

# Mecanismos de Reutilização em Sistemas de Informação

## *Reuse Mechanisms in Information Systems*

Jugurta Lisboa Filho  
Cirano Iochpe

### **Resumo**

---

*A idéia da reutilização, que é inerente ao desenvolvimento de sistemas computacionais, pode implicar no aumento de produtividade e qualidade em todas as etapas do desenvolvimento de software. Este artigo tem como objetivo identificar os principais mecanismos de reutilização aplicados nas etapas iniciais do processo de desenvolvimento de sistemas de informação (early reuse). Inicialmente são descritas algumas pesquisas sobre reutilização de especificação de requisitos desenvolvidas, principalmente, no início da década de 90. Em seguida são descritos novos mecanismos surgidos para auxiliar na reutilização de projetos orientados a objetos como padrões (patterns), com ênfase nos padrões de análise e frameworks. O estudo também descreve os mecanismos atuais de apoio à reutilização na área de Sistemas de Informação Geográfica (SIG).*

*Palavras-chave: Reutilização, Padrões de análise, Framework, SIG.*

### **Abstract**

---

*The idea of reuse that is inherent to the computation system development can result in productivity and quality increase, in all the software development phases. This paper identifies and presents the mainly reuse mechanisms that may be applied in the early stage of the information system development. Researches about requirement analysis that were developed in the 90's are described. After that, new mechanisms that support reuse in object-oriented software design are investigated. It includes design and analysis patterns as well as frameworks. The paper also describes current mechanisms that support the reuse in Geographic Information Systems (GIS).*

*Key words: Reuse, Analysis Patterns, Framework, GIS.*

---

Jugurta Lisboa Filho é Professor Assistente Universidade Federal de Viçosa, do Departamento de Informática - Campus Universitário. 36571-000, Viçosa, MG, Brasil. Fone/Fax: (31) 891-2397

E-mail: [jugurta@dpi.ufv.br](mailto:jugurta@dpi.ufv.br)

Cirano Iochpe é Professor Adjunto da Universidade Federal do Rio Grande do Sul - Instituto de Informática. Caixa Postal 15064 -91501-970, Porto Alegre, RS, Brasil. Fone: (51) 316-6820 Fax: (51) 3191576. E-mail: [iochpe@inf.ufrgs.br](mailto:iochpe@inf.ufrgs.br)

# 1. Introdução

A idéia da reutilização é inerente ao desenvolvimento de sistemas computacionais. Durante as décadas de 60 e 70, a idéia de reutilização centrava-se, basicamente, na reutilização de códigos de programas (ou subprogramas). Bibliotecas de funções, escritas em diversas linguagens de programação, foram criadas e disponibilizadas para que pudessem ser reutilizadas. A busca por mecanismos voltados a facilitar a reutilização de software teve como resultado uma série de eventos científicos dedicados ao tema. Em 1983 ocorreu o primeiro grande congresso na área, o *Workshop on Reusability in Programming* (Perlis, 1983 Apud [NEI 94]).

Como um dos principais instrumentos de aumento da produtividade dos programadores e da qualidade dos sistemas desenvolvidos, a idéia de reutilização sempre esteve associada à criação de mecanismos que possibilitem a administração da complexidade dos sistemas. Por exemplo, o conceito de Tipo Abstrato de Dados (TAD) foi desenvolvido para possibilitar que estruturas de dados pudessem ser reutilizadas [EMB 87]. Para usar um TAD, o programador necessita conhecer apenas a interface e a funcionalidade de cada um dos subprogramas disponíveis. Outro exemplo de instrumento voltado à reutilização é o paradigma da orientação a objetos que tem, como um de seus princípios fundamentais, possibilitar a reutilização de classes existentes através de mecanismos como especialização e polimorfismo. Atualmente, na área de Engenharia de Software, pesquisas sobre reutilização exploram conceitos como *framework* e padrões [COL 96].

Em cada uma das fases do ciclo de vida de um sistema, existe a possibilidade de se reutilizar resultados obtidos anteriormente. Um exemplo que pode-se citar é o emprego de bibliotecas de funções de interface homem-máquina, durante a fase de codificação dos programas. O objetivo deste artigo é revisar alguns dos principais instrumentos que podem ser aplicados nas tarefas desenvolvidas nas fases iniciais do ciclo de vida.

O restante do artigo está organizado como segue. A Seção 2 apresenta os principais avanços obtidos pela pesquisa em reutilização de especificação de requisitos. Na Seção 3 é des-

critado o conceito de padrões e relacionados diversos tipos de padrões existentes. A Seção 4 descreve a técnica de reutilização baseada em *frameworks*. A Seção 5 descreve as abordagens de reutilização existentes na área de Sistemas de Informação Geográfica (SIG). Por último, a Seção 6 apresenta comentários, deficiências e vantagens das técnicas apresentadas.

## 2. Reutilização de Especificação de Requisitos

Nesta seção são apresentados alguns resultados de pesquisas sobre reutilização no campo da Engenharia de Software, pesquisas estas desenvolvidas até meados da década de 90.

Segundo Neighbors [NEI 94], no processo de desenvolvimento de software, quanto mais cedo forem identificados e criados recursos reutilizáveis, maior será o impacto da reutilização nas fases posteriores. Portanto, a reutilização de modelos durante a especificação de requisitos é o primeiro momento em que o projetista pode empregar algum tipo de recurso reutilizável, isto é, qualquer recurso existente que possa ser útil na construção de um sistema computacional. Para Edelweiss [EDE 94], o uso de abordagens de reutilização durante a fase de especificação de requisitos apresenta as seguintes vantagens:

- redução do custo do desenvolvimento da especificação;
- redução do custo de verificação e validação da especificação;
- aumento da produtividade no desenvolvimento de especificações;
- aumento da qualidade das especificações;
- padronização de especificações;
- facilidade de comunicação entre equipes que utilizam a mesma biblioteca.

Zirbes [ZIR 95] demonstrou, através de experimentos realizados com diferentes técnicas de análise de requisitos, que a modelagem baseada na reutilização de especificações de sistemas análogos, produz resultados mais completos e corretos do que a modelagem sem o apoio da reutilização sistemática. Os resultados da experiência desenvolvida por Zirbes mostram que os ganhos obtidos com o empre-

go da reutilização não estão apenas em ganhos de qualidade, mas também em produtividade onde, embora o tempo de desenvolvimento medido durante o experimento não tenha apresentado um ganho muito significativo, o tempo total dedicado ao sistema é reduzido por diversos fatores indiretos, dentre eles, a redução no tempo de conhecimento do domínio e do tempo de manutenção, decorrentes de projetos de melhor qualidade.

Os objetivos da pesquisa sobre reutilização são “reunir, organizar e facilitar o acesso a todo o conhecimento sobre o desenvolvimento de software em um determinado domínio” [ZIR 95]. Desta forma, um fator fundamental em qualquer abordagem baseada em reutilização é a existência de um conjunto de recursos que possam ser reutilizados. Com isto, pressupõe-se a existência de um repositório (biblioteca) para disponibilizar esses recursos. Dois dos principais problemas que surgem na abordagem de reutilização, são: (1) a escolha de um método para classificação e organização dos recursos disponíveis; (2) a existência de mecanismos que auxiliem o projetista a localizar os possíveis recursos reutilizáveis em seu projeto.

Uma forma simples de se organizar uma biblioteca de recursos reutilizáveis é por meio de domínios. Um domínio, no contexto da Engenharia de Software, engloba um conjunto de aplicações caracterizadas por um conjunto comum de dados e objetivos relacionados [PET 91]. A descrição do recurso reutilizável, adicionada às informações auxiliares para localizá-lo e manipulá-lo, constituem a infraestrutura básica para o processo de reutilização. Portanto, o processo de desenvolvimento de software com base em reutilização deve considerar algumas etapas adicionais, capazes de manter e consultar a infra-estrutura de recursos disponíveis. Segundo Zirbes [ZIR 95], o processo de reutilização constitui-se das seguintes etapas (Figura 1):

- construção da biblioteca de recursos: esta atividade inclui a identificação, escolha, classificação e armazenamento dos recursos reutilizáveis;
- reutilização dos recursos da biblioteca: esta atividade inclui a busca, seleção e adaptação dos recursos utilizáveis, de acordo com as necessidades do software sendo desenvolvido.

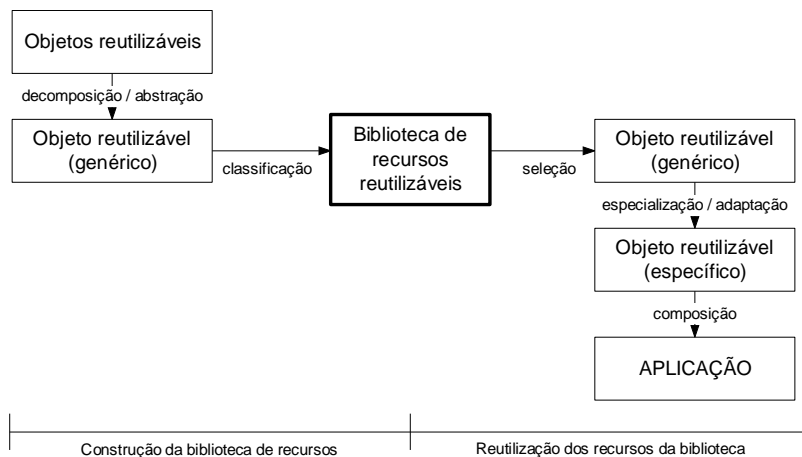


FIGURA 1 - Processo de reutilização de software

O processo de seleção de recursos reutilizáveis torna-se cada vez mais complexo, à medida em que a biblioteca aumenta de tamanho. Assim, é necessário o uso de métodos que auxiliem a localização de recursos reutilizáveis. Um deles, o método de seleção

por analogia, foi desenvolvido por Maiden em sua tese de doutorado [MAI 92]. O processo de reutilização usando analogia possui duas etapas. Na primeira etapa é identificado um conjunto de domínios abstratos representando famílias de sistemas com características afins.

Na segunda etapa é aplicado o raciocínio por analogia para reconhecer a semelhança entre o problema a ser definido e os domínios existentes, compreender esta semelhança e então, transferir o conhecimento sobre o domínio.

Além do método de seleção por analogia, uma série de outros métodos que auxiliam a localização de recursos reutilizáveis são descritos na literatura [GIR 90]. Alguns destes métodos utilizam técnicas de inteligência artificial para auxiliar o projetista a identificar e selecionar recursos reutilizáveis nos sistemas sendo desenvolvidos (Wood, 1988 Apud [PIM 98]). Ferramentas automatizadas são utilizadas para dar apoio ao uso desses métodos. Entre elas pode-se citar: o sistema Catalog [FRA 87] que permite classificar recursos reutilizáveis através do método de classificação de software baseado em vocabulário livre; o sistema ROSA - *Retrieval Of Software Artifacts* [GIR 96] propõe a classificação e recuperação automática baseada em vocabulário extraído de textos escritos em linguagem natural; a ferramenta CART - *Computer-Aided Reuse Tool* [LIA 93] utiliza um método de classificação baseado em facetas, isto é, grupos de classes elementares formando um esquema, para organizar uma biblioteca de software orientado a objetos.

Devido ao uso crescente do paradigma de orientação a objetos no desenvolvimento de software, novos instrumentos de reutilização têm sido propostos na literatura. As seções seguintes apresentam as tendências atuais em termos de abordagens de reutilização.

### 3. Padrões

A abordagem de padrões<sup>1</sup> é ainda uma idéia muito recente no campo da Ciência da Computação. Desta forma, não existe um consenso sobre uma definição para o termo "padrão" neste contexto. Segundo Gamma [GAM 94], o uso de padrões está diretamente relacionado com a idéia de reuso em Engenharia de Software e teve origem no trabalho do arquiteto Christopher Alexander que, em [ALE 77], definiu:

*"Cada padrão descreve um problema que ocorre diversas vezes em nosso*

*ambiente e, então, descreve a solução desse problema de tal forma que você a reutilize milhares de vezes sem ter que fazer a mesma coisa duas vezes"* (pag. x).

As idéias de Alexander deram origem ao desenvolvimento de pesquisas sobre padrões de projeto que, posteriormente, foram estendidas para diversos outros tipos de padrões.

A abordagem de padrões começou a ser conhecida após a publicação do livro *Design patterns: elements of reusable object-oriented software* [GAM 94], que apresenta um catálogo de soluções para problemas recorrentes em projeto de sistemas orientados a objetos. Neste livro, Erich Gamma, Richard Helm, Ralph Johnson e John Vlissides, que se tornaram conhecidos como a *gang* dos quatro ("*gang of four*" ou GoF), definem:

*"Um padrão apresenta a essência de uma solução para um problema recorrente, em um contexto específico"* (pag. 2-3).

Esta definição compreende as idéias fundamentais de um padrão. A expressão "uma solução para um problema" significa que cada padrão identifica um problema e apresenta uma solução para ele. O termo "essência de uma solução" significa que somente os elementos essenciais são descritos, deixando os aspectos específicos para serem detalhados pelo projetista, dado que aspectos específicos normalmente não são reutilizados. O termo "problema recorrente" significa que os padrões devem ser descritos para problemas que já ocorreram diversas vezes e irão ocorrer novamente. Por último, "em um contexto específico" significa que a solução completa é válida para um contexto particular.

Buschmann [BUS 96], classifica os padrões em três categorias: padrões de arquitetura; padrões de projeto; e idiomas (também conhecido por padrões de implementação [PRE 95]). As duas primeiras categorias incluem os padrões relacionados com a fase de projeto do sistema, mas são aplicadas em problemas de diferentes escalas. Os padrões de projeto são mais abstratos (e menores) do que os padrões de arquitetura, enquanto que idiomas são padrões escritos em alguma linguagem de programação.

---

<sup>1</sup> Do inglês Pattern.

Uma quarta categoria de padrões, introduzida por Fowler [FOW 96], inclui os padrões de análise. Estes padrões são usados para descrever soluções empregadas durante as fases de análise de requisitos e modelagem conceitual dos dados. Os padrões de análise refletem estruturas conceituais representativas do domínio da aplicação e não soluções computacionais.

Fowler [FOW 97] define um padrão de análise como:

*“Uma idéia que se provou útil em um contexto prático e que, provavelmente, será útil em outras situações similares” (pag xv).*

A maioria dos padrões de análise publicados até o momento foram projetados, principalmente, para solucionar problemas de aplicações comerciais [COA97, FOW97, HAY95]. No entanto, a idéia de padrões de análise pode ser usada para aumentar a qualidade e a produtividade no desenvolvimento de aplicações não-convencionais como, por exemplo, as aplicações de SIG. Por exemplo, os padrões de análise podem ser usados para documentar como os projetistas de um determinado domínio de aplicação modelam fenômenos geográficos e seus relacionamentos. Padrões de projeto podem ser usados para documentar como os dados sobre o relevo de uma região são representados através de um Modelo Numérico de Terreno (MNT). Por último, idiomas podem definir como um MNT deve ser implementado em cada software de SIG específico. As subseções seguintes descrevem, em maiores detalhes, as características dos padrões de análise e de projeto.

### 3.1 Padrões de Análise

Devido à abordagem de padrões ter sido introduzida na área de Engenharia de Software, por projetistas de sistemas orientados a objetos, a categoria de padrões mais conhecida atualmente é a dos padrões de projeto. No entanto, diversas pesquisas vêm sendo realizadas buscando-se padrões que auxiliem projetistas de banco de dados a reutilizar soluções para problemas de modelagem de dados [COA97, HAY95, FOW97].

Segundo Coad [COA 92], as técnicas tradicionais de reutilização em análise orientada a objetos baseiam-se nos mecanismos de herança. Assim, para reutilizar classes pré existen-

tes, o projetista necessita saber como localizar as classes adequadas a seu problema e derivar novas subclasses a partir dessas classes. Dessa forma, durante a modelagem de uma nova aplicação, cada classe funciona como um simples bloco reutilizável.

Um padrão de análise (ou padrão conceitual, na terminologia de Riehle [RIE 96]) descreve um conjunto de classes, possivelmente pertencentes a diferentes hierarquias de classes, e as associações existentes entre elas. Padrões de análise podem ser vistos, portanto, como uma forma de descrever subesquemas de projetos mais complexos, os quais ocorrem com frequência durante o processo de modelagem de muitas aplicações. O uso de padrões melhora, de forma significativa, o tempo de desenvolvimento de novas aplicações, uma vez que a reutilização ocorre através de subesquemas e não através de classes isoladas [HAY 96].

Geralmente, um padrão de análise apresenta a solução do problema de uma forma mais sugestiva do que prescritiva, fornecendo um modelo e a discussão do por que a solução é proposta desta forma, suas vantagens e desvantagens. Segundo Fowler [FOW 96], a contribuição realmente importante de um padrão não é o modelo fornecido como solução, mas sim, o raciocínio que está por trás desta solução.

Hay [HAY 95] descreve um conjunto de padrões de análise, que ele batizou de padrões de modelos de dados, que são soluções de modelagem conceitual de dados para aplicações comerciais. Para ilustrar um padrão de análise, a seguir é mostrado o padrão *Localizações Geográficas*, extraído de [HAY 95], com algumas adaptações. O padrão está descrito segundo a forma proposta por Coplien [COP 95], sendo que os diagramas foram convertidos para a notação UML [BOO 98].

#### 3.1.1 Exemplo de um Padrão de Análise

NOME:

Localizações Geográficas

PROBLEMA:

Descreve uma solução alternativa para o problema da modelagem de endereços, sendo útil tanto para pessoas, como para organizações.

CONTEXTO:

O padrão descreve uma alternativa de solução para o problema da modelagem de endere-

ços, servindo tanto para pessoas físicas como para pessoas jurídicas. É comum, em muitas situações da vida real, um mesmo endereço estar associado a mais de uma pessoa (ou organização) e vice-versa. Nestes casos, quando o endereço é modelado simplesmente na forma de atributos, uma série de problemas conhecidos podem ocorrer como, por exemplo, informação incompleta ou a inconsistência do banco de dados.

**FORÇAS:**

- Frequentemente é necessário ordenar/ selecionar endereços por bairros, cidades ou por outro tipo de divisão onde o endereço está localizado. Assim, cada local tem de estar em uma localização geográfica, o que significa que cada localização geográfica pode ser a localização de um ou mais locais.
- Para emissão de etiquetas de endereçamento, o endereço pode ser tratado

como um único objeto de informação. Se o propósito do endereço é apenas para emissão de etiquetas, a escolha da abordagem de atributos pode ser considerada.

**SOLUÇÃO:**

A Figura 2 mostra o diagrama de classes que compõe o padrão. A classe PESSOA é uma generalização das classes PESSOA FÍSICA e PESSOA JURÍDICA. A classe PESSOA possui uma associação LOCALIZA-SE (de cardinalidade m:n) com a classe ENDEREÇO. Esta associação pode incluir atributos como período e tipo de locação. A classe ENDEREÇO é que deve conter os atributos que caracterizam o endereço de uma instância de PESSOA. A classe LOCALIZAÇÃO GEOGRÁFICA é usada para criar uma estrutura hierárquica de localizações (ex.: País, Estado, Município, Bairro, etc), sendo que cada instância de ENDEREÇO deve estar associada a uma instância de LOCALIZAÇÃO GEOGRÁFICA.

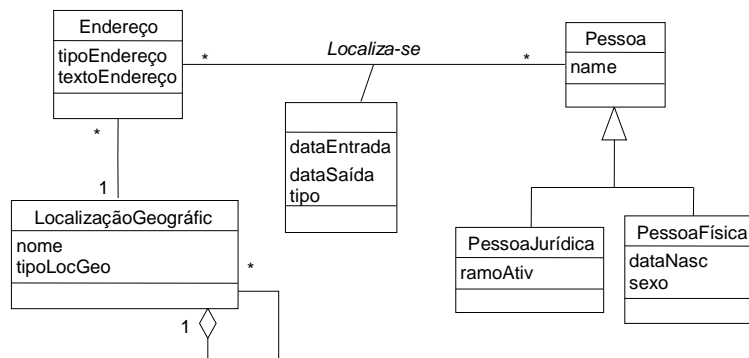


FIGURA 2 - Padrão de análise *LocalizaçõesGeográficas*

De acordo com Gerth [GER 96], a notação gráfica dos modelos de dados não são suficientes para a descrição dos padrões. A notação gráfica somente captura o resultado final da modelagem, representando-a na forma de um grafo de classes e associações. A descrição de um padrão, no entanto, deve também incluir a documentação das decisões tomadas, escolhas feitas, alternativas identificadas e restrições ao uso do padrão.

Segundo Hay [HAY 95], um padrão de análise pode e deve ser customizado para situações específicas. Por exemplo, o padrão *LocalizaçõesGeográficas* mostra uma situação (existem alternativas diferentes para a modelagem de informações sobre endereço) na qual pode-

se observar que não existe o modelo totalmente correto, mas sim, o mais adequado para um problema em um determinado contexto. Outro ponto a ser observado é que a definição de um padrão pode estar associada ou incluir outros padrões. Por exemplo, a generalização da classe PESSOA é uma solução definida em outro padrão (PARTY), descrito tanto em [HAY 95], como em [FOW 97]. Por sua vez, o padrão *Localizações Geográficas* poderá ser reutilizado no projeto de outros padrões mais abrangentes.

**3.2 Padrões de Projeto**

Da mesma forma, como não existe uma fronteira distinta, entre as atividades desenvolvidas durante as fases de análise e de proje-

to do processo de desenvolvimento de software, torna-se difícil definir, exatamente, as diferenças entre os padrões de análise e os padrões de projeto. O objetivo desta Seção não é apresentar em detalhes as características dos padrões de projeto, mas sim, o de evidenciar as principais diferenças entre um padrão de projeto e um padrão de análise. São elas:

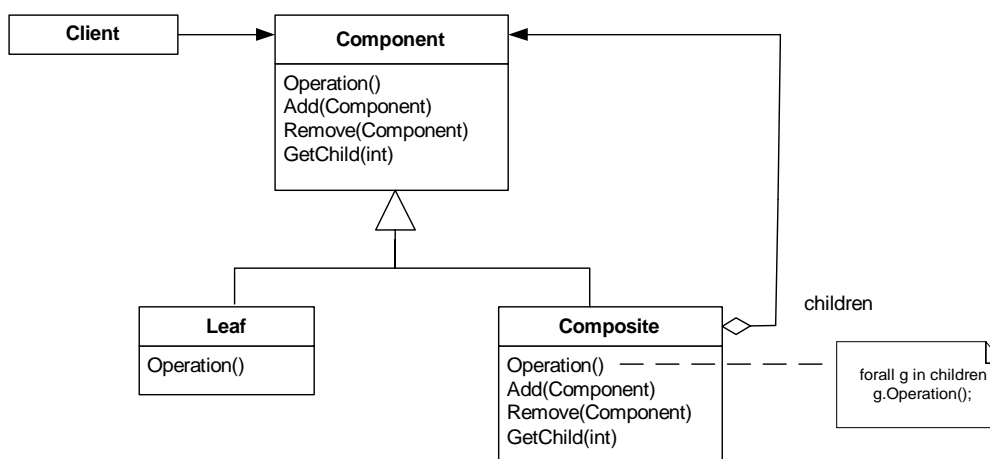
- Um padrão de análise é um padrão cuja forma é descrita através de termos e conceitos pertencentes ao domínio da aplicação. Já, a forma de um padrão de projeto é descrita através de construtores genéricos de projeto de software (ex.: objeto, agregação) [RIE 96].

- Padrões de análise não são escritos para servirem a um propósito geral. É importante encontrar o equilíbrio entre obter um padrão muito abstrato e um padrão muito especializado. Por exemplo, apresentar como solução para o problema de rede de transportes, que seus componentes podem ser modelados através de elementos do tipo conexão e ligação, em uma estrutura de rede é apresentar a solução de forma altamente abstrata, não auxiliando muito o projetista. No entanto, se o padrão de análise identifica os principais tipos de componentes de uma rede de transporte e como estes componentes podem (ou não) se relacionar, este padrão torna-se realmente útil para ser reutilizado. Um padrão de projeto descreve a

estrutura e as dinâmicas de seus componentes, esclarecendo suas responsabilidades. Assim, os padrões de projeto podem ser vistos como complementares aos padrões de análise.

- Ao contrário dos padrões de análise, os padrões de projeto existentes são apresentados de forma bem estruturada e organizados separadamente em catálogos. Por exemplo, os padrões de Gamma [GAM 94] apresentam quatro elementos essenciais: nome do padrão, o problema, a solução e as conseqüências. No entanto, outros itens de descrição podem ser adicionados como motivação, participantes, sinônimos e outros padrões relacionados. A classificação dos padrões de Gamma é feita segundo dois critérios ortogonais: (1) *Propósito*, incluindo se o padrão está relacionado com o processo de criação de objetos, com a composição de classes ou com as interações comportamentais dos objetos; (2) *Escopo*, indicando se o padrão se aplica a classes ou a objetos. Os padrões de análise, por sua vez, são apresentados de forma mais explicativa, apresentando soluções alternativas, vantagens e desvantagens de cada um.

A Figura 3 ilustra o diagrama de classes do padrão de projeto *Composite* [GAM 94] e a Figura 4 mostra um exemplo de uso desse padrão no diagrama de classes do *framework* conceitual **GeoFrame** [LIS 99].



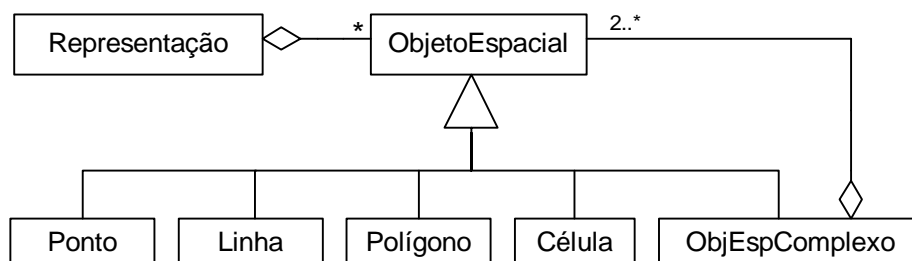


FIGURA 4 - Exemplo de uso do padrão de projeto *Composite*

## 4. Frameworks

O uso de *frameworks*<sup>2</sup> caracteriza uma técnica de reutilização de software orientado a objetos [JOH 97]. Segundo Pree [PRE 94], *framework* constitui um avanço real em termos de reutilização de software, uma vez que a reutilização ocorre não apenas a partir de pequenos blocos de programas, mas sim de todo um sistema (ou parte de um sistema), incluindo a reutilização do projeto.

Souza [SOU 98] define um *framework* como:

*“Um projeto genérico em um domínio que pode ser adaptado a aplicações específicas, servindo como um molde para a construção de aplicações”.*

Esta definição fornece uma visão bem mais abrangente sobre a potencialidade de um *framework* do que as definições apresentadas pelos autores mais ligados à programação orientada a objetos. Por exemplo, Wirfs-Brock e Johnson definem *framework* como sendo:

*“Um esqueleto de implementação de uma aplicação ou de um subsistema de aplicação, em um domínio de problema particular” [WIR 90].*

*“Um projeto reutilizável de um programa ou parte de um programa, expresso como um conjunto de classes” [JOH 92].*

Como um instrumento de reutilização, um *framework* não necessita estar implementado em uma linguagem de programação para fornecer a solução parcial a uma família de pro-

blemas. A vantagem de se ter o *framework* parcialmente implementado é que a solução final está bem mais próxima de ser atingida pelo desenvolvedor da nova aplicação. No entanto, o *framework* fica restrito a uma linguagem de programação específica ou necessita ser fornecido em diversas outras linguagens.

Uma das principais características de um *framework* é que o fluxo de controle entre o *framework* e sua aplicação cliente é invertido. Normalmente, quando uma biblioteca de classes é utilizada, o controle das chamadas às rotinas disponíveis é feito pelo programa que está utilizando a biblioteca. No caso de um *framework*, os métodos desenvolvidos para a aplicação é que são chamados pelos métodos do *framework*. Isto evidencia que, além de reutilizar todo um conjunto de classes, o conhecimento sobre o projeto do software também é reutilizado.

Padrões de projeto podem ser usados tanto na construção como na documentação de *frameworks* [JOH 92, MEU 97]. Um simples *framework* pode ser construído a partir de diversos padrões de projeto (também chamados de micro arquiteturas).

Outra forma de distinguir padrões de projeto e *frameworks* é que, enquanto um *framework* pode ser apresentado como um software inacabado, os padrões representam o conhecimento sobre como o software foi desenvolvido. Além disso, a complexidade dos problemas a serem resolvidos pelos *frameworks* são bem maiores do que os problemas resolvidos pelos

<sup>2</sup> O termo *framework* não será traduzido, neste texto, devido a seu uso freqüente pela comunidade de Engenharia de Software no Brasil. Uma possível tradução para *framework* é arcabouço.



padrões. Conseqüentemente, a construção de um *framework* é um processo bem mais complexo do que a definição de um padrão. Silva [SIL 96] descreve e compara uma série de métodos para construção de *frameworks*.

Um *framework* é composto por um conjunto de classes, muitas das quais abstratas, ou seja, não possuem instâncias. As classes de um *framework* podem ser classificadas como pontos fixos, isto é, não podem ser alteradas, ou como pontos adaptáveis, onde são feitas as mudanças e extensões necessárias à aplicação. Segundo Quadros [QUA 97], os pontos fixos determinam a arquitetura da aplicação, definindo sua estrutura geral, as responsabilidades e colaborações entre as classes e o fluxo de controle do *framework*, não podendo ser alterados pelo projetista da aplicação.

De acordo como as aplicações são derivadas a partir de um *framework*, este pode ser classificado em caixa-preta ou caixa-branca. Os pontos adaptáveis de um *framework* caixa-branca fornecem classes incompletas, as quais devem ser estendidas com as especificidades da aplicação. Um *framework* caixa-preta fornece um conjunto de classes alternativas para cada ponto adaptável. Neste caso o projetista deve escolher uma das alternativas disponíveis, mas não pode alterar suas propriedades. Um *framework* caixa-branca requer que o projetista tenha um maior conhecimento sobre o funcionamento do *framework*, enquanto que um *framework* caixa-preta requer do projetista um maior conhecimento sobre o domínio da aplicação.

As três abordagens de reutilização apresentadas até aqui (de especificação de requisitos, de padrões e *frameworks*) podem ser empregadas no desenvolvimento de aplicações em geral. A seção seguinte discute as características das abordagens de reutilização existentes atualmente na área de SIG.

## 5. Abordagens de Reutilização em SIG

Na área de Sistemas de Informação Geográfica (SIG) o conceito de reutilização tem sido aplicado, basicamente, no compartilhamento de dados. Observa-se, atualmente, um rápido

crescimento da demanda por dados geoespaciais digitais [WEB 98]. Por sua vez, a todo momento, muitos dados geoespaciais estão sendo produzidos, tanto por instituições governamentais como por instituições privadas. Dados geoespaciais, neste caso, incluem os mais variados tipos de coleções de dados georreferenciados como, por exemplo, imagens de satélite, fotografias aéreas, amostragem geológica, observações biológicas, dados de redes de infra-estrutura e dados censitários.

A coleta de dados primários, ou seja, dados usados na produção de outros dados, tanto para aplicações ambientais como para aplicações urbanas, é uma tarefa de custo muito elevado. Com o objetivo de reduzir estes custos, as instituições necessitam de mecanismos eficientes para facilitar a reutilização de dados já existentes, visto que muitos conjuntos de dados georreferenciados podem atender a diversas aplicações de SIG, implementadas em instituições diversas.

Além da reutilização de dados, algumas pesquisas sobre mecanismos de reutilização de modelagem de processos espaciais têm sido realizadas [MAR 97, MAR 98]. Segundo Marr [MAR 97], modelagem do processo espacial é o procedimento de delinear como os dados geográficos devem ser manipulados a fim de se obter resultados desejados.

Com relação a reutilização de projeto de banco de dados geográficos, porém, muito pouco foi feito até o momento [LIS 98]. Alguns padrões<sup>3</sup> de metadados prevêm a documentação sobre informações semânticas associadas aos dados, mas com o intuito de facilitar o entendimento dos dados a serem reutilizados.

Metadados são dados sobre dados [ISO 97]. A existência de conjuntos de dados geoespaciais é documentada através de conjuntos de metadados. O termo metadados geoespaciais é usado para especificar metadados referentes a conjuntos de dados georreferenciados [SHE 95].

A importância dos metadados geoespaciais reside no fato de possibilitarem ao usuário avaliar o grau com que os dados satisfazem às necessidades de uma determinada aplicação. A informação prestada pelos metadados provê uma substancial economia de recursos na pesquisa e busca de dados existentes bem como no monitoramento, controle e aquisição dos mesmos. Metadados expressam informações

---

<sup>3</sup> Do inglês *standard*.

sobre o conteúdo, a qualidade, a atualidade e outras características dos dados [WEB 98].

Para facilitar a busca e o acesso a conjuntos de dados existentes, alguns países estão especificando e institucionalizando padrões para a descrição uniforme dos metadados. Os padrões de metadados estabelecem categorias e elementos a serem contemplados para descrever adequadamente os dados. Entre os padrões de metadados mais conhecidos, pode-se citar: o padrão americano CSDGM [FED 97], o padrão canadense SAIF [BRI 95], o padrão CEN-TC287 da comunidade européia [CEN 96] e o padrão australiano ANZLIC [ANZ 97]. Estes padrões estabelecem uma terminologia comum aos fenômenos espaciais, classes de dados e seus modelos de informação, além de regras para produção de metadados.

A efetiva reutilização de dados geoespaciais digitais depende de diversos fatores, entre eles pode-se citar a interoperabilidade dos dados e a vontade de compartilhamento dos dados entre as instituições (de forma comercial ou não). Para que um maior número possível de usuários possa reutilizar dados geoespaciais existentes, é necessário que os catálogos de metadados estejam disponíveis para consulta. Alguns padrões já estão operacionais (ex.: CSDGM), dispendo inclusive de programas para cadastramento de metadados e esquemas para disponibilizar os metadados através da Internet [VIL 99].

Uma vez elaborado um conjunto de metadados, é necessário torná-lo acessível através de ferramentas de busca, o que vem suprir o primeiro objetivo dos metadados que é a localização dos dados. Conjuntos de dados geoespaciais são localizados através dos metadados. No entanto, estes dados não estão, necessariamente, disponíveis para serem obtidos automaticamente (ex.: *download* via ftp). Pode-se, por exemplo, localizar um conjunto de dados para uma determinada região geográfica de interesse. Através dos metadados de qualidade, avalia-se a adequabilidade desse conjunto para o uso planejado. Por último, pode ser necessário, caso o dado não esteja disponível, fazer o contato direto com o fornecedor do dado a fim de obtê-lo.

## 6. Conclusões

As abordagens de reutilização em sistemas de informação, embora apresentem diversas

vantagens quando realmente aplicadas no desenvolvimento de sistemas, na prática não têm sido muito utilizadas. Alguns fatores podem justificar o pouco uso desses instrumentos:

- ausência de uma metodologia de projeto de sistemas que induza o projetista a buscar por possíveis recursos reutilizáveis;
- baixa disponibilidade de recursos reutilizáveis;
- dificuldade de localização dos recursos reutilizáveis existentes. Inexistência de um mecanismo de apoio a busca dos recursos existentes;
- falta de cultura de trabalho em cooperação. É mais fácil para o projetista procurar recursos disponíveis para serem reutilizados do que disponibilizar seus próprios recursos para outros projetistas.

A abordagem de padrões (*pattern*) é ainda muito pouco conhecida. O projetista necessita, no entanto, estudar diversos padrões e exemplos de utilização para que possa identificar possíveis aplicações desses padrões. O treinamento dos projetistas sobre o uso de padrões pode ser compensado pelo ganho posterior em produtividade e qualidade dos sistemas desenvolvidos. Além disso, uma vez treinado, o projetista estará apto a aplicar padrões em diversos sistemas. Uma outra vantagem decorrente do uso de padrões é o da unificação de vocabulário entre os diversos membros de uma equipe de desenvolvimento.

Os problemas com o uso de *frameworks* são semelhantes aos encontrados no uso de padrões, porém, em escala diferente. Devido a sua maior abrangência, o uso de *frameworks* exige um conhecimento muito maior por parte do projetista, quando comparado ao uso de padrões.

O alto custo e a complexidade dos processos de aquisição de dados georreferenciados tornam a reutilização de dados geoespaciais um fator de grande importância. Toda aplicação de SIG utiliza-se de um conjunto de dados primários (ex.: hidrografia, vegetação ou divisas municipais). Desta forma, os conjuntos de dados digitais referentes a esses dados primários podem, na maioria das vezes, ser reutilizados por grupos de usuários trabalhando em aplicações diversas sobre a mesma região geográfica.

No Brasil, têm sido realizadas algumas iniciativas isoladas para disponibilizar dados

georreferenciados. Por exemplo, o projeto *GeoMinas* (<http://www.mg.gov.br/geominas>), onde pode-se obter dados primários sobre o Estado de Minas Gerais. Um segundo exemplo é o projeto do grupo de geoprocessamento da Companhia de Desenvolvimento do Planalto Central – CODEPLAN (<http://www.gdf.gov.br/codeplan/geopro>). Estas iniciativas apenas fornecem os conjuntos de dados que podem ser reutilizados, mas não fornecem nenhum mecanismo que auxilie a busca e localização dos dados disponíveis. Um sistema experimental de busca a metadados geoespaciais é disponibilizado pelo Serviço Geológico do Brasil (CPRM) em (<http://www.cprm.gov.br>) [ALM 99].

Estudos no sentido da adoção de um padrão internacional para elaboração de metadados geoespaciais, no Brasil, têm sido realizados [RIB 96, WEB 98, ALM 99]. No entanto, para adoção de um padrão existente, ou mesmo para a definição de um padrão nacional, é necessário o envolvimento de toda a comunidade de usuários e produtores de dados, o que deve ser feito sob a coordenação de algum órgão responsável.

## Referências Bibliográficas

- [ALE 77] ALEXANDER, C. et al. **A Pattern Language**. New York: Oxford University, 1977.
- [ALM 99] ALMEIDA, L. F. B. Padrão CPRMg para intercâmbio de dados gráficos digitais. In: GISBRASIL '99 CONGRESSO E FEIRA PARA USUÁRIOS DE GEOPROCESSAMENTO DA AMÉRICA LATINA, 5, 1999, Salvador. Anais... Salvador: Fator GIS, 1999.
- [ANZ 97] ANZLIC THE AUSTRALIA NEW ZEALAND LAND INFORMATION COUNCIL. **Core Metadata Elements for Land and Geographic Directories in Australia and New Zealand**. 1997. Disponível em <<http://www.anzlic.org.au/metaelem.htm>>
- [BOO 98] BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. **The Unified Modeling Language User Guide**. Addison-Wesley, 1998.
- [BUS 96] BUSCHMANN, F. et al. **Pattern-Oriented Software Architecture: A System of Patterns**. New York: John Wiley & Sons, 1996.
- [CEN 96] CEN EUROPEAN COMMITTEE FOR STANDARDIZATION. **Geographic Information – Data Description Language – Conceptual Schema Language**. Brussels: CEN, 1996. (Report CR 287005).
- [COA 97] COAD, P. **Object Models: Strategies, Patterns, and Applications**. 2.ed. New Jersey: Yourdon Press, 1997.
- [COL 96] COLEMAN, D. et al. **Object-Oriented Development: the fusion method**. New Jersey: Prentice-Hall, 1996.
- [COP 95] COPLIEN, J. O. A generative development-process pattern language. In: COPLIEN J. O. and SCHMIDT D. (Eds.) **Pattern Languages of Program Design**, Reading: Addison-Wesley, 1995.
- [EDE 94] EDELWEISS, N. **Sistemas de Informação de Escritórios: um Modelo para Especificações Temporais**. Porto Alegre: CPGCC da UFRGS, 1994. Tese de Doutorado.
- [EMB 87] EMBLEY, D. W.; WOODFIELD, S.N. A knowledge structure for reusing abstract data types. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 9., 1987, Monterey. **Proceedings...** New York: IEEE Computer Society Press, 1987.
- [FED 97] FEDERAL GEOGRAPHIC DATA COMMITTEE. **Content Standard for Digital Geospatial Metadata**. Washinton, D.C.: FGDC, 1997. Disponível em <<http://www.fgdc.gov>>
- [FOW 96] FOWLER, M. Analysis patterns and business objects. **ACM SIGPLAN Notices**, New York, v.31, n.10, 1996. Trabalho apresentado na OOPSLA, 11., 1996, San Jose, US.
- [FOW 97] FOWLER, M. **Analysis Patterns: Reusable Object Models**. Menlo Park, CA: Addison Wesley Longman, 1997.
- [FRA 87] FRAKES, W. B. et al. Information retrieval and software reuse. In: ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE

- ON RESEARCH AND DEVELOPMENT IN INFORMATION RETRIEVAL, 1989, Cambridge. **Proceedings...** New York: ACM, 1989.
- [GAM 94] GAMMA, E. et al. **Design Patterns: Elements of Reusable Object-Oriented Software**. Reading, MA: Addison Wesley, 1994.
- [GER 96] GERTH, Th.; SCHACHTSCHABEL, R.; SCHÖNEFELD, R. Using patterns in design and documentation of software. In: THE WHITE OO NIGHTS -WOON, 1996, St. Petersburg, Russia. **Proceedings...** St. Petersburg: [s.n.], 1997.
- [GIR 90] GIRARDI, M. R. **Uma ferramenta de apoio à reutilização de software no desenvolvimento orientado a objetos**. Porto Alegre: CPGCC da UFRGS, 1990. Dissertação de Mestrado.
- [GIR 96] GIRARDI, M. R. **Classification and retrieval of software and their description in natural language**. Genève: Université de Genève, 1996. PhD Thesis.
- [HAY 95] HAY, D. C. **Data Model Patterns: Conventions of Thought**. New York: Dorset House Publishing, 1995.
- [HAY 96] HAY, D. C. Using data model patterns for rapid application development. In: INTERNATIONAL ORACLE USER WEEK, 1996. **Proceedings...** [S.l.]: Oracle, 1996.
- [ISO 97] ISO/TC211 TERMINOLOGY PROJECT TEAM. **Collections of terms and definitions from ISO/TC211 geographic information/geomatics**. Disponível em <<http://www.statkart.no/isotc211/terms/terms211.html>>
- [JOH 92] JOHNSON, R. E. Documenting frameworks using patterns. In: OBJECT-ORIENTED PROGRAMMING SYSTEMS, LANGUAGES AND APPLICATIONS CONFERENCE - OOPSLA, 1992, Vancouver. **Proceedings...** Vancouver: ACM, 1992.
- [JOH 92] JOHNSON, R. E. Documenting frameworks using patterns. In: OBJECT-ORIENTED PROGRAMMING SYSTEMS, LANGUAGES AND APPLICATIONS CONFERENCE - OOPSLA, 1992, Vancouver. **Proceedings...** Vancouver: ACM, 1992.
- [JOH 97] JOHNSON, R. E. **Components, Frameworks, Patterns.**, [S.l.:s.n.], 1997. White paper
- [LIA 93] LIAO, H. C. et al. Software reuse based on a large object oriented library. **Software Engineering Notes**, New York, v.18, n.1, p.74-80, 1993.
- [LIS 98] LISBOA F, J.; IOCHPE, C.; BEARD, K. Applying Analysis Patterns in the GIS Domain. In: ANNUAL COLLOQUIUM OF THE SPATIAL INFORMATION RESEARCH CENTRE, 10., 1998, Dunedin, NZ. **Proceedings...** Dunedin: SIRC, University of Otago, 1998.
- [LIS 99] LISBOA F, J.; IOCHPE, C. Specifying analysis patterns for geographic databases on the basis of a conceptual framework. In: ACM SYMPOSIUM ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS, 7., 1999, Kansas City, USA. **Proceedings...** Kansas City: ACM, 1999.
- [MAI 92] MAIDEN, N. A. **Analogical specification reuse during requirements analysis**. London: Department of Business Computing, City University, School of Informatics, 1992. PhD Thesis.
- [MAR 97] MARR, A. J.; PASCOE, R. T.; BENWELL, G. Interoperable GIS and spatial process modelling. In: GEOCOMPUTATION, 2., 1997, Dunedin, New Zealand. **Proceedings...** Dunedin: University of Otago, 1997.
- [MAR 98] MARR, A. J. et al. Towards the development of a documentation structure for modelling spatial process. In: GEOCOMPUTATION, 3., 1998, Bristol-UK. **Proceedings...** Bristol: University of Bristol, 1998.
- [MEU 97] MEUSEL, M.; CZARNECKI, K.; KÖPF, W. A model for structuring user documentation of object-oriented framework using patterns and hypertext. In: EUROPEAN CONFERENCE ON OBJECT-ORIENTED PROGRAMMING – ECOOP, 1997, Finland. **Proceedings...** Finland: Springer-Verlag, 1997. (Lecture Notes in Computer Science, v.1241)
- [NEI 94] NEIGHBORS, J. M. An assessment of reuse technology after ten years. In: INTERNATIONAL CONFERENCE ON SOFTWARE REUSE, 3., 1994, Rio de Janeiro. **Proceedings...** Los Alamitos, California: IEEE Computer Society Press, 1994. 235p. p.6-13.
- [PET 91] PETERSON, A. S. Model-based software reuse terminology. **Software**

- Engineering Notes**, New York, v.16, n.2, p.45-51, 1991.
- [PIM 98] PIMENTA, A. **Especificação formal de uma ferramenta de reutilização de especificações de requisitos**. Porto Alegre: CPGCC da UFRGS, 1998. Dissertação de Mestrado.
- [PRE 94] PREE, W. Meta patterns - a means for capturing the essentials of reusable object-oriented design. In: EUROPEAN CONFERENCE ON OBJECT-ORIENTED PROGRAMMING - ECOOP, 1994. **Proceedings...** Berlim: Springer Verlag, 1994.
- [PRE 95] PREE, W. **Design Patterns for Object-Oriented Software Development**. Reading, MA: Addison-Wesley, 1995.
- [QUA 97] QUADROS, E. M.; RUBIRA, C. M. F. Construção de um framework para sistemas controladores de trens utilizando padrões de projeto e metapadrões. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 11., 1997, Fortaleza. **Anais...** Fortaleza: SBC, 1997.
- [RIB 96] RIBEIRO, G. P.; SOUZA, J. M.; FREITAS, A. L. B. Digital geospatial metadata: a brazilian case federal databases. In: IEEE METADATA CONFERENCE, 1., 1996, Maryland. **Proceedings...** Maryland: IEEE, 1996.
- [RIE 96] RIEHLE, D.; ZÜLLIGHOVEN, H. Understanding and using patterns in software development. **Theory and Practice of Object Systems**, [S.l.], v.2, n.1, 1996.
- [SHE 95] SHELLEY, E. P.; JOHNSON, B. D. Metadata: concepts and models. In: NATIONAL CONFERENCE ON THE MANAGEMENT OF GEOSCIENCE INFORMATION AND DATA, 3, 1995, Adelaide, Australia. **Proceedings...** Adelaide: [s.n.], 1995.
- [SIL 96] SILVA, R. P.; PRICE, R. T. Em direção a uma metodologia para o desenvolvimento de frameworks de aplicação orientados a objetos. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 10., 1996, São Carlos. **Anais...** São Carlos: SBC, 1996.
- [SOU 98] SOUZA, C. R. B. **Um framework para editores de diagramas cooperativos baseados em anotações**. Campinas: Unicamp, 1998. Dissertação de Mestrado.
- [VIL 99] VILLAVICENCIO, F.; IOCHPE, C.; LISBOA F., J. Um servidor de consultas à clearinghouse em português para distribuição de metadados geográficos. In: GISBRASIL '99 CONGRESSO E FEIRA PARA USUÁRIOS DE GEOPROCESSAMENTO DA AMÉRICA LATINA, 5, 1999, Salvador. **Anais...** Salvador: Fator GIS, 1999.
- [WEB 98] WEBER, E. J.; LISBOA F., J.; IOCHPE, C.; HASENACK, H. Geospatial metadata in Brazil: an experience in data documentation of an environmental GIS application. In: INT. CONFERENCE & EXHIBITION ON GEOGRAPHIC INFORMATION - GIS Planet, 1998, Lisbon, Portugal. **Proceedings...** Lisbon: USIG, 1998.
- [WIR 90] WIRFS-BROCK, R.; JOHNSON, R. E. Surveying current research in object-oriented design. **Communication of the ACM**, New York, v.33, n.9, 1990.
- [ZIR 95] ZIRBES, S. F. **A reutilização de modelos de requisitos de sistemas por analogia: experimentação e conclusões**. Porto Alegre: CPGCC da UFRGS, 1995. Tese de Doutorado.

