

**ESTUDO COMPARATIVO ENTRE AS
NOTAÇÕES ADOTADAS EM DIFERENTES
MODELOS DE DADOS PARA
REPRESENTAÇÃO DE ABSTRAÇÕES E
CONSTRUTORES SEMÂNTICOS.**

**MYRIAN RENATA BARROS ARAUJO
CAETANO TRAINA JR.**

Nº 74

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
Mai./1998

SYSNO	<u>967962</u>
DATA	<u> / / </u>
ICMC - SBAB	

Instituto de Ciências Matemáticas de São Carlos
Universidade de São Paulo
São Carlos

Estudo Comparativo entre as Notações Adotadas
em Diferentes Modelos de Dados para
Representação de Abstrações e
Construtores Semânticos

Myrian Renata Barros Araujo
e_mail: mrenata@icmssc.sc.usp.br

Caetano Traina Jr.
e_mail: caetano@icmssc.sc.usp.br

ICMSC - USP
13560 - 970

Caixa Postal 668
São Carlos SP

Maio de 1998

Sumário

Capítulo 1

Introdução

1.1 Considerações iniciais	5
1.2 Organização do trabalho.....	5

Capítulo 2

Modelos Considerados

2.1 Introdução	6
2.2 Modelo Entidade - Relacionamento (ME-R).....	6
2.3 OOA e OOD de Coad/Yourdon	7
2.4 OMT	8
2.5 Fusion.....	8
2.6 SIRIUS	9

Capítulo 3

Comparação Entre os Elementos Básicos dos Modelos

3.1 Introdução	10
3.2 Descrição das Tabelas	10
3.2.1 Entidade/Objeto e Conjunto/Classe	10
3.2.2 Associação entre Elementos	13
3.2.3 Tipos de Atributos	16
3.2.4 Cardinalidade / Multiplicidade	21
3.2.5 Abstrações	23
3.3 Considerações finais	28

Referências Bibliográficas.....	29
---------------------------------	----

Lista de tabelas

Tabela 1) Elementos que representam “alguma coisa” ou um “conjunto de coisas.....	12
Tabela 2) Associações entre elementos.....	14
Tabela 3) Tipos de atributos existentes nos modelos estudados.....	18
Tabela 4) Representação de cardinalidade / multiplicidade.....	22
Tabela 5) Abstrações existentes nos modelos.....	25

Capítulo 1

Introdução

1.1 Considerações iniciais

Modelos de dados, de maneira geral, possuem elementos de modelagem e construtores semânticos similares, tanto conceitualmente como notacionalmente. No entanto, muitas vezes a representação de tais elementos e construtores semânticos não é clara e nem única, dificultando a compreensão quando estudam-se mais de um modelo. Para facilitar o estudo de vários modelos, este relatório tem como objetivo ilustrar e comparar as representações gráficas utilizadas nos elementos que compõem alguns dos mais conhecidos modelos, bem como seus construtores semânticos.

1.2 Organização do Trabalho

Este trabalho está organizado em três capítulos. No primeiro tem-se uma introdução geral do trabalho aqui apresentado. O segundo capítulo apresenta uma breve descrição dos modelos de dados que são abordados no texto. E finalmente o terceiro capítulo tem como objetivo apresentar uma comparação notacional entre os modelos estudados. Para facilitar, um conjunto de cinco tabelas é apresentada no final do capítulo 1. Cada tabela apresenta um conjunto de elementos fundamentais para os modelos. A tabela cinco, em particular, ilustra as abstrações que os modelos suportam.

Capítulo 2

Modelos Considerados

2.1 Introdução

Este capítulo compõe-se de uma breve introdução aos modelos de dados aqui apresentados. Modelos como ME-R, ME-RX, OOA, OMT, Fusion e o modelo SIRIUS.

2.2 Modelo Entidade - Relacionamento (ME-R)

Este modelo foi proposto, nos anos 70, por Peter Chen [Chen_76] com objetivo de representar informações básicas, tanto sintaticamente como semanticamente, sobre uma realidade. Para que isto seja possível, este baseia-se na adoção de uma visão natural do mundo real que pode ser descrita através de *Conjuntos de Entidades e Relacionamentos*. Uma característica importante é que o ME-R [Elmasri_94] é intuitivo, o que é facilitado pela representação gráfica das informações modeladas.

O ME-R rapidamente teve grande aceitação de uso e desde então tem sido bastante explorado. Com as novas aplicações que foram sendo desenvolvidas, como bases de dados para projeto de engenharia (CAD/ CAM), bases de dados para armazenar imagens e gráficos, aplicações multimídia entre outras, novos elementos foram necessários para suportá-las. Assim novos conceitos “semânticos” de modelagem foram adicionados ao ME-R, como: *especialização, generalização*, entre outros, ou seja, o ME-R foi estendido, sendo denominado ME-RX (Modelo Entidade - Relacionamento Estendido) [Elmasri_94] e [Traina_96].

2.3 OOA e OOD de Coad/Yourdon

A OOP (Object Oriented Programming - Programação Orientada a Objeto) baseada em objetos foi discutida pela primeira vez no final dos anos 70 por pessoas que trabalhavam com a linguagem SIMULA [Dahl_70]. Nessa época, ela era uma parte importante da linguagem Smaltalk [Ingalls_78] desenvolvida na Xerox PARC. Enquanto isso, o resto do mundo usava linguagens tais como COBOL [ANSI_78] e [Sammet_69] e FORTRAN [ANSI_78], [Rosen_67] e [Sammet_69], e métodos de decomposição funcional para o tratamento de problemas de projeto e implementação. Havia pouca discussão sobre OOD (Object Oriented Design - Projeto Orientado a Objetos), e virtualmente nenhuma discussão sobre OOA (Object Oriented Analysis - Análise Orientada a Objetos).

Algumas alterações ocorreram na década passada, tornando-se nos anos noventa, fatores importantes, tais como: os conceitos básicos de um enfoque baseado em objetos que tiveram uma década para amadurecimento, e a atenção mudou gradualmente de considerações sobre codificação, para considerações sobre projeto e análise. Note que era difícil pensar em codificação orientada a objetos quando as linguagens disponíveis eram COBOL, FORTRAN ou C [Kernighan_78], ficando essa tarefa mais fácil com as linguagens Smaltalk e Ada [Wegner_80]. Outro aspecto, é que os atuais sistemas interativos e on-line dedicam muito mais atenção à *interface com o usuário* do que os antigos sistemas de processamento batch baseados em textos. Bill Joy, da Sun Microsystems, argumenta que aproximadamente 75% da codificação em um sistema moderno pode estar relacionadas à interface com o usuário. Segundo Coad/Yourdon [Coad_93] e [Coad_96], a experiência mostra que um enfoque baseado em tais sistemas, desde a análise até o projeto e a implementação, é uma forma mais natural de abordar tais sistemas voltados para usuários. Finalmente, a complexidade funcional é menos considerada agora do que antigamente, tornando a modelagem de dados prioritária.

Os objetivos fundamentais da OOA são: a compreensão do domínio do problema; o desenvolvimento de uma comunicação interpessoal; e a evolução contínua e reutilização. Por outro lado, os objetivos fundamentais do OOD são: melhorar a produtividade; aumentar a qualidade e elevar a manutenibilidade.

2.4 OMT

O método OMT denominado Object Modeling Technique (Técnica de Modelagem de Objetos) foi criado por Rumbaugh e outros [Rumbaugh_91] no início dos anos noventa, sendo um dos métodos mais completos desenvolvidos até então. Ele serve como divisor de águas, pois a partir deste modelo houve uma evolução mais concreta dos modelos orientado a objetos.

O método OMT estende desde a análise até o projeto e a implementação. Este método reuniu os conceitos importantes para uma modelagem orientada a objetos, introduzindo uma notação gráfica intuitiva que é utilizada ao longo de todo o desenvolvimento do modelo, de modo que a informação adicionada em uma etapa do desenvolvimento não é necessariamente perdida na(s) etapa(s) seguinte(s). Este modelo serviu de base para a criação de outros modelos, como Fusion .

2.5 Fusion

O método para análise e projeto orientado a objetos Fusion foi criado por Coleman e outros [Coleman_94] para tentar reduzir ou eliminar algumas deficiências que os métodos orientados a objetos publicados anteriormente apresentam, tais como os modelos, OMT [Rumbaugh_91] e Booch [Booch_94]. Muitos daqueles métodos surgiram a partir de métodos tradicionais de visão “funcional” e são considerados, segundo os autores, como a “primeira geração” dos métodos orientado a objetos. O desenvolvimento orientado a objetos tenta explorar os aspectos intuitivos durante as interações com o mundo real, uma vez que os objetos do software orientado a objetos estão ligados aos objetos do mundo físico.

O método Fusion integra e estende abordagens existentes, oferecendo uma notação consistente de processo de desenvolvimento, que inicia no processo de análise e segue até à implementação, através de uma notação específica para cada um de seus diferentes modelos. O nome Fusion (em português significa: fusão) indica que o método foi criado como uma fusão a partir de notações e conceitos propostos por outros métodos, estendendo e integrando essas abordagens numa nova geração.

2.6 SIRIUS

O modelo SIRIUS é um Modelo de Dados Orientado a Objetos baseado em Abstrações de Dados [Biajiz_96], ou seja, os conceitos do modelo encontram-se organizados semanticamente suportando tais abstrações de dados fundamentais que são: abstração de classificação, de generalização e de associação. Onde a abstração de generalização é denominada generalização ampla para expressar sua forma mais abstrata. Esta por sua vez é especializada em duas outras abstrações, que são: abstração de generalização restrita e a categorização. Além da abstração de generalização ampla, a abstração de associação também é especializada em outras duas abstrações que são: a abstração de agregação e a de composição. Seu desenvolvimento utiliza como ponto de partida um metamodelo que permite representar de maneira uniforme os elementos essenciais de qualquer modelo de dados orientado a objetos. Elementos como: objetos, atributos, característica de atributos, tipos de objetos entre outros.

O desenvolvimento do modelo SIRIUS teve como objetivo atender as necessidades de suporte à construção de software para ambientes de projetos de engenharia como CAD [Ishikawa_93] e [Kempe_95], ambientes estes que necessitam representar informações que requerem uma alta complexidade estrutural de dados como, por exemplo: imagens, documentos, dados temporais, etc. Enfim, SIRIUS possui recursos para que modelar sistemas não convencionais.

Capítulo 3

Comparação Entre os Elementos Básicos dos Modelos

3.1 Introdução

Esta seção tem como objetivo efetuar uma comparação dos elementos conceituais básicos dos modelos apresentados anteriormente. Este capítulo inclui cinco tabelas, onde cada tabela aborda um tipo de conceito básico, avaliando sua notação em cada modelo. Em cada tabela apresenta-se um texto que ajuda a ressaltar os pontos em comum entre os modelos, bem como as distinções e particularidades de cada modelo.

3.2 Descrição das Tabelas

3.2.1 Entidade/Objeto e Conjunto/Classe

A **Tabela 1)** apresenta os elementos que representam “alguma coisa”, ou seja, elementos representáveis do mundo real. Estes podem ser representados por entidades (quando considera-se Modelos Semânticos) e/ou objetos (quando considera-se Modelos Orientados a Objetos). Conceitualmente, entidades e objetos são similares. No entanto, a informação sobre uma entidade (como utilizada no ME-R e ME-RX) é fornecida restrita a sua estrutura, através dos atributos que estão associados a esta. Já o conceito de objetos dos demais modelos, além de fornecerem informações estruturais sobre os objetos através dos atributos, também oferecem informações a respeito do comportamento do objeto, através das operações (métodos ou serviços) que estão associadas ao objeto. Nos modelos OOA, OMT e Fusion, estas operações são mostradas separadamente, o que não ocorre em SIRIUS, onde o comportamento está “embutido” nos atributos das divisões dinâmica, sendo expresso através das características de atributos:

- *regras* que são atributos que possuem uma condição e uma ação. A condição permite que a ação

seja executada sempre que for feita alguma operação do SGBD sobre o objeto, ou quando for recebida alguma mensagem pelo objeto, oriunda da execução de outra ação ou procedimento na base de dados;

- *procedimento* são atividades que devem ser exercidas de uma forma pré-determinada pelo sistema, solicitadas pelos aplicativos para execução imediata ou postergada.

Estes atributos apresentam os aspectos dinâmicos do modelo, ou seja, as operações ou métodos, fornecendo informações a respeito do comportamento do objeto modelado.

Nessa tabela também é ilustrado o conceito de classe, o qual corresponde a conjuntos de objetos que possuem característica e comportamento comuns. Em OMT, os atributos e as operações são definidas sobre uma classe, sendo portanto válidos para uma instância de objeto. Em OOA uma classe é representada por um conjunto uniforme de atributos e serviços, onde o serviço é um comportamento específico que um objeto deve exibir, indicando qual comportamento será oferecido por um objeto em uma classe. No modelo Fusion, a representação de uma classe se dá através do nome da classe e os atributos que caracterizam os objetos pertencentes a esta classe. Em SIRIUS, a representação de classe é diferente, pois esta é representada através de tipos de objetos, onde o nome do tipo é escrito na parte superior à esquerda, na representação do objeto.

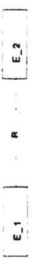
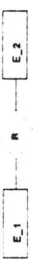
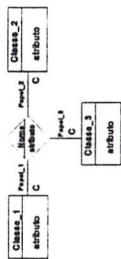
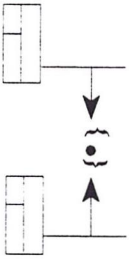
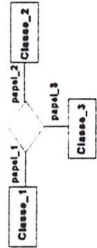
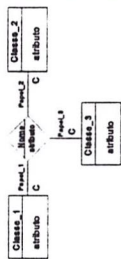
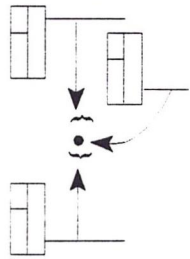
	ME-R	ME-RX	OOA	OMT	Fusion	SIRIUS
Rel. Binário/Associação						
Rel./As. Ternário						

Tabela 2) Associações entre elementos

3.2.2 Associação entre Elementos

Na **Tabela 2** encontra-se a representação de associação (ou relacionamento) entre os elementos vistos na **Tabela 1**.

Basicamente, os relacionamentos entre os modelos estudados são muito similares, podendo diferenciar, eventualmente, no nome utilizado. Por exemplo, os modelos MER, ME-RX e SIRIUS utilizam o termo relacionamento, já os modelos OOA e OMT empregam o termo associação, mas ambos os termos representam os mesmos conceitos. Pode-se ressaltar também algumas peculiaridades de um modelo, como por exemplo, o método OMT refere-se ao nome ligação para expressar a associação entre instâncias de objetos, e esta ligação é representada de modo análogo à associação entre classes. Já o modelo SIRIUS, representa instâncias do conjunto de relacionamentos entre objetos através de duas chaves ({ }) envolvendo o objeto primário (simbolizado pelo círculo). SIRIUS possui também a particularidade de representar o conceito de relacionamento triplo, um tipo especial de relacionamento onde a palavra “triplo” não provém do número de objetos participantes, mas sim por identificar três maneiras de um objeto poder participar do relacionamento [Biajiz_92].

Os modelos têm em comum o fato de poderem representar relacionamentos ternários, relacionamentos quaternários, etc (exceto OOA, pois este não suporta esta forma de relacionamento). O método OMT representa estes relacionamentos acrescentando um losango que conecta todas as classes envolvidas no relacionamento. Os demais modelos não lançam mão de notações adicionais para representar esta forma de relacionamento, apenas conectam os elementos que participam do relacionamento ao símbolo de relacionamento. Em SIRIUS, quando existe um relacionamento onde mais de dois objetos estão envolvidos, significa que mais do que dois objetos possuem atributos com características de relacionamento, e que se associam via o objeto primário (objeto que representa o relacionamento e que é indicado pelo círculo). Outra característica comum entre os modelos é que todos, com exceção de OOA, podem apresentar atributos de relacionamentos. Geralmente são apresentados de modo análogo à representação de atributos de objetos (ou entidades), porém neste caso, estes atributos representam características do relacionamento.

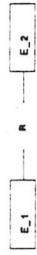
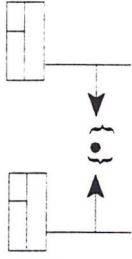
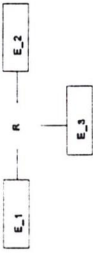
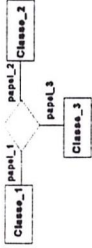
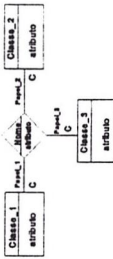
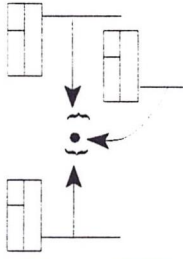
	ME-R	ME-RX	OOA	OMT	Fusion	SIRIUS
Rel. Binário/Associação						
Rel./As. Ternário						

Tabela 2) Associações entre elementos

3.2.3 Tipos de Atributos

A **Tabela 3** contém alguns tipos de atributos encontrados nos modelos, como os atributos monovalorados que estão presentes em todos os modelos e são representados de modo análogo pela maioria deles; diferindo no ME-R e ME-RX, onde estes atributos são representados através de elipses, e em SIRIUS, onde são indicados por setas dispostas ortogonalmente a uma reta vertical que se origina no objeto.

Os atributos multivalorados e compostos são tratados na fase de modelagem apenas pelos modelos ME-R, ME-RX e SIRIUS. Os demais modelos tratam estes atributos nas fases posteriores a análise (geralmente na fase de projeto), pois consideram que a análise deve ser efetuada em um alto nível de abstração, onde “detalhes” são inseridos só depois que a modelagem for considerada satisfatória.

O ME-R e ME-RX não fazem restrições quanto ao uso de atributos multivalorados e compostos, no entanto, ambos não possuem uma notação padrão, e por isto tais atributos tem sido representados de varias maneiras por diversos autores [Traina_96]. É interessante notar que em um diagrama de Entidade - Relacionamento, nem sempre indicam-se os elementos que compõem um atributo composto, e quando estes são indicados explicitamente, podem ser representados por uma das maneiras ilustradas na **Tabela 3**. Atributos multivalorados são representados nos diagramas através de uma elipse dupla ou uma barra após o nome do atributo, ou ainda, indicando-se através de uma linha dupla.

SIRIUS trata atributos multivalorados distintamente e ainda subdivide estes atributos em:

- *listas*, que corresponde a uma seqüência de elementos (no caso uma seqüência de atributos), onde podem ser definidas operações próprias de conjuntos, tais como união e intersecção ou ordenação e tratamento de repetição de valores.
- *conjuntos*, onde são definidas operações como união e intersecção, mas não ordenação. Neste caso não admite-se repetição de elementos.
- *vetores*, onde estes representam estruturas de arranjos de dados unidirecionais indexado onde estes dados formam um conjunto finito e ordenado de elementos.

Um atributo composto em SIRIUS é representado por uma tupla, formada pela agregação de mais do que um atributo. Existe também outro tipo especial de atributo que é formado pela agregação de atributos. o chamado atributo de atributo, formado por um atributo “base” e seus sub

atributos estes não sendo tão relevante para a modelagem a ponto de serem representados individualmente. A representação de atributos que possuem esta característica também podem ser encontrada na **Tabela 3**.

Os atributos de relacionamento são tratados de modo análogo aos atributos (de entidades e/ou objetos) pelos modelos, ME-R, ME-RX, Fusion e SIRIUS. Este tipo de atributo não é suportado por OOA. O modelo OMT possui uma categoria de atributo que é o atributo de ligação, onde é tratado de modo análogo ao atributo de objeto e classes.

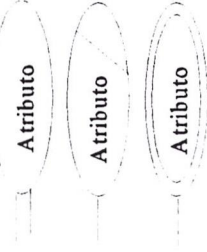
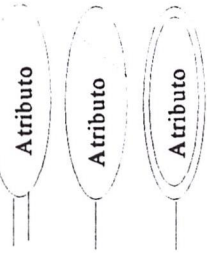
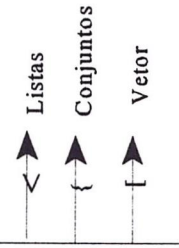
	ME-R	ME-RX	OOA	OMT	Fusion	SIRIUS
Atributo			atributos são representados por escrito, na segunda região de objetos e classe	atributos são representados por escrito, na segunda região de objetos e classe	atributos são representados por escrito, na segunda região de objetos e classe	
Atr. multivalorado						

Tabela 3) Tipos de atributos existentes nos modelos estudados

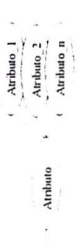
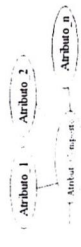
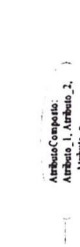
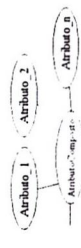
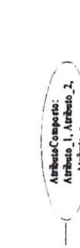
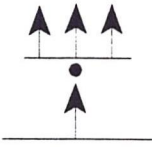
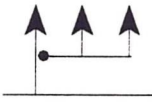
	ME-R	ME-RX	OOA	OMT	Fusion	SIRIUS
Atr. composto	  	  				
Atr. de atributo						

Tabela 3) Tipos de atributos existentes nos modelos estudados (continuação)

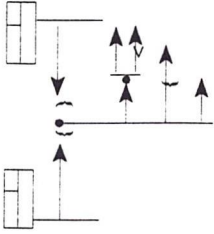
	ME-R	ME-RX	OOA	OMT	Fusion	SIRIUS
Atr. de relacionamento						

Tabela 3) Tipos de atributos existentes nos modelos estudados (continuação)

3.2.4 Cardinalidade/Multiplicidade

A **Tabela 4** ilustra como são utilizados os conceitos de *cardinalidade* (determina quantas vezes no mínimo uma entidade ou objeto pode ocorrer em um relacionamento), e *multiplicidade* (ou *participação*) (determina um intervalo mínimo e máximo de ocorrências de um objeto em um relacionamento). De modo geral não existe uma distinção precisa de qual conceito um determinado modelo está utilizando, porque alguns modelos misturam tais conceitos.

Os modelos ME-R, ME-RX e Fusion empregam o termo cardinalidade. No entanto analisando-se mais atentamente, nota-se que Fusion considera também intervalos numéricos, que a rigor deveria ser classificado como multiplicidade. Por outro lado, o modelo OMT utiliza o termo multiplicidade, mesmo quando se refere à ocorrência de apenas um objeto em um relacionamento. SIRIUS utiliza o conceito de multiplicidade para indicar os limites mínimo e máximo de valores instanciados que um determinado atributo pode assumir, não empregando-se o conceito de cardinalidade.

Observa-se que apesar de não existir sempre uma coerência formal entre o conceito (cardinalidade/multiplicidade) e o nome utilizado para este, os modelos indicam a (as) quantidade (s) de ocorrência entre os elementos que estão envolvidos em um certo relacionamento, sem causar impacto para a modelagem.

	ME-R	ME-RX	OOA	OMT	Fusion	SIRIUS
Cardinalidade/Multiplicidade			N 0, 1 0, N N, M	exatamente um zero ou muitos zero ou um um ou mais 1-2, 4 especificação numérica	Zero ou mais (default) Zero ou mais Um ou mais Valor Numérico Intervalo Numérico	(0-1) (0-N) (1-1) (N-M)

Tabela 4) Representação de cardinalidade / multiplicidade

3.2.5 Abstrações

A **Tabela 5** ilustra os tipos de abstração que um modelo pode conter. O ME-R suporta implicitamente alguns tipos de abstrações como: abstração de classificação, que pode ocorrer através do conjunto de entidades e de relacionamento e abstração de agregação, que pode ocorrer através do conjunto de relacionamento. No entanto, não existe uma notação para representar estas abstrações neste modelo.

A Abstração de generalização é análoga conceitualmente para cada modelo, no entanto, sua representação notacional pode diferir de modelo para modelo.

Basicamente, todos os modelos representam esta abstração através de uma classe (ou entidade) genérica, a qual é especializada através de um critério de generalização. Este critério indica a forma como irá ocorrer a especialização, unindo a classe genérica às classes (ou entidades) especializadas. Este critério possui uma notação distinta para cada modelo, onde a principal diferença se dá nas restrições colocadas, onde estas dependem da semântica, e do que se pretende representar na modelagem.

Os modelos OOA e OMT não possuem restrições associadas à abstração de generalização. Fusion indica a restrição de sobreposição, ou seja, quando uma superclasse pode ser formada apenas por uma subclasse (disjuntas representado pelo triângulo cheio), ou quando uma superclasse pode ser formada por mais de uma subclasse (não disjunta representada por um triângulo vazio). ME-RX possui duas restrições, exclusão mútua/sobreposição e especialização total/parcial. Na exclusão mútua, a intersecção de entidades detalhe é vazia (disjuntas). Esta restrição é indicada por dois arcos que cruzam os segmentos de retas que unem as entidades detalhe ao critério de generalização. Na sobreposição, a intersecção de entidades detalhe pode não ser vazia (não disjuntos). Na especialização total, a união das entidades detalhe forma a entidade abstrata, o que é representado por uma seta “grossa” que une o critério de generalização à entidade abstrata. Na especialização parcial, a união de entidades detalhe é diferente da entidade abstrata [Traina_96].

SIRIUS trata a abstração de generalização (generalização ampla) especializando-a na abstração de generalização restrita e categorização. Porém estas duas últimas abstrações “herdam” os conceitos da generalização ampla, e o critério de generalização é representado de maneira distinta para cada abstração. Na generalização restrita (ilustrada na tabela 3 mas sendo referenciada apenas por generalização) o critério é representado através de um trapézio dividido ao meio onde na parte superior têm-se as restrições e na parte inferior têm-se as restrições e na parte inferior o nome do

critério utilizado na especialização. Na categorização o critério de generalização é representado através de um trapézio invertido onde na parte superior têm-se as restrições e na parte inferior o nome do critério.

Tanto a generalização restrita quanto a categorização possuem dois tipos de restrições que são:

- *disjunção* (D) ou *sobreposição* (S), a qual é conceitualmente análoga à restrição de exclusão mútua/sobreposição do ME-RX.
- *participação total* (T) ou *participação parcial* (P), análoga à restrição de especialização total/parcial do ME-RX

As restrições de SIRIUS são indicadas na primeira linha do trapézio (generalização restrita) ou trapézio invertido (categorização) no centro do diagrama da abstração de generalização (generalização ampla).

Abstração de agregação corresponde à idéia de que elementos de modelagem podem associar-se, formando outros objetos que representam essa associação. Estes objetos correspondem ao todo e os objetos que formam o “todo” são considerados parte do “todo”. Esta abstração pode ser representada por uma entidade agregação que é formada por entidades componentes, por triângulos que indicam quais Classes-&-Objetos são parte da Classe-&-Objeto “toda”, por losangos indicando qual classe-parte forma uma classe-todo e por classe agregada formada por outras classes que estão associadas à classe agregada, para os modelos ME-RX, OOA, OMT e Fusion respectivamente. A agregação em SIRIUS é diferente dos demais modelos, pois em SIRIUS uma agregação é representada através de: tuplas, relacionamentos e atributos de atributos.

Abstração de Classificação, tem como objetivo “classificar” os elementos utilizados em uma modelagem. Isto é feito no ME-R e ME-RX implicitamente, pois um conjunto de entidades “classifica” uma determinada entidade. Analogamente para o OOA, OMT e Fusion, esta abstração se dá através do conceito de classe de objeto. Em SIRIUS, a abstração de classificação ocorre através de classificação/instanciação de objetos.

Abstração de Composição não é suportada pelos modelos OOA, OMT e Fusion. No ME-RX, esta abstração corresponde à idéia de que os elementos de modelagem podem compor outros elementos, formando objetos complexos (compostos). Em SIRIUS, a idéia da abstração de composição é análoga. porém. utiliza o conceito de colônia para ilustrar a abstração de composição

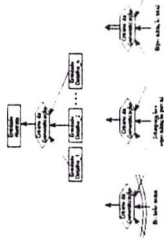
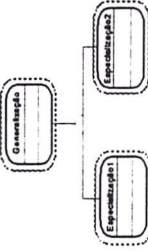
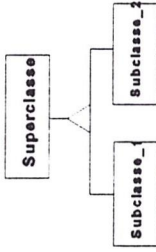
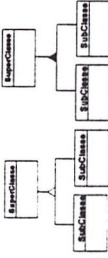
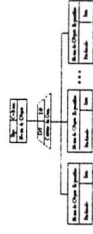
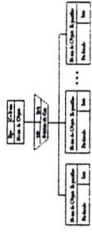
	ME-R	ME-RX	OOA	OMT	FUSION	SIRIUS
Generalização						
Categorização						

Tabela 5) Abstrações existentes nos modelos (continuação)

	ME-R	ME-RX	OOA	OMT	FUSION	SIRIUS
Agregação						A agregação em SIRIUS é representada através de: tuplas, relacionamentos e atributos de atributos
Classificação		Não é representado explicitamente. Implicitamente um conjunto de entidades “classifica” uma determinada entidade	Esta abstração se dá através do conceito de classe de objeto	Esta abstração se dá através do conceito de classe de objeto	Esta abstração se dá através do conceito de classe de objeto	Esta abstração se dá através de classificação

Tabela 5) Abstrações existentes nos modelos (continuação)

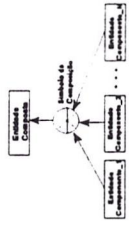
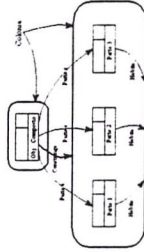
	ME-R	ME-RX	OOA	OMT	FUSION	SIRIUS
Composição						

Tabela 5) Abstrações existentes nos modelos (continuação)

3. 3 Considerações Finais

As notações aqui apresentadas não são únicas pois estas podem ser ilustradas de maneira distintas por mais de um autor. Um único elemento do modelo e/ou um construtor semântico pode apresentar algumas vezes mais de uma variação notacional.

No entanto, sabendo-se claramente qual(is) o(s) conceito(s) que deseja ser ilustrado em uma modelagem a representação notacional pode apresentar algumas pequenas variações não acarretando grandes problemas no entendimento da modelagem. A comparação notacional feita anteriormente serve também para que o leitor se acostume com as diversas representações, facilitando deste modo o emprego correto das mesmas.

Referências Bibliográficas

- [ANSI_74] American National Standard Programming Language COBOL (ANS X3.23-1974). New York: American National Standards Institute, 1974.
- [ANSI_78] American National Standard Programming Language FORTRAN (ANS X3.9-1978). New York: American National Standards Institute, 1978.
- [Biajiz_92] Biajiz, M. - "MRO - Uma Atualização do Modelo de Representação de Objetos e Uma Abordagem Formal", Dissertação de Mestrado em Ciências da Computação - ICMSC - USP, São Carlos, SP, julho de 1992.
- [Biajiz_96] Biajiz, M. - "Representação de Modelos de Dados Orientados a Objetos através de Parametrização de Abstrações", Tese de Doutorado, São Carlos - SP, Setembro de 1996.
- [Booch_94] Booch, G.; "Object-Oriented Analysis and Design with Applications", The Benjamin/Cummings Publishing Company, 1994.
- [Chen_76] Chen, Peter P. S.; "The Entity-Relationship Model - Toward a Unified View of Data", ACM Transactions on Database Systems, vol.1, nº1, pag. 9 - 36, March 1976.
- [Coad_93] Coad, P.; Yourdon, E. - "Projeto baseado em objetos", Editora Campus, 1993.
- [Coad_96] Coad, P.; Yourdon, E. - "Análise baseada em objetos", Editora Campus, 1996.
- [Colemanl_94] Coleman, D.; et ali - "Object-Oriented Development The Fusion method", Prentice Hall Object-Oriented Series, 1994.
- [Dahl_70] Dahl, O. J. ; Myhrhaug, B. et alli - "SIMULA 67 Common Base Language" Publication

N. S - 22, Oslo: Norwegian Computing Center, Out., 1970.

[Elmasri_94] Elmasri, R.; Navathe, S. B. - "Fundamentals of Database Systems", Addison-Wesley Publishing Company, 1994, 2ª Edição.

[Ingalls_78] Ingalls, D. H. "The Smaltalk-76 Programming System Design and Implementation" Conference Record of the 5th Annual ACM Symp. on Principles of Programming Languages, pp. 9 - 16, Jan. 1978.

[Ishikawa_93] Ishikawa, H. et alli - "The Model, Language, and Implementation of an Object-Oriented Multimedia Knowledge Base Management System", ACM Transactions on Database Systems, vol.18, n^o1, pp. 1 - 50, March 1995.

[Kempe_95] Kempe J. et alli - "Benchmarking Object-Oriented Database Systems for CAD*", Lectures Notes Computer Science, n^o978, September 1995.

[Kernighan_78] Kernighan, B. W. e Ritchie, D. M. - "The C Programming Language" Englewood Cliffs, N. J.: Prentice-Hall, 1978.

[Rosen_67] Rosen, S. - "Programming Systems and Languages", New York, McGraw-Hill, 1967.

[Rumbaugh_91] Rumbaugh, J. Et ali - "Object-Oriented Modeling and Design", PrenticeHall, 1991.

[Sammet_69] Sammet, J. E. - "Programming Languages: History and Fundamentals" Englewood Cliffs, N.J.: Prentice-Hall, 1969.

[Traina_96] Traina Jr. - "Sistemas de Banco de Dados Conceitos Gerais e Motivação", Notas de aula do curso de banco de dados.

[Wegner_80] Wegner, P. - "Programming with Ada: An Introduction by Means of Graduated Examples", Englewood Cliffs, N. J.: Prentice-Hall, 1980.