

XIV Conferencia Latinoamericana de Informática
17avas. Jornadas Argentinas de Informática
e Investigación Operativa
Buenos Aires, Setiembre 1988.

EM BUSCA DE UM SISTEMA GENERALIZADO DE GERENCIA DE DADOS, MODELOS, E INTERFACES COM O USUARIO.

MARIA HELENA BRAZ*

RUBENS NASCIMENTO MELO**

Dep. de Informática - PUC/RJ
R. Marquês de S. Vicente 225
22453 Gávea - Rio de Janeiro
BRASIL
Telef. (021) 5299385

SUMARIO

Com o objetivo de definir as características de um sistema genérico para Gerência integrada de Dados, de Modelos e de Interfaces com o Usuário, considerado importante para dar suporte ao desenvolvimento de aplicações como Sistemas de Apoio à Decisão (SAD) onde estes três aspectos assumem igual relevância, procurou-se estabelecer uma analogia de conceitos entre Sistemas de Gerência de Dados, de Modelos e de Interfaces com o Usuário, e propõe-se o paradigma de objetos como forma de integrar estes três sistemas.

* Eng. Eletrotécnica (I.S.T. 1979), Mestre em Pesquisa Operacional e Engenharia de Sistemas (I.S.T., 1984); Prof. Ext. Dept. de Inf. PUC-RJ; Engenharia de Software, Sistemas de Apoio à Decisão, Interfaces com o Usuário.

** Dr.Sc ITA 1976; Prof. Associado Dept. de Inf. PUC-RJ; Banco de Dados, Engenharia de Software, Sistemas Gráficos Interativos.

I - INTRODUÇÃO.

Até há alguns anos, um programador de aplicação tinha que incluir na "lógica" de seu programa, além dos procedimentos específicos para resolver o problema em si, também o código correspondente à gerência dos dados e todos os comandos de entrada e saída (E/S) necessários para a comunicação com o usuário.

A primeira evolução no sentido de diminuir a complexidade das tarefas de programação correspondeu à criação de bibliotecas de subrotinas que ofereciam funções previamente implementadas sendo posteriormente ligadas ao código desenvolvido. Foi o caso das bibliotecas com funções matemáticas que apoiavam a linguagem FORTRAN.

A segunda evolução correspondeu à criação de Sistemas de Gerência de Banco de Dados (SGBD), retirando da aplicação a necessidade de gerenciar os dados. A integração com os procedimentos computacionais da aplicação passou a ser feita através das linguagens de manipulação de dados oferecidas pelo SGBD.

Em paralelo, e com o desenvolvimento da tecnologia, surgiram os sistemas interativos, baseados em novos dispositivos de E/S, para os quais a interface tomou uma nova importância pois passou a ser possível ao usuário final interagir diretamente com o sistema e até controlar a execução das aplicações. Também nesta situação a primeira evolução para simplificar o uso destes novos dispositivos consistiu na criação de pacotes gráficos oferecendo funções de alto nível e aliviando a programação de detalhes de controle desses dispositivos.

Posteriormente, a necessidade de desenvolver aplicações com interfaces de qualidade recorrendo a prototipação sugeriu que, à semelhança do que acontecera para os dados, se deveria procurar gerenciar as interfaces exteriormente às aplicações de tal modo que se pudessem adaptar livremente as interface sem necessariamente alterar os módulos da aplicação.

Por outro lado o aparecimento de aplicações tipo Sistemas de Apoio à Decisão (SAD), veio salientar a importância dos Modelos como recurso que, à semelhança dos dados e interfaces poderia ser também gerenciado com vantagens do ponto de vista da consistência, segurança, etc.. e garantindo um

desenvolvimento mais simples destas aplicações.

O conceito de Modelo, familiar na área de SAD, corresponde a uma abstração do mundo real que se caracteriza por depender do domínio do problema que se pretende resolver. Por exemplo, um Modelo num SAD poderia ser um módulo de programa que permite realizar projeções de vendas.

Repare-se que geralmente os Modelos assentam em ferramentas computacionais do tipo das já anteriormente armazenadas em bibliotecas mas distinguem-se destas por terem associada uma lógica deduzida da realidade modelada. Como exemplo poderemos dizer que o algoritmo Simplex é a ferramenta sobre a qual se implementa um Modelo para maximização de lucro atendendo a determinadas restrições espessas por expressões lineares.

Verifica-se então, que uma forma de simplificar o processo de desenvolvimento de aplicações será recorrer a Sistemas de Gerência de Dados, Modelos e Interfaces, capazes de isolar dentro do possível, os detalhes de controle de cada um destes aspectos.

O desenvolvimento de aplicações como SAD onde estes três aspectos - Dados, Modelos e Interfaces - são fundamentais aponta ainda a necessidade da sua integração. Neste sentido, este trabalho procura analisar as características de Dados, Modelos e Interfaces de forma a encontrar analogias permitindo definir um Sistema Generalizado para Gerência comum destes aspectos.

II - ANALOGIA ENTRE GERENCIA DE DADOS, GERENCIA DE MODELOS E GERENCIA DE INTERFACES.

A criação dos SGBD foi, como se sabe, fortemente motivada por questões como: (1) Redundância e Consistência; (2) Integridade e Segurança; (3) Uso compartilhado; (4) Normalização; (5) Independência lógica e física.

Estas questões motivaram igualmente o desenvolvimento de Sistemas de Gerência para Modelos e Interfaces.

No caso dos Modelos, redundância significa a existência de múltiplas versões do mesmo Modelo desenvolvidas independentemente. Inconsistência surge devido a problemas de validade e precisão que podem fazer com que dois Modelos

aparentemente idênticos apresentem resultados distintos. Será necessário ter acesso às condições de aplicação do Modelo garantindo que ele é usado corretamente. Além disso, será necessário criar mecanismos garantindo a integridade e segurança dos Modelos gerenciados, que consistirá fundamentalmente em assegurar o uso correto e proteger contra eventuais alterações de parâmetros ou outros elementos que possam invalidar o Modelo.

O uso compartilhado será ainda um benefício pois atualmente a criação de Modelos é geralmente feita numa base pessoal e após o uso do Modelo este é descartado o que traz problemas de produtividade e custo para a organização. São exemplos desta situação os Modelos criados recorrendo a planilhas de cálculo.

O problema da normalização está fortemente ligado com redundância e inconsistência. Na realidade a normalização dos Modelos evita a existência de diferentes Modelos para uma mesma situação garantindo que todos os usuários obtêm resultados semelhantes no seu uso.

A independência lógica e física, que foi um objetivo fundamental alcançado pelos SGBD, corresponderá, neste ambiente, à definição de Modelos de forma lógica e independente da descrição da sua implementação física.

A descrição de um Modelo e a sua solução devem ser funções separadas permitindo que todos os usuários possam descrever Modelos com uma linguagem única e que a sua representação seja transformada para o nível apropriado à sua solução.

Do ponto de vista das organizações será necessário prever a existência de novas funções ligadas à Gerência de Modelos. Tal como surgiu o Administrador de Dados e o Administrador de Base de Dados para gerenciar dados, também para os Modelos se poderá pensar na existência de um Administrador de Modelos e um Administrador de Base de Modelos.

Naturalmente será de prever uma grande cooperação entre as Administrações de Dados e Modelos, sendo possível considerar pelo menos na fase inicial que a administração de dados já existente englobe os Modelos. No entanto com a crescente aceitação dos Modelos como recurso exigindo conhecimento

profundo em áreas como Estatística, Pesquisa Operacional, Otimização, etc, deverá necessariamente se dissociar alcançando uma colocação equivalente à administração de dados na estrutura da empresa.

Tal como a Administração de Dados, a Administração de Modelos deverá ter, à sua disposição, ferramentas de Software para lhe dar suporte. Em [Dolk 85] são indicados como fundamentais um Sistema Dicionário/Diretório (SDD) e um SGM correspondentes ao Dicionário de Dados e ao SGBD.

O SDD será a ferramenta fundamental para o AM pois conterá informações sobre os Modelos existentes e fornecerá documentação básica sobre eles. O SGM será importante para o ABM pois permitirá armazenar, manipular e controlar os Modelos. Ambos poderão compor um SGM genérico dependendo do grau de integração alcançado.

A compatibilidade com o SGBD será fundamental pois os Modelos necessariamente utilizarão os dados do BD devendo o SGM ser desenvolvido atendendo ao SGBD disponível.

Para conseguir independência lógica, o usuário do SGM deverá poder especificar os Modelos numa Linguagem de Descrição de Modelos (LDM). Ao usar esta linguagem o usuário deverá preocupar-se com a descrição do Modelo esquecendo a representação necessária à respectiva solução.

Para operar com os Modelos o usuário terá à sua disposição uma Linguagem para Manipulação de Modelos (LMM) que permita atualização, resolução, consulta, ligação, podendo ser uma linguagem específica ou estar embutida numa linguagem de programação já existente.

Outro componente do SGM será o Sistema de Controle de Modelos (SCM) que gerenciará o acesso, o armazenamento e consulta ao nível físico de implementação.

Qualquer destes componentes, LDM, LMM e SCM têm equivalência direta com os componentes de um SGBD (Linguagem de Descrição de dados, Linguagem de Manipulação de Dados e Sistema de Controle de Base de Dados), no entanto, no caso dos Modelos, será ainda necessário considerar um novo componente que é a Biblioteca de Soluções contendo os algoritmos utilizáveis na solução dos Modelos, por exemplo, o algoritmo do Simplex para resolução de Modelos de programação linear.

Esta biblioteca deverá conter um conjunto de algoritmos adequado às aplicações pretendidas pela organização e servindo as classes de Modelos existentes.

Na fig.1 ilustramos as ligações entre os componentes de SGM associando-os com os diferentes níveis de usuários.

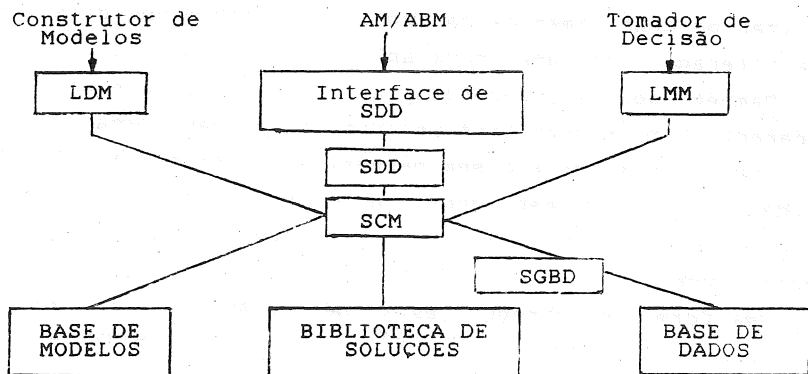


fig.1 - Os Componentes de um SGM.

Apesar da analogia estabelecida entre Dados e Modelos para ilustrar o que será um SGM é conveniente notar que os Modelos são, na realidade, entidades bem mais complexas que os Dados envolvendo parâmetros e relações sofisticadas entre eles, geralmente expressas de forma algébrica.

Uma interface, por sua vez, é descrita por diálogos apoiados em técnicas de interação. Também neste caso poderemos seguir uma abordagem idêntica à anterior e verificar que os Sistemas de Gerência de Interfaces com o Usuário (SGIU), atualmente propostos, procuram resolver o mesmo tipo de questões que motivaram quer os SGBD quer os SGM.

Para as Interfaces também será importante evitar redundância e garantir consistência. Dispondo de um SGIU o projetista de interfaces poderá recorrer a técnicas de interação, ícones, janelas, etc... previamente definidos para montar as suas interfaces aproveitando rotinas já existentes. Quanto à consistência, ela significa que a interface criada deverá apresentar uma uniformidade nas formas de interação que terá como vantagem a possibilidade dos usuários recorrerem à analogia para exacutarem novas situações de interação.

Isolando a interface das funções da aplicação garante-se também a possibilidade de usar partes de uma interface em outras aplicações semelhantes, isto é, o uso compartilhado de componentes das interfaces.

A normalização neste caso também será importante pois garantirá consistência entre diferentes interfaces. A existência de interfaces normalizadas torna possível ao usuário utilizar os conhecimentos já adquiridos, sobre uso de sistemas, para interagir com uma nova aplicação.

Também aqui a independência lógica e física é um objetivo fundamental pois permitirá recorrer a uma gama cada vez maior de dispositivos de E/S sem necessitar alterar as interfaces criadas. Estas corresponderão, no nível lógico, à definição de diálogos e de estilos de interação independentes de dispositivos.

Uma forma de conseguir esta independência lógica será, de forma semelhante ao que foi criado para dados e Modelos, criar uma Linguagem de Descrição de Diálogos LDDg cujo uso permite especificar o diálogo correspondente à interface pretendida, ignorando a forma como as técnicas de interação irão implementar tal diálogo.

Para manipular diálogos o projetista de interfaces necessitará também de uma Linguagem de Manipulação de Diálogos LMDg que irá permitir a atualização, execução, ligação, consulta dos diálogos criados.

Outro componente do SGIU será o Sistema de Controle de Diálogos SCD que, a partir da especificação de um diálogo na LDDg, é capaz de gerenciar a execução dos módulos de aplicação e o acesso e consulta ao nível interno de implementação das técnicas de interação.

Tal como no caso dos Modelos, também aqui é necessário prever a existência de uma biblioteca de técnicas de interação que corresponderá à biblioteca de soluções.

Na fig.2 apresentamos os componentes do SGIU associando-os com diferentes níveis de usuário.

Qualquer dos componentes LDDg, LMDg e SCD têm equivalência direta com os componentes já descritos anteriormente para os SGBD e SGM. Para completar tal equivalência será necessário considerar também a existência de um Sistema Dicionário de

Dialogos e Módulos de Estilo (SDDME) que, de forma semelhante aos seus congêneres, conterà informação sobre todos os diálogos e técnicas de interação organizadas em módulos de estilo, fornecendo a documentação básica do SGIU. A responsabilidade deste componente caberá ao Administrador de Diálogos (AD) que deverá estabelecer as normas de construção garantindo a segurança, consistência e normalização do SDDME.

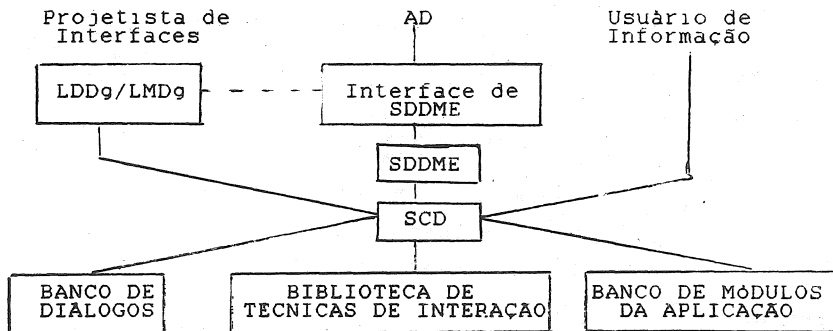


fig.2 - Os Componentes de um SGIU.

III - INTEGRAÇÃO DE DADOS, MODELOS E INTERFACES EM SAD.

Como vimos anteriormente a principal motivação para o desenvolvimento de Sistemas de Gerência consiste essencialmente em procurar simplificar a complexidade das tarefas ligadas ao desenvolvimento de aplicações.

Em aplicações dirigidas para níveis mais altos de gerência na empresa, que necessitam recorrer a Modelos para satisfazer as suas necessidades em termos de tratamento de informação, o desenvolvimento será beneficiado pela existência de um SGM. Como estas aplicações necessitam igualmente manipular dados é necessário integrar o SGM com o SGBD disponível para permitir que um Modelo possa acessar ou gerar dados cuja gerência é da responsabilidade do SGBD. Outro aspecto importante será a integração em termos de uso isto é, permitir que o acesso a Dados e Modelos seja feito de forma uniforme através de interface generalizada e flexível (SGIU).

A necessidade de integração de Dados e Modelos tem sido notada recentemente e certas propostas [Liang85] visam integrar

Dados e Modelos ao nível das consultas e uso do sistema.

Seguindo a analogia com a área de BD, os SGM podem ser divididos também em três níveis: externo, lógico e físico, correspondentes respectivamente aos três níveis de usuários previstos, a saber: Os Usuários de Informação, os Construtores de Modelos e os Construtores de Ferramentas.

No nível externo os Modelos são vistos de forma a refletir a visão do usuário de informação a quem interessa fundamentalmente as saídas que atendem às entradas disponíveis e às necessidades existentes. Essa visão externa pode ser representada, no modelo relacional, como duas relações - INPUT e OUTPUT - do tipo:

```
INPUT( Nome do Modelo, Nome da Entrada),  
OUTPUT( Nome do Modelo, Nome da Saída).
```

Considerando, por exemplo, o Modelo Vendas dado por:

Faturação = Preço * Quantidade

o seu esquema externo seria, usando estas relações, representado por:

INPUT		OUTPUT	
Nome do Modelo/Nome da Entrada	Nome do Modelo/Nome da Saída	Nome do Modelo/Nome da Entrada	Nome do Modelo/Nome da Saída
Vendas	Preço	Vendas	Faturação
Vendas	Quantidade		

que é possível armazenar numa base de dados relacional facilitando a sua manipulação e integrando-a com os dados que poderão pertencer a essa base. Outra vantagem é isolar este nível dos níveis lógico e físico pois qualquer mudança terá um impacto muito limitado neste nível.

No nível lógico os Modelos não são mais vistos como dados mas sim como subrotinas e tratados como programas escritos em linguagem de muito alto nível. Neste nível devem ser definidas duas linguagens, uma para definição (LDM) e outra para manipulação (LMM). A especificação de um Modelo com essas linguagens terá a seguinte configuração lógica conforme sugerido em [Liang85]:

```
NAME: <Nome do Modelo>  
OUTPUT: <Saídas,...>  
INPUT: <Entradas,...>  
TOOL: <Opcional,...>  
MODEL: <Opcional,...>  
BEGIN  
  USE <Tool1>  
  IF <Condição> THEN  
    LINK <Modelo>  
  ELSE  
    USE <Tool2>  
END
```

que no caso do exemplo anterior (Modelo Vendas) seria:

```
NAME: Vendas
OUTPUT: Faturação
INPUT: Preço, Quantidade
BEGIN
    Faturação = Preço * Quantidade
END.
```

Analisando esta forma de organização verifica-se que quaisquer mudanças a nível de ferramentas não afetam este nível bem como mudanças na sequência lógica de computação de ferramentas ou Modelos não afeta o esquema externo já apresentado.

No nível físico é mantido um banco de subrotinas que podem ser usadas através de um único comando USE. As ferramentas não podem se comunicar entre si e não devem conter informação relativa à formulação lógica dos Modelos.

As principais vantagens desta integração residem na possibilidade de adaptação às características dos diferentes usuários e na independência e modularidade que é proporcionada pelos níveis considerados. Por outro lado garante também a integração, ao nível externo, com o SGBD, pois, os usuários de informação farão consultas de forma semelhante nos dois casos.

Por outro lado para que os sistemas sejam adaptados adequadamente aos usuários finais, com perfis possivelmente diferentes, é preciso que as interfaces possam ser definidas, prototipadas, avaliadas e ajustadas. São as interfaces que controlam as aplicações que por sua vez são responsáveis pela execução de modelos e manipulação de dados verificando-se ser necessário não só integrar as interfaces com os modelos como também com os dados.

Um caso típico de aplicação onde tal é necessário são os SAD, estes sistemas procuram apoiar tomadores de decisão na resolução de problemas pouco estruturados e são fundamentalmente compostos por (1) Banco de Dados, (2) Base de Modelos e (3) Interface de usuário que, combinados adequadamente, formam um SAD específico para certa classe de tomadores de decisão. Um SAD é, em geral, obtido de um ambiente Gerador de SAD (GSAD) que contém facilidades como os SGBD, SGM e SGIU e que permite configurar diferentes SAD específicos para diferentes classes de usuários e aplicações.

Na fig.3 ilustramos um ambiente GSAD onde os vários

Sistemas de Gerência estão representados.

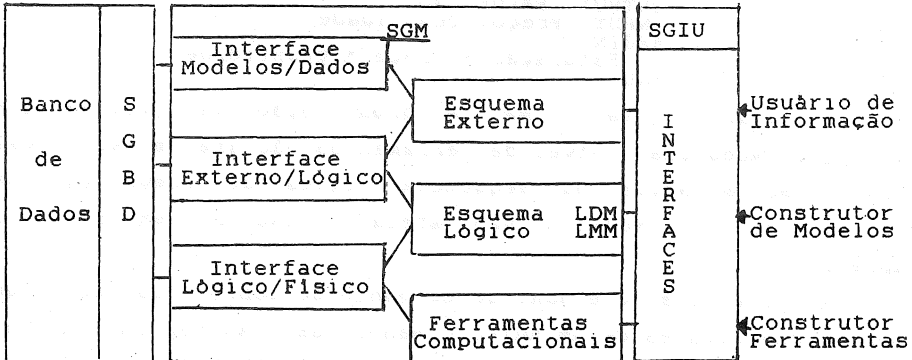


fig. 3 - Um ambiente GSAD integrado.

Para que SAD específicos sejam configurados a partir de um GSAD, este deve ter as qualidades necessárias para tornar disponíveis a diferentes Tomadores de Decisão, possivelmente com perfis diferentes, os vários modelos e dados existentes na Empresa. O ambiente GSAD deve portanto ter possibilidade de definição, armazenamento e manipulação de Modelos e Dados bem como de Diálogos que podem ser ajustados aos perfis dos usuários em questão. Se dispusermos de um SGBD, um SGIU e um SGM o desenvolvimento deste tipo de aplicações só será facilitado se estes sistemas estiverem integrados, pois, no caso contrário, será necessário desenvolver interfaces para comunicação entre eles e perder-se-á a vantagem de simplificar o processo de desenvolvimento.

IV - UMA REPRESENTAÇÃO INTEGRADORA.

Para conseguir uma integração real entre SGM, SGBD e SGIU não basta, no entanto, olhar para as semelhanças entre conceitos que foram estabelecida anteriormente. Apesar dessas semelhanças não é evidente como poderão ser tratados de forma uniforme entidades tão distintas como são Dados, Modelos e Diálogos.

Vimos, na proposta de T.Liang [Liang85], que os modelos poderiam ser vistos, pelo menos ao nível externo, como relações e através de uma representação única (relações) era feita a integração destes com os Dados. Esta proposta não contempla, no

entanto, o caso dos diálogos, pois limita-se a considerar apenas uma linguagem para consultas aos dados e modelos sem atender à necessidade de integração com o SGIU.

Para estender esta proposta, promovendo a integração de Dados, Modelos e Diálogos será necessário encontrar uma forma de representação suficientemente flexível que torne viável gerenciar estes três componentes num único ambiente, facilitando a posterior construção de ambientes específicos de desenvolvimento de aplicações como, por exemplo, um GSAD.

Esse ambiente genérico para gerência de Dados, de Modelos e de Interfaces com o Usuário seria um Sistema de Gerência de Banco de Dados Não Convencional que teria funções para armazenamento e acesso a entidades dos três tipos.

Um paradigma que se mostra adequado à construção de tal sistema é a orientação para objetos. Neste paradigma as entidades são vistas como objetos capazes de evoluir ao longo do tempo e interagir entre si.

Um objeto é então uma forma de representação capaz de modelar simultaneamente os aspectos estáticos e dinâmicos das entidades a representar, através da definição, respectivamente, dos seus atributos e do seu comportamento face aos eventos que podem ocorrer durante a sua existência.

A criação de um sistema de Banco de Dados orientado para Objetos parece a melhor resposta para a construção de um Sistema Generalizado de Gerência de Dados, de Modelos e de Interfaces com o Usuário e dará suporte ao posterior desenvolvimento de aplicações onde estes três aspectos estão presentes simultaneamente. Um sistema de gerência deste tipo está sendo desenvolvido no projeto EITIS do Departamento de Informática da PUC-RJ [Melo 87, 88].

V - CONCLUSOES.

Este trabalho abordou o problema de integração de aspectos de gerência de Dados, Modelos e Interfaces, em áreas de aplicação onde os três aspectos podem ser vistos como recursos que podem e devem ser gerenciados de maneira centralizada como é o caso dos SAD.

A analogia com a área de Gerência de Banco de Dados foi

considerada como base para o estabelecimento de um enfoque comum tanto nos aspectos de Administração como nos referentes à Arquitetura para estes Sistemas de Gerência, tendo-se mostrado a equivalência que é possível estabelecer do ponto de vista dos conceitos básicos entre sistemas de gerência de Dados, de Modelos e de Interfaces com o usuário. No entanto, se estas semelhanças indicam ser possível a construção de um Sistema Generalizado de Gerência de Dados, de Modelos e de Interfaces com o Usuário, nada adiantam sobre qual a melhor representação que garante a construção de tal sistema.

Foi proposto então, o paradigma de objetos como forma de integrar, do ponto de vista da representação, entidades tão diversas como são os Dados, os Modelos e os Diálogos.

BIBLIOGRAFIA

[Blann86] - R.W.Blanning, "An Entity-Relationship Approach to Model Management", Decision Support Systems, N.2, 1986.

[Braz 88] - Maria Braz, "Sistemas de Gerência de Modelos", Monografia do Departamento de Informática da PUC-RJ, 1988.

[Dolk 84] - D.R.Dolk, B.R.Konsynski - "Knowledge Representation for Model Management Systems", IEEE Transactions on Software Engineering, Vol.SE-10, N.6, 1984.

[Dolk 85] - Daniel Dolk, Benn Konsynski - "Model Management in Organizations", Information & Management, N.9, 1985.

[Dolk 86] - Daniel R. Dolk - "Data as Models: An Approach to Implementing Model Management", Decision Support Systems, N.2 1986.

[Liang85] - T.Liang - "Integrating Model Management with Data Management in Decision Support Systems", Decision Support Systems, N.1, 1985.

[Melo 87] - Rubens Melo - "EITIS - Um ambiente de ferramentas integradas para desenvolvimento de sistemas interativos", I Simpósio de Engenharia de Software, Petrópolis, 1987.

[Melo 88] - R.Melo, R.Lanzelotte, M.A.Rezende, P.Gualandi - "OMS-EITIS - Um sistema de Gerenciamento de Objetos para o projeto EITIS", 1988, a ser publicado.

[Rozo 88] - M.A.Rozo - "Ambiente integrado para Sistemas Interativos: Criação de Sistemas de Apoio à Decisão", Tese de Mestrado PUC-RJ, 1988.