental importância para a definição dos estilos utilizados.

A distinção clara entre os diversos estilos apresentados e das características do sistema permite ao arquiteto desenvolver uma arquitetura clara e específica que servirá como estrutura geral na fase de implementação do sistema, evitando dessa forma ambigüidades e redundâncias de projeto.

O uso de estilos arquiteturais de software proporciona para os projetistas de software maior segurança, inclusive no reúso, manutenção, confiabilidade e redução nos custos de projetos de software. Permite também uma maior interação entre as várias pessoas envolvidas no projeto do sistema, tais como, cliente, projetista, usuário, etc. A arquitetura de software fornece um mecanismo de comunicação comum entre essas pessoas.

Os estilos arquiteturais de software permitem ao projetista uma melhor visão de como os componentes e conectores poderão se comportar no sistema e essa visão pode evitar inúmeros problemas no desenvolvimento e implementação do sistema.

REFERÊNCIAS

BASS, L.; CLEMENTS, P.; KAZMAN, Rick. **Software architecture in practice**. Estados Unidos: Addison Wesley, 1998.

GARLAN, David; SHAW, Mary. **An introduction software architecture**. School of Computer Science Carnegie Mellon University Pittisburgh, 1994.

MENDES, Antônio. **Arquitetura de software**: desenvolvimento orientado para arquitetura. Rio de Janeiro: Campus, 2002.

Abstract

This paper presents Architectural Styles, a sub-area of Software Architecture, that aims to develop software systems based on preestablished architectural patterns. These systems will make possible the reuse of components' configurations that belong to a system that has characteristics that are particular to other systems. The main traits and a that has characteristics and classification of Architectural Styles, as well as the presentation of a case study, using the heterogeneous styles of a purchasing system throught the Internet, are showed in this paper.

Key words: *Architectural styles; software architecture; software components.*

