Particionamento e Agregação Estruturais na Análise Orientada a Objetos

Guillermo Bustos Reinoso
Carlos A. Heuser
UFRGS/Informática
Caixa Postal 15064
91501-970 Porto Alegre RS
gbustos|heuser@inf.ufrgs.br

Resumo

O artigo propõe uma forma de estruturar modelos orientados a objetos em diversos níveis de abstração, com o objetivo de controlar a complexidade. Com base nessa forma de estruturação, é proposta uma técnica para executar a análise orientada a objetos. A técnica combina uma estratégia *top-down*, que inclui a identificação de *clusters* e definição de interdependência entre *clusters*, e uma estratégia *bottom-up*, onde são definidos critérios de agregação e prioridades entre estes critérios. Ambas as estratégias são integradas em termos conceituais, de notação e de procedimento geral.

1 INTRODUÇÃO

A análise orientada a objetos (AOO) tem sido tema de intensa pesquisa nos últimos anos. A maioria das propostas que existem na literatura coincidem, em maior ou menor grau, quanto à necessidade de contar com mecanismos para estruturar o sistema em modelagem em diferentes níveis de abstração, diminuindo sua complexidade.

A obtenção de modelos estruturados pode dar-se segundo duas estratégias. Na estratégia *top-down*, o sistema é particionado em componentes mais simples. Na estratégia *bottom-up* as classes são agregadas para constituir o sistema. Devem existir critérios objetivos e práticos que permitam particionar sistemas ou agregar classes [BUS 93], ainda mais no caso de sistemas grandes e complexos onde a estruturação do modelo é imprescindível.

Ao considerar uma técnica de AOO surgem imediatamente algumas questões:

1) O particionamento inicial deve acontecer na perspectiva da organização da qual o sistema de informação faz parte (sistemas e subsistemas) ou na perspectiva do software orientado a objetos (classes e objetos)?

2) Estas duas visões são incompatíveis?

Além disso, surge a questão de que processo de análise deve ser seguido para obter o modelo. Pode-se considerar uma decomposição do sistema *antes* da identificação das classes (a estratégia *top-down*) ou uma composição *após* a identificação destas (a estratégia *bottom-up*). Em outras palavras, pode-se considerar o *particionamento* do sistema em subsistemas ou a *agregação* de classes em *clusters* de algum tipo.

As técnicas existentes na literatura não resolvem essas questões. A maioria enfatiza a agregação de classes (a estratégia *bottom-up*), como por exemplo os assuntos ("subjects") de Coad & Yourdon [COA 92], os "clusters" de Nerson [NER 92], os domínios de Bailin [BAI 89], os "ensembles" de De Champeaux (descritos em [WIR 90] e [FIC 92]), os módulos de Rumbaugh et

al. [RUM 91] e os objetos compostos de Booch [BOO 91]. Algumas propostas combinam o particionamento e a agregação, com uma clara preponderância desta última, como por exemplo os subsistemas de Jacobson et al. [JAC 92], os subsistemas de Shlaer & Mellor [SHL 92] e as visões de alto-nível ("high-level views") de Embley et al. [EMB 92]. Apenas as áreas de interesse ("areas of concern") de Reenskaug et al. (descrito em [WIR 90]) tratam exclusivamente do particionamento (a estratégia *bottom-up*).

O presente trabalho propõe uma forma de estruturar modelos orientados a objetos, bem como uma técnica para executar a AOO. A principal diferença entre as técnicas existentes e a proposta deste trabalho, é a de que no presente trabalho é proposta uma integração explícita de ambas as estratégias, em termos de conceitos, em termos de notação, bem como em termos de processo de modelagem, incluindo uma aplicação recursiva de ambas as estratégias e a definição de um *cluster*-sistema, que representa o sistema como um todo.

Inicialmente, o artigo apresenta um conjunto de conceitos e uma notação gráfica para o particionamento/agregação de modelos orientados a objetos (OO). A seguir são discutidas estratégias para obter modelos orientados a objetos estruturados da forma proposta.

O componente estático básico utilizado em ambas as estratégias é o *cluster*. Um *cluster* é definido como um conjunto de classes de objetos associadas conforme estruturas estáticas específicas. Entre estas estruturas estão as hierarquias de generalização/especialização, de composição, e associações de participação (*membership*).

2 ESTRUTURANDO MODELOS OO

O conceito central para a estruturação de modelos OO em diversos níveis de abstração usado na presente proposta é o de *cluster*, um agregado de classes de objetos. Um modelo OO assume uma estrutura hierarquizada, com um *cluster* conceitual que representa o sistema como um todo, sucessivos níveis de abstração, descendendo na hierarquia, que acrescentam progressivamente mais detalhes ao modelo em forma de *clusters* relacionados; e um último nível, que está representado por classes de objetos com atributos, correspondendo ao maior nível de detalhe possível do modelo. A figura 1 mostra a estrutura hipotética geral de um modelo OO segundo a presente proposta.

A notação usada baseia-se na notação OMT de Rumbaugh et al [RUM 91]. OMT prevê apenas a representação *plana*[1] de modelos OO. Entretanto, os conceitos aqui expostos não estão restritos a uma notação em particular e podem ser facilmente adaptados para outras notações.

Os conceitos para a construção de modelos OO estruturados são os seguintes:

1. Os *clusters* ou agregados de classes de objetos são representados por um retângulo de cantos arredondados com o intuito de distingui-los das classes comuns (representadas por retângulos). Dentro do símbolo representativo do *cluster* pode ser apresentada uma lista de nomes de seus componentes (nomes de classes e/ou de *subclusters*). A figura 1 mostra exemplos de *clusters*: *cluster*-sistema, *cluster* 1, *cluster* 2 e *cluster* n; e de classes: classes 1.1 à 1.n, e classes 2.1 à 2.n. Um *cluster* é identificado por um nome representativo. Em determinadas situações[2] o nome do *cluster* pode ser derivado do nome de um de seus componentes, considerado como mais relevante.
2. Uma *classe agregada de associações* é representada por um losango com um círculo externo. Uma classe de associações deste tipo agrega duas ou mais classes de associações comuns. A representação é mostrada entre o *cluster 2* e o *cluster n* na figura 1.

---

[1] Por representação plana entende-se aquela que não é hierarquizada através de mecanismos de particionamento e/ou agregação.

[2] A escolha de nomes de *clusters* é tratada mais detalhadamente abaixo, quando for discutida a estratégia *bottom-up*.

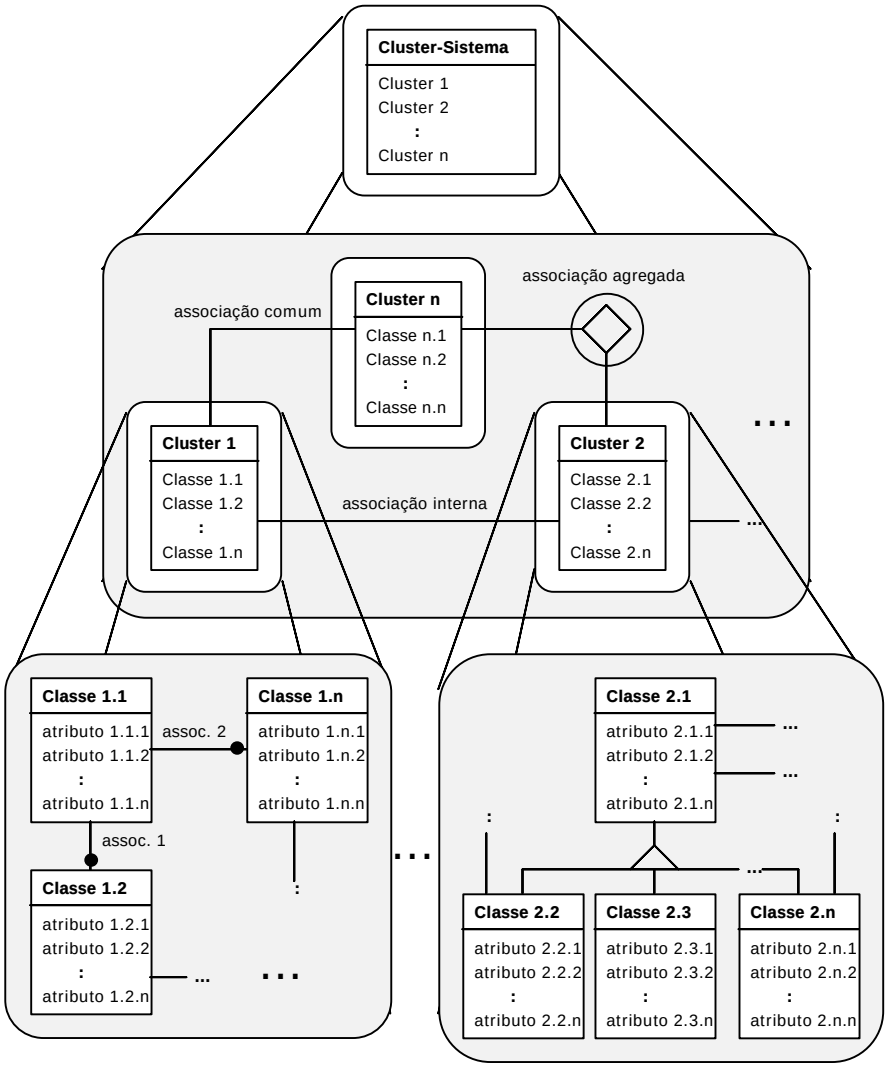

Figura 1 *- Estrutura hierárquica geral de um modelo de AOO*

3. As classes de associações entre *clusters* ou entre um *cluster* e uma classe de objetos podem ser representadas de duas formas:

   - Quando na associação participam as mesmas classes que dão o nome ao *cluster*, a classe de associações é representada na forma usual e é denominada *associação comum*.
   - Quando na associação participa pelo menos uma classe que não a que dá o nome ao *cluster*; ou quando o nome de um dos *clusters* não é derivado de nenhum dos componentes, estas classes de associações são denominadas *classes internas de associações* e sua representação é por linhas descontínuas. Por convenção, seu nome inclui o nome do *cluster* concatenado através de um ponto com o nome da classe participante.

   Exemplos de classes de associações são mostradas nas figuras 1 e 2. Na figura 2, que corresponde a uma porção simplificada do problema da IFIP [OLL 82], o *cluster* Avaliador é composto pelas classes Avaliador (que dá o nome ao *cluster*) e Relatório, cuja associação é mostrada na parte inferior. Os atributos do *cluster* Avaliador correspondem aos nomes das classes componentes. Na associação avalia, que é do tipo comum, participa a classe Avaliador, e portanto é representada na forma convencional no nível do *cluster*. Na associação Avaliador.Relatório relata, que é do tipo interno, participa a classe Relatório. Portanto sua representação é com linhas descontínuas e seu nome inclui também o nome do *cluster* e da classe associada.

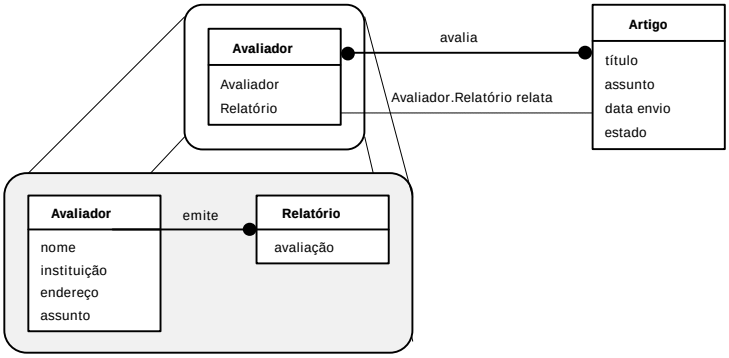

***Figura 2** - Exemplo específico da representação das associações entre clusters e componentes*

3 PROPOSTA DE PARTICIONAMENTO/AGREGAÇÃO

A técnica para obtenção de modelos OO estruturados consiste na complementação de duas estratégias: a *top-down* que conduz ao particionamento do sistema sob análise, e a *bottom-up* que resulta na agregação das classes identificadas na análise. Ambas as estratégias devem ser aplicadas quando for necessário, e cada uma delas pode modificar os resultados da outra com a finalidade de melhorar a construção, a semântica e a especificação do sistema de objetos.

3.1 ESTRATÉGIA *TOP-DOWN*

A estratégia *top-down* objetiva particionar o sistema nos seus principais componentes, os *clusters* . Há duas questões a considerar: a identificação inicial de *clusters* e a identificação da interdependência entre *clusters*.

3.1.1 Identificação Inicial de *Clusters*

Quando da identificação inicial de *clusters*, o problema centra-se em definir os principais componentes do sistema. Tais componentes poderão ser, e provavelmente o serão, modificados posteriormente quando da agregação das classes constituintes. Assim, pode afirmar-se que os *clusters* definidos nesta fase são apenas *clusters candidatos*, sujeitos a posteriores particionamentos e/ou agregações.

É preciso destacar *que todo particionamento do sistema deve ser realizado com o reconhecimento da existência de um conjunto difuso de componentes em cada divisão ou partição identificada*.

Em termos gerais, os conceitos que devem ser identificados e definidos como *clusters* caem nas seguintes categorias:

1 *Conceitos Hierárquicos*:

Estes conceitos correspondem às abstrações das hierarquias do tipo generalização/especialização, composição, agregação e participação. O conceito representa toda a hierarquia e não apenas à classe do topo da mesma. As hierarquias correspondem a:

- Um *conceito geral*, que eventualmente virá a ser refinado em uma hierarquia de generalização/especialização de classes.

  Por exemplo, no caso do sistema de conferências da IFIP, pode definir-se o *cluster* Pessoa sem considerar ainda, neste nível de abstração, um relacionamento hierárquico entre as classes Pessoa, Autor e Moderador. A figura 3 mostra o *cluster* Pessoa e a hierarquia de classes que ele abstrai. Nesta seção, os modelos dos exemplos apresentam os *clusters* com atributos (nomes dos componentes do *cluster*), apenas como uma referência. Na estratégia *top-down*, não é necessário indicar os atributos na definição dos *clusters*, ao contrário do que acontece na estratégia *bottom-up* onde são obrigatórios.

- Um *conceito composto* como um todo que virá a ser particionado eventualmente em uma hierarquia de composição.

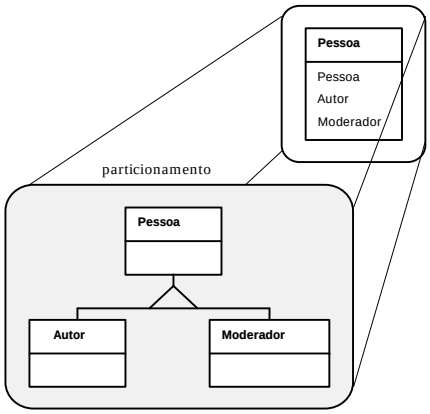

Figura 3 *- O conceito geral representado pelo cluster* Pessoa

Este conceito descreve o todo composto de partes sem fazer menção ainda aos componentes. Por exemplo, pode ser definido o *cluster* Sessão sem considerar ainda, que uma instância da classe Sessão poderia estar composta de instâncias da classe Artigo Aceito no caso do problema da IFIP. A figura 4 mostra este exemplo.

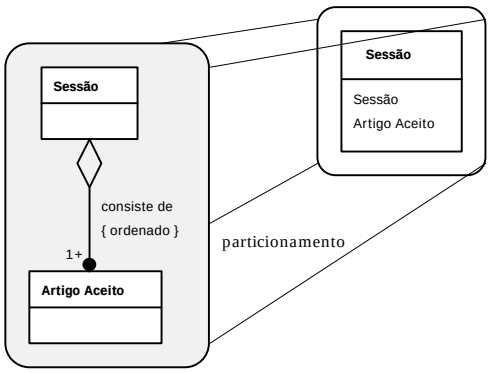

Figura 4 *- O conceito composto representado pelo cluster* Sessão

- Um *conceito associativo* que descreve uma associação de outros conceitos, que virá a ser refinado eventualmente em uma classe associativa entre outras classes ou ainda entre *clusters*.

  Por exemplo, pode ser definido um *cluster* Contrato que posteriormente será refinado em uma classe associativa Contrato que relaciona Empresa Contratante e Empresa Contratada. A figura 5 mostra este exemplo.

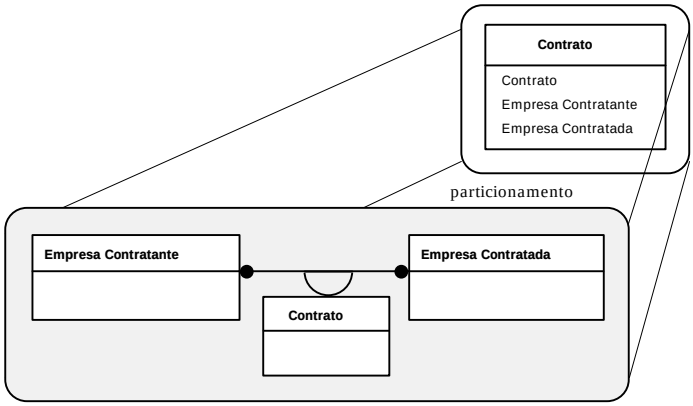

Figura 5 *- O conceito associativo representado pelo cluster* Contrato

- Um conceito de participação que pode ser refinado em um relacionamento de *membership*.

Este conceito descreve uma classe conjunto sem fazer referência explícita aos elementos participantes deste conjunto.

2 *Conceito de Dominância*:

Este conceito corresponde à situação na qual existe, por algum critério importante do ponto de vista do sistema modelado, uma preponderância de uma classe ou *cluster* sobre as classes ou *clusters* com os quais associa-se diretamente. Tal classe é denominada de *dominante*[3] em relação às restantes. Por exemplo, pode ser definido um *cluster* Estado, e este *cluster* vir a ser refinado em uma classe Estado relacionada às classes Município e Capital através de associações denominadas possui. Neste exemplo, as classes Município e Capital podem ser consideradas como de menor relevância que a classe Estado a qual estão associadas. A figura 6 mostra este exemplo.

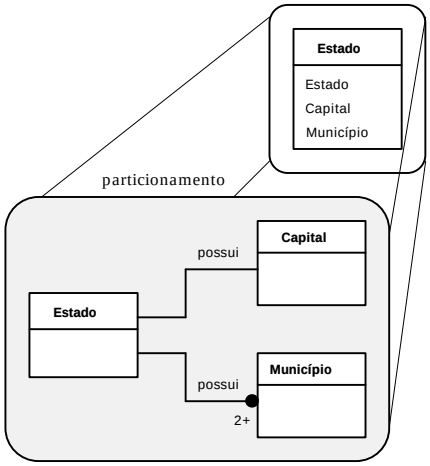

Figura 6 *- O conceito de dominância representado pelo* cluster *Estado*

3 *Conceito de Multi-relacionamento*:

Este conceito refere-se a um *cluster* que eventualmente será refinado como um relacionamento complexo entre várias classes ou *clusters*. Geralmente, os multi-relacionamentos expressam um conceito mais ou menos difuso e são representados por associações do tipo ternário ou ainda superiores. Por exemplo, pode ser definido o *cluster* Matrícula que será refinado nas classes Aluno, Disciplina e Semestre associadas por um relacionamento ternário denominado *matrícula*. A figura 7 mostra este exemplo.

4 *Conceito de Categoria*:

Este critério faz referência a um *cluster* cujos componentes potenciais caem dentro de uma mesma categoria. Estas categorias podem ser funcionais, lógicas, procedurais, temporais, etc. O relevante aqui é a categoria e não os potenciais componentes pertencentes à categoria. Por exemplo, pode ser definido o *cluster* Correspondência que será particionado nas classes Carta Intenção e Resposta Convite, para o caso do problema da IFIP. É importante destacar que o conceito não implica em nenhum tipo de associação entre as classes pertencentes à categoria.

3.1.2 A Interdependência Entre os *Clusters*

Após a identificação inicial dos *clusters* candidatos, podem ser examinadas as interdependências existentes entre estes *clusters*. Algumas regras para a identificação da interdependência entre *clusters* podem ser consideradas:

---

[3] Na seção de estratégia *bottom-up* é fornecida uma definição mais rigorosa deste conceito.

- Considerar que cada *cluster encapsula* ou esconde elementos que são irrelevantes para os *clusters* restantes. Isto é, os *subclusters* ou classes componentes não devem ser conhecidos para outros *clusters*. As ideias e conceitos expressados neste primeiro esboço do sistema devem possuir níveis de abstração semelhantes, e dar uma boa visão dos principais assuntos que trata o sistema, sem acrescentar excessivos detalhes.

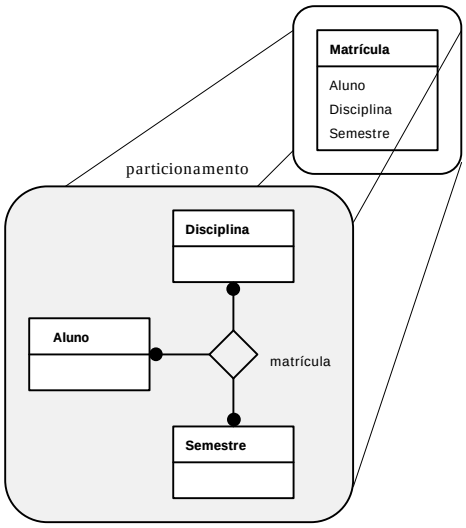

***Figura 7** - O conceito de multi-relacionamento representado pelo cluster Matrícula*

- Tratar os *clusters candidatos* como possuindo um comportamento paralelo, isto é, que todos os *clusters* iniciais possam mudar seus estados, ativando ações e transições em forma naturalmente concorrente. Isto garante uma adequada divisão das ideias e conceitos do sistema em grupos coesos e relativamente independentes. Quanto maior a concorrência, maior será a independência e maior também a coesão dos *clusters*.
- O ideal é que os *clusters* iniciais não possuam classes em comum. Porém, ao iniciar o particionamento, pode ser difícil de evitar que a mesma classe seja definida em mais de um *cluster*[4]. Ao identificar esta situação os *clusters* iniciais devem ser redefinidos, particionando novamente ou agregando algumas das classes já identificadas. As classes em comum são também um bom mecanismo para identificar os relacionamentos entre os *clusters*. A existência de classes comuns em dois *clusters* indica a existência de um relacionamento *implícito* entre estes *clusters*, ao contrário das associações que representam relacionamentos explícitos. Deve-se procurar um compromisso adequado entre a múltipla especificação e a minimização dos relacionamentos. A figura 8 mostra um exemplo de uma classe Aluno que é utilizada como relacionamento implícito entre os *clusters* Matrícula e Bolsa. Já na figura 9, a mesma classe Aluno é mostrada em apenas um dos *clusters* o que implica em um relacionamento explícito (ou *associação interna*) entre os *clusters* Matrícula e Bolsa.
- No caso de estimar-se a existência de mais de uma associação em paralelo entre os *clusters* e/ou classes, pode definir-se uma associação agregada. Por exemplo, pode ser definida uma associação agregada estipulação entre os *clusters* Contrato e Serviço, que pode vir a ser refinada nas associações paralelas inclui e exclui como é mostrado na figura 10.

3.2 ESTRATÉGIA *BOTTOM-UP*

A estratégia *bottom-up* permite complementar a estratégia *top-down* descrita na seção anterior. O propósito desta estratégia é definir *clusters* através de mecanismos de agregação de classes e associações já definidas.

---

[4] Isto pode ocorrer particularmente com as classes ditas dominantes.

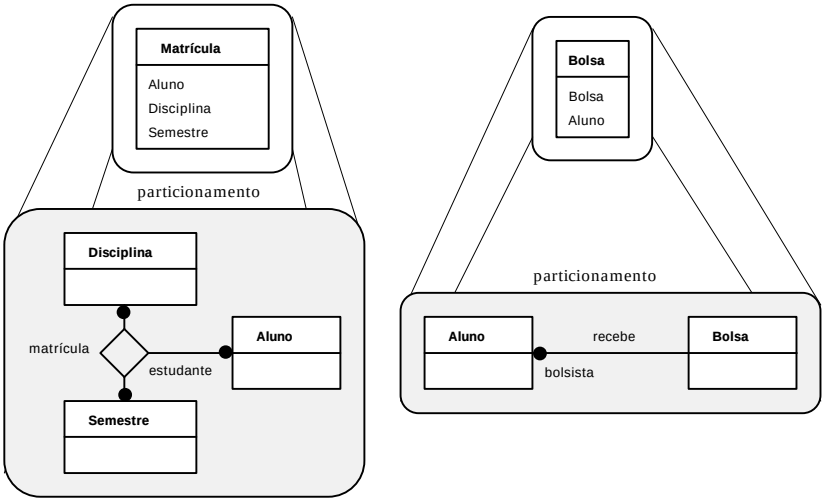

Figura 8 *- Representação repetida de uma classe em comum com relacionamento implícito entre os clusters*

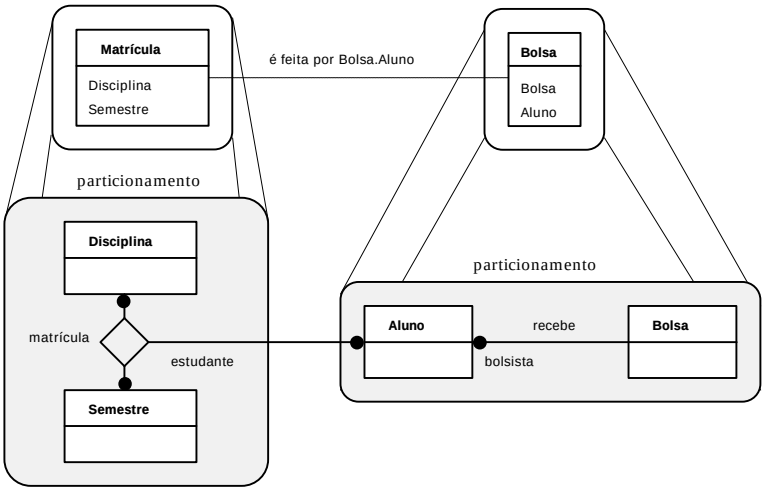

Figura 9 *- Representação de uma classe em comum com relacionamento explícito entre os clusters*

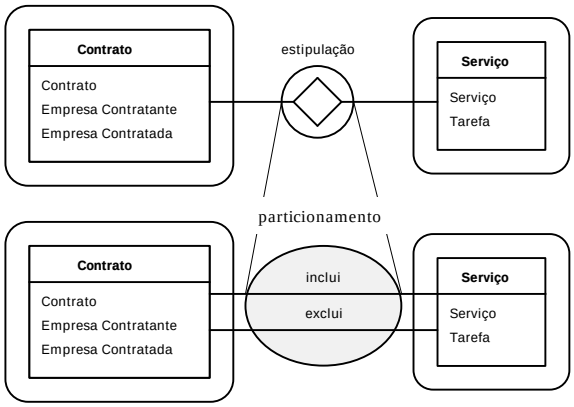

Figura 10 *- Definição e particionamento de uma associação agregada em associações paralelas*

Para agregar classes de objetos existem diversos critérios, isto é, a partir da identificação de certos conceitos, estruturas e associações, é possível definir abstrações (*clusters*) que escondam alguns dos detalhes definidos no nível inferior. Os componentes a ser agregados também podem ser, por sua vez, *clusters*, sendo portanto *subclusters* em relação ao agregado definido. Assim, quando nesta seção se faz referência a um *componente* entende-se que é uma classe ou *subcluster* indistintamente.

3.2.1 Os Critérios de Agregação

Os critérios de agregação que permitem definir *clusters* são indicados a seguir. Como convenção, os atributos dos *clusters* correspondem sempre aos nomes dos componentes agregados.

1. *Agregação por Dominância*:

   O critério baseia-se no conceito de *dominância* de um componente sobre outros a ele associados diretamente. Um componente é dito *dominante* se as cardinalidades das associações são do tipo um-para-muitos (1:n), isto é, associações de uma instância do componente dominante com muitas instâncias dos componentes a ele associados. A dominância é tão mais forte quanto mais associações deste tipo o componente possua. Os relacionamentos muitos-para-muitos (n:n) também podem ser tratados, para os efeitos da dominância, como associações 1:muitos sempre que exista um componente claramente dominante em relação aos restantes.

   Como convenção, o *cluster* assume o nome do componente dominante. A figura 11 mostra a configuração típica de esta situação, com o caso de uma classe dominante Produto e as classes Fornecedor, Depósito e Linha de Produção.

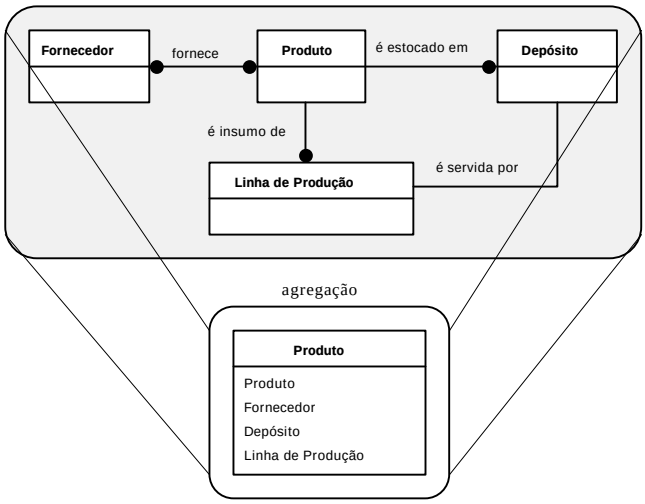

Figura 11 *- Exemplo de agregação pelo critério de dominância*

2. *Agregação por Abstração*:

   O critério baseia-se nas diferentes tipos de abstrações encontradas nos modelos OO. As abstrações são generalização/especialização, composição, agregação (classes associativa) e participação (*membership*). Como convenção, os *clusters* assumem os nomes dos componentes dos topos das respectivas hierarquias. Estas abstrações já foram descritas na seção da estratégia *top-down* e são as seguintes:

   - *generalização/especialização*: como exemplo ver a figura 3, concebida no sentido inverso, isto é, como uma agregação da hierarquia das classes Pessoa, Autor e Moderador para o *cluster* Pessoa;
   - *composição*: como exemplo ver a figura 4, concebida no sentido da agregação, isto é, da hierarquia das classes Sessão e Artigo Aceito para o *cluster* Sessão;
   - *agregação*: como exemplo ver a figura 5, concebida no sentido da agregação da hierarquia das classes Contrato, Empresa Contratante e Empresa Contratada; e
   - *participação*.

3. *Agregação por Restrição de Integridade*:

   Corresponde ao caso em que uma restrição de integridade do modelo relaciona implicitamente os componentes participantes. É aconselhável que estes componentes permaneçam juntos em um *cluster* para que a restrição possa ser facilmente reconhecida. A figura 12 mostra um exemplo das classes Pós-Graduando, Bolsa e Salário associadas e sob a restrição *"Pós-Graduando recebe Bolsa ou percebe Salário mas não ambos"*. Este conjunto de classes, por estarem submetidas à mesma restrição constituem o *cluster* Manutenção Estudante.

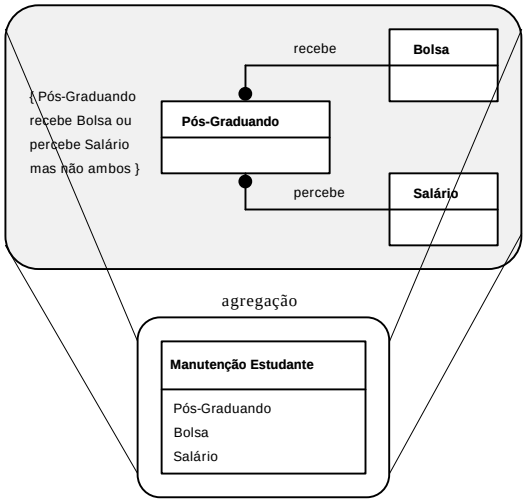

Figura 12 *- Exemplo de agregação pelo critério de restrição*

4. *Agregação por Associações Simples*:

   Este critério baseia-se na existência de associações simples (binárias) de diversas cardinalidade (1:1, 1:n, n:n) entre componentes. A idéia é agregar para manter classes associadas dentro do mesmo cluster. Obviamente, este critério permite múltiplas agregações, conflitantes entre si. A figura 13 mostra, para um exemplo simples, alguns possíveis *clusters*: Plano de Disciplinas, Manutenção Aluno, Turma e Conclusão construídos a partir de diversas associações simples.

5. *Agregação por Associação Complexa*:

   Este critério baseia-se nas associações complexas (de grau maior que dois) entre componentes. Como convenção, os *clusters* agregados preservam o nome do multi-relacionamento. Como exemplo, ver a figura 7, concebida no sentido da agregação, isto é, do relacionamento complexo matrícula entre as classes Aluno, Disciplina e Semestre para o *cluster* Matrícula.

3.2.2 A Prioridades entre os Critérios

Podem existir conflitos quando da agregação, pois mais de um critério pode se aplicar a um conjunto de classes e associações. Por exemplo, pode existir um componente dominante que também faz parte de uma hierarquia de generalização/especialização. Deve agregar-se por dominância ou por abstração de generalização?

Para resolver estes e outros conflitos é definida uma classificação dos critérios de agregação em níveis de *coesão* dos *clusters*. A coesão de um *cluster* é uma medida intuitiva da importância de manter as classes reunidas em um *cluster*. Os critérios de agregação são listados abaixo por ordem decrescente de coesão. O primeiro critério é o de mais alta coesão, o último o de mais baixa.

1 *Agregação por dominância*: É a mais alta coesão devido à dependência (funcional ou de existência) dos componentes dominados em relação ao dominante.

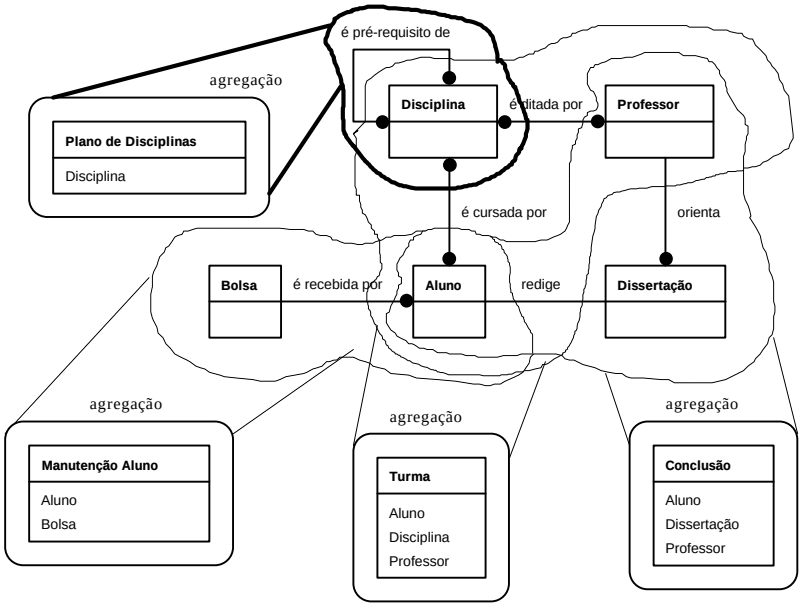

Figura 13 *- Exemplos de agregações pelo critério de associação simples*

2 *Agregação por abstração*: É a segunda maior coesão devido ao relacionamento hierárquico estabelecido entre os componentes na generalização/especialização, composição, agregação e participação.

3 *Agregação por restrição*: Existe uma coesão moderada entre os componentes relacionados através de relacionamentos restritos.

4 *Agregação por associação simples*: Existe um sub-espectro dentro deste nível, que vai da maior coesão nas associações binárias 1:1, para as associações binárias 1:n, e finalmente, com a menor coesão, as associações binárias n:n.

5 *Agregação por associação complexa*: É a segunda menor coesão devido a ausência de dominância de qualquer dos componentes que participam no relacionamento ternário ou superior.

6 *Agregação sem associação*: A coesão é apenas decorrente do fato de que os componentes participam de um mesmo *cluster* definido inicialmente (*cluster* candidato) através de algum mecanismo de particionamento da estratégia *top-down* (por exemplo, segundo o conceito de categoria).

3.3 A COMBINAÇÃO DAS ESTRATÉGIAS

Na prática o construtor do modelo OO dificilmente seguirá estritamente uma das estratégias, *top-down* e *bottom-up*. Assim, abaixo é sugerida uma seqüência de passos para executar a modelagem que combina as duas estratégias:

1 Identificar e definir inicialmente os *cluster* candidatos: Centrar-se nos conceitos hierárquicos (geral, composto, associativo, e de participação), de dominância, de multi-relacionamento e de categoria para identificar os *clusters* componentes. Evitar, no possível, que os *clusters* venham a ter classes em comum.

2 Refinar os *clusters* anteriores com a revisão de suas interdependências: Levar em conta que *clusters* devem ocultar detalhes irrelevantes no nível de abstração considerado, possuindo comportamento concorrente, incluindo e excluindo classes em comum (eventualmente definindo novos *clusters* a partir de hierarquias de papéis), e definindo associações agregadas, onde estimadas.

3 Aplicar os passos 1 e 2 em forma recursiva montando uma hierarquia de *clusters* até chegar às classes atômicas.4 Trabalhando sobre as classes identificadas nos passos anteriores, localizar os componentes dominantes em cada *cluster* já definido (possuem maior coesão).

5 Agregar os componentes formando *clusters* (que podem não concordar com as definições anteriores). Usar os demais critérios para agregar componentes com menor coesão. Para melhorar a legibilidade do modelo baseado em ambas estratégias (particionamento/agregação) aplicar as seguintes regras (em ordem de precedência):

   1 Os componentes do *cluster* potencial devem pertencer a um mesmo *cluster* previamente particionado

   2 Se existe conflito entre potenciais *clusters* de um mesmo nível de coesão, devem manter-se estes componentes sem agregar dentro dos clusters previamente definidos, e tentar um possível particionamento.

6 Aplicar os passos 4 e 5 em forma recursiva até obter um modelo completo. Se necessário retornar aos passos 1 e 2 de forma recursiva, já que as estratégias são complementares.

7 Verificar a consistência do modelo após sucessivas aplicações dos passos anteriores. As associações entre os componentes devem verificar-se em cada nível definido.

8 Finalmente, definir o *cluster* de mais alto nível, isto é, o *cluster* que representa o sistema como um todo. Neste nível de abstração não interessa a coesão, pode até ser mínima, já que este *cluster-sistema* é puramente conceitual.

4 FUNDAMENTAÇÃO DA PROPOSTA

Conforme mencionado, a notação gráfica para modelos planos (sem o conceito de *cluster*) usada no trabalho baseia-se em OMT [RUM 91] Quanto às convenções próprias do particionamento/agregação, adaptou-se a definida por Batini et al. [BAT 92] para o projeto conceitual de banco de dados.

A estratégia *top-down* apresentada utiliza os conceitos da técnica de *clustering* de Teorey et al. [TEO 89], e a idéia de aplica-la à AOO de Embley et al. [EMB 92]. O conceito de subsistema como primeiro particionamento do sistema é apresentado por Shlaer & Mellor [SHL 92] e Jacobson et al. [JAC 92]. Quanto à interdependência entre os *clusters*, o conceito de comportamento paralelo dos *clusters* é tomado dos *ensembles* de De Champeaux, descritos em [WIR 90] e [FIC 92]. A idéia de associações agregadas representando associações comuns paralelas é de Embley et al. [EMB 92].

A estratégia *bottom-up* é a mais desenvolvida na literatura. Os critérios de agregação, prioridades e procedimento são adaptados da proposta de *clustering* para o modelo entidade-relacionamento de Teorey et al. [TEO 89], e, em menor grau, da técnica *OSA* de Embley et al. [EMB 92]. O conceito de *dominância* é adaptado da definição inicial de Feldman & Miller [FEL 86], e as definições de Teorey et al. [TEO 89], Huffman & Zoeller [HUF 90] e também de Embley et al. [EMB 92].

5 CONCLUSÕES E COMENTÁRIOS

Em termos gerais, pode afirmar-se que a proposta do presente trabalho é mais completa que as similares encontradas na literatura. Baseia-se em heurísticas de modelagem e portanto não é formal. Pode ser entendida como uma extensão das técnicas de *clustering* da modelagem semântica de dados, com uma integração de mecanismos de particionamento para a AOO.

A base da proposta está em combinar as estratégias *top-down* e *bottom-up* para a AOO, sem dar maior ênfase a uma delas, porém fornecendo uma ordem básica. O analista fica livre para definir e redefinir *clusters* de acordo a conceitos e interdependências para particionar o sistema, e critérios e prioridades para agregar as classes do sistema. Isto lhe confere uma maior flexibilidade na construção dos modelos OO.

Infelizmente, pela própria natureza *plana* da colaboração dos objetos, é difícil desenvolver uma estratégia *pura* de particionamento dos sistemas sob modelagem. Geralmente, os conceitos para particionar são definições simétricas e complementares dos conceitos para agregar, o que dificulta às vezes sua aplicação. Isto é particularmente verdadeiro para a definição inicial dos *clusters* candidatos. De acordo com Shlaer & Mellor [SHL 92], dado que o particionamento do sistema é realizado considerando os componentes do domínio do problema, tem-se um problema de *bootstrapping*, pois deve-se criar abstrações de componentes ainda não bem identificados. A maneira de superar este conflito proposta neste trabalho está na combinação recursiva e sucessiva de ambas as estratégias.

Finalmente, este trabalho deve ser considerado como apenas uma primeira proposta no sentido da integração dos mecanismos de particionamento/agregação. Falta ainda um maior desenvolvimento dos conceitos, notações e procedimentos, e uma adequada validação na aplicação a problemas reais. Um aspecto que não é abordado nos critérios considerados é o efeito das perspectivas dinâmica e funcional [RUM91] sobre os critérios para executar o particionamento/agregação. No presente trabalho, tratou-se unicamente o aspecto estático.

REFERÊNCIAS BIBLIOGRÁFICAS

[BAI 89] BAILIN, Sidney. An Object-Oriented Requirements Specification Method. *Communications of the ACM*, New York, v.32, n.5, p.608-623, May. 1989.

[BAT 92] BATINI, Carlo; CERI, Stefano; NAVATHE, Shamkant. *Conceptual Database Design: An Entity-Relationship Approach*. Redwood City: Benjamin/Cummings, 1992.

[BOO 86] BOOCH, Grady. Object-Oriented Development. *IEEE Transactions on Software Engineering*, New York, v.12, n.2, p.211-221, Feb. 1986.

[BOO 91] BOOCH, Grady. *Object Oriented Design with Applications*. Redwood City: Benjamin/Cummings, 1991.

[BUS 93] BUSTOS, Guillermo. *Estudo Comparativo de Técnicas de Análise Orientada a Objetos*. Trabalho Individual I nº 317, Porto Alegre: CPGCC-UFRGS, Mar. 1993.

[BUS 94] BUSTOS, Guillermo. *Aplicação Comparativa de Técnicas de Análise Orientada a Objetos*. Trabalho Individual II a ser publicado, Porto Alegre: CPGCC-UFRGS, 1994.

[COA 92] COAD, Peter; YOURDON, Edward. *Análise Baseada em Objetos*. Rio de Janeiro: Campus, 1992.

[DEC 92] DE CHAMPEAUX, Dennis; LEA, Doug; FAURE, Penelope. The Process of Object-Oriented Design. *ACM SIGPLAN Notices*, New York, v.27, n.10, p.45-62, Oct. 1992. Trabalho apresentado na Annual Conference on Object-Oriented Programming, Systems, Languages, and Applications - OOPSLA '92, 7., 1992, Vancouver.

[EMB 92] EMBLEY, David; KURTZ, Barry; WOODFIELD, Scott. *Object-Oriented System Analysis: A Model-Driven Approach*. Englewood Cliffs: Prentice-Hall, 1992.

[FEL 86] FELDMAN, P.; MILLER, D. Entity Model Clustering: Structuring a Data Model by Abstraction. *The Computer Journal*, Cambridge, v.29, n.4, p.348-360, Aug. 1986.

[FIC 92] FICHMAN, Robert; KEMERER, Chris. Object-Oriented and Conventional Analysis and Design Methodologies: Comparison and Critique. *IEEE Computer*, Los Alamitos, v.25, n.10, p.22-39, Oct. 1992.

[HUF 90] HUFFMAN, Scott; ZOELLER, Randal. A Rule-Based System Tool for Automated ER Model Clustering. In: LOCHOVSKY, F. H. (ed.) *Entity-Relationship Approach to Database Design and Querying*, Amsterdam: North-Holland, 1990, p.221-236.

[JAC 92] JACOBSON, Ivar; CHRISTERSON, Magnus; JONSSON, Patrik; ÖVERGAARD, Gunnar. *Object-Oriented Software Engineering: A Use Case Driven Approach*. Wokingham: Addison-Wesley, 1992.

[JAL 89] JALOTE, Pankaj. Functional Refinement and Nested Objects for Object-Oriented Design. *IEEE Transactions on Software Engineering*, New York, v.15, n.3, p.264-270, Mar. 1989.

[NER 92] NERSON, Jean-Marc. Applying Object-Oriented Analysis and Design. *Communications of the ACM*, New York, v.35, n.9, p.63-74, Sep. 1992.

[OLL 82] OLLE, T. William. Comparative Review of Information Systems Design Methodologies: Problem Definition. In: IFIP WG 8.1 WORKING CONFERENCE ON COMPARATIVE REVIEW OF INFORMATION SYSTEMS DESIGN METHODOLOGIES, 1982, Noordwijkerhout. *Proceedings...* Amsterdam: North-Holland, 1982. p.8-9.

[RUM 91] RUMBAUGH, James; BLAHA, Michael; PREMERLANI, William; EDDY, Frederick; LORENSEN, William. *Object-Oriented Modeling and Design*. Englewood Cliffs: Prentice-Hall, 1991.

[SHL 92] SHLAER, Sally; MELLOR, Stephen. *Object Life Cycles: Modeling the World in States*. Englewood Cliffs: Prentice-Hall (Yourdon Press), 1992.

[TEO 89] TEOREY, Toby; WEI, Guangping; BOLTON, Deborah; KOENIG, John. ER Model Clustering as an Aid for User Communication and Documentation in Database Design. *Communications of the ACM*, New York, v.32, n.8, p.975-987, Aug. 1989.

[WIR 90] WIRFS-BROCK, Rebecca; JOHNSON, Ralph. Surveying Current Research in Object-Oriented Design. *Communications of the ACM*, New York, v.33, n.9, p.104-124, Sep. 1990.