

Universidade Federal do Rio Grande do Sul
Instituto de Informática
Curso de Pós-Graduação em Ciência da Computação

**Uma Proposta de
Particionamento/Agregação Estrutural
para a Análise Orientada a Objetos**

por

Guillermo Bustos Reinoso

T.I. n° 412 CPGCC-UFRGS Abril 94

Trabalho Individual III

Prof. Carlos A. Heuser
Orientador

Porto Alegre, abril de 1994.

SUMÁRIO

Lista de Figuras.....	4
Lista de Tabelas.....	6
Resumo	7
Abstract.....	8
Introdução	9
1. Análise Orientada a Objetos	11
1.1. Propostas de Particionamento/Aggregação na AOO	14
1.1.1. As Áreas de Interesse	15
1.1.2. Os Assuntos	15
1.1.3. Os Clusters	16
1.1.4. Os Domínios de Bailin.....	16
1.1.5. Os Domínios e os Subsistemas de Shlaer & Mellor	17
1.1.6. Os Ensembles.....	18
1.1.7. Os Módulos.....	19
1.1.8. Os Objetos Compostos.....	19
1.1.9. Os Subsistemas de Jacobson et al.	20
1.1.10. As Visões de Alto-Nível	20
1.2. A Comparação das Propostas	21
2. Proposta de Particionamento/Aggregação	23
2.1. A Notação e Convenções.....	24
2.2. Particionamento/Aggregação na Modelagem Estática	27
2.2.1. Estratégia Top-Down	27
2.2.1.1. Os Domínios	27
2.2.1.2. A Identificação Inicial dos Clusters	30
2.2.1.3. A Interdependência Entre os Clusters	38
2.2.1.4. O Procedimento Geral.....	41
2.2.2. Estratégia Bottom-Up	42

2.2.2.1. Os Critérios de Agregação	43
2.2.2.2. A Prioridades entre os Critérios	46
2.2.2.3. O Procedimento Geral	48
2.2.3. A Combinação das Estratégias.....	48
2.2.4. As Bases da Proposta	51
2.3. O Particionamento/Agregação na Modelagem Dinâmica.....	52
2.4. A Avaliação da Proposta.....	53
3. Conclusões e Comentários.....	55
Anexo 1 - Notação Utilizada Conforme a Metodologia OMT	58
Anexo 2 - Definição do Problema da IFIP	62
Anexo 3 - Particionamento/Agregação para o Problema da IFIP	66
Referências Bibliográficas.....	72

LISTA DE FIGURAS

Figura 1.1 - Estrutura de níveis dos domínios de Bailin [BAI 89]	17
Figura 2.1 - (a) Representação de uma classe; (b) Representação de um cluster	24
Figura 2.2 - (a) Representação de uma associação ternária; (b) Representação de uma associação agregada	25
Figura 2.3 - Exemplo da representação das associações entre clusters e componentes	26
Figura 2.4 - Exemplo de domínios no espaço do problema (extraído de Shlaer & Mellor [SHL 92])	29
Figura 2.5 - Exemplo específico para o conceito geral representado pelo cluster Pessoa	31
Figura 2.6 - Exemplo específico para o conceito composto representado pelo cluster Sessão	32
Figura 2.7 - Exemplo específico para o conceito associativo representado pelo cluster Contrato	33
Figura 2.8 - Exemplo específico para o conceito de participação representado pelo cluster Grupo de Trabalho	34
Figura 2.9 - Exemplo específico para o conceito de dominância representado pelo cluster Estado	35
Figura 2.10 - Exemplo específico para o conceito de multi-relacionamento representado pelo cluster Matrícula	36
Figura 2.11 - Exemplo específico para o conceito de categoria representado pelo cluster Correspondência	37
Figura 2.12 - Exemplo de uma representação repetida de uma classe em comum com relacionamento implícito entre os clusters	39
Figura 2.13 - Exemplo de uma única representação de uma classe em comum com relacionamento explícito entre os clusters	39
Figura 2.14 - Exemplo da definição do cluster Aluno a partir dos papéis de uma classe (vide figuras 2.12 e 2.13)	40
Figura 2.15 - Exemplo específico de definição e particionamento de uma associação agregada em associações paralelas	41
Figura 2.16 - Exemplo específico de agregação pelo critério de dominância	43
Figura 2.17 - Exemplo específico de agregação pelo critério de restrição	45

Figura 2.18 - Exemplos específicos de agregações pelo critério de associação simples	46
Figura 2.19 - Estrutura hierárquica final do modelo de AOO	50

LISTA DE TABELAS

Tabela 1.1 - Comparação das propostas de particionamento/agregação	22
Tabela 2.1 - Comparação dos conceitos da proposta.....	54

RESUMO

O trabalho consiste em uma proposta específica para particionar e agregar estruturalmente as classes na análise orientada a objetos.

Inicialmente, são revistos os principais questionamentos levantados em relação a análise orientada a objetos, entre os quais encontra-se as técnicas de particionamento/*clustering*. Diversas propostas encontradas na literatura são brevemente descritas e comparadas entre si.

A proposta é apresentada com uma notação e convenções derivadas da notação *OMT* de Rumbaugh et al., contudo outras notações também podem ser usadas. A técnica é dividida em uma estratégia *top-down*, incluindo definição de domínios e identificação e definição de interdependência entre *clusters*. Na estratégia *bottom-up* são definidos critérios de agregação e prioridades entre estes critérios. Ambas as estratégias são integradas em termos conceituais, de notação e de procedimento geral.

Finalmente, a técnica é comparada brevemente com as outras propostas e aplicada ao caso do problema da IFIP.

Conclui-se que a proposta é mais completa que as suas similares, porém ainda é informal e fortemente baseada em heurísticas. Deve ser aperfeiçoada e complementada com o aspecto dinâmico da modelagem orientada a objeto.

Palavras-chave: *análise orientada a objetos, modelagem orientada a objetos, análise top-down, clustering de classes.*

ABSTRACT

This work presents an specific technique of structural partitioning and class aggregation (clustering) in object-oriented analysis.

First, the main problems of object-oriented analysis are revised, among them the partitioning/aggregation techniques. Several proposals found in bibliography are described and compared each other.

Rumbaugh's *OMT* notation is used with some modifications in this technique, but it's possible to adapt others. The technique is divided in a top-down strategy, including domains definitions, clusters identification and interdependence. And in a bottom-up strategy, where criteria and priorities among these criteria are defined. Both strategies are integrated in conceptual terms, notation and general procedure.

Finally, the technique is compared to others and applied to the IFIP problem.

Concluding, this proposal is more complete than the similar ones, however it's still informal and strongly heuristic based. It must be improved and must include the dynamic aspect of object-oriented modeling.

Keywords: *object-oriented analysis, object-oriented modeling, top-down analysis, class clustering.*

INTRODUÇÃO

A análise orientada a objeto tem sido um tema de forte pesquisa nos últimos anos. Diversas propostas existem na literatura que abordam esta fase do desenvolvimento de sistemas orientados a objetos desde diversas perspectivas. A maioria delas, em menor o maior grau, coincidem em torno à necessidade de contar com mecanismos para particionar os sistemas sob modelagem, aplicando uma estratégia *top-down*, ou agregar as classes de objetos constituintes do sistema, aplicando uma estratégia *bottom-up*.

Existe um trabalho mais consolidado para esta última estratégia a partir das técnicas de *clustering* dos modelos semânticos de dados, principalmente para modelos entidade-relacionamento estendidos. Quanto à estratégia *top-down* na análise orientada a objetos, quase não é tratada.

Considerando a escassa atenção dada à estratégia *top-down* nas propostas de análise orientada a objetos, e a relativa maturidade das técnicas de *clustering*, este trabalho centra seus esforços em desenvolver ambas estratégias e integrá-las. Para este efeito são estabelecidos os seguintes objetivos:

- Estudar as propostas de particionamento e/ou agregação nas técnicas de análise orientada a objetos encontradas na literatura.
- Adaptar as técnicas de *clustering* desenvolvidas para a modelagem semântica de dados para a modelagem estrutural orientada a objetos.
- Propor critérios, procedimentos e notações para particionar sistemas sob modelagem estrutural orientada a objetos.
- Integrar as estratégias *top-down* e *bottom-up* em uma proposta de particionamento/agregação estrutural para a análise orientada a objetos.
- Esboçar a base sobre a qual seja possível definir futuramente mecanismos de particionamento/agregação na modelagem dinâmica orientada a objetos integrada à modelagem estrutural.

- Avaliar a proposta em termos comparativos e através da aplicação a um caso específico.

O conteúdo do presente trabalho encontra-se dividido nos capítulos descritos brevemente a seguir.

O *capítulo 1* apresenta os principais questionamentos da análise orientada a objetos e sobre a conveniência de utilizá-la. Entre os questionamentos encontra-se os mecanismos de particionamento/agregação. A seguir, são revistas e sucintamente comparadas as propostas destes mecanismos encontradas na literatura.

No *capítulo 2*, é descrita em detalhe a proposta de particionamento/agregação estrutural deste trabalho em termos de notações, estratégia *top-down*, estratégia *bottom-up* e integração de ambas as estratégias. São descritas também as idéias básicas para realizar o particionamento/agregação na modelagem dinâmica. No fim do capítulo, a proposta é avaliada comparativamente.

Finalmente, o *capítulo 3* apresenta as conclusões e alguns comentários relativos ao presente trabalho.

1. ANÁLISE ORIENTADA A OBJETOS

Entender-se-á por Análise Orientada a Objetos (AOO) *a construção de modelos do espaço do problema, identificando e especificando um conjunto de objetos semânticos que interagem e comportam-se conforme os requisitos estabelecidos para o sistema de objetos* [MON 92]. Os objetos ditos *semânticos* nada mais são do que aqueles objetos cujo significado é determinado pelo domínio específico da aplicação. Desta definição desprende-se que a AOO é centrada na modelagem conceitual dos domínios dos problemas, e não considera portanto detalhes da implementação das soluções para tais problemas.

É claro que sendo uma análise *orientada a objetos* deverá observar este paradigma. A orientação a objetos sustenta-se, entre outros, nos conceitos de identidade, classificação, polimorfismo, herança e encapsulamento. O encapsulamento propõe que sejam separados nos objetos os aspectos externos dos aspectos internos. Os aspectos externos são conhecidos por outros objetos, em compensação, os aspectos internos relativos à implementação, são de conhecimento exclusivo do próprio objeto. O encapsulamento não é restrito à orientação a objetos, porém é nela que ganha mais força porque os objetos combinam internamente a estrutura de dados (atributos) e os processos que a manipulam (métodos). A questão que pode ser levantada é a seguinte: faz sentido encapsular ao nível da modelagem conceitual? Dado que a maior vantagem do encapsulamento revela-se nas fases de implementação: quais benefícios poderiam se esperar do encapsulamento na fase de análise? Considerando a diversidade de propostas de AOO possíveis de encontrar na literatura¹, estas questões ainda não possuem adequadas respostas. Concluindo então, *não pode afirmar-se ainda, que a AOO é vantajosa porque utiliza o encapsulamento.*

Quanto à essência do paradigma de orientação a objetos, isto é, à visão da realidade composta por objetos é freqüentemente indicada como *mais natural*² do que visão funcional. A pergunta que surge é se existe realmente uma visão mais natural. A resposta, segundo Constantine [CON 89], é que não existem *classes de objetos* no universo físico real. São construções que existem apenas na mente dos observadores, e conclui que os *objetos* não são mais reais ou naturais do que as *funções*. Ao anterior,

¹ Vide [BUS 93] para uma classificação e comparação geral.

² Por exemplo, Coad & Yourdon [COA 92] argumentam neste sentido para recomendar a AOO face a análise funcional.

Loy [LOY 90] acrescenta que esta variação da controvérsia *forma vs. função* não está resolvida e provavelmente nunca será resolvida. Assim, não pode afirmar-se que uma ou outra visão é a forma *mais natural* em que as pessoas vêm a realidade do universo. Isto tem uma consequência para AOO: *o fato de que AOO veja os domínios como compostos de objetos não é uma vantagem comparativa em relação a outros tipos de análises fora deste paradigma*³.

Em outra perspectiva, a reusabilidade é indicada como uma das principais vantagens da utilização do desenvolvimento orientado a objetos. Porém, de maneira paradoxal, pode ser constatado que esta *não existe na AOO*, é deficiente no nível de projeto e razoável no nível de implementação. As principais causas por esta omissão podem ser encontradas no fato de que a maioria das propostas de técnicas de AOO utilizam, em maior ou menor grau, procedimentos, diagramas e notações criados fora do paradigma e com uma forte orientação apenas para a modelagem de sistemas e, portanto, não consideram a reusabilidade como um dos seus objetivos.

Na AOO, a reusabilidade pode tomar duas formas [FIC 92]: 1) como o reuso de especificações prévias de análise e projeto orientados a objetos e devidamente armazenadas em bibliotecas conceituais para tais efeitos; e 2) como o reuso de abstrações de componentes de programas já implementados fora do paradigma da orientação a objetos (engenharia reversa). Como um primeiro passo no sentido da reusabilidade a nível da AOO, parece mais adequada a 2ª forma. A 1ª forma só poderá ser atingida após a análise e desenvolvimento de muitos sistemas sob este paradigma. Assim, hoje em dia *não pode afirmar-se que a AOO é vantajosa porque apresenta reusabilidade*.

A principal vantagem que pode ser indicada no sentido de utilizar a AOO na modelagem é que ela fornece uma visão uniforme dos sistemas baseada nos objetos, que pode e deveria ser mantida nas seguintes fases de projeto e implementação, isto é, todo o ciclo de vida do sistema deve ser orientado a objetos. O *gap semântico*⁴ se vê diminuído, assim por exemplo, os objetos identificados no domínio do problema são mapeados diretamente nos objetos especificados no domínio da solução durante o projeto. O mesmo acontece com estes objetos ao serem mapeados nos objetos codificados na

³ Isto é particularmente aplicável por exemplo à análise estruturada [YOU 90], que baseia sua visão em funções.

⁴ Como *gap semântico* se define a *distância* entre o domínio do problema no mundo real e o domínio da solução na implementação computacional [BUS 93].

implementação. Isto é conhecido como *continuidade estrutural (structural continuity)* [DEC 92] nos sucessivos mapeamentos entre a AOO, o projeto orientado a objetos e sua implementação também orientada a objetos. Pode afirmar-se então que, *a AOO é vantajosa se tanto o projeto como a implementação são também conduzidos sob o mesmo paradigma.*

Outro problema diz relação com o particionamento do sistema sob modelagem e/ou a agregação das classes componentes do sistema modelado. Conforme exposto em [BUS 93], devem existir critérios objetivos e práticos que permitem particionar sistemas ou agregar classes, mais ainda em se tratando de sistemas grandes e complexos onde uma decomposição *top-down* é altamente desejável. As primeiras questões que surgem são: 1) o particionamento inicial deve ser 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)?, e 2) são incompatíveis estas visões? Neste sentido, pode questionar-se se deve ser realizada uma decomposição *antes* da identificação das classes (estratégia *top-down*) ou uma composição *após* a identificação destas (estratégia *bottom-up*). Ou em outras palavras, se o sistema é particionado em subsistemas ou as classes são agregadas em *clusters* de algum tipo. Na literatura abundam propostas da estratégia *bottom-up*: módulos [RUM 91], assuntos (*subjects*) [COA 92], objetos compostos (*composite objects*) [BOO 91] e *clusters* [NER 92] são alguns exemplos. Concluindo, *as técnicas de AOO ainda não possuem critérios objetivos, padronizados e consistentes para o particionamento da complexidade dos sistemas analisados e/ou para a agregação das classes e objetos identificados.*

Em síntese, os questionamentos à AOO expostos acima são os seguintes:

- encapsulamento na modelagem conceitual
- visão de domínios compostos por objetos vs. compostos por funções
- reusabilidade dos modelos de requisitos
- continuidade estrutural no ciclo de vida orientado a objetos
- critérios de particionamento/agregação

É exatamente neste último questionamento que o presente trabalho se insere, com uma proposta específica para o particionamento e a agregação na AOO, definindo critérios mais objetivos e consistentes que os disponíveis na literatura. Na seguinte seção são abordadas algumas das técnicas alternativas propostas.

1.1. PROPOSTAS DE PARTICIONAMENTO/AGREGAÇÃO NA AOO

Na literatura existem várias propostas que visam abordar o problema do particionamento da complexidade nos sistemas sob a AOO ou a agregação das classes na especificação obtida. Geralmente estas propostas fazem parte de técnicas específicas para esta fase, que incluem procedimentos, diagramas e notações para tal efeito.

Entre as propostas existentes, serão brevemente discutidas as seguintes:

1. Áreas de interesse (*areas of concern*) da técnica OORASS (*Object-Oriented Role Analysis, Synthesis and Structuring*) de Reenskaug et al., descrito em [WIR 90].
2. Assuntos (*subjects*) da técnica OOA (*Object-Oriented Analysis*) de Coad & Yourdon [COA 92].
3. *Clusters* da técnica OOA&D (*Object-Oriented Analysis and Design*) de Nerson [NER 92].
4. Domínios (*domains*) da técnica *Object-Oriented Requirements Specification Method* de Bailin [BAI 89].
5. Domínios (*domains*) e subsistemas (*subsystems*) da técnica de AOO de Shlaer & Mellor [SHL 92].
6. *Ensembles* propostos por De Champeaux e descritos em [WIR 90] e [FIC 92].
7. Módulos (*modules*) da técnica de análise da metodologia OMT (*Object Modeling Technique*) de Rumbaugh et al. [RUM 91].
8. Objetos compostos (*composite objects*) da proposta de Booch [BOO 91].
9. Subsistemas (*subsystems*) da técnica de análise da metodologia OOSE (*Object-Oriented Software Engineering*) de Jacobson et al. [JAC 92].
10. Visões de alto-nível (*high-level views*) da técnica OSA (*Object-Oriented System Analysis*) de Embley et al. [EMB 92].

1.1.1. As Áreas de Interesse

As áreas de interesse ou *areas of concern* é uma estratégia proposta por Reenskaug et al. (descrito em [WIR 90]) para o particionamento da complexidade dos sistemas abordados com a técnica OORASS (*Object-Oriented Role Analysis, Synthesis and Structuring*).

A técnica OORASS consiste em três operações críticas: análise, síntese e estruturação. Estas operações baseiam-se nas propriedades de encapsulamento, herança e enlace dinâmico (*dynamic binding*) da orientação a objetos. A análise descreve os sub-problemas através do encapsulamento do comportamento nos objetos em um modelo de objetos denominado Modelo de Papéis (*Role Model*).

O Modelo de Papéis separa o domínio do problema em *áreas de interesse* sobrepostas. Cada área é modelada como uma estrutura de objetos que interagem. Por sua vez, cada objeto é definido por abstração como um papel conforme o seu propósito na estrutura do Modelo de Papéis. Posteriormente, os Modelos de Papéis individuais são integrados graças à composição de objetos que desempenham diferentes papéis nos diversos modelos.

1.1.2. Os Assuntos

Os assuntos ou *subjects* fazem parte da técnica de Coad & Yourdon *Object-Oriented Analysis - OOA* [COA 92]. Os assuntos têm um objetivo específico dentro da proposta destes autores: proporcionar uma visão geral de um modelo extenso da OOA.

Segundo Coad & Yourdon os assuntos possuem duas motivações: mecanismos de orientação para a leitura da especificação da OOA e organização de pacotes de trabalhos em projetos extensos.

Os assuntos são identificados a partir das hierarquias de generalização/especialização (estruturas Gen-Espec na terminologia dos autores) e de composição (estruturas Todo-Parte) dentro da modelagem estrutural. Como candidatos a assuntos podem ser indicadas as classes dos topos das diferentes hierarquias, depois estes

assuntos candidatos são refinados procurando interdependências mínimas (considerando as conexões das ocorrências) e interações mínimas (considerando as conexões de mensagens). Os assuntos podem ter interseções e uns podem conter outros.

1.1.3. Os *Clusters*

O conceito de *cluster* faz parte da técnica de análise e projeto orientados a objetos de Nerson [NER 92]. Os *clusters* correspondem a agrupamentos de classes que representam os principais focos de interesse do sistema.

Na fase de análise, os *clusters* permitem agrupar as classes de acordo a critérios de *proximidade* tais como a funcionalidade dos subsistemas, os níveis de abstração ou os locais específicos dos usuários finais.

Segundo Nerson, é comum começar definindo um único *cluster* geral e, após ter realizado alguns agrupamentos de classes, continuar com novos *clusters*.

1.1.4. Os Domínios de Bailin

Os domínios ou *domains* correspondem a uma proposta que faz parte da técnica *Object-Oriented Requirements Specification Method* de Bailin [BAI 89]. Eles são concebidos para organizar as *entidades* (objetos) em diagramas entidade-relacionamento separados. Cada diagrama deve representar então um domínio específico. O conjunto de todos os domínios e seus correspondentes diagramas fornece uma visão completa das entidades do sistema.

O autor indica que podem existir diferentes *níveis* de domínios de entidades, assim por exemplo a figura 1.1 representa uma hierarquia de tipos de domínios [BAI 89], em que cada domínio baseia-se no domínio de mais baixo nível.

Na fase inicial de identificação de entidades, interessa o domínio de aplicação e possivelmente alguns domínios tecnológicos.

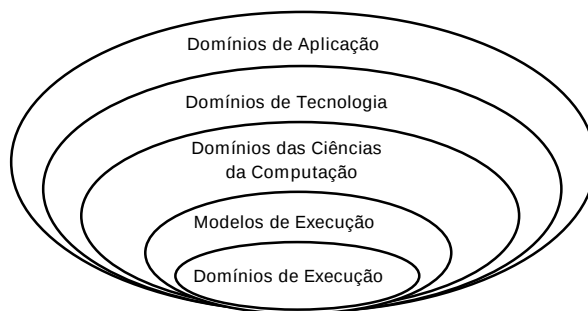


Figura 1.1 - Estrutura de níveis dos domínios de Bailin [BAI 89]

1.1.5. Os Domínios e os Subsistemas de Shlaer & Mellor

Os conceitos de domínio ou *domain* e de subsistema ou *subsystem* fazem parte da técnica de AOO de Shlaer & Mellor [SHL 92].

Os domínios são definidos como um *mundo* separado, e *habitado* por um conjunto de objetos que comportam-se de acordo com regras e políticas próprias do domínio. Os tipos de domínios possíveis, segundo os autores, são: domínios de aplicação, domínios de serviço, domínios de arquitetura e domínios de implementação. Nos modelos de multidomínios, os objetos são definidos em exatamente um domínio e não precisam da existência de objetos nos outros domínios. Cada um dos domínios, como um todo, pode interagir com outro(s) domínio(s) através de *pontes* (*bridges*) que definem relacionamentos cliente-servidor.

Tanto os domínios como as pontes são definidos previamente à identificação dos objetos ou modelagem de informação na terminologia de Shlaer & Mellor [SHL 90].

Já os subsistemas constituem-se em um mecanismo de particionamento de grandes domínios compostos pelos *clusters*⁵ de objetos existentes nos modelos. Os relacionamentos inter-subsistemas (*spanning relationships*), que devem ser minimizados, descrevem as associações entre objetos pertencentes a diferentes subsistemas.

⁵ Um *cluster*, segundo Shlaer & Mellor [SHL 92], é um grupo de objetos interconectados por muitos relacionamentos. O número de relacionamentos entre *clusters* é sempre menor que o encontrado ao interior deles.

O propósito dos subsistemas é particionar o trabalho de análise para conseguir abordar a sua complexidade. O problema contraditório de *particionar o sistema para abordar sua complexidade depois de tê-la abordado*, é resolvido por uma tentativa inicial de definição de subsistemas que pode ser refinada logo após a construção dos modelos.

Em síntese, a hierarquização que Shlaer & Mellor propõem nos modelos de objetos é a seguinte: o modelo global é composto de modelos dos domínios (1 ou mais), os domínios são compostos de subsistemas, os subsistemas são compostos por *clusters*, e os *clusters* são compostos de objetos.

1.1.6. Os *Ensembles*

O conceito de *ensemble* foi introduzido por De Champeaux como parte de sua pesquisa em AOO na Hewlett-Packard [WIR 90], [FIC 92].

Os *ensembles* são análogos aos objetos convencionais. Também são definidas as classes de *ensembles* que possuem seu correspondente nas classes convencionais. Um *ensemble* é um agrupamento *plano* de objetos (ou outros *ensembles*) que *naturalmente* estão juntos porque participam de relacionamentos todo-parte. Estes *ensembles* podem possuir atributos, estados e transições e a capacidade de interagir com outros objetos ou *ensembles* [FIC 92]. A principal diferença com os objetos é que os *ensembles* possuem paralelismo interno, enquanto os objetos são máquinas de estados finitos [WIR 90].

O propósito principal dos *ensembles* é esconder detalhes de um conjunto de objetos ou *subensembles* que são irrelevantes fora do próprio *ensemble*. Possuem também mecanismos que permitem intermediar nos *triggers* e nas mensagens entre objetos ou *ensembles* externos e os próprios componentes do *ensemble* [WIR 90].

1.1.7. Os Módulos

Os módulos ou *modules* são um mecanismo próprio da fase de análise da metodologia *OMT* (*Object Modeling Technique*) de Rumbaugh et al. [RUM 91]. Os autores definem um módulo como um conjunto de classes que capturam algum subconjunto lógico de um modelo de objetos total.

Na técnica de análise OMT as classes são representadas em um modelo de objetos, cujo diagrama pode possuir várias *páginas* de extensão. Um módulo agrupa classes em uma ou mais destas páginas. Cada módulo deve ser definido com o intuito de reduzir o número de *linhas cruzadas*.

Os autores sugerem uma estrutura de *estrela* para organizar os módulos, isto é, um único módulo contém o topo da estrutura das classes de mais alto nível. Os módulos restantes expandem cada classe de alto nível em uma hierarquia de generalização/especialização e acrescentam associações às classes de nível mais baixo.

1.1.8. Os Objetos Compostos

Os objetos compostos (*compound*, *composite* ou *aggregate objects*) são propostos na técnica de projeto orientado a objetos de Booch [BOO 91]. Um objeto composto é construído por uma abstração de outros objetos em um relacionamento de composição (ou *containing relationship* segundo o autor), isto é, o objeto composto *contém* os objetos componentes.

Booch indica que os objetos compostos são úteis como um meio de diminuir o número de objetos visíveis em um determinado nível de abstração ao encapsular os objetos componentes. Contudo, este mecanismo permite a inclusão de conexões indesejáveis no nível dos objetos compostos devidas aos objetos componentes.

1.1.9. Os Subsistemas de Jacobson et al.

Os subsistemas ou *subsystems* fazem parte da técnica de análise da metodologia *OOSE* (*Object-Oriented Software Engineering*) de Jacobson et al. [JAC 92]. Os autores definem um subsistema simplesmente como um grupo de objetos já identificados.

O propósito dos subsistemas é reduzir a complexidade do modelo da análise. Os subsistemas também podem ser compostos por subsistemas. O último nível nesta hierarquia está composto por *pacotes de serviço* (*service packages*) que são atômicos.

Os critérios para a divisão conceitual do sistema em subsistemas podem ser o funcional ou a procura da mínima interação entre os subsistemas. Os subsistemas podem se comunicar através de relacionamentos do tipo *dependeDe* (*dependsOn*), onde um objeto de um subsistema interage com um objeto de outro subsistema.

1.1.10. As Visões de Alto-Nível

As visões de alto-nível ou *high-level views* fazem parte da proposta OSA (*Object-Oriented System Analysis*) de Embley et al. [EMB 92]. Este conceito é introduzido pelos autores como um mecanismo de abstração para reduzir a complexidade dos sistemas sob análise.

As visões são divididas em virtude do aspecto do modelo que abstraem: visões das *classes de objetos* e dos *conjuntos de relacionamentos* para o aspecto estático representado no modelo de objeto-relacionamento (*Object-Relationship Model*); e visões de *estados de alto-nível* e de *transições de alto nível* para o aspecto dinâmico representado no modelo objeto-comportamento (*Object-Behavior Model*).

As visões das classes de objetos e dos conjuntos de relacionamentos correspondem a agregados de classes de objetos e conjuntos de relacionamentos de mais baixo nível. O mesmo acontece com os estados e transições de alto-nível que agrupam estados, transições e restrições de níveis mais baixos em estados e transições conceituais.

Cada componente de modelagem de alto-nível da técnica OSA é um elemento que pode ser usado no lugar de um componente básico. Assim a construção dos modelos estático e dinâmico pode começar a partir de qualquer nível de abstração e utilizando uma estratégia *top-down* e/ou *bottom-up*.

1.2. A COMPARAÇÃO DAS PROPOSTAS

A tabela 1.1 mostra a seguir uma comparação simples, em base a alguns critérios e características relevantes para este trabalho, das diferentes propostas de particionamento/agregação nas técnicas de AOO. Os critérios utilizados são:

- *estratégia*: diz relação à estratégia utilizada para compor/decompor, isto é, se é essencialmente *top-down* e portanto particiona (decompõe) o sistema previamente, ou é maiormente *bottom-up* e agrega (compõe) as classes já identificadas.
- *dimensão*: verifica a dimensão da modelagem em que é realizado o particionamento/agregação, isto é, se centra-se no aspecto estático da estrutura das classes ou considera de alguma forma também o aspecto comportamental da interação dos objetos.
- *abrangência*: está relacionado com o *escopo* ou abrangência que o conceito ou construção proposto possui ao longo do ciclo de vida do sistema orientado a objetos: nível conceitual, nível de projeto e nível de implementação.

CARACTERÍSTICAS PROPOSTAS	ESTRATÉGIA	DIMENSÃO	ABRANGÊNCIA	COMENTÁRIOS
1. ÁREAS DE INTERESSE (REENSKAUG ET AL.)	Particiona- mento	Estática/Dinâ- mica	Nível concei- tual	Muito pouco do- documentado na li- teratura
2. ASSUNTOS (COAD & YOURDON)	Agregação	Estática, inclu- indo aspectos dinâmicos no nível de servi- ço	Nível concei- tual	Pouco elabo- rado
3. CLUSTERS (NERSON)	Agregação	Estática	Níveis con- ceitual e de projeto	Definição muito intuitiva e pouco precisa
4. DOMÍNIOS (BAILIN)	Agregação	Estática	Nível concei- tual	Maior ênfase na documentação
5. DOMÍNIOS (SHLAER & MELLOR)	Particiona- mento	Estática, inclu- indo aspectos dinâmicos (mo- delos de esta- dos e proces- sos)	Nível concei- tual	Conceito mais amplo e elabo- rado

6. <i>ENSEMBLES</i> (DE CHAMPEAUX)	Agregação	Estática/Dinâmica	Níveis conceitual, de projeto e de implementação	Conceituação mais objetiva, porém de critério mais de implementação
7. MÓDULOS (RUMBAUGH ET AL.)	Agregação	Estática	Nível conceitual	Centrado mais na documentação
8. OBJETOS COMPOSTOS (BOOCH)	Agregação	Estática	Nível conceitual, de projeto e implementação	Restrito à abstração por composição
9. SUBSISTEMAS (JACOBSON ET AL.)	Agregação/ Particionamento	Estática/Dinâmica	Nível conceitual	Conceito muito amplo e pouco elaborado
10. SUBSISTEMAS (SHLAER & MELLOR)	Particionamento/Agregação	Estática, incluindo aspectos dinâmicos (modelos de estados e processos)	Nível conceitual	Conceito mais preciso que inclui representações
11. VISÕES DE ALTO-NÍVEL (EMBLEY ET AL.)	Particionamento/Agregação	Estática/Dinâmica	Nível conceitual	Conceito mais geral e dividido conforme as dimensões da modelagem

Tabela 1.1 - Comparação das propostas de particionamento/agregação

2. PROPOSTA DE PARTICIONAMENTO/AGREGAÇÃO

Os mecanismos de particionamento e/ou agregação são altamente desejáveis na análise sob diferentes paradigmas e portanto também para o caso da AOO, como foi indicado anteriormente.

Mais especificamente, para o procedimento de análise e a especificação dos sistemas de software, os mecanismos de particionamento/agregação:

- permitem particionar o trabalho de análise e modelagem quando realizada em equipes paralelas
- devem favorecer divisões *naturais* do sistema para facilitar a reintegração dos componentes
- devem poder ser aplicados inicialmente (geralmente através de uma estratégia *top-down*) a fim de contribuir na abordagem da complexidade
- fornecem abstrações dos modelos que escondem os detalhes irrelevantes nos distintos níveis dentro da hierarquia, assim o modelo pode ser apresentado segundo diferentes perspectivas⁶
- tornam os modelos mais estáveis e corretos, devido a que a própria estrutura hierarquizada obtida com estes mecanismos facilita sua posterior manutenibilidade
- favorecem o desenvolvimento de estratégias de reusabilidade ao nível de especificação de requisitos, sendo preciso para isto definir bibliotecas conceituais cujos componentes reusáveis podem ser justamente as unidades resultantes do particionamento e/ou da agregação

⁶ Por exemplo, os níveis mais altos podem ser úteis para gerentes e usuários finais, os níveis médios para analistas e projetistas, e os níveis mais baixos para projetistas e programadores.

2.1. A NOTAÇÃO E CONVENÇÕES

Como base para a representação *plana*⁷ do aspecto estrutural da modelagem orientada a objetos é utilizada a notação, ligeiramente estendida, da *modelagem de objetos* proposta por Rumbaugh et al. (vide Anexo 1 - "Notação Utilizada Conforme a Metodologia OMT" para uma descrição completa). Contudo, os conceitos aqui expostos não estão restritos a uma notação em particular e podem ser facilmente adaptados para qualquer outra representação.

A seguir é apresentada a notação definida para os construtores de agregação e suas associações⁸:

1. Os *clusters* ou agregados de classes⁹ de objetos são representados por uma caixa externa de cantos arredondados com o intuito de distingui-los das classes comuns. Os *clusters* podem também apresentar atributos, que correspondem aos nomes dos componentes (classes e/ou *subclusters*). A figura 2.1 mostra graficamente estas notações.

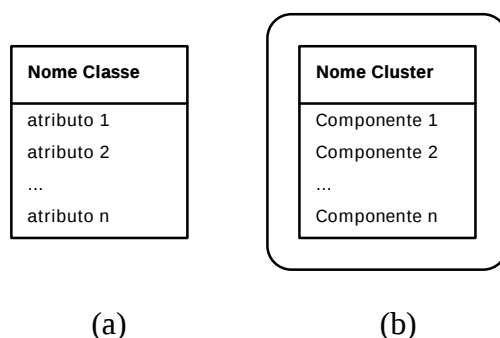


Figura 2.1 - (a) Representação de uma classe; (b) Representação de um *cluster*

⁷ Por representação plana entender-se-á aquela que não é hierarquizada através de mecanismos de particionamento e/ou agregação.

⁸ Todos os conceitos subjacentes a esta representação são definidos nas próximas seções.

⁹ No decorrer do presente trabalho os termos *classe de objetos*, *classe*, e *objeto* serão tratados indistintamente.

2. As associações agregadas são representadas por um losango com um círculo externo para diferencia-lo das associações ternárias¹⁰. A figura 2.2 mostra estas notações.

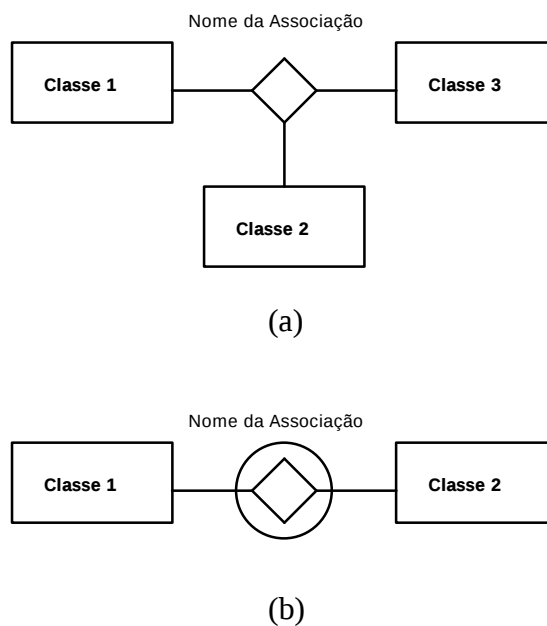


Figura 2.2 - (a) Representação de uma associação ternária; (b) Representação de uma associação agregada

3. As associações entre *clusters* ou entre um *cluster* e uma classe podem ser representadas de duas formas:
- Quando na associação participam as mesmas classes que dão o nome ao *cluster*¹¹ esta associação 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

¹⁰ Sendo que nas associações ternárias participam justamente três classes, e que isto deveria ser suficiente para distingui-las das associações agregadas, que consistem em uma (auto-relacionamento) ou duas classes participantes; julga-se importante dar maior ênfase ainda a estas últimas e assimilá-las, no possível, à representação dos *clusters* para uma melhor associação de idéias.

¹¹ Esta situação é denominada *clustering por dominância* e é definida mais adiante.

de nenhuma dos componentes, estas associações são denominadas *associações internas* e sua representação é por linhas descontínuas. Por convenção, seu nome inclui o nome do *cluster* concatenado ao nome da classe participante separado por um ponto.

Estes tipos de associações são representadas no exemplo da figura 2.3., que corresponde a uma porção simplificada do problema da IFIP [OLL 82], descrito no Anexo 2 - "Definição do Problema da IFIP". Nesta figura, o *cluster Avaliador* é configurado 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 destas 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.

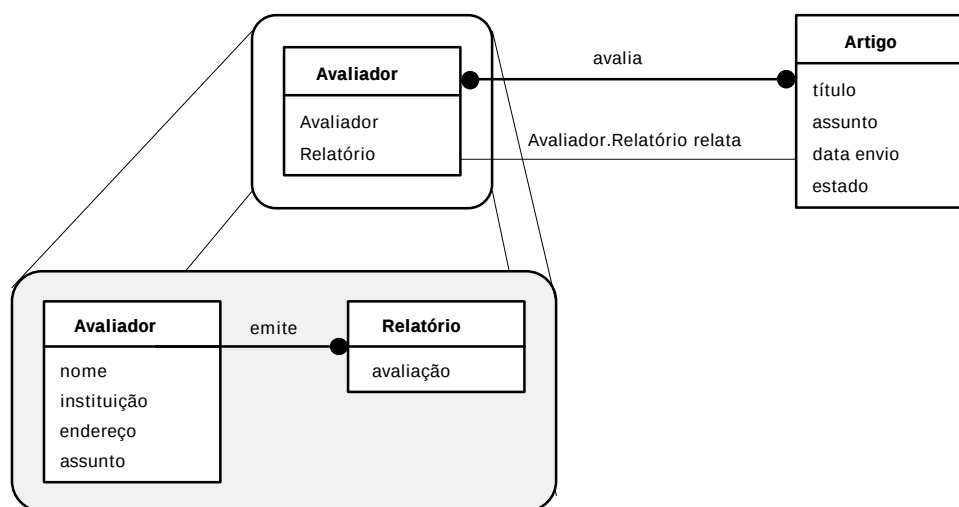


Figura 2.3 - Exemplo da representação das associações entre *clusters* e componentes

2.2. PARTICIONAMENTO/AGREGAÇÃO NA MODELAGEM ESTÁTICA

A proposta de particionamento/agregação para a AOO consiste essencialmente na complementação de duas estratégias: uma *top-down* que conduz ao particionamento do sistema sob análise, e a outra *bottom-up* que resulta na agregação das classes identificadas na análise. Ambas 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.

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

2.2.1. Estratégia Top-Down

A estratégia *top-down* permite, pelo menos na fase inicial da análise, particionar o sistema nos seus principais componentes, geralmente denominados subsistemas. Porém, nesta proposta não será feita distinção entre estes componentes identificados inicialmente, e os construídos e/ou agregados posteriormente, todos são *clusters*.

Prévio à identificação dos *clusters* iniciais, é necessário observar o espaço do problema¹² para identificar, eventualmente, a existência de mais de um domínio.

2.2.1.1. Os Domínios

Um *domínio* é um mundo real, hipotético ou abstrato, habitado por um conjunto distinto de objetos que comportam-se de acordo com regras e políticas características do domínio. Cada domínio possui um conhecimento especializado próprio e forma, portanto, um todo coeso e separado. Desta forma, os domínios tendem a repre-

¹² Espaço é considerado um conceito mais amplo que domínio, assim um espaço pode conter mais de um domínio.

sentar um conjunto de idéias muito próximas em termos das classes de objetos, e as regras e políticas que regem o seu comportamento.

Ao examinar o espaço do problema é possível perceber se existe mais de um domínio. Geralmente as visões dos usuários, sempre que não sejam funcionais, podem dar um indício no sentido da identificação destes domínios. Os domínios devem então representar as *divisões naturais* do espaço do problema. Naturais no sentido dos usuários que vivem a realidade do espaço do problema.

Os domínios podem ser classificados em aplicação, serviço, arquitetura e implementação. Os domínios de aplicação correspondem àqueles que são centrais para o espaço do problema em questão do ponto de vista do usuário. Os domínios de serviço fornecem mecanismos gerais e utilidades funcionais para os domínios de aplicação. Os domínios de arquitetura e de implementação, mais voltados às fases seguintes da análise, dizem relação a estruturas para o gerenciamento de dados e controle para o sistema, para o domínio de arquitetura; e linguagens de programação, redes de computadores, sistemas operacionais, e bibliotecas de classes, para o domínio de implementação.

O conceito de domínios de aplicação e de serviço podem ser úteis para identificar os próprios domínios no espaço do problema. A figura 2.4 mostra um exemplo de domínios para um problema de gerenciamento de ferrovias. Cada elipse representa um domínio diferente. As setas indicam relacionamentos cliente-servidor, isto é, a origem é o domínio cliente, e o destino é domínio servidor.

Deve tentar-se diminuir no possível, as interações entre os domínios, ou melhor, entre os objetos que compõem os diferentes domínios. Um bom critério de identificação e definição dos domínios passa também pela procura da minimização das interdependências entre estes domínios. Os domínios corretamente identificados e definidos (já *habitados* por objetos) determinam as seguintes restrições para os objetos:

- um objeto é definido em exatamente um domínio;
- todos os objetos em um domínio requerem da existência de objetos no mesmo domínio; e

- os objetos em um domínio não requerem da existência dos objetos em um domínio diferente.

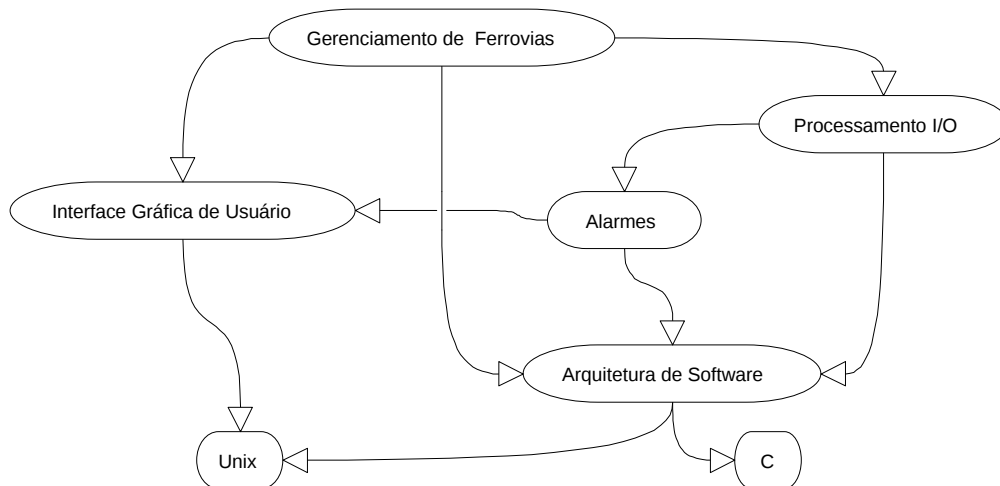


Figura 2.4 - Exemplo de domínios no espaço do problema (extraído de Shlaer & Mellor [SHL 92])

Finalmente, ainda é possível tentar uma decomposição adicional dos domínios em subdomínios, porém este nível adicional pode incorrer em um particionamento funcional, o qual é indesejável em um sistema de objetos. Assim qualquer relação funcional entre os domínios, ou entre os subdomínios (se existirem), deve ser descartada. *Os domínios e os subdomínios não são funções.* O particionamento inicial e posteriores não seguem um padrão funcional, senão estrutural conforme as associações entre as classes dos objetos. Isto também é válido para os *clusters*.

2.2.1.2. A Identificação Inicial dos *Clusters*

Quando da identificação inicial dos *clusters*, o problema centra-se em poder definir os principais componentes do sistema. Tais componentes poderão ser, e o mais provável é que assim seja, modificados posteriormente através da agregação das classes constituintes que eventualmente redefinirão os limites destes mesmos

componentes. Assim, pode afirmar-se que os *clusters* definidos no início são apenas *clusters candidatos*, sujeitos a posteriores particionamentos e/ou agregações.

Muitos dos conceitos aqui descritos são simétricos em relação às duas estratégias e, portanto, são tratados tanto na seção da estratégia *top-down* como *bottom-up*. No caso desta seção, o sentido dado é do conceito (representado pelo *cluster*) às classes ou eventualmente *clusters*, isto é, refere-se ao particionamento do *cluster* em seus componentes.

Uma consideração importante em relação a esta estratégia é *que todo particionamento do sistema o de seus componentes deve ser realizado com o reconhecimento da existência de um conjunto difuso de elementos em cada divisão ou partição identificada*. Este fato implica que nas definições dos conceitos para identificar os *clusters* candidatos, seja considerado implicitamente os componentes de cada um destes *clusters* potenciais.

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. Estas hierarquias podem possuir uma relativa profundidade, isto é, podem apresentar mais de um nível de abstração, sempre que correspondam ao mesmo conceito. Para os conceitos hierárquicos devem ser definidos *clusters* que representem:
 - um conceito geral, que eventualmente virá a ser refinado em uma hierarquia de generalização/especialização de classes ou associações de *clusters*. A hierarquia pode possuir vários níveis, assim uma classe pode ser uma subclasse (especialização) em relação a um nível superior, e também uma superclasse (generalização) em relação a um nível inferior. Este conceito pode até corresponder à classe do topo da hierarquia a ser definida, mas não necessariamente. Por exemplo, pode definir-se o *cluster* **Pessoa** sem se preocupar em representar, neste nível de abstração, um relacionamento hierárquico da classes **Pessoa**, **Autor** e **Moderador** no caso do problema da IFIP. Isto é, não

interessa distinguir ainda entre **Autores** e **Moderadores** que podem ser tratados genericamente como **Pessoas**. O *cluster* representa a hierarquia das classes **Pessoa**, **Autor** e **Moderador** e não apenas a classe mais abstrata. A figura 2.5 mostra o conceito geral como *cluster*¹³ e a hierarquia que representa. Trata-se de um particionamento e portanto o sentido é do *cluster* **Pessoa** para a hierarquia das classes **Pessoa**, **Autor** e **Moderador**.

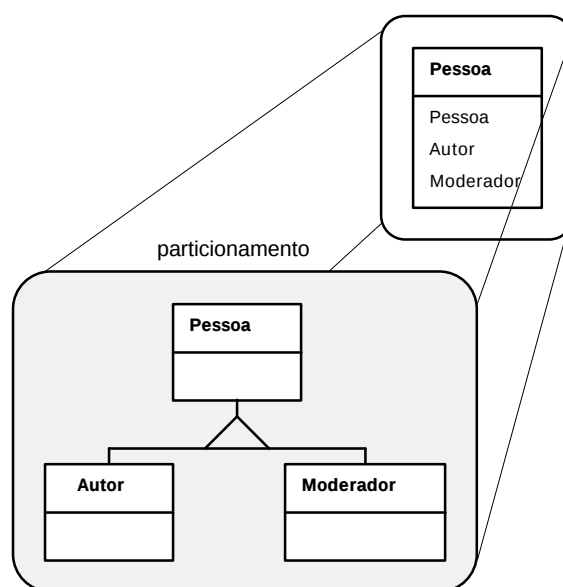


Figura 2.5 - Exemplo específico para o conceito geral representado pelo *cluster* **Pessoa**

- um conceito composto como um todo que virá a ser particionado eventualmente em uma hierarquia de composição ou associação entre *clusters*. Nada impede que este tipo de hierarquia possua mais de um nível, assim é possível ter composições, cujos componentes são, por sua vez, também composições. Este conceito descreve o todo composto de partes sem fazer menção ainda aos componentes. Também pode corresponder diretamente com a classe do topo da hierarquia. Por exemplo, pode ser definido o *cluster* **Sessão** sem se preocupar em representar, neste nível de abstração, que uma instância da classe **Sessão** poderia estar composta de instâncias da classe **Artigo Aceito** no caso do

¹³ Nesta seção todas as figuras apresentam os *clusters* com atributos (nomes dos componentes do *cluster*), apenas como uma referência. Não é necessário indicar os atributos na definição dos *clusters* na estratégia *top-down*, ao contrário do que acontece na estratégia *bottom-up* onde são obrigatórios.

problema da IFIP. Isto é, ainda não é relevante mostrar a composição de **Sessões** por **Artigos Aceitos** e suficiente com indicá-la como um todo. O *cluster* representa a hierarquia das classes **Sessão** e **Artigo Aceito** e não apenas a classe mais abstrata. A figura 2.6 mostra o conceito composto como *cluster* e a hierarquia que representa. Trata-se também de um particionamento.

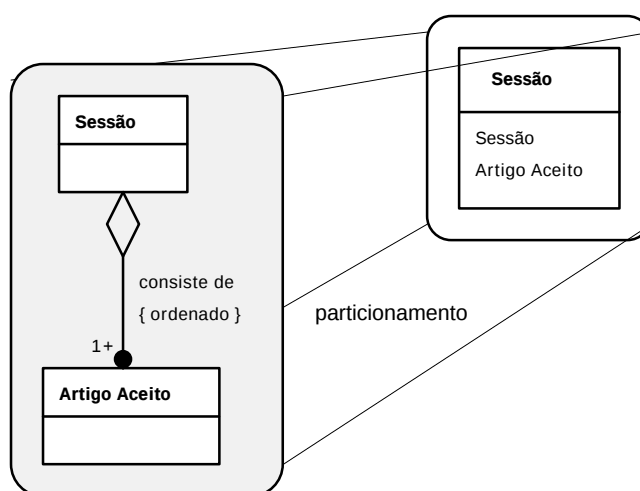


Figura 2.6 - Exemplo específico para o conceito composto representado pelo *cluster* **Sessão**

- um conceito associativo que descreve um relacionamento ou associação de outros conceitos, que virá a ser refinado eventualmente em uma classe associativa entre outras classes ou ainda entre *clusters*. Por sua vez, estas classes ou *clusters* podem também eventualmente corresponder a classes associativas, gerando assim uma hierarquia mais profunda. Este conceito descreve uma hierarquia de agregação e o *cluster* pode corresponder diretamente à classe associativa ou agregada. Por exemplo, pode ser definido um *cluster* **Contrato** que posteriormente será refinado em uma classe associativa **Contrato** que relaciona **Empresa Contratante** e **Empresa Contratada**. Isto é, não é relevante ainda a existência das classes **Empresa Contratante** e **Empresa Contratada** neste nível de abstração em que apenas interessa o **Contrato** como um todo. O *cluster* representa a hierarquia das classes **Contrato**, **Empresa Contratante** e **Empresa Contratada** e não apenas a classe associativa. A figura 2.7 mostra o conceito associativo como *cluster* e a hierarquia que representa. Como trata-se de um

particionamento, o sentido da figura é do *cluster* **Contrato** às classes **Contrato**, **Empresa Contratante** e **Empresa Contratada**.

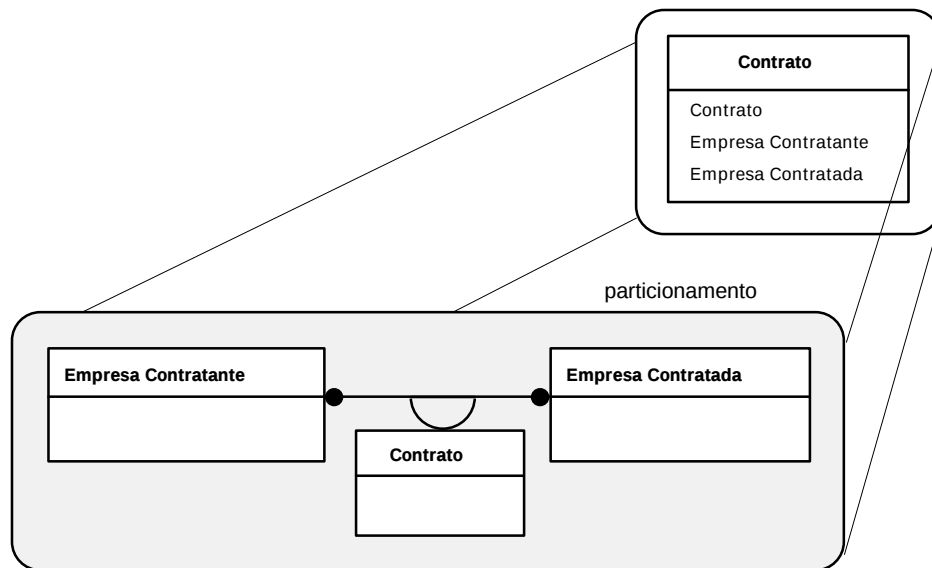


Figura 2.7 - Exemplo específico para o conceito associativo representado pelo *cluster* **Contrato**

- um conceito de participação que pode ser refinado em um relacionamento de *membership* ou uma associação entre *clusters*. Uma hierarquia com mais níveis é possível representar no caso em que existam conjuntos, cujos elementos sejam também, por sua vez, conjuntos. Este conceito descreve uma classe conjunto sem fazer referência explícita aos elementos participantes neste conjunto. Por exemplo, pode ser definido um *cluster* **Grupo de Trabalho** que posteriormente será particionado em uma classe-conjunto **Grupo de Trabalho** e os elementos participantes da classe-membro **Membro de Grupo** no caso do problema da IFIP. É irrelevante portanto, a participação do **Membro de Grupo** no conjunto **Grupo de Trabalho** neste nível de abstração, já que apenas interessam os **Grupos de Trabalho**. O *cluster* representa a hierarquia das classes **Grupo de Trabalho** e **Membro de Grupo** e não apenas a classe mais abstrata. A figura 2.8 mostra o conceito de participação como *cluster* e a hierarquia que representa no sentido do particionamento.

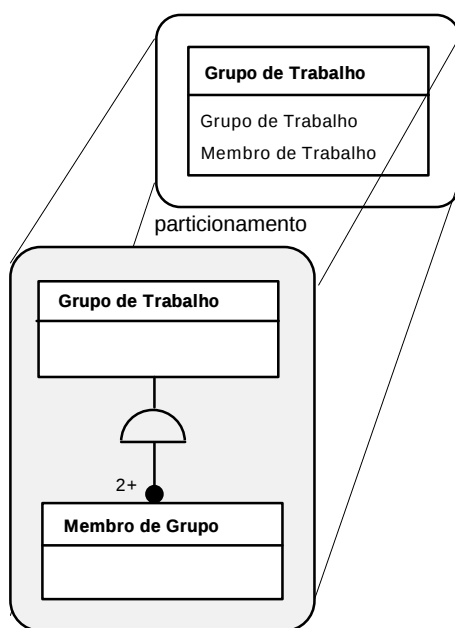


Figura 2.8 - Exemplo específico para o conceito de participação representado pelo *cluster* **Grupo de Trabalho**

2. *Conceito de Dominância*: Este conceito corresponde à situação na qual existe uma importância relativa de uma classe ou *cluster* sobre as classes ou *clusters* com os quais associa-se diretamente e que podem não estar definidas ainda. Tal classe é denominada de *dominante*¹⁴ em relação às restantes. O conceito de dominância permite abstrair e definir um *cluster* entendendo que de entre as classes que o compõem existe uma mais importante, e que as outras correspondem a detalhes irrelevantes em relação à esta classe, neste nível de abstração. 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** são detalhes de menor relevância que o representado pela classe **Estado** ao qual estão associados, no nível de abstração em questão. O *cluster* representa ao conjunto de classes **Estado**, **Município** e **Capital** e não apenas a classe dominante. A figura 2.9 mostra o conceito de dominância como *cluster* e as classes e associações que representa. Trata-se de um particionamento e o sentido é do *cluster* para as classes.

¹⁴ Na seção de estratégia *bottom-up* é fornecida uma definição mais rigorosa deste conceito.

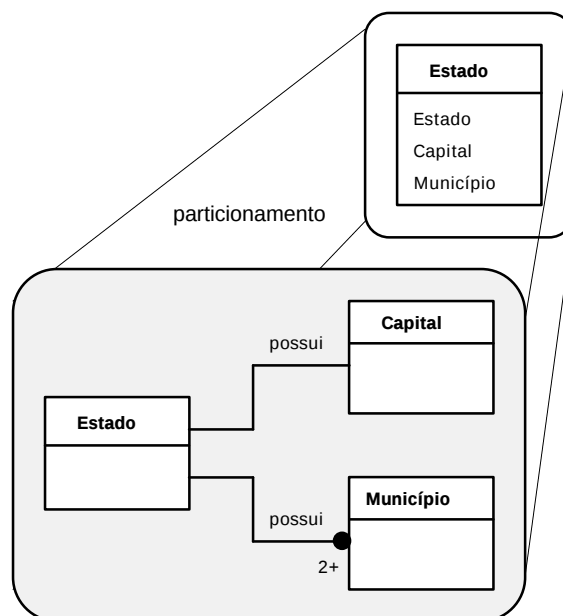


Figura 2.9 - Exemplo específico para o conceito de dominância representado pelo *cluster Estado*

3. *Conceito de Multi-relacionamento*: Este conceito refere-se a um *cluster* que eventualmente será refinado como um complexo relacionamento entre várias classes ou *clusters*. Geralmente, estes relacionamentos complexos expressam um conceito mais ou menos difuso e são representados por associações do tipo ternário ou ainda superiores. Neste caso, o relevante é a existência do relacionamento complexo, passando a um segundo plano as classes associadas. Por exemplo, pode ser definido o *cluster Matrícula* que será refinado nas classes **Aluno**, **Disciplina** e **Semestre** associadas por um relacionamento ternário denominado *matrícula*. Assim, é irrelevante a existência das classes **Aluno**, **Disciplina** e **Semestre** neste nível de abstração, passando a ganhar maior destaque a existência de um complexo **Matrícula**. O *cluster* representa o conjunto das classes **Aluno**, **Disciplina** e **Semestre** e sua associação *matrícula* e não apenas a associação. A figura 2.10 mostra o conceito de multi-relacionamento como *cluster* e as classes e associação que representa.

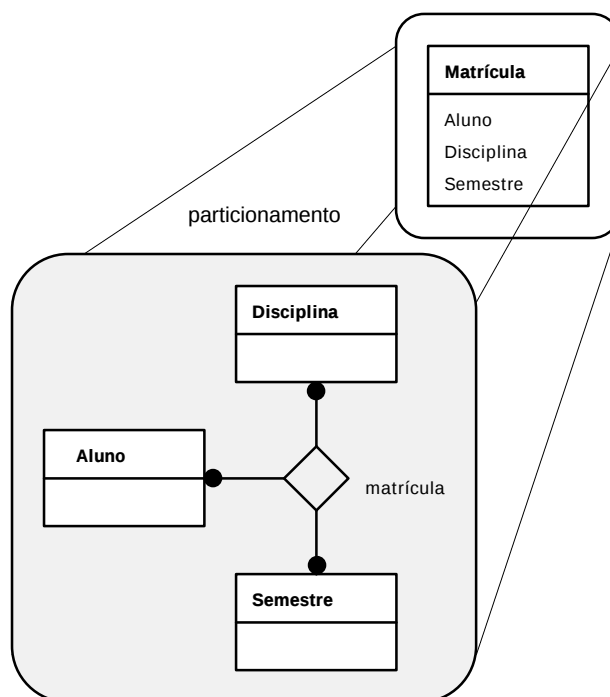


Figura 2.10 - Exemplo específico para o conceito de multi-relacionamento representado pelo *cluster* **Matrícula**

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, de procedimento, temporais, etc. O relevante aqui é a categoria e não os potenciais componentes pertencentes à categoria. Estes *clusters* são os que possuem a maior probabilidade de serem redefinidos durante a aplicação da estratégia *bottom-up*, porque geralmente não apresentam nenhum tipo de associação entre seus componentes. 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. Ambas classes podem ser consideradas como pertencentes à categoria de correspondências. O mais importante é a categoria representada pelo *cluster* **Correspondência** do que os componentes individuais¹⁵. A figura 2.11 mostra o conceito de categoria como *cluster* e as classes que representa.

¹⁵ Ainda é possível configurar uma hierarquia de generalização/especialização com a superclasse **Correspondência** e as subclasses **Carta Intenção** e **Resposta Convite**.

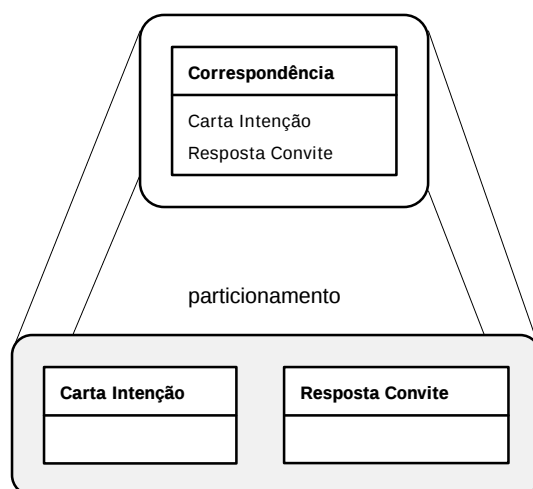


Figura 2.11 - Exemplo específico para o conceito de categoria representado pelo *cluster* **Correspondência**

Finalmente, é recomendado também utilizar os papéis dos usuários baseados em objetos para particionar inicialmente o sistema. Para isto deve observar-se os papéis que desempenham os diversos usuários do sistema. Podem existir grupos de usuários com responsabilidades vinculadas a grupos específicos de objetos: os *clusters* candidatos. Em outras situações, os papéis dos usuários podem estar baseados em funções ou considerações temporais. As interações dos diferentes usuários com o sistema podem seguir certas seqüências ou fases, e estarem vinculadas a um mesmo conjunto de objetos. Neste caso, esta guia não é útil porque inevitavelmente levaria a ter que definir *clusters* com muitas classes de objetos em comum.

2.2.1.3. A Interdependência Entre os *Clusters*

Após a identificação inicial dos *clusters* candidatos, podem ser examinadas as interdependências existentes entre estes *clusters*. As seguintes são algumas considerações para precisar mais a definição tentativa dos *clusters*:

- 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 idéias 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.

- Tratar os *clusters candidatos* como possuindo um comportamento em 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 idéias 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, inevitavelmente pode ser definida a mesma classe em mais de um *cluster*. Para evitar uma sobre especificação, isto deve ser evitado, modificando os *clusters*, particionando novamente ou agregando algumas das classes já identificadas. Contudo, as classes em comum são também um bom mecanismo para simplificar os relacionamentos entre os *clusters*. Ao incluir as classes comuns em cada *cluster* é obviada a necessidade de representar externamente os relacionamentos entre estes *clusters*, tornando a classe um relacionamento implícito. Um adequado compromisso entre a sobre especificação e a minimização dos relacionamentos deve ser procurada. A figura 2.12 mostra um exemplo de uma classe **Aluno** que é utilizada como relacionamento implícito entre os *clusters* **Matrícula** e **Bolsa**. Na figura 2.13 a mesma classe **Aluno** é mostrada em apenas um dos *clusters* o que implica em um relacionamento externo entre os *clusters* **Matrícula** e **Bolsa**.
- Às vezes, os papéis desempenhados por uma classe em diferentes *clusters* podem definir uma hierarquia de generalização, com cada papel como subclasse e a classe que desempenha os papéis como superclasse. Isto pode definir um *cluster* agregado dos *clusters* onde participa a classe com papéis, ou um *cluster* separado, ao mesmo nível de abstração, que agrega os papéis e se relaciona com os *clusters* onde eram desempenhados os papéis. Por exemplo, os papéis de *estudante* e *bolsista* desempenhados pela classe **Aluno**, nas figura 2.12 e 2.13, podem constituir uma hierarquia como a mostrada na figura 2.14., que neste caso, define um *cluster* **Aluno** relacionado com o *cluster* **Matrícula** e a classe

Bolsa (ao retirar a classe **Aluno** com o papel *bolsista*, esta classe ficou isolada e não constitui um *cluster*).

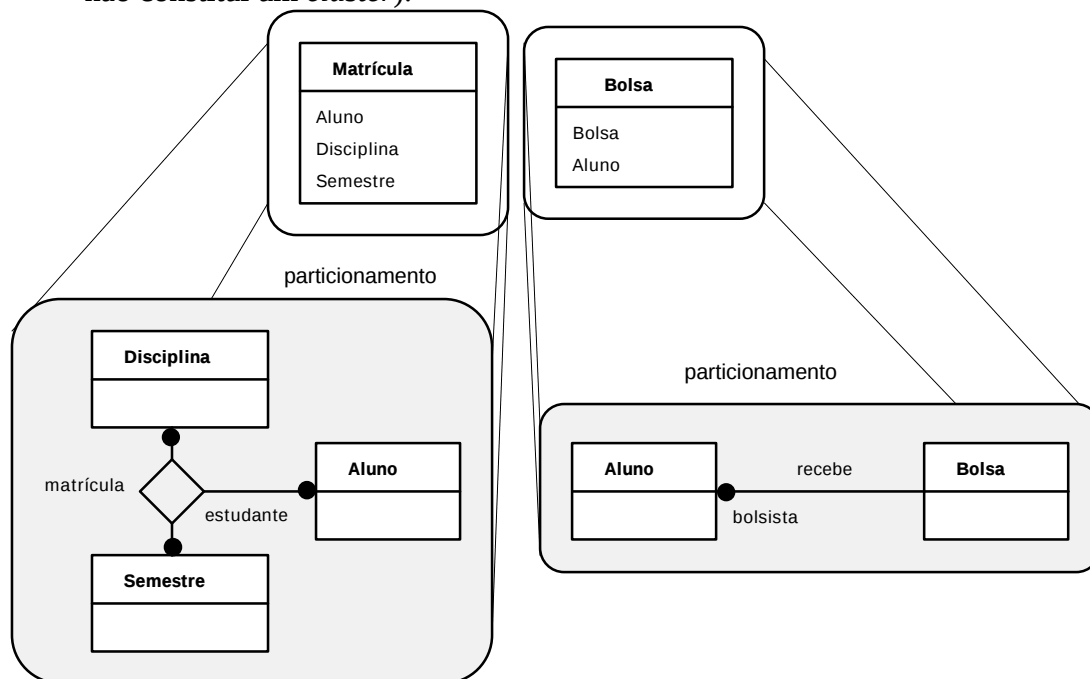


Figura 2.12 - Exemplo de uma representação repetida de uma classe em comum com relacionamento implícito entre os *clusters*

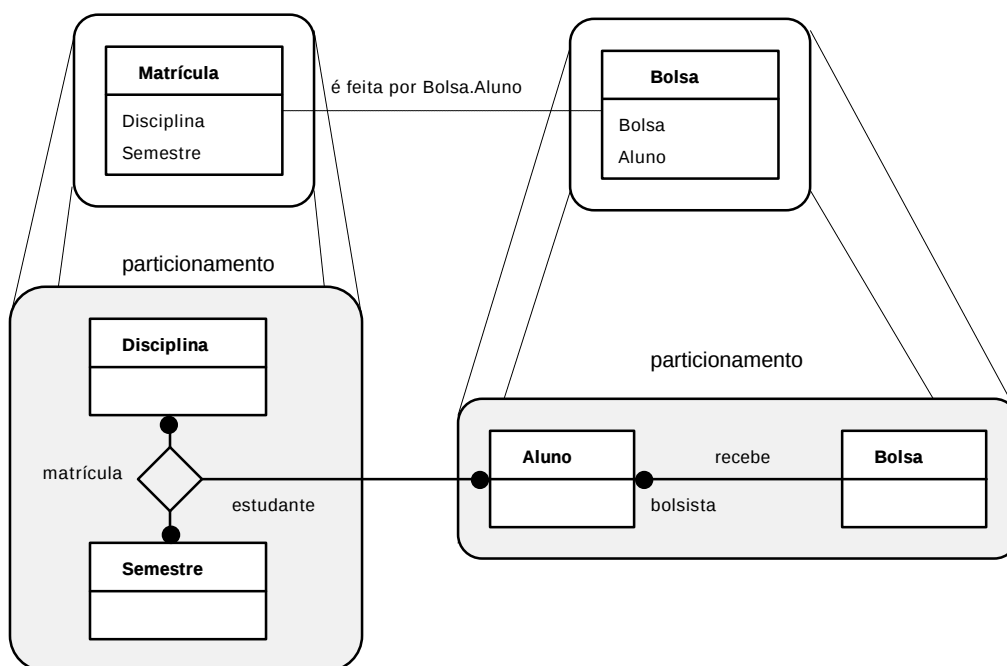


Figura 2.13 - Exemplo de uma única representação de uma classe em comum com relacionamento explícito entre os *clusters*

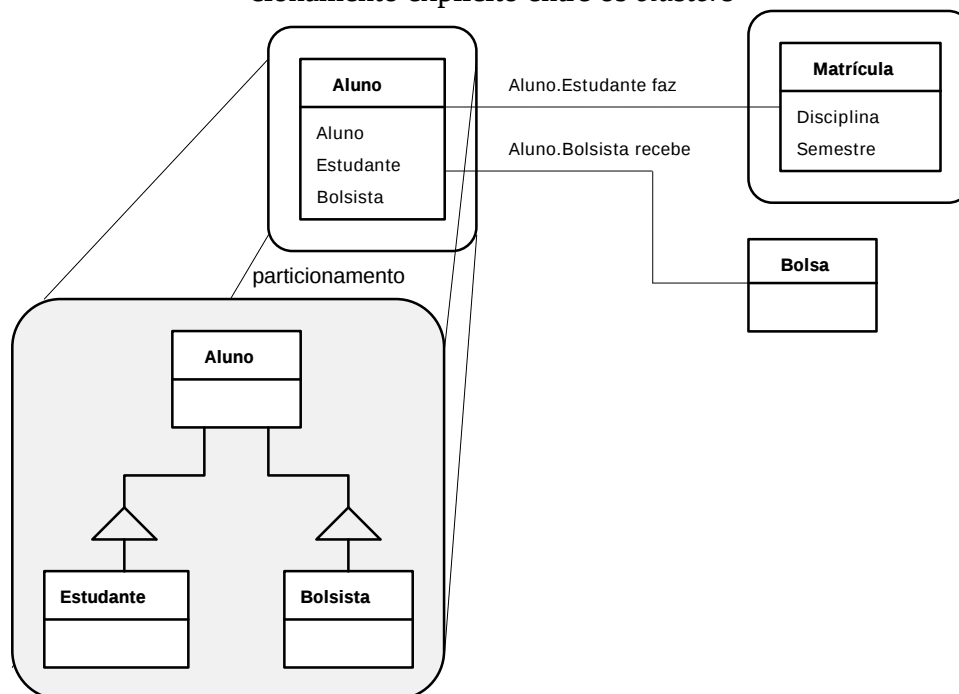


Figura 2.14 - Exemplo da definição do *cluster Aluno* a partir dos papéis de uma classe (vide figuras 2.12 e 2.13)

- 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 ou relacionamento agregado. 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 2.15.

2.2.1.4. O Procedimento Geral

A estratégia *top-down* descrita nas seções anteriores pode ser sintetizada no seguinte procedimento geral:

1. Identificar a existência de mais de um domínio no espaço do problema: Os domínios devem ser todos coesos e relativamente independentes, possuindo conhe-

cimento próprio. Deve evitar-se um particionamento funcional. Os seguintes passos devem ser realizados para cada domínio em separado.

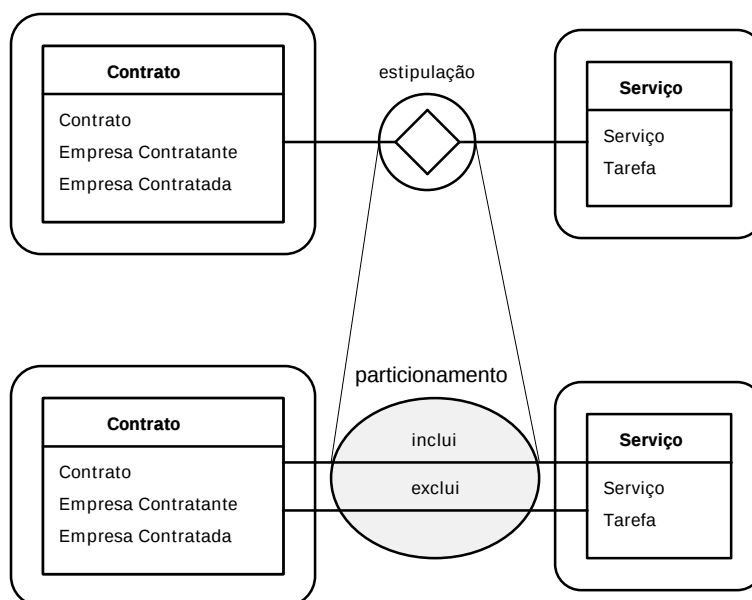


Figura 2.15 - Exemplo específico de definição e particionamento de uma associação agregada em associações paralelas

2. 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 de cada domínio. Evitar, no possível, que os *clusters* venham a ter classes em comum.
3. Refinar os *clusters* anteriores com a revisão de suas interdependências: Considerar os *clusters* como escondendo detalhes irrelevantes externamente, 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.
4. Aplicar os passos anteriores em forma recursiva: Esta estratégia deve ser combinada com a estratégia *bottom-up* com o propósito de complementar seus critérios na definição dos *clusters*, classes, associações comuns e agregadas.

Na prática, pode ser necessário uma *blitz* de identificação de objetos para poder definir e redefinir os *clusters* candidatos através da aplicação das estratégias *top-down* e *bottom-up*.

2.2.2. 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 *clustering* ou agregação de classes e associações já definidas.

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. Claramente, sendo esta uma estratégia *bottom-up*, os componentes a serem 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.

2.2.2.1. 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. Um componente diz-se que é *dominante* se as cardinalidades entre este componente e os componentes com os quais associa-se diretamente são do tipo 1:muitos, isto é, associações de muitas instâncias dos outros componentes com uma instância do componente dominante. A dominância é mais clara quanto mais associações deste tipo o componente possua. Os relacionamentos muitos:muitos 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 2.16 mostra a configuração típica de esta situação, com o caso de uma classe dominante **Produto** e as classes **Fornecedor**, **Depósito** e **Linha de Produção**. Trata-se de uma agregação e o sentido é das classes para o *cluster*.

2. *Agregação por Abstração*: O critério baseia-se nas diferentes abstrações construídas por hierarquia. 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 2.5, 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 2.6, 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 2.7, concebida no sentido da agregação da hierarquia das classes **Contrato**, **Empresa Contratante** e **Empresa Contratada**; e
- *participação*: como exemplo ver a figura 2.8, concebida como agregação da hierarquia das classes **Grupo de Trabalho** e **Membro de Grupo**.

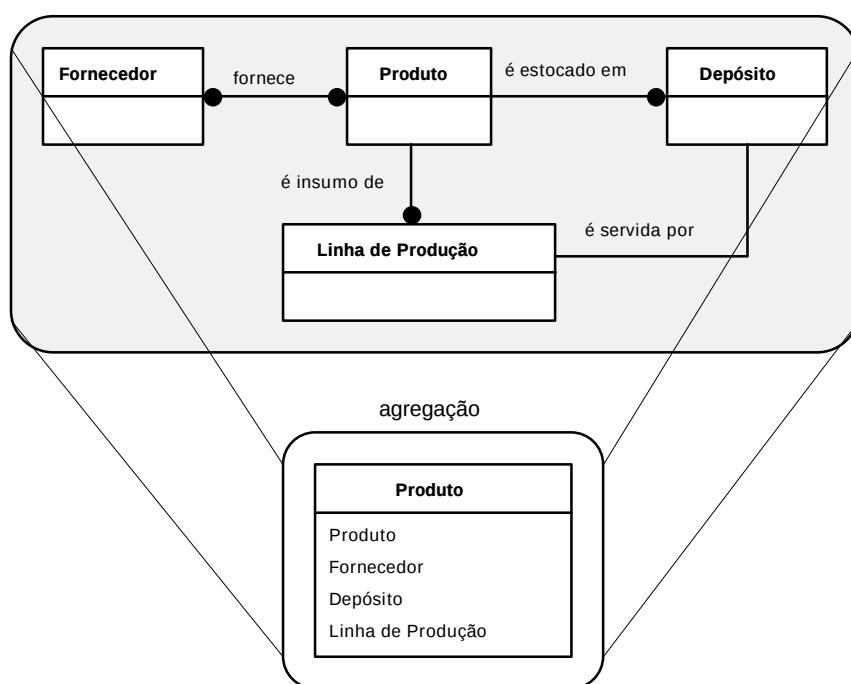


Figura 2.16 - Exemplo específico de agregação pelo critério de dominância

3. *Agregação por Restrição*: Corresponde ao caso em que uma restrição do modelo relaciona implicitamente os componentes participantes. É claro que estes componentes devem permanecer juntos para que a restrição faça sentido e possa ser facilmente reconhecida. A figura 2.17 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*. Como é uma agregação, o sentido dado é das classes **Pós-Graduando**, **Bolsa** e **Salário** para o *cluster Manutenção Estudante*.

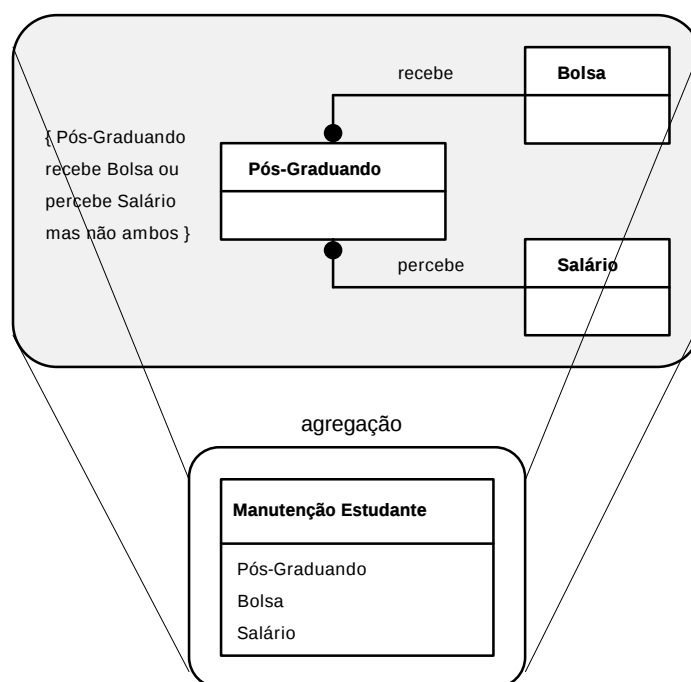


Figura 2.17 - Exemplo específico de agregação pelo critério de restrição

4. *Agregação por Associações Simples*: Este critério baseia-se nas associações simples (unárias e binárias) de diversa cardinalidade (1:1, 1:muitos, muitos:muitos) entre componentes para agregar preservando tais associações. a figura 2.18 mostra alguns possíveis *clusters*: **Plano de Disciplinas**, **Manutenção**

Aluno, **Turma** e **Conclusão** construídos a partir de diversas associações. Todos estes exemplos são no sentido da agregação, isto é, das classes e associações para os *clusters*.

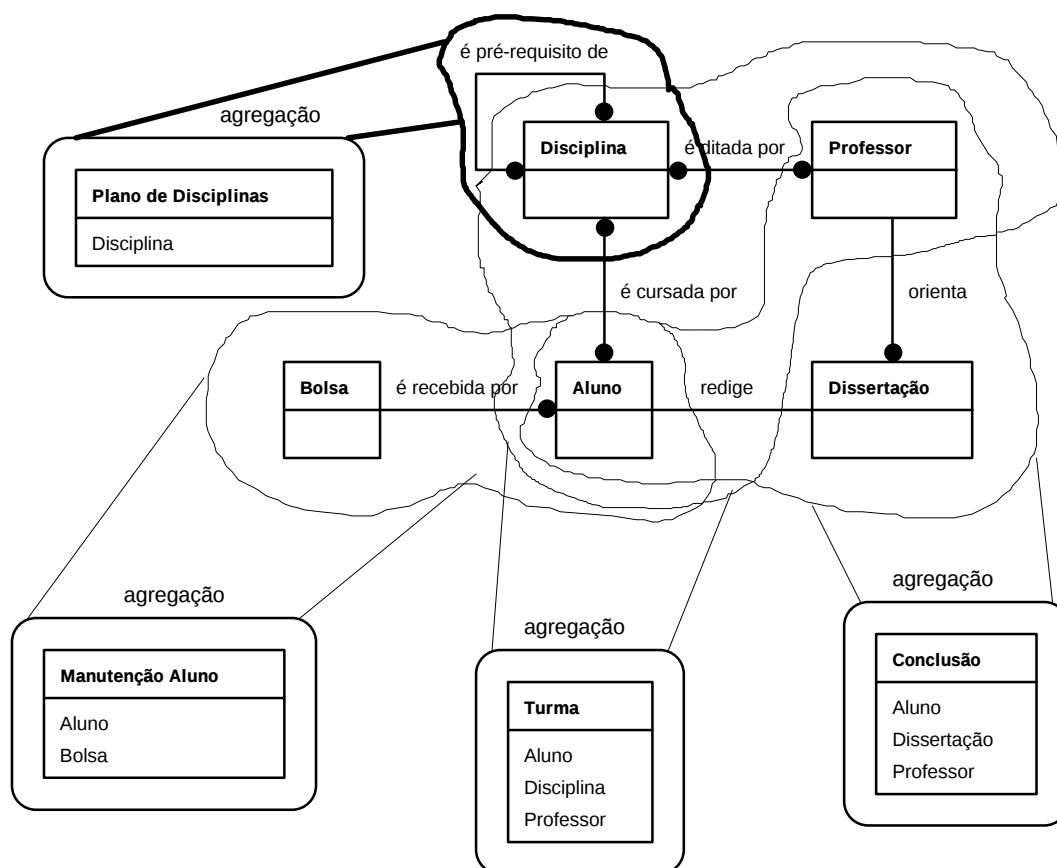


Figura 2.18 - Exemplos específicos de agregações pelo critério de associação simples

5. *Agregação por Associação Complexa*: Este critério baseia-se nas associações complexas (ternárias ou ainda superiores) entre componentes, independentemente de sua cardinalidade. Como convenção, os *clusters* agregados preservam o nome do multi-relacionamento. Como exemplo, ver a figura 2.10, concebida no sentido da agregação, isto é, do relacionamento complexo *matrícula* entre as classes **Aluno**, **Disciplina** e **Semestre** para o cluster **Matrícula**.

2.2.2.2. A Prioridades entre os Critérios

É comum que existam conflitos na hora de aplicar mais de um critério a um conjunto de componentes suscetíveis de serem agregados em um *cluster*. Por exemplo, pode existir um componente dominante que também faz parte de uma hierarquia de generalização/especialização. Deve agregar-se por dominância ou abstração por generalização?

Para resolver estes e outros conflitos é definido o conceito de *coesão* dos *clusters*. A coesão de um *cluster* representa a *força* interna que relaciona os componentes do mesmo. Baseado neste conceito, é possível contar com um espectro da coesão em função dos critérios de agregação. Os níveis de coesão, apresentados de maior a menor, são os seguintes:

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

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

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

Nível 4 - Agregação por associação simples: Existe um sub-espectro dentro deste nível, que vai da maior coesão nas associações unárias, passando pelas associações binárias 1:1, associações binárias 1:muitos, e finalmente, com a menor coesão, as associações binárias muitos:muitos.

Nível 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.

Nível 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).

2.2.2.3. O Procedimento Geral

É claro que a aplicação da estratégia *bottom-up* só é possível após algum tipo de particionamento inicial, a partir do qual possam ser identificados e definidos os diversos componentes. Com este considerando, esta estratégia, cujos critérios e prioridades foram descritos nas seções anteriores, pode ser sintetizada no seguinte procedimento geral:

1. Localizar os componentes dominantes em cada domínio ou *cluster* já definido (possuem maior coesão).
2. Agregar os componentes formando *clusters*. Usar os outros 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): 1º) os componentes do *cluster* potencial devem pertencer a um mesmo domínio ou *cluster* previamente particionado; e 2º) se existe conflito entre potenciais *clusters* de um mesmo nível de coesão, devem manter-se estes componentes sem agregar dentro dos domínios ou *clusters* previamente definidos, e tentar um possível particionamento.
3. Aplicar os passos anteriores em forma recursiva: Esta estratégia deve ser combinada com a estratégia *top-down* com o propósito de complementar seus critérios na definição dos *clusters*, classes, associações comuns e agregadas.

2.2.3. A Combinação das Estratégias

Com o propósito de usufruir das vantagens de ambas as estratégias, pode ser definido o seguinte procedimento integrado:

1. Aplicar o procedimento geral da estratégia *top-down*. Quando seja necessário redefinir e agregar segundo os critérios da estratégia *bottom-up*.

2. Aplicar o procedimento geral da estratégia *bottom-up*. Quando seja necessário redefinir e particionar segundo os critérios da estratégia *top-down*.
3. Combinar aplicando sucessiva e recursivamente ambas estratégias para redefinir componentes, associações e *clusters*.
4. Verificar a consistência do modelo após sucessivas aplicações das estratégias *top-down* e *bottom-up*. As associações entre os componentes devem verificar-se em cada nível definido.
5. 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.

O resultado da aplicação combinada de ambas as estratégias é um modelo hierarquizado, 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; e um último nível, que está representado por classes de objetos com atributos, correspondendo ao maior nível de detalhe possível do modelo. A figura 2.19 mostra uma hipotética estrutura final do modelo de AOO resultante da aplicação da proposta de particionamento/agregação.

O Anexo 3 - "Particionamento/Agregação Para o Problema da IFIP" apresenta um exemplo de um modelo hierarquizado para a definição do problema da IFIP, conforme descrição do anexo 2, resultante da aplicação da proposta e de acordo com as notações mostradas no trabalho.

2.2.4. As Bases da Proposta

A proposta que apresentada nas seções anteriores possui as bases descritas na literatura que são indicadas a seguir.

Em relação à notação base para o aspecto estático, é utilizada a proposta de Rumbaugh et al. [RUM 91] (vide Anexo 1 - "Notação Utilizada Conforme a Metodo-

logia *OMT*") como parte da metodologia *OMT* (*Object Modeling Technique*) acrescida de algumas extensões da técnica *OSA* (*Object-Oriented System Analysis*) de Embley et al. [EMB 92]. Quanto às convenções próprias do particionamento/agregação, existem diversas propostas na literatura para representar os *clusters*, tais como as da técnica *OSA* de Embley et al. [EMB 92] (*clusters* como caixas cinzas), as de Teorey et al. [TEO 89] para o modelo entidade-relacionamento (caixas com bordas grossas), ou ainda as da proposta de aninhamento de diagramas entidade-relacionamento de Carlson & Ji [CAR 90] (caixas com dupla linha). Qualquer uma delas pode ser adotada, porém neste trabalho optou-se pela definida por Batini et al. [BAT 92] para o projeto conceitual de banco de dados (caixas externas de cantos arredondados).

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]. A seção dedicada aos domínios dentro desta estratégia é baseada na proposta similar de Shlaer & Mellor [SHL 92]. Todas as definições são extraídas do texto destes autores.

Quanto à interdependência entre os *clusters*. O conceito de comportamento paralelo dos *clusters* é tomada dos *ensembles* de De Champeaux, descritos em [WIR 90] e [FIC 92]. O conceito de papéis como especializações em uma hierarquia é usada por Reenskaug et al. e descrita em [WIR 90]. A idéia de associações agregadas representando associações comuns paralelas é de Embley et al. [EMB 92]. A representação de classes comuns nos *clusters* é originada de uma idéia na representação de módulos de Rumbaugh et al. [RUM 91], e nos subsistemas de Shlaer & Mellor [SHL 92].

Em relação à estratégia *bottom-up*, é a mais desenvolvida na literatura. O aqui exposto, em termos de critérios de agregação, prioridades e procedimento, é adaptado 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].

2.3. O PARTICIONAMENTO/AGREGAÇÃO NA MODELAGEM DINÂMICA

Para uma modelagem conceitual completa orientada a objetos, é preciso acrescentar uma visão da dimensão dinâmica dos objetos componentes do sistema. Geralmente, o comportamento do sistema de objetos é representado em termos de funções ou ações, eventos, estados e transições.

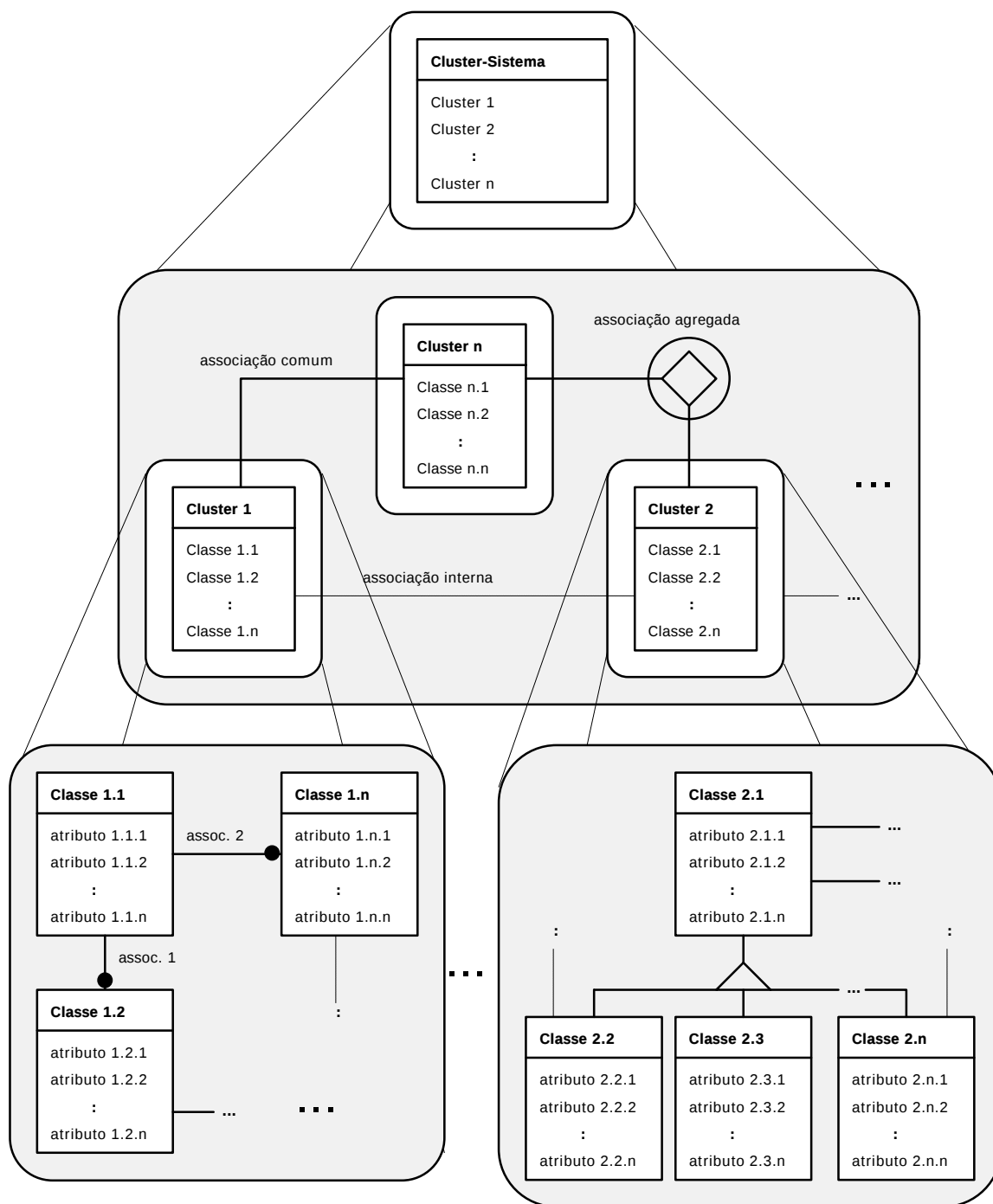


Figura 2.19 - Estrutura hierárquica final do modelo de AOO

Neste trabalho não será apresentada uma proposta específica para abordar este aspecto, porém algumas idéias são apresentadas a seguir:

- *Estratégia de particionamento/agregação:* A idéia básica é utilizar a hierarquização estrutural, definida na seção do particionamento/agregação, para o aspecto dinâmico e vice-versa. Em forma ideal, o particionamento/agregação estrutural e funcional deve ser realizado em conjunto, aportando cada aspecto critérios para definir *clusters* de componentes e funções.
- *Particionamento funcional:* O particionamento das funções mais abstratas não será realizado conforme a clássica decomposição funcional, senão através de uma orientação por eventos, como na análise essencial de sistemas [MCM 91]. Nesta visão, as funções são decompostas em relação à resposta que oferecem a eventos externos específicos.
- *Classes de eventos:* Para manter uma consistência nos níveis de abstração dos aspectos estrutural (*cluster*, *subcluster*, classes e atributos das classes) e funcional (funções abstratas, subfunções e métodos das classes), e considerando que o particionamento funcional será realizado por eventos, é necessário introduzir o conceito de *classes de eventos*. Os eventos devem ser classificados conforme as funções abstratas nos diversos níveis da hierarquia. Uma primeira classificação corresponde em dividir os eventos que relacionam o ambiente com o sistema em estímulos (eventos externos de entrada ao sistema) e respostas (eventos internos de saída para o ambiente). Os eventos também sofrem particionamento/agregação em concordância com as funções abstratas e os *clusters*.
- *Mapeamento das funções:* As funções abstratas definidas nos níveis mais altos da hierarquia devem ser mapeadas em definições precisas do comportamento dos objetos definidos no último nível da hierarquia. Este mapeamento deveria ser gradual, conforme o detalhe vai aumentando ao descer pela hierarquia dos *clusters*. A modelagem dinâmica dos objetos do último nível deve consistir em eventos, estados, transições e ações.
- *Modelagem dinâmica dos objetos:* Ainda deve ser definido o modelo mais adequado para representar o comportamento dos objetos. Entre os modelos mais conhecidos estão: diagramas de transição de estados [YOU 90], redes de Petri [HEU 90], *statecharts* [HAR 87], redes de estados (*state nets*) [EMB 92], e modelos de estados [SHL 92]. Também é possível introduzir modificações e/ou ex-

COMBINAÇÃO DAS ESTRATÉGIAS:	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não	Não
1. APLICAÇÃO RECURSIVA DE UMA OU AMBAS ESTRATÉGIAS	Não	Apenas para <i>bottom-up</i>	Não	Não	Não	Sim	Não	Não	Não	Não	Sim
2. DEFINIÇÃO DO CLUSTER-SISTEMA	Sim	Não	Sim	Não	Não	Não	Não	Não	Não	Não	Não

Tabela 2.1 - Comparação dos conceitos da proposta

3. CONCLUSÕES E COMENTÁRIOS

Primeiramente, devem ser examinados os objetivos estabelecidos inicialmente para verificar se foram plenamente atingidos:

- Foram estudadas onze propostas de mecanismos de particionamento/agregação para a AOO.
- Foi adaptada a proposta básica de *clustering* de Teorey et al. [TEO 89], para a modelagem estrutural como é descrito na seção da estratégia *bottom-up*.
- Foram adaptados, estendidos e definidos conceitos, procedimentos e notações para abordar o particionamento da modelagem estrutural dos sistemas orientados a objetos.
- Foi proposta uma integração, em termos de procedimentos, recursividade, conceitos e notação, das estratégias *top-down* e *bottom-up*.
- Foram definidas algumas idéias básicas que virão a definir mecanismos de particionamento/agregação para o aspecto dinâmico em concordância com o aspecto estrutural da modelagem orientada a objetos.
- A proposta foi avaliada em termos comparativos e através da aplicação ao caso do problema da IFIP.

Pelo indicado acima, os objetivos foram atingidos satisfatoriamente no desenvolvimento deste trabalho.

Em termos gerais, pode afirmar-se que a proposta do presente trabalho é mais completa e rigorosa que as similares encontradas na literatura. É baseada fortemente em heurísticas de modelagem e portanto fica longe de ser 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.

O central da proposta está em combinar as estratégias *top-down* e *bottom-up* para a AOO, sem dar maior ênfase a nenhuma 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 orientados a objetos.

Considerando as referências bibliográficas, existe uma clara fraqueza na aplicação da estratégia *top-down* na AOO, geralmente por considera-la vinculada à decomposição funcional clássica, claramente contrária ao conceito de encapsulamento defendido no paradigma de orientação a objetos. O mesmo não acontece em se tratando da estratégia *bottom-up*, que tem herdado em forma natural os desenvolvimentos na área de modelagem semântica de dados.

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(s) domínio(s) do espaço do problema, tem-se um problema *bootstrapping*, isto é, no início deve ser construído um modelo plano do sistema de objetos para poder particioná-lo com maior certeza, assim deve lidar-se com a complexidade do(s) domínio(s) em primeiro lugar, quando o que se pretende com o particionamento é justamente abordar tal complexidade. Uma maneira de superar esta circularidade está na combinação recursiva e sucessiva de ambas as estratégias.

Neste mesmo sentido, é curioso observar que existem algumas propostas, notavelmente a de Shlaer & Mellor [SHL 92], que distinguem os componentes do primeiro particionamento como *subsistemas*. Os restantes subcomponentes caem na categoria de *clusters* ou algum outro construtor. Desde o ponto de vista dos conceitos de modelagem e da especificação dos modelos hierarquizados orientados a objetos, não existe nenhuma diferença entre os componentes do primeiro nível de decomposição e os restantes.

Outra contribuição do trabalho está na definição explícita do denominado *cluster-sistema*, que corresponde ao *cluster* do topo da hierarquia e representa o sistema de objetos como um todo. Este *cluster* conceitual não é novo, e já foi citado por Jalote

[JAL 89], Booch [BOO 86], e mais recentemente por De Champeaux [DEC 92]. Contudo, não tinha sido definido em nenhuma das propostas de AOO encontradas na literatura.

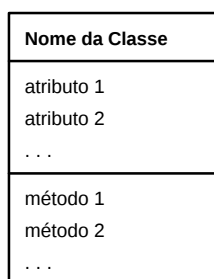
Quanto à notação, é preciso reiterar que os conceitos aqui apresentados independem da notação utilizada e podem ser facilmente utilizados com outras sintaxes. O mesmo acontece com as convenções adotadas para a definição de *clusters* e associações entre os mesmos.

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. Isto fica mais em evidência, sendo esta uma proposta apenas para o aspecto estrutural da modelagem orientada a objetos.

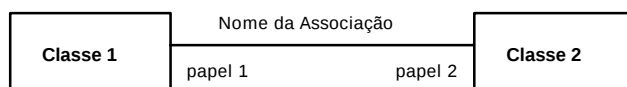
ANEXO 1 - NOTAÇÃO UTILIZADA CONFORME A METODOLOGIA OMT

A notação utilizada é baseada em sua maioria na proposta por Rumbaugh et al. [RUM 91] para a modelagem de objetos (modelagem estrutural).

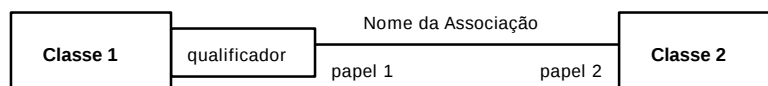
Classe:



Associação entre a Classe 1 e a Classe 2:



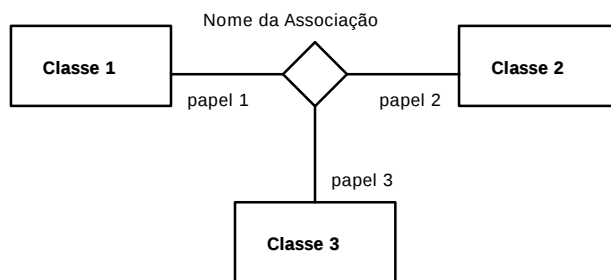
Associação qualificada entre a Classe 1 e a Classe 2 (qualificador é atributo da Classe



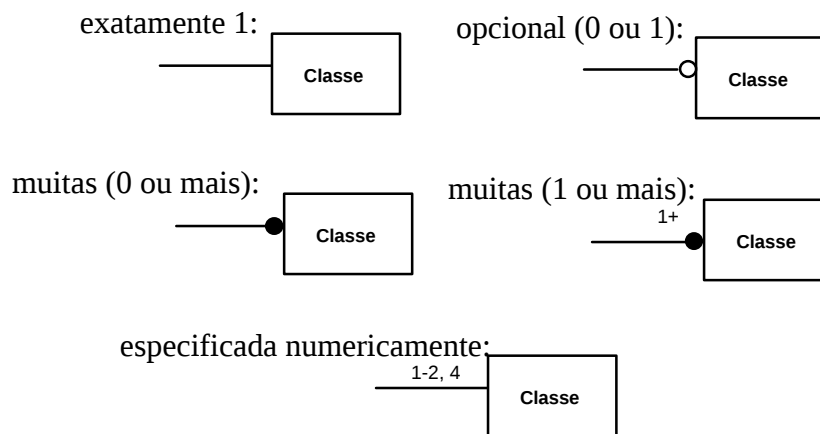
Associação como Classe (agregação ou classe relacional)



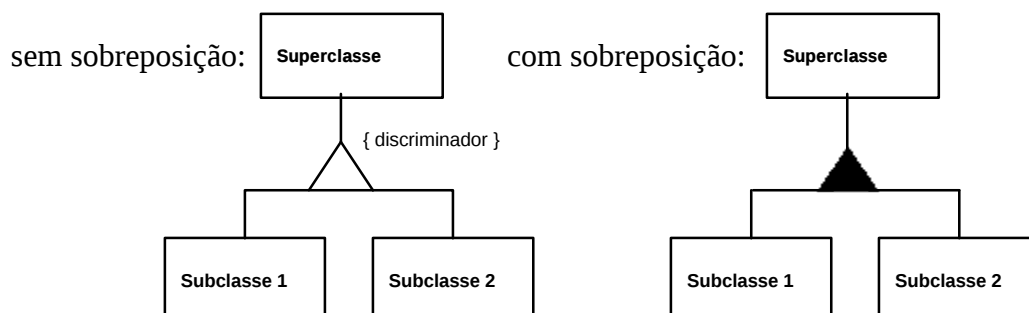
Associação ternária:



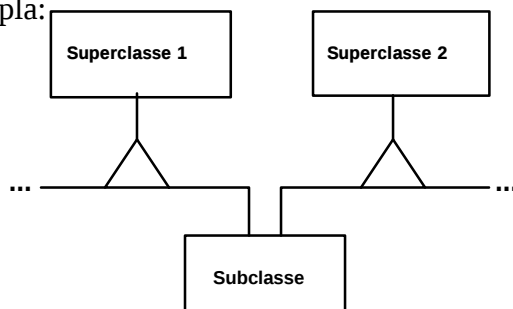
Multiplicidade das associações:



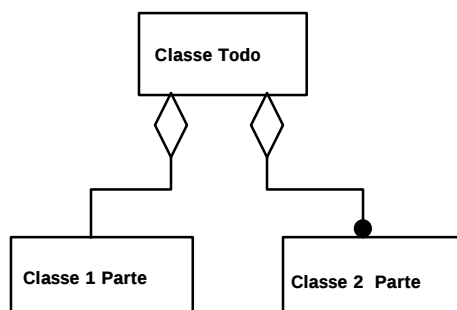
Generalização (é-um, é-subclasse-de, ou ainda é-subconjunto-de):



Herança múltipla:

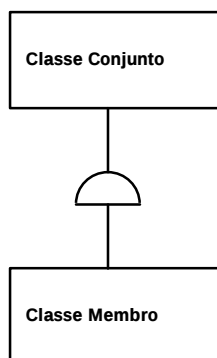


Composição (é-composto-de, é-parte-de):



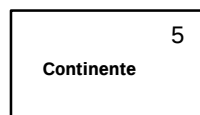
Utilizando o conceito de relacionamento *membership* [DAV 90], *é-membro-de* [POT 88] ou *é-elemento-de* [MAT 89] da modelagem semântica de dados, que também é utilizado na técnica OSA de Embley et al. [EMB 92], é definida uma notação específica adicional mostrada a seguir:

Associação de Participação (membership)

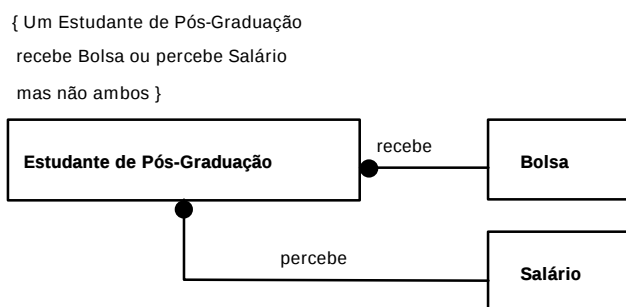


Finalmente, adaptando a notação da técnica *OSA* de Embley et al. [EMB 92], é possível acrescentar diferentes restrições em forma de anotações, que venham aumentar a semântica do modelo. Exemplos deste tipo de restrições são:

Número de Objetos na Classe:



Anotação Textual da Restrição:



ANEXO 2 - DEFINIÇÃO DO PROBLEMA DA IFIP

O problema da IFIP (*International Federation for Information Processing*) proposto por Olle [OLL 82], foi apresentado inicialmente em 1982, como base para uma conferência para o estudo comparativo de metodologias de projeto de sistemas de informação. O caso da IFIP é um problema amplamente divulgado na literatura e muito utilizado até hoje, permitindo comparações com outras aplicações já desenvolvidas de técnicas de modelagem não orientadas a objetos (como por exemplo as apresentadas em [IFI 82]), e orientadas a objetos (vide [BUS 94] como referência).

O texto integral da definição do problema, a partir da referência [OLL 82], é o seguinte:

“ESTUDO COMPARATIVO DE METODOLOGIAS DE PROJETO DE SISTEMAS DE INFORMAÇÃO

Definição do Problema

1. Antecedentes

Uma Conferência de Trabalho da IFIP é uma conferência internacional que tenta reunir especialistas de todos os países da IFIP para discutir tópicos técnicos de interesse específico para um ou mais Grupos de Trabalho da IFIP. O procedimento usual, que será considerado para este propósito, é uma conferência convidada que não é aberta para qualquer um. Para tais conferências, parte do problema é assegurar que todos os membros relacionados com o(s) Grupo(s) de Trabalho e Comitê(s) Técnico(s) da IFIP sejam convidados, mesmo que eles não possam comparecer. Além disto, é importante assegurar suficientes assistentes à conferência de tal maneira que a viabilidade financeira seja atingida sem exceder o máximo indicado pelas facilidades disponíveis.

As políticas da IFIP sobre Conferências de Trabalho sugerem a nomeação de um Comitê de Programa para tratar com o conteúdo técnico da conferência e um Comitê de Organização para tratar das matérias financeiras, preparação do local, convites e/ou publicidade. É claro

que estes comitês precisam trabalhar juntos, possuem necessidades comuns de informação e precisam manter registros de informação consistentes e atualizados.

2. Sistema de informação a ser projetado

O sistema de informação a ser projetado deve suportar as atividades dos Comitês de Programa e Organização vinculados à preparação de uma Conferência de Trabalho da IFIP. O envolvimento destes dois comitês é considerado análogo a duas entidades organizacionais em uma estrutura corporativa que usam informação em comum.

Devem ser suportadas as seguintes atividades dos comitês.

Comitê de Programa:

1. Preparar uma lista de pessoas para enviar a chamada de trabalhos.
2. Cadastrar as cartas de intenção em resposta à chamada.
3. Cadastrar os artigos de contribuição recebidos.
4. Distribuir os artigos entre aqueles que realizam a avaliação.
5. Coletar os relatórios dos avaliadores e selecionar os artigos para inclusão no programa.
6. Agrupar os artigos aceitos em sessões para apresentação e selecionar o moderador para cada sessão.

Comitê de Organização:

1. Preparar uma lista de pessoas que receberão convite para a conferência.
2. Imprimir convites prioritários para Representantes Nacionais, membros do Grupo de Trabalho e membros de grupos de trabalho associados.
3. Assegurar de que todos os autores dos artigos aceitos recebam um convite.
4. Assegurar de que os autores de artigos rejeitados recebam um convite.
5. Evitar o envio de convites duplicados para qualquer pessoa.
6. Registrar a aceitação dos convites.

7. Gerar a lista final dos participantes.

3. Fronteiras do sistema

Deve ser notado que os aspectos financeiros e de cronograma do trabalho do Comitê de Organização, a planificação de ambos os comitês, as acomodações em hotéis para os participantes e as matérias relativas à preparação de cópias dos anais tem sido omitidos deste exercício, porém uma submissão pode incluir algum ou todos estes aspectos adicionais se os autores sentem-se motivados.”

Esta definição do problema é a única fonte consultada para a aplicação da proposta de particionamento/agregação na AOO. Por isto é necessário acrescentar alguns supostos para garantir a completeza, consistência e a não ambigüidade do problema. Estes supostos são os seguintes:

Geral

- Dado que o tempo é um fator crítico na preparação da conferência, deve ser permitido o maior paralelismo possível entre as atividades das comissões, evitando estabelecer seqüências artificiais das mesmas.

Objetos e atributos

- A definição dos moderadores e avaliadores não é indicada explicitamente na definição do problema, porém é necessária sua inclusão para os efeitos de consistência dos modelos.
- O programa da conferência de trabalho é dividido em sessões agrupadas por assuntos. Estes mesmos assuntos de especialização são válidos para os artigos submetidos, como para os moderadores e avaliadores. Porém, dado que os assuntos não estão definidos explicitamente no texto do problema, não serão identificados como uma classe independente e relacionada com as sessões, moderadores, avaliadores e artigos, e ficarão como atributos.

Relacionamento entre objetos

- Entre as pessoas que recebem a chamada de trabalhos, existem alguns que não participam da conferência, portanto não recebem convite.
- Não todas as pessoas que recebem uma chamada de trabalhos enviam uma carta de intenção.
- Não todas as pessoas que receberam algum tipo de convite, enviam respostas a este convite.
- Não todos os moderadores disponíveis necessariamente moderam alguma sessão.
- Não todos os avaliadores de trabalhos necessariamente avaliam algum artigo.

Valoração de atributos

- A carta de intenção como resposta à chamada de trabalhos pode indicar o desejo de participar ou não na conferência.
- A resposta ao convite recebido para a conferência pode ser uma aceitação do mesmo ou uma negativa.
- Os artigos que já foram aceitos ou rejeitados para participar da conferência não podem mais mudar o seu estado, isto é, um artigo aceito não poder ser rejeitado e vice-versa.

ANEXO 3 - PARTICIONAMENTO/AGREGAÇÃO PARA O PROBLEMA DA IFIP

Neste anexo é apresentado um modelo hierarquizado, conforme a proposta deste trabalho, para o problema da IFIP, segundo a descrição dada no anexo 2. Pelas próprias dimensões do problema, a hierarquia obtida não é nem muito larga (quantidade máxima de *clusters* nos níveis da hierarquia), nem muito profunda (número de níveis na hierarquia), porém é um bom exemplo para ilustrar os conceitos e notações em uma aplicação.

Ao aplicar o procedimento da estratégia *top-down* foi encontrado um único domínio: o domínio de aplicação do problema da IFIP. Os *clusters* candidatos identificados são: **Participante** (segundo o conceito geral), **Correspondência** (segundo o conceito de categoria) e **Programa** (segundo o conceito composto). Existem associações paralelas entre elas, o que permite definir associações agregadas. Os *clusters* foram assim particionados: **Participante** (Participante, Convidado Prioritário), **Correspondência** (Carta Intenção, Resposta Convite) e **Programa** (Sessão, Artigo, Artigo Aceito, Relatório).

Ao aplicar o procedimento da estratégia *bottom-up* não foi localizada classe dominante nenhuma. Os *clusters* iniciais foram confirmados, as hierarquias foram previamente refinadas, especialmente a do novo *cluster* **Pessoa**, ex-**Participante** que ficou **Pessoa** (Pessoa, Receptor Chamada Trabalhos, Convidado, Participante, Moderador, Autor, Avaliador, Convidado Prioritário, Membro Grupo, Representante Nacional). Os componentes do *cluster* **Programa** possuem um conflito para serem agregados por abstração: existe uma hierarquia de generalização/especialização e uma hierarquia de composição com a classe **Artigo Aceito** em comum, além disso não foi possível particioná-lo.

O resultado da combinação de ambas estratégias, incluindo a definição do *cluster*-sistema, é apresentado nas seguintes figuras. Por razões de compreensão, o modelo é apresentado em forma *top-down*:

- A figura 1 mostra o *cluster*-sistema **Conferência IFIP**, os atributos indicam os componentes: **Pessoa**, **Correspondência** e **Programa**.

- A figura 2 mostra o particionamento do *cluster*-sistema em seus componentes e suas associações agregadas, este *cluster* possui uma coesão de nível 4 (associação simples).
- A figura 3 mostra o particionamento do *cluster* **Pessoa**, corresponde a uma complexa hierarquia de generalização/especialização. A coesão deste *cluster* é de nível 2 (abstração por generalização/especialização).
- A figura 4 mostra o particionamento do *cluster* **Correspondência**, cuja coesão é de nível 6 (sem associação). Não parece útil generalizar as classes **Carta Intenção** e **Resposta Convite**, como foi sugerido anteriormente.
- A figura 5 mostra o particionamento do *cluster* **Programa**. Este *cluster* possui uma coesão de nível 2 (abstração por generalização/especialização e por composição). Como existem duas abstrações, sem existir nenhum tipo de dominância, foi mantido o nome do *cluster* como definido inicialmente.
- Finalmente, a figura 6 mostra o modelo plano do problema para fins de comparação.

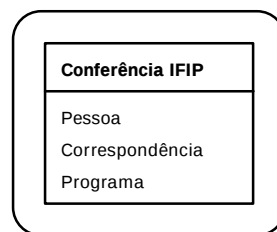


Figura 1 - Cluster-sistema Conferência IFIP

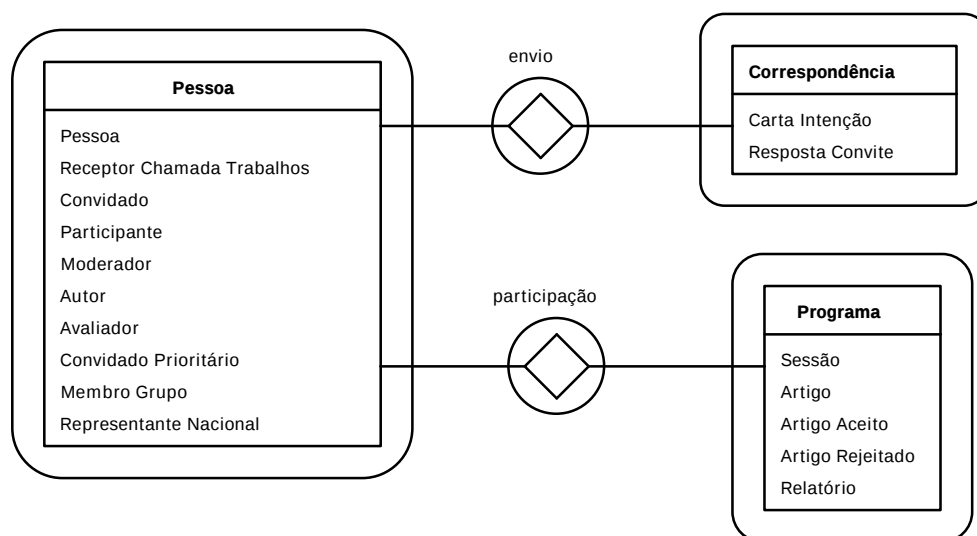


Figura 2 - Particionamento do *cluster*-sistema Conferência IFIP

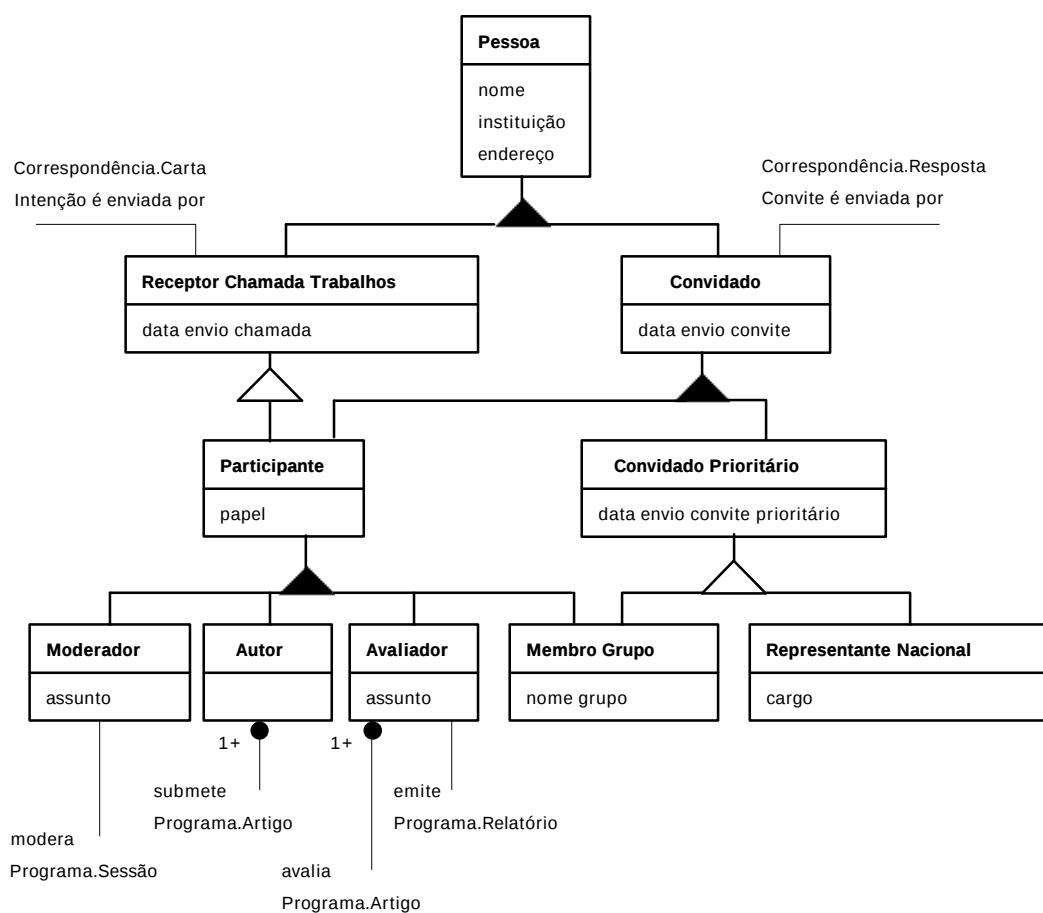


Figura 3 - Particionamento do *cluster* Pessoa

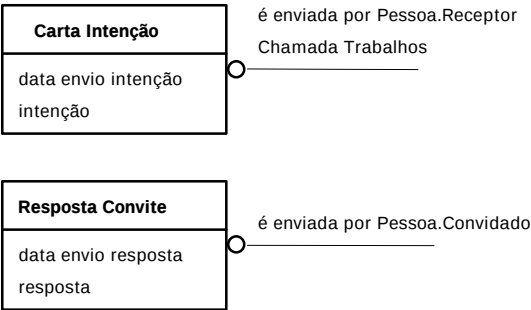


Figura 4 - Particionamento do *cluster* Correspondência

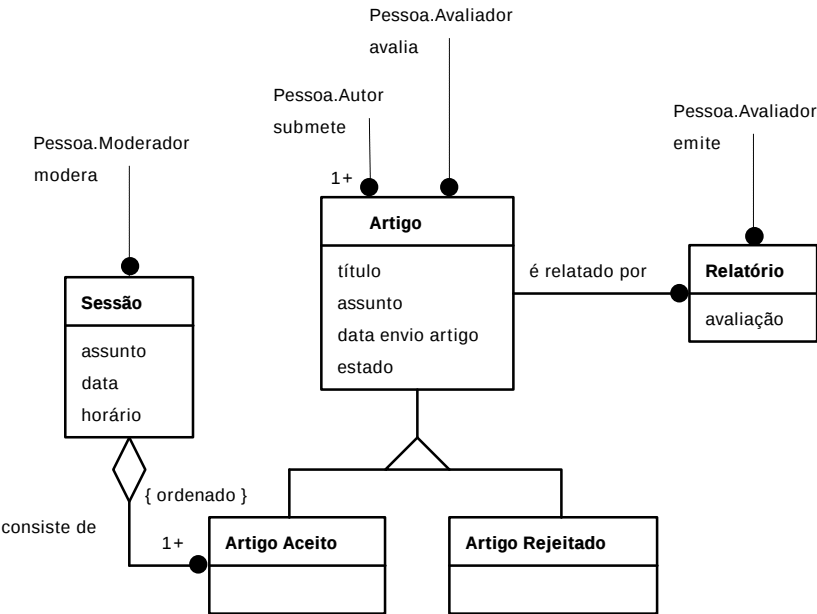


Figura 5 - Particionamento do *cluster* Programa

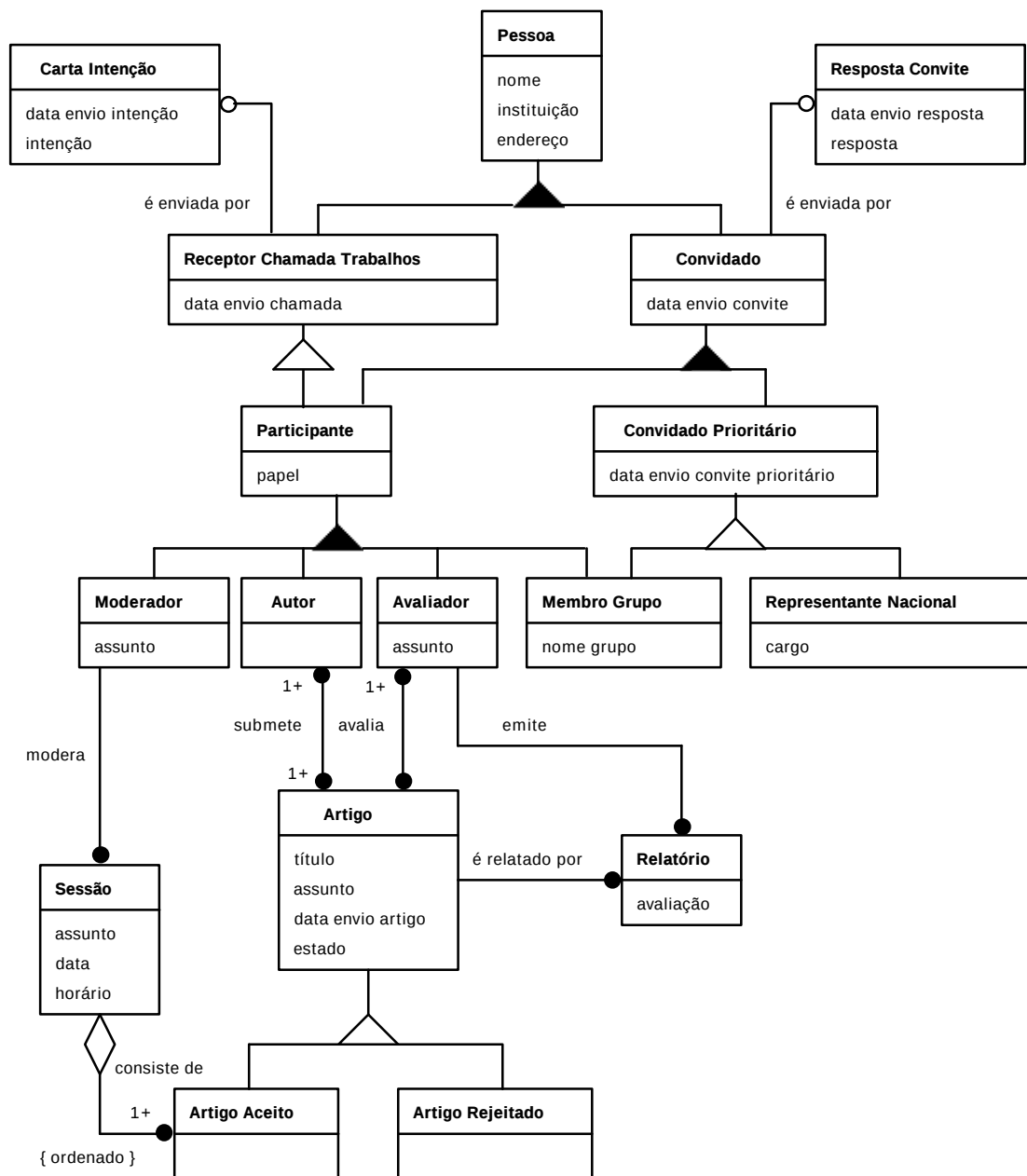


Figura 6 - Modelo plano do problema da IFIP

A seguir é mostrado o dicionário de dados, segundo a metodologia *OMT*, para este problema. A definição das classes ajuda a aumentar a semântica mostrada nos diagramas.

DICIONÁRIO DE DADOS DO PROBLEMA DA IFIP

Artigo - Um trabalho submetido à conferência por um ou mais autores.

Artigo Aceito - Um artigo submetido à conferência e aprovado pela Comissão de Programa para ser apresentado em alguma sessão.

Artigo Rejeitado - Um artigo submetido à conferência não aceito pela Comissão de Programa para ser apresentado.

Autor - Uma pessoa que submete um ou mais artigos à conferência.

Avaliador - Uma pessoa especialista em um assunto técnico que avalia artigos submetidos à conferência.

Carta Intenção - Uma carta que envia um receptor de chamada de trabalhos em que indica sua intenção ou não de participar da conferência.

Convidado - Uma pessoa que recebe um convite para a conferência. Pode ser um autor de artigo aceito ou rejeitado, avaliador de artigos, moderador de sessões, membro de grupo ou algum representante nacional.

Convidado Prioritário - Uma pessoa que recebe um convite prioritário para comparecer à conferência. Pode ser membro de grupos de trabalhos ou algum representante nacional.

Membro Grupo - Uma pessoa que faz parte de um grupo de trabalho da IFIP associado à conferência.

Moderador - Uma pessoa que modera as sessões que fazem parte do programa da conferência.

Receptor Chamada Trabalhos - Uma pessoa que trabalha na área de pesquisa da conferência de trabalho. Recebe a chamada de trabalhos porque é um potencial autor de um ou mais artigos.

Relatório - Uma avaliação realizada por algum avaliador sobre um artigo determinado.

Representante Nacional - Uma pessoa que desempenha algum cargo vinculado à área de pesquisa no país em que é realizada a conferência.

Resposta Convite - Uma carta enviada por uma pessoa que recebe um convite. Nela expressa a aceitação ou negativa para comparecer à conferência.

Sessão - Um agrupamento por assunto de artigos a ser apresentados na conferência. Possui data e horário específicos.

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. 470p.
- [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. 580p.
- [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.
- [CAR 90] CARLSON, C. Robert; JI, Wenguang. The Nested Entity-Relationship Model: A Pragmatic Approach to E-R Comprehension and Design Layout. In: LOCHOVSKY, F. H. (ed.) **Entity-Relationship Approach to Database Design and Querying**, Amsterdam: North-Holland, 1990, p.43-57.
- [COA 92] COAD, Peter; YOURDON, Edward. **Análise Baseada em Objetos**. Rio de Janeiro: Campus, 1992. 225p.

- [CON 89] CONSTANTINE, Larry. Object-Oriented and Structured Methods: Toward Integration. **American Programmer**, New York, v.2, n.7/8, Aug. 1989.

- [DAV 90] DAVIS, James; BONNELL, Ronald. Modeling Semantics with Concepts Abstraction in the EARL Data Model. In: LOCHOVSKY, F. H. (ed.) **Entity-Relationship Approach to Database Design and Querying**, Amsterdam: North-Holland, 1990, p.95-110.

- [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. 302p.

- [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.

- [HAR 87] HAREL, David. Statecharts: A Visual Formalism for Complex Systems. **Science of Computer Programming**, Amsterdam, v.8, n.3, Jun. 1987.

- [HEU 90] HEUSER, Carlos. **Modelagem Conceitual de Sistemas: Redes de Petri**. Kapelusz: Buenos Aires, 1990.

- [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.

- [IFI 82] IFIP WG 8.1 WORKING CONFERENCE ON COMPARATIVE REVIEW OF INFORMATION SYSTEMS DESIGN METHODOLOGIES, 1982, Noordwijkerhout. **Proceedings...** Amsterdam: North-Holland, 1982.

- [JAC 92] JACOBSON, Ivar; CHRISTERSON, Magnus; JONSSON, Patrik; ÖVERGAARD, Gunnar. **Object-Oriented Software Engineering: A Use Case Driven Approach**. Wokingham: Addison-Wesley, 1992. 528p.

- [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.

- [LOY 90] LOY, Patrick. A Comparison of Object-Oriented and Structured Development Methods. In: THAYER, Richard; DORFMAN, Merlin (eds.). **System and Software Requirements Engineering**, Los Alamitos: IEEE Computer Society Press, 1990.

- [MAT 89] MATTOS, Nelson. Abstraction Concepts: The Basis for Data and Knowledge Modeling. In: BATINI, C. (ed.) **Entity-Relationship Approach**, Amsterdam: North-Holland, 1989, p.473-492.

- [MCM 91] McMENAMIN, Stephen; PALMER, John. **Análise Essencial de Sistemas**. São Paulo: McGraw-Hill, 1991. 567p.

- [MON 92] MONARCHI, David; PUHR, Gretchen. A Research Typology for Object-Oriented Analysis and Design. **Communications of the ACM**, New York, v.35 n.9, p.35-47, Sep. 1992.

- [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.
- [POT 88] POTTER, Walter; TRUEBLOOD, Robert. Traditional, Semantic, and Hyper-Semantic Approaches to Data Modeling. **IEEE Computer**, Los Alamitos, v.21, n.6, p.53-63, Jun. 1988.
- [RUM 91] RUMBAUGH, James; BLAHA, Michael; PREMERLANI, William; EDDY, Frederick; LORENSEN, William. **Object-Oriented Modeling and Design**. Englewood Cliffs: Prentice-Hall, 1991. 500p.
- [SHL 90] SHLAER, Sally; MELLOR, Stephen. **Análise de Sistemas Orientada para Objetos**. São Paulo: McGraw-Hill, 1990. 178p.
- [SHL 92] SHLAER, Sally; MELLOR, Stephen. **Object Life Cycles: Modeling the World in States**. Englewood Cliffs: Prentice-Hall (Yourdon Press), 1992. 251 p.
- [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.
- [YOU 90] YOURDON, Edward. **Análise Estruturada Moderna**. Rio de Janeiro: Campus, 1990. 836p.