Uma formalização em IMC: o modelo DODM

Marcelo Soares Pimenta
Carlos Alberto Heuser
UFRGS/CPGCC-DI
Caixa Postal 1501
90001 Porto Alegre RS
e-mail: heuser@ufrgs.anrs.br (bitnet)

Resumo

O trabalho apresenta a descrição formal dos tipos de objetos do DODM, o modelo de dados do SGBD orientado à engenharia de software DAMOKLES, usando a abordagem de formalização IMC (*Information Modeling by Composition*) e a respectiva linguagem IMCL. A intenção é permitir uma melhor compreensão do DODM, bem como testar a aplicabilidade de IMC a casos reais.

Abstract

The object types of DODM, the data model of DAMOKLES, a non-conventional DBMS for software engineering environments, are formalized, by using the IMC (*Information Modeling by Composition*) approach and its language IMCL. The work aims to clarify the underlying concepts of DODM, as well as to test the practical usability of IMC.

Este projeto foi desenvolvido com o apoio financeiro do CNPq, da FINEP e da SID/Informática (Projeto ESTRA)

1 Introdução

Uma abordagem de modelagem de dados define como a informação pode ser representada e manipulada por um sistema de gerência de banco de dados (SGBD) e compreende:

- Os tipos de objetos atômicos e os construtores para objetos compostos
- Os operadores sobre os objetos de diversos tipos

Neste trabalho, é apresentada uma descrição formal dos tipos de objetos da abordagem DODM (*Design Object Data Model*), a abordagem de modelagem de dados do SGBD DAMOKLES. DAMOKLES (*DAtabase Management system Of KarLsruhe for Environments for Software engineering*) foi criado para suportar ambientes de engenharia de software. DODM é uma abordagem destinada a modelagem dos objetos que aparecem neste tipo de ambientes [Dittrich86]. Um resumo das características de DAMOKLES pode ser encontrado em [Pimenta89a].

A formalização é feita através da abordagem IMC (*Information Modeling by Composition*) e da respectiva linguagem IMCL. O objetivo da presente formalização é duplo:

- Deseja-se obter uma descrição clara, precisa e não ambígua dos modelos de dados criados com DODM, permitindo detectar partes incompletas, mal definidas ou inconsistentes da documentação existente. Por outro lado, tem-se, com a formalização, a base para a comparação de DODM com outras abordagens não convencionais de modelagem.
- Deseja-se adquirir experiência em casos reais com a ferramenta de formalização IMC.

2 A técnica de formalização IMC

A técnica de formalização IMC vem sendo desenvolvida desde o início da década de 70 [DurchholzRichter74], basicamente voltada para a formalização de abordagens de modelagem de dados [Richter81].

Esta técnica de formalização se distingue de outras de uso mais difundido (p.ex.: VDM [BjornerJones82]) por suportar o conceito de *ponto*, que corresponde a um objeto dentro de um contexto (objeto componente dentro de objeto composto). Exatamente por estar dotada deste conceito, a técnica IMC foi preferida para a formalização em questão, pois o conceito torna a abordagem particularmente atraente para a formalização de abordagens de modelagem não-convencionais, onde aparecem objetos com grandes e variados níveis de embutimento.

Em sua versão atual [DurchholzRichter88], IMC foi dotada de uma linguagem conhecida por IMCL. A aplicação mais importante de IMC/IMCL foi a formalização da estrutura de documentos para a proposta de norma *ISO Office Document Architecture (ODA)* [ISO88].

A linguagem IMCL é uma linguagem de lógica de predicados de primeira ordem. O universo de discurso de um modelo IMC é composto de:

- um universo não vazio de *entidades* dos seguintes tipos:
  - construções
  - pontos
  - conjuntos de pontos
  - a entidade UNDEF ("indefinido")
- *funções* do universo de entidades para o universo de entidades, i.é operadores sobre entidades do universo
- *relações* entre entidades do universo

Uma *construção* é um objeto que pode ser:

- um *átomo*, construção que não é composta de outras construções
- um *composto*, construção composta:
  - como um conjunto: uma *coleção*
  - como uma função: uma *nominação*
  - como uma seqüência: uma *catenação* (catenações não aparecem na formalização presente neste trabalho)

Note-se que a terminologia especial (coleção ao invés de conjunto, ...) é usada apenas para distinguir conjuntos de construções de conjuntos de entidades em geral.

A fim de permitir a referência a construções componentes dentro de uma construção composta, independentemente da forma de compósição, foi introduzido um conceito especial. Este conceito é o objeto conceitual que representa a idéia intuitiva de apontar para um diagrama de estrutura de informação e dizer "aqui", em frase como "este componente tem que ser excluído" ou "este componente deve ser substituído por outro objeto". O problema fundamental é que nem sempre o "aqui" é identificado unicamente pelo componente propriamente dito (p.ex.: em uma palavra, a mesma letra pode aparecer diversas vezes), mas sim pelo contexto em que o componente aparece (no caso da palavra, a posição da letra). Para tratar de forma abstrata com a idéia do "aqui" é usado em IMC o conceito de *ponto*.

O conceito de ponto permite que seja feita a distinção entre uma construção componente considerada e as suas diversas aparições dentro de uma construção composta. Considerando a cadeia de caractcres "data" (uma catenação) como exemplo, tem-se três construções componentes: 'd', 'a' e 't'. Enquanto 'd' e 't' aparecem cada uma em um ponto somente, 'a' aparece em dois pontos, na segunda e quarta posição da palavra. Assim "data" tem quatro pontos componentes, mas somente três construções componentes.

Uma construção fora de qualquer contexto é dita como estando no seu *ponto próprio*.

A linguagem IMCL provê uma notação textual para construções, para conjuntos de pontos, e para a entidade indefinida (a prática demonstrou que basta notação para conjuntos de

pontos, não sendo necessária notação especial para pontos individuais). Além da linguagem textual, existe uma linguagem gráfica, a *notação em caixas*, que representa construções e pontos de forma bastante intuitiva. Esta notação, entretanto, tem semântica precisa e pode ser convertida diretamente para a notação textual da linguagem.

Alguns exemplos são dados na figura 1 para introduzir a notação de caixas. Adicionalmente, para os átomos, é dada a notação textual correspondente.

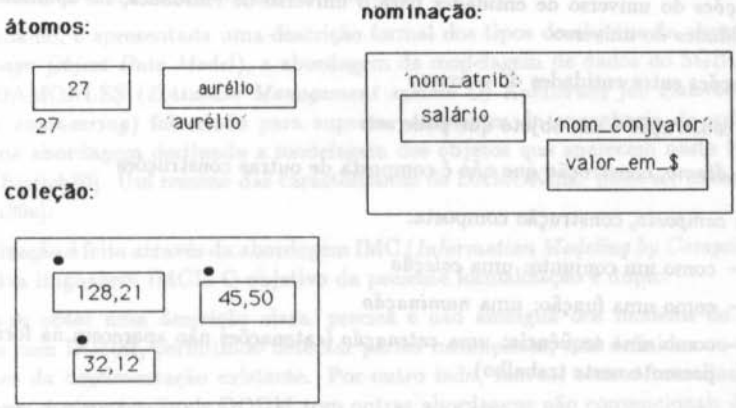

Figura 1: Exemplo da notação gráfica de IMCL

IMCL é uma linguagem de predicados de primeira ordem. Como tal, ela contém *fórmulas* e *termos* compostos da maneira usual. Ela possui todos os quantificadores e símbolos lógicos de uma linguagem de primeira ordem. Adicionalmente, possui os símbolos predicativos e funcionais específicos de IMC, dos quais são listados abaixo apenas alguns deles, os necessários à compreeensão desta formalização.

Fórmulas de IMCL

Abaixo estão algumas fórmulas com semântica pré-definida. O usuário de IMCL pode introduzir novos símbolos predicativos.

IsAtom(t) t é um átomo
IsNom(t) t é uma nominação
IsCol(t) t é uma coleção

$t_1 = t_2$ a entidade t_1 é igual a entidade t_2
$t_1 \neq t_2$ a entidade t_1 é diferente da entidade t_2

$t_1 \in t_2$ a construção t_1 é elemento da coleção t_2
$t_1 \notin t_2$ a construção t_1 não é elemento da coleção t_2
$t_1 \hat{\in} t_2$ a entidade t_1 é um conjunto unitário de pontos e está contida no conjunto de pontos t_2
$t_1 \subseteq t_2$ t_1 é subconjunto ou igual a t_2 (se aplica a coleções e conjuntos de pontos)

Termos de IMCL

Abaixo estão alguns termos com semântica pré-definida. É facultado ao usuário de IMCL introduzir novos símbolos funcionais. Se o argumento de uma função não for do tipo requerido, o resultado é UNDEF. Como não há na linguagem termos que designam pontos individuais, sempre que na descrição abaixo constar "o ponto t" leia-se "o conjunto unitário de pontos t".

C t Se t designa um ponto, então C t designa a *construção* no ponto t
N t Se t designa o ponto de um componente imediato de uma nominação, então N t designa o *nome* no ponto t
COLC t Se t designa um conjunto de pontos, então COLC t designa a coleção das construções nos pontos de t
NAMESET t Se t designa uma nominação, então NAMESET t designa a coleção dos nomes dos componentes da nominação

$\hat{}t$ se t designa uma construção, então $\hat{}t$ designa o *ponto próprio* da construção
$t.$ Se t designa um conjunto de pontos sem pontos atômicos, $t.$ designa o conjunto de todos os pontos de componentes imediatos de t

$[t_1; t_2; t_3]$ Se t_i designa uma construção, o termo designa a coleção de todas as construções t_i
$[\,]$ Designa a coleção vazia
$[n_1 : c_1; n_2 : c_2]$ Se n_i e c_i designam construções e todos n_i são diferentes, o termo designa a nominação de componentes c_i sob nomes n_i
$[:]$ Designa a nominação vazia

A figura 2 apresenta um exemplo de uma construção, no caso, uma nominação. Há três pontos componentes sob nomes 'A', 'B' e 'C'. A construção sob nome 'A' é uma coleção, cujas construções componentes são, por sua vez, nominações. Os símbolos $\triangle_i$ indicam pontos dentro da construção em questão. Considerando-se que a variável ref designa a construção referência da figura 2, o ponto $\triangle_4$ é designado em IMCL pelo seguinte termo:

$$\hat{}ref.\text{'A'}.\,<\,\text{C}\;xs.\text{'A'} = 12\,>.\text{'B'}$$

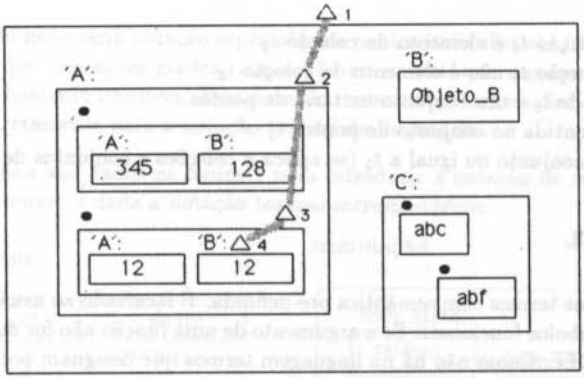

Figura 2: Exemplo de construção com pontos indicados(Δ_i)

Este termo pode ser entendido da seguinte forma:

$\hat{}ref$	o ponto próprio da construção referência (o pontoΔ_1)
$\hat{}ref.$	o conjunto dos (três) pontos componentes da construção referência (pontos componentes do ponto Δ_1)
$\hat{}ref.$'A'	o ponto do componente de nome 'A' (o ponto Δ_2)
$\hat{}ref.$'A'. $<$ C xs.'A' $= 12 >$	o ponto componente da coleção em Δ_2, que satisfaz a condição $<$ C xs.'A' $= 12 >$; xs é uma variável pré-definida, que referencia cada um dos pontos componentes de Δ_2; a condição especifica que devem ser selecionados os pontos que tem um componente 12 sob nome 'A' (o ponto Δ_3 é selecionado)
$\hat{}ref.$'A'. $<$ C xs.'A' $= 12 >$.'B'	o ponto do componente de nome 'B' dentro de Δ_3 (o ponto Δ_4)

Usando o termo que designa o ponto Δ_4 é possível escrever o termo que designa a *construção* neste ponto (a construção 12):

$$C\ \hat{}ref.\text{'A'}. < C\ xs.\text{'A'} = 12 > .\text{'B'}$$

e o *nome* neste ponto ('B'):

$$N\ \hat{}ref.\text{'A'}. < C\ xs.\text{'A'} = 12 > .\text{'B'}$$

3 Formalização do DODM

Por limitações de espaço, a formalização do modelo DODM aqui apresentada não é completa. Entretanto, a porção da formalização apresentada deve ser suficiente, tanto para demonstrar

a utilização de IMC na formalização de modelos de dados, quanto para apresentar de forma introdutória a abordagem DODM.

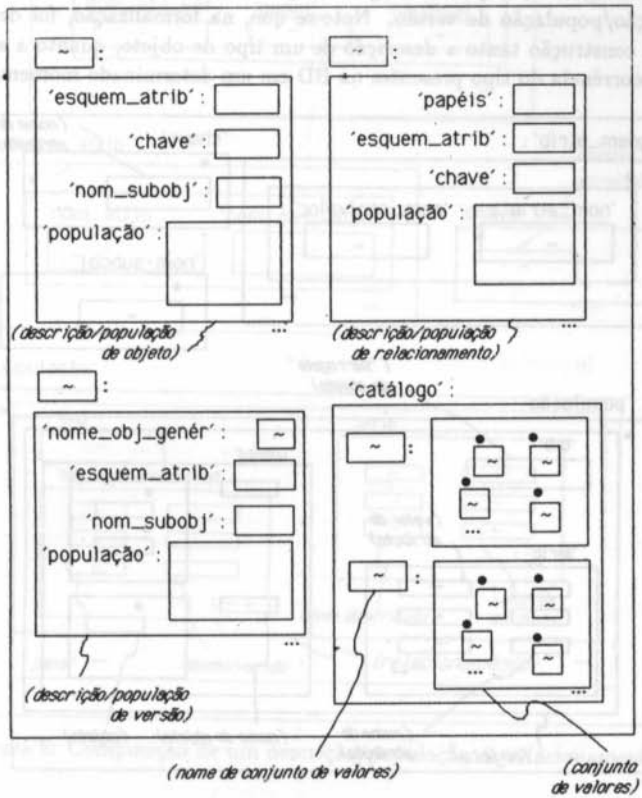

Figura 3: Composição (dois níveis) de um modelo DODM

A figura 3 é uma representação diagramática da composição das construções de tipo modelo DODM. Esta figura exibe apenas dois níveis de embutimento. Os átomos estão indicados através de um til. A ocorrência repetida (zero a n vezes) de subestruturas é indicada por três pontos. Como se verifica na figura, todo modelo DODM contém ao menos um componente sob nome 'catálogo', no qual estão indicados os diversos conjuntos de valores com o respectivo nome. Adicionalmente, um modelo DODM pode conter diversos componentes na função de descrição e população de *objeto*, de *relacionamento* e de *versão*. A palavra *descrição* é usada no sentido de "informações que descrevem (especificam) um tipo (um conjunto) de construções" e a palavra *população* é usada no sentido de "conjunto de construções de um tipo presentes em um contexto em um determinado momento".

A composição de uma construção do tipo descrição/população de objeto é apresentada na figura 4. Já a composição de uma construção do tipo descrição/população de relacionamento é apresentada na figura 5, enquanto a figura 6 apresenta a composição de uma construção do tipo descrição/população de versão. Note-se que, na formalização, foi decidido juntar em uma única construção tanto a *descrição* de um tipo de objeto, quanto a sua *população* (conjunto de ocorrência do tipo presentes na BD em um determinado momento).

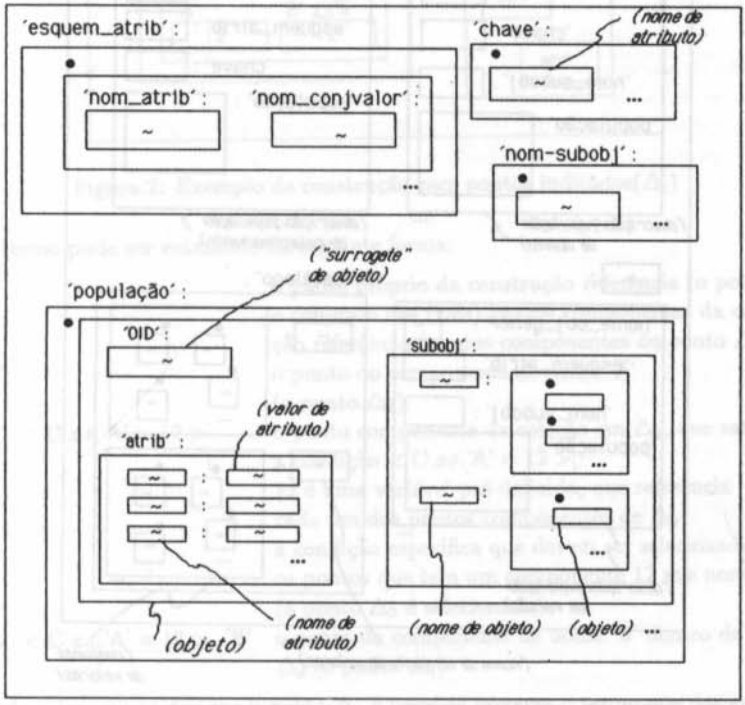

Figura 4: Composição de um modelo descrição/população de objeto

Com os diagramas apresentados, tem-se uma visão geral da composição de um modelo DODM. Entretanto, a linguagem diagramática usada não permite especificar as restrições de integridade entre os diversos componentes de uma construção. Para tal, é necessário recorrer à linguagem textual IMCL, o que é feito, a título de exemplo, na fórmula abaixo, que define o predicado É_modelo_DODM.

Note-se que esta fórmula não é completa. Para compreendê-la, é necessário conhecer a semântica dos predicados É_catálogo, É_descr/pop_objeto, É_descr/pop_relac e É_descr/pop_versão. Estes predicados definem tanto a composição detalhada, quanto as restrições de integridade entre os diversos componentes das construções dos tipos definidos. A definição destes predicados encontra-se em [Pimenta89b]. Incluídos na fórmula abaixo,

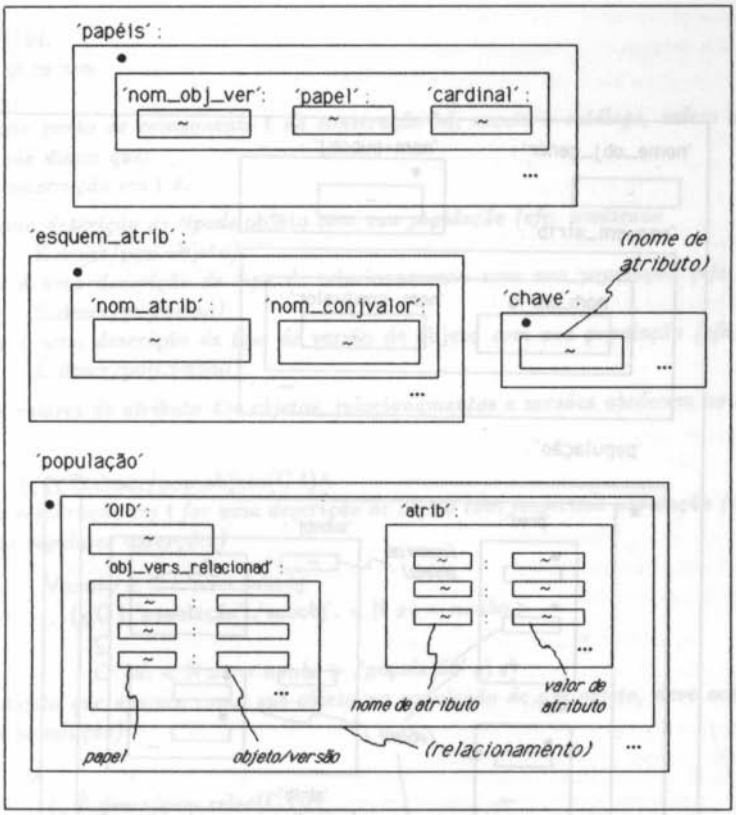

Figura 5: Composição de um descrição/população de relacionamento

vão, entre parênteses e em itálico, comentários, que pretendem diminuir o impacto do formalismo.

$\forall bd$

$(_0$ É_modelo_DODM$(bd) \Longleftrightarrow$

- IsNom $(bd) \wedge$

  $\exists cv \,\widehat{\in}\, \hat{}\, bd.$

  $(_1$ N $cv =$ 'catálogo' $\wedge$ É_catálogo (C $cv) \wedge$

(Uma entidade bd do universo de discurso é um modelo de dados DODM se, e somente se, é uma nominação com um componente de nome catálogo e tipo (cfe. predicado É_catálogo*). Opcionalmente, podem haver outros componentes conforme definição abaixo.)*

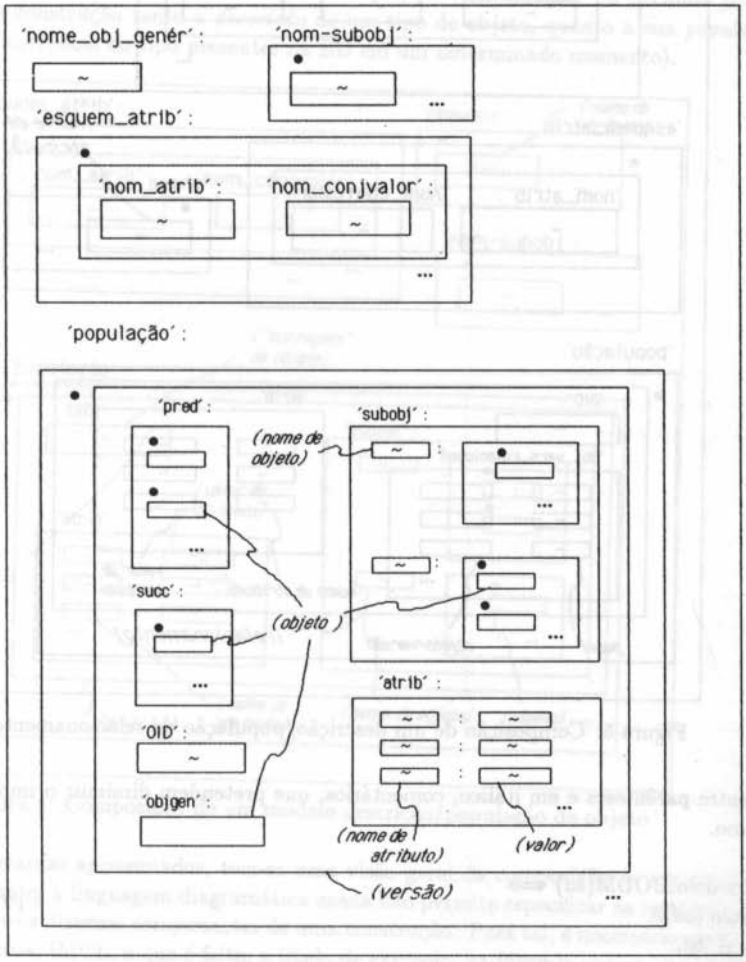

Figura 6: Composição de um descrição/população de versão

$\forall t \,\widehat{\in}\, \hat{}bd.$
$(_2\ t \neq cv \Longrightarrow$
$(_3$

(Para todo ponto de componente t da construção bd, exceto o catálogo, valem as fórmulas abaixo, que dizem que:

- *a construção em t é:*

  uma descrição de tipode objeto *com sua população (cfe. predicado* É_descr/pop_objeto*),*
  ou é uma descrição de tipo de relacionamento *com sua população (cfe. predicado* É_descr/pop_relac*),*
  ou é uma descrição de tipo de versão de objeto *com sua população (cfe. predicado* É_descr/pop_versão*).*

- *os valores de atributo dos objetos, relacionamentos e versões obedecem ao catálogo*

$)$

$(_4\ (_5$ É_descr/pop_objeto(C t)$\wedge$

(Caso a construção em t for uma descrição de objeto com respectiva população (ver figura 4) valem as seguintes asserções)

$\forall nsubo \in$ C t.'nom_subobj'
$(_6$ C t.'população'..'subobj'. $<$ N $xs = nsubo >$
$\subseteq$
C $\hat{}bd.$ $<$ N $xs = nsubo >$.'população' $_6)\ _5)$

(Todo objeto que aparece como sub-objeto na população de um objeto, deve ocorrer na sua própria população)

$\vee$
$(_5$ É_descr/pop_relac(C t)$\wedge$

(Caso a construção em t for uma descrição de relacionamento com respectiva população (ver figura 5) valem as seguintes asserções)

$\forall ptpapel \,\widehat{\in}\, t$.'papéis'.
$(_6 \exists\ esqext \ \widehat{\in}\ bd.$ $<$ N $xs =$ C $ptpapel$.'nom_objet' $>$
$(_7$ É_descr/pop_objeto(C $esqext$)$\vee$
É_descr/pop_versão(C $esqext$) $_7)\wedge$

(Todo nome de descrição/população de objeto ou versão mencionado em papéis *existe como descrição/população de objeto ou versão definido no modelo)*

COLC t.'população'..'obj_ver_relacionad'. $<$ N $xs =$ C $ptpapel$.'papel' $>$
$\subseteq$
C $\hat{}bd.$ $<$ N $xs =$ C $ptpapel$.'nom_obj_ver' $>$.'população' $_6)\ _5)$

(Todo objeto ou versão que participa em um relacionamento é elemento de sua própria população)

$\vee$

$(_5$ É_descr/pop_versão(C t)$\wedge$

(Caso a construção em t for uma descrição de versão com respectiva população (ver figura 6) valem as seguintes asserções)

$\exists$ *descrpop* $\hat{\in}$ ˆ*bd*.

$(_6$ N *descrpop* = C t.'nome_obj_genér'$\wedge$

(É_descrição/ext_objet(C *descrpop*)$\vee$

É_descrição/ext_versão(C *descrpop*))$\wedge$

(O nome de objeto genérico associado à versão é um nome de descrição/população de objeto ou versão presente no modelo)

COLC *descrpop*.'esquem_atrib'..'nom_atrib'

$\subseteq$

COLC t.'esquem_atrib'..'nom_atrib'$\wedge$

(O conjunto de atributos do objeto genérico está incluído no conjunto de atributos da versão)

C *descrpop*.'nom_subobj'

$\subseteq$

C t.'nom_subobj' $_6)\wedge$

(O conjunto de nomes de sub-objeto do objeto genérico está incluído no conjunto de nomes de sub-objetos da versão)

$(_6$ COLC ˆt.'população'.'objgen'

$\subseteq$

C ˆ*bd*. < *Nxs* = C ˆt.'nome_obj_genér' > .'população' $_6)\wedge$

(O conjunto de objetos presentes na população como objetos genéricos das versões associadas está incluído na sua própria população)

$\forall$*ptnomobj* $\hat{\in}$ t.'população'..'subobj'.

$(_6$ C *ptnomobj* $\subseteq$ C ˆ*bd*. < N *xs* = N *ptnomobj* > .'população' $_6)\wedge$

(Cada sub-objeto presente na população de uma versão, é elemento de sua própria população)

C t.'população'..'pred' $\subseteq$ C t.'população'$\wedge$

C t.'população'..'succ' $\subseteq$ C t.'população' $_5)$

(Uma versão e suas versões predecessoras e sucessoras são versões do mesmo tipo; por isto, as predecessoras e sucessoras estão incluídas na população de seu tipo de versão.)

$_4)$

$\wedge$

COLC t.'esquem_atrib'..'nom_conjvalor' $\subseteq$ NAMESET(C *cv*)$\wedge$

(Cada nome de conjunto de valores que aparece no descrição/população de um objeto, de uma versão ou de um relacionamento, como domínio de um atributo, deve estar presente no catálogo*)*

$\forall atr \,\hat{\in}\, t.$'população'..'atrib'.
($_4$ C $atr \in$
C $cv.$
$<$ N $xs =$ C $t.$
'descrição_atrib'.
$<$ C $xs.$'nom_atrib' = N $atr >$.
'nom_conjvalor' $>$ $_4$)

(Cada valor de atributo em 'população'*, é um valor do conjunto de valores (o domínio do atributo) associado ao nome de atributo em* 'atributos'*)*

$_3$) $_2$) $_1$) $_0$)

4 Conclusões

Os objetos do modelo DODM foram formalizados usando-se a abordagem IMC e sua linguagem IMCL, tanto na forma textual, quanto na forma diagramática (notação de caixas).

Para a elaboração da formalização da abordagem DODM, foi necessário remover ambigüidades existentes nas descrições informais disponíveis na literatura. Entre outros, os seguintes tópicos não puderam ser compreendidos com base na documentação existente, tendo sido feitas as suposições indicadas:

- O mesmo tipo de objeto pode ser declarado mais que uma vez, com diversos papéis, na composição de um outro tipo de objeto? Foi assumido que pode aparecer uma única vez.

- Os atributos e sub-objetos de um tipo de versão contém necessáriamente os atributos e sub-objetos de seu objeto genérico? Como nas aplicações usuais do conceito de versão isto ocorre, foi assumido que sim.

A técnica IMC, desenvolvida em uma época na qual existiam somente os modelos convencionais, mostrou-se, pela generalidade de seus conceitos, adequada também à formalização de modelos de dados não convencionais, orientados a ambientes de engenharia de software como é o caso do DODM.

Entretanto, o trabalho deixa, ainda, questões em aberto em relação à comparação de IMC com outros formalismos de objetivos análogos (p.ex. VDM). Como já foi mencionado, uma diferença básica entre IMC e os demais formalismos está no conceito de ponto, que, dentro do conhecimento dos autores, é original de IMC. Em relação à necessidade e conveniência deste conceito surgem diversas questões: Existem inconvenientes em utilizar uma técnica de formalização desprovida do conceito de ponto, e se existem, quais são? Será o conceito de ponto realmente necessário em uma técnica de formalização? Por que VDM consegue uma difusão tão ampla, mesmo desprovida deste conceito? Portanto, seria necessário, agora, estudar se este conceito é efetivamente necessário e quais as vantagens advindas da sua existência em uma técnica de formalização.

Agradecimentos

Os autores agradecem ao Dr. G. Richter por sua valiosa contribuição na revisão de versões preliminares deste texto.

Referências bibliográficas

[BjornerJones82] Bjorner, D. and Jones, C.B.: *Formal Specification and software development.* Englewood Cliffs, Prentice-Hall, 1982.

[Dittrich86] Dittrich, K.R. et al: DAMOKLES – A database system for software engineering environments. in: International Workshop on advanced programming environments. Trondheim (Norway), June 16-18, 1986. *Proceedings.* Berlin, Springer-Verlag, 1986. p 353–71. (LNCS 244).

[DurchholzRichter74] Durchholz, R. and Richter, G.: Concepts for Database Management Systems. in: Klimbie, J.W. and Koffeman, K.L.: *Data Base Management.* North-Holland, Amsterdam (1974) pp.97–122.

[DurchholzRichter88] Durchholz, R. and Richter, G.: *Information Modelling by Composition* – The IMC/IMCL Reference Manual. GMD internal report, January 1988 (Publication in preparation).

[ISO88] ISO TC97/SC18/WG3 N901: Formal Specification of ODA Document Strucutures. February 1988.

[Pimenta89a] Pimenta, M.S. O sistema DAMOKLES. in: Price, R.T. and Golendziner, L.G.: *Bancos de dados para aplicações não convencionais.* Buenos Aires, Kapelusz, 1989 (IV EBAI).

[Pimenta89b] Pimenta, M.S.: *Descrição Formal do DODM em IMC.* Trabalho Individual. Porto Alegre, UFRGS/CPGCC, 1989.

[Richter81] Richter, G.: Utilization of Data Access and Manipulation in Conceptual Schema Definitions. *Information Systems* **6** (1981) 53–71.