

Automação das funções de biblioteca e pacotes de software: características e vocações

CARLOS HENRIQUE MARCONDES *

Programas de computador usados na automação das bibliotecas brasileiras têm sido desenvolvidos com ferramentas de software (gerenciadores de bases de dados, linguagens) nem sempre adequados à manipulação de informações e procedimentos característicos de uma biblioteca. Este trabalho se propõe a analisar o problema de representação e modelagem de funções biblioteconômicas em termos de sua complexidade, e a relacionar estas características com as funcionalidades e vocações de pacotes de software disponíveis para a automação de bibliotecas.

1 - INTRODUÇÃO

O processo de automação de uma biblioteca deve ser encarada como meio para que melhores, mais rápidos e mais acurados serviços informativos sejam prestados aos usuários. Muitas vezes esta afirmação tem sido esquecida e a automação tem sido encarada como um modismo, como um fim em si mesma, como o passaporte para a tão falada "modernidade". Estamos ainda engatinhando em relação à automação das bibliotecas brasileiras. O mercado de pacotes de **software** é restrito - o único artigo publicado em periódico especiali-

* Professor Assistente, Depto. Documentação, Univ. Federal Fluminense

zado em informática sobre **software** para aplicações em bibliotecas(3) é de 1991, relaciona somente seis programas, sendo que alguns destes programas/empresas já saíram do mercado. É também deformado pela presença do Micro-ISiS, um gerenciador de bases de dados documentais com aplicação em automação de bibliotecas que é distribuído pela UNESCO/IBICT praticamente de graça.

Um número significativo de bibliotecas está automatizado, mas a maioria desenvolveu com recursos próprios seus aplicativos, de forma artesanal, sem qualquer preocupação com integração, padronização, compatibilidade e intercâmbio de informações. O resultado são sistemas isolados, que não conseguem trocar informações (intercâmbio) entre si, como conclui SAYÃO et al (7:231): *"No entanto, o desenvolvimento de um software bibliográfico completo, que dê suporte às funções básicas de catalogação e recuperação de dados bibliográficos, é uma questão ainda por ser desenvolvida"*.

Esta situação se deve, em parte, ao desconhecimento das necessidades de uso e, conseqüentemente, da modelagem e da manipulação de informações bibliográficas por parte do pessoal de processamento de dados, responsável, neste período, pelo desenvolvimento desses sistemas. Esse desconhecimento faz com que o profissional de processamento de dados "simplifique" o problema, tendendo a resolvê-lo com os modelos e metodologias por ele conhecidos e com os quais está acostumado a trabalhar. São as observações comuns, muitas vezes marcadas por um tom autoritário, do tipo "o campo TÍTULO só pode ter no máximo 100 posições", ou "só existem campos para 3 AUTORES".

Por outro lado, o profissional bibliotecário, por falta de um conhecimento específico em informática e mesmo por questões culturais mais complexas, como a supervalorização da técnica e da tecnologia e o poder originário de quem as manipula, considera, de maneira ingênua, que o computador, uma vez na biblioteca, "resolverá todos os problemas". Geralmente toma uma atitude omissa, de "entregar a Deus" (e aos profissionais de informática) as soluções de automação. Tende por isso a simplificar e minimizar a importância da formalização de uma situação-problema para sua correta comunicação ao pessoal de informática, de modo que ela possa ser representada no computador. Nas palavras de MEY (3:77): *"Precisamos, sim, da conscientização necessária a qualquer usuário, para expressar corretamente nossas necessidades..."*; e *"...não exprimimos corretamente o que desejamos"*.

Formalizar a solicitação ou encomenda de programas para automação de bibliotecas é questão da maior importância para um esforço de desenvolvimento de software e propostas neste sentido foram levantadas por MARCONDES (5). O presente trabalho endereça uma outra questão, ou seja, critérios para a escolha e/ou avaliação das ferramentas de **software** (linguagens, gerenciadores de bases de dados) adequadas à implementação da automação das diversas funções dentro de uma biblioteca.

Para isso, o trabalho se propõe a discutir esta questão sob o enfoque dos problemas de modelagem e representação de funções biblioteconômicas em computador. A introdução segue-se a segunda parte, onde a solução de problemas com o uso do computador é vista sob o enfoque formal e cognitivo, como elaboração/formalização de modelos; a terceira parte discute a implementação desses modelos no espaço computacional; na quarta parte são propostos critérios de complexidade na representação desses modelos no espaço computacional; a quinta parte discute pacotes de software e linguagens de programação vistos como um elenco de representações possíveis e analisa os modelos em que os pacotes comerciais estão baseados; finalmente, a sexta parte apresenta as conclusões no tocante ao uso desses pacotes na automação de bibliotecas brasileiras.

2 - AUTOMAÇÃO COMO UMA QUESTÃO DE REPRESENTAÇÃO DE PROBLEMAS NO ESPAÇO COMPUTACIONAL

Mesmo que de forma intuitiva, um computador é conhecido como uma "máquina que processa dados". Processar significa organizar, ordenar, agregar, relacionar dados, de modo que estes ganhem (novos) significados para um usuário. Processa-se dados, ou seja, usa-se o computador para resolver problemas da vida real, da nossa atividade enquanto seres humanos, profissionais, num dado contexto social e econômico. A motivação para processar dados através do computador, é intenção, na verdade, de resolver problemas da vida real.

A solução de um problema se dá em dois planos: no primeiro momento, a elaboração da solução, o problema é resolvido conceitualmente, ou seja, ele é compreendido, relacionando-se e analisando-se todos os seus aspectos, identificando-se todas as variáveis envolvidas, para finalmente construir mentalmente sua solução. Esta solução mental é o que se denomina um "modelo" do problema, um modelo conceitual do mesmo, ou seja, no plano das idéias, que

formaliza a compreensão que se tem do problema e da solução adotada. É desejável que um modelo tenha um caráter instrumental, servindo assim como uma ferramenta através da qual o homem interfere na realidade concreta.

Uma vez obtido um modelo conceitual do problema, um segundo momento então seria a utilização deste na elaboração de um procedimento, uma série ordenada de passos ou etapas que conduza (ou que se espera que conduza) à solução prática do mesmo.

Exemplificando, a fórmula da Física $V=AT$ serve para relacionar a velocidade de um corpo que cai em queda livre com o tempo de queda livre e com a aceleração da gravidade. Esta fórmula é um modelo (um dos possíveis, já que um problema pode ter mais de uma solução), ou seja um recorte e uma explicação de uma parte do mundo real, que nos permite encontrar soluções para determinados problemas. Ao usarmos lápis e papel e calcularmos a fórmula para valores de massa de alguns corpos, estamos implementando e utilizando concretamente o modelo na solução de problemas.

Ao se utilizar o computador na solução de um problema o que se faz é reproduzir nele um modelo da situação-problema e os passos para sua solução; está-se representando praticamente o modelo. Diz-se que o modelo está sendo implementado. A solução de problemas com o uso do computador é pois uma questão de representar (implementar) neste as condições do problema, as informações nele envolvidas e os passos para sua solução; nas palavras de TAKAHASHI (8:4): *"Todo o programa de aplicação em computador representa um **modelo**, no espaço das Soluções, de um problema (ou mais exatamente, da resolução de um problema) do mundo real. A construção de um programa envolve um processo de mapeamento de aspectos de objetos pertencentes ao Espaço de Problemas para **representações abstratas** no Espaço de Soluções, de tal maneira que operações sobre estas representações abstratas correspondam a operações no mundo real"*.

3 - CONSTRUÇÃO DE MODELOS EM COMPUTADOR E SUA COMPLEXIDADE

Ao processar dados o computador está simulando um problema da vida real. Estas simulações, ou modelos, ou representações, têm então que ser construídas com os elementos básicos de construção de modelos que um computador dispõe. Estes elementos, como a

expressão "processamento de dados" indica, são processos e dados. Um computador portanto simula modelos de situações-problema do mundo real representando-os através de processos que atuam sobre dados. Além de ser um bom modelo da situação-problema, o modelo construído no espaço computacional, dado o seu caráter instrumental, vai ser usado por pessoas (usuários) interessadas na solução de um problema real; portanto, é de se esperar que, como um instrumento ou ferramenta, seja "prático" e "fácil" de ser usado.

Na verdade o processo de construção de representações ou modelos de situações-problema do mundo real em um computador é mais complexo e envolve outros tipos de questões; a primeira delas se refere a quanto é acurada nossa visão do problema real e quão correta a solução que propomos para o mesmo, enfim, a nossa capacidade de formulação, conceitualização e formalização do problema (ou seja, de construção de um modelo conceitual do mesmo). Vai-se pressupor aqui que estas questões estejam satisfatoriamente equacionadas. A elaboração de modelos conceituais de sistemas computacionais é endereçada por metodologias como Análise Estruturada (4) e o Modelo Entidades-Relacionamentos (1).

De que maneira, a nível computacional, pode ser complexa a representação de modelos de problemas? Para isso tem-se que analisar como podem ser combinados os elementos básicos de construção de modelos, dados e processo, disponíveis no espaço computacional, e como um possível usuário poderá usar esta "ferramenta", ou seja, o modelo construído com estes elementos.

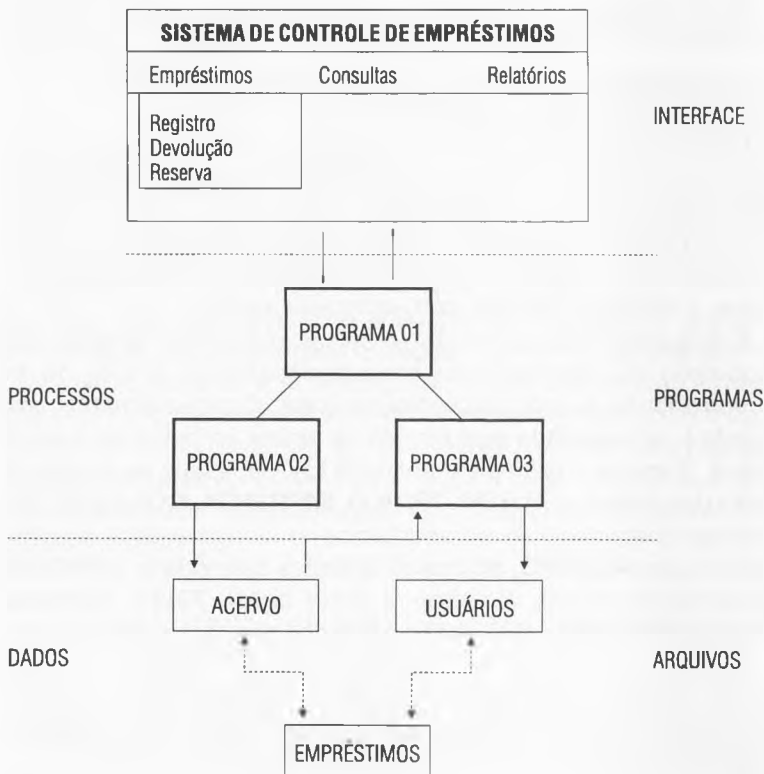
Entidades do mundo real (entes, coisas, objetos de interesse para o problema) são representadas computacionalmente através de dados, sob a forma de arquivos computacionais. Conceitualmente, uma entidade é representada pela eleição de alguns atributos para representá-la. Tome-se o exemplo a entidade **LIVRO**: este é representado através dos atributos: **AUTOR, TÍTULO, IMPRENTA, COLAÇÃO**, etc. No espaço computacional representamos um livro através de registros de um arquivo **ACERVO**, em que os atributos da entidade **LIVRO** são representados através de campos como **autor, Título, Imprenta, Colação**. Operações sobre essas entidades são representadas através de processos, sob a forma de programas, que criam, alteram ou excluem registros (entidades) de arquivos, representando o registro de um livro no acervo, sua vinculação a um usuário como empréstimo, recuperando o registro de um livro sobre um dado assunto, etc.

Operações sobre entidades do mundo real podem ser represen-

tadas computacionalmente através de processos ou o que comumente se conhece como programas. Programas são textos de instruções ou procedimentos atuando sobre dados, ou seja, sobre representações de entidades do mundo real. São compostos de instruções lógicas, comparações, decisões, procedimentos alternativos em função de determinada informação (dado) recuperada de algum arquivo.

Além desse tipo de instrução, muitos processos podem ainda ter, como componentes, operações básicas sobre os dados propriamente ditos. Assim, um programa realiza operações básicas de inclusão, exclusão e alteração em registros de um arquivo. Estas operações podem ainda ser vistas como compostas de operações mais elementares, de leitura e gravação de registros.

FIGURA 1



Os elementos construtores de modelos no espaço computacional se dispõem como na Fig. 1. Com estes elementos em mente, pode-se propor critérios de complexidade para modelos conceituais de situações-problemas do mundo real e suas representações computacionais. Apresenta-se a seguir o esquema, que servirá para se avaliar a complexidade das funções encontradas numa biblioteca:

Complexidade representacional de modelos computacionais

- De representação de entidades:
 - De representação de entidades;
 - De representação de relacionamentos entre entidades;
- De representação de processos:
 - Algoritmicamente complexos;
 - De operações sobre entidades e seus relacionamentos (Inclusão, Exclusão e Alteração);
- De uso do modelo (Interação, Interfaces, Documentação, Treinamento).

4 - AS FERRAMENTAS DE SOFTWARE E SEU POTENCIAL DE REPRESENTAÇÃO DE MODELOS

Uma vez formalizado um problema, isto é, concebido seu modelo conceitual, uma segunda questão refere-se a capacidade de representar bem esta situação-problema e a solução proposta no espaço computacional. Esta segunda questão depende fortemente de que instrumentos ou ferramentas de representação se dispõem. No espaço computacional estas ferramentas são as linguagens de programação através das quais se escrevem os algoritmos dos processos aos quais os dados vão ser submetidos no decorrer da solução do problema, ou os Sistemas Gerenciadores de Bases de Dados (SGBDs) através dos quais os arquivos (que vão conter os dados) vão ser tratados, ou mesmo os pacotes aplicativos a serem utilizados. Todas estas ferramentas implementam um elenco (limitado) de representações possíveis para os dados - seres relevantes do contexto da situação-problema em foco, por exemplo, um livro ou um usuário - e um elenco de processos aos quais estes dados serão submetidos.

Exemplificando-se esta questão, um pacote aplicativo, uma planilha eletrônica tipo Lotus 1-2-3 possui em si um elenco (limitado) de representações de dados que ele é capaz de manipular e um elenco de processos (ou manipulações) sobre estes dados que este aplicativo

é vocacionado para realizar. Ou seja, um aplicativo de planilha eletrônica representa bem informações dispostas em forma tabular, em células que são a intersecção de linhas com colunas, totalizando ou calculando o resultado de uma célula ou conjuntos de células em função do conteúdo de outras.

Mas uma planilha como o Lotus 1-2-3 não representa bem um problema tal como recuperar informações de um banco de dados ou processar textos. Seu potencial (limitado) de representação não atende bem a estes tipos de problemas.

OS SGBDs se constituem no software básico para suportar o armazenamento de grandes quantidades de informação e poder recuperá-la segundo diferentes critérios e cruzamentos como é característico da maior parte das funções bibliotecárias. São os candidatos naturais para o uso na manipulação de uma base de dados contendo o acervo de uma biblioteca. Este dado é confirmado na pesquisa de SAYÃO et al (7:224).

O mercado brasileiro sofre um **boom** deste tipo de pacote voltado para aplicações comerciais, a partir da década de 80, devido principalmente à disseminação dos microcomputadores, embora tenha havido um crescimento significativo deste tipo de pacote para equipamentos de maior porte. Títulos de origem estrangeira oriundos dos grandes produtores mundiais de software, como os popularíssimos DBASE III/IV, PARADOX, FOXBASE, DATAFLEX e até o nacional DIALOG, etc, para microcomputadores PC-compatíveis, somam-se a títulos como ADABAS, DB2, SUPRA, ORACLE, etc, para equipamentos maiores.

Este tipo de pacote obedece a uma lógica própria, a de uma ferramenta de produtividade, com o objetivo primeiro de simplificar o desenvolvimento de aplicações. Desenvolve-se a estrutura de uma base de dados declarando-se os campos componentes e suas características e o pacote oferece de antemão mecanismos de incluir, excluir, alterar recuperar a informação poupando o esforço de desenvolvê-los.

De uma maneira geral, todos estes pacotes estão calcados no chamado "Modelo Relacional" desenvolvido pelo pesquisador americano Eugen Codd (2). No "Modelo Relacional", as informações são vistas na forma de tabelas, em que as linhas são os registros individuais de informação e as colunas são os diferentes campos. Nesse modelo todos os registros são idênticos, isto é, possuem o mesmo **lay-out**; além disso, não são permitidos campos com ocorrências

multivaloradas. Estas características vão de encontro a necessidades muito específicas das aplicações (problemas) comerciais, onde a informação é geralmente uniforme, como os registros de um arquivo de estoque, ou o cadastro de funcionários, ou o cadastro de uma mala direta. O **lay-out** fixo uniforme também otimiza outra característica do que se chama aqui neste trabalho de informação comercial, a **necessidade de atualização constante**; isto porque a memória de massa de um computador, o disco magnético, é modularizado em unidades de tamanho fixo, as chamadas trilhas, setores e cilindros. Além disso, quando da atualização de um registro de **lay-out** fixo, nunca acontece o fato da informação não caber na sua localização original no disco magnético, porque a informação não muda de tamanho.

O que aconteceu no Brasil, e ainda acontece, pelos motivos expostos na primeira parte deste trabalho, é que este tipo de ferramenta, isto é, os SGBDs comerciais foram utilizados para o desenvolvimento de aplicações biblioteconômicas, sem nenhuma avaliação mais aprofundada de sua adequação. É interessante notar que este tipo de "confusão" não existe entre os grandes fornecedores de **software** e equipamentos: a IBM tem o DB2 para aplicações comerciais e o STAIRS para aplicações bibliográficas, a BULL tem o IDSII para aplicações comerciais e o MISTRAL para aplicações bibliográficas.

5 - COMPLEXIDADE DAS FUNÇÕES DE UMA BIBLIOTECA

Conforme a proposta do trabalho, o que se propõe aqui é analisar os problemas ou funções (ou modelos de situações-problema) existentes numa biblioteca segundo critérios de complexidade propostos e daí extrair subsídios para a escolha de construtores de modelos computacionais (linguagens, gerenciadores de bases de dados ou pacote aplicativos) que abriguem mais amigavelmente os diferentes tipos de complexidade das funções (situações-problema) de uma biblioteca.

As funções bibliotecárias ao serem representadas no espaço computacional, o serão em termos de informações que são manipuladas (ou transformadas) por processos, utilizadas por usuários através de uma interface. A complexidade pode residir nas próprias informações, no relacionamento entre elas, no processo a que são submetidas as informações, na interface entre o modelo (programa) e o usuário ou numa combinação desses tipos.

Um problema (em nosso caso a automação de uma função bibliotecária) a ser representado num espaço computacional pode ser complexo do ponto de vista da **representação de entidades** que compõem o problema. Exemplo característico desse tipo de complexidade é a representação de informações tipicamente bibliográficas; tome-se por exemplo o **lay-out** de um formato bibliográfico e se constatará facilmente como é complexo um arquivo que tem que manipular campos de tamanho variável, número variável de ocorrências de valores de um campo, como o caso de Autores ou Assuntos, campos opcionais, campos codificados, subcampos dentro de campos, indicadores, etc. Exemplo típico de uma função bibliotecária complexa, segundo este ponto de vista, é a automação de um catálogo ou a criação de uma base de dados para emissão de uma bibliografia.

O segundo tipo de complexidade será aqui definido como **complexidade de representação de processo**; este tipo de complexidade pode ser de dois subtipos: os processos a que as informações são submetidas são **algoritmicamente complexos**, envolvem cálculos, totalizações, decisões, testes, diversas alternativas; este tipo de complexidade é menos freqüente nas funções bibliotecárias, mas um exemplo seria um relatório estatístico de uso de periódicos ou de empréstimos solicitados por tipos de usuários.

O segundo subtipo refere-se aos casos em que a informação é submetida a processos de armazenamento, atualização, criação de índices ou recuperação, todos processos vinculados **às operações sobre as entidades e seus relacionamentos**, que alteram informações sobre entidades e seus relacionamentos, operações estas ligadas ao registro da informação na memória de massa de um computador. Esse tipo de complexidade é diretamente proporcional à complexidade da **representação de entidades**, mencionada anteriormente. Daí porque é tão complexo, do ponto de vista computacional, armazenar a informação bibliográfica, tipicamente de tamanho variável, na memória de massa de um computador, que é otimizada e modularizada em unidades de tamanho fixo, chamados setores, trilhas, cilindros, etc.

Um terceiro tipo de complexidade proposto será definido como complexidade de **representação de relacionamentos** entre agregados de informações. Em termos menos rebuscados esse é o tipo de complexidade encontrado quando diferentes entidades, representadas em diferentes arquivos, guardam entre si relações semânticas. E o caso de uma lista de periódicos para aquisição, relacionados ao

registro de seu fornecedor e simultaneamente relacionados a um fundo ou verba de aquisição.

Em termos do registro da informação, este tipo de complexidade implica em percorrer cadeias de registros de informações relacionadas para a realização de uma operação sobre as mesmas, ou seja, diversos acessos a memória de massa.

Os tipos de complexidades descritos fornecem um parâmetro para a análise das funções bibliotecárias com vistas a sua automação. A seguir é apresentado um exemplo do uso possível do esquema proposto.

- **Kardex Automatizado** para uma pequena biblioteca especializada, que precisa controlar as assinaturas de vários periódicos.

Suponhamos que o sistema seja exclusivamente para uso interno da biblioteca. Temos o registro do Periódico relacionado ao registro do seu Fornecedor e também relacionado ao registro de sua Coleção composta dos diferentes volumes e números existentes na biblioteca.

Fornecedor: Nome, endereço, telefone, fax,
1 relacionado a 1

Periódico: ISSN, Título, Periodicidade, Data-Chegada, Valor-Assinatura.

1 relacionado a n

Coleção: {Volume, Número, Data-Publicação} ("n" vezes).

Esta função é complexa do ponto de vista do relacionamento entre agregados de informações. Além disso, é complexa também pelas operações de atualização que implica, como atualização da data de chegada do último número e dos volumes, números chegados.

No entanto, esta função é simples do ponto de vista da representação computacional das entidades envolvidas. Do ponto de vista das operações sobre entidades e seus relacionamentos, a operação mais freqüente, o registro de novos números recebidos, resume-se à inclusão de uma instância (um registro) da entidade **Coleção**.

Isto, aliado ao fato do sistema ser unicamente para uso interno da biblioteca, indica que na sua automação poderá ser usado em SGBD comercial como por exemplo o DBASE III, uma ferramenta bastante conhecida pelos profissionais de informática.

6 - CONCLUSÕES

O profissional bibliotecário, por deficiência na sua formação, enfrenta o problema de escolher um pacote ou assistir ao desenvolvimento de um aplicativo para automação de sua biblioteca, com uma série de idéias simplistas, pré-concebidas e, na maioria das vezes, equivocadas. Em primeiro lugar, desconhece os problemas envolvidos na representação de modelos de situações-problemas em computador endereçados anteriormente; imagina que, dado um programador ou analista, usando qualquer ferramenta de representação (ou melhor, não importando qual seja a ferramenta usada) ele poderia automatizar qualquer biblioteca; não consegue entender quando um analista ou programador lhe diz que certa situação é "impossível" de ser representada, porque não tem a noção do problema de que representação no ambiente computacional se faz através de ferramentas (**software**, linguagens, pacotes) que podem ser mais ou menos adequadas para representar tal e qual situação-problema; na verdade a situação-problema é "impossível" de ser representada com uma dada ferramenta dentro de um modelo (paradigma) válido para situações comerciais. Além disso, por falta de conhecimento, curva-se diante do poder dos que detêm a tecnologia, quando estes afirmam que algumas coisas são "impossíveis", não tendo argumentos para contrapor a esta afirmação.

Por outro lado, o profissional de informática médio tem também sérias deficiências na sua formação para enfrentar problema que fujam ao modelo encontrado num ambiente comercial típico. A disseminação do micro-computador fez com que surgissem entre nós os famosos "cursinhos" de programação e análise, de curta duração, formando profissionais aos montes, com um tratamento superficial de questões como modelagem, estudos de usuários, etc. Este profissional está preocupado fundamentalmente com a forma e pouco com o conteúdo; considera o conteúdo problema de seu usuário. Por isso seus modelos são formais. Trabalha com modelos e ferramentas de representação convencionais, de ampla disseminação comercial, mas que têm limitações de modelagem de problemas que extrapolem aqueles encontrados num ambiente de sistemas comerciais; tenta enquadrar todas as situações-problemas neste modelo; carece ainda da visão crítica das limitações de suas ferramentas para representar outras situações-problemas que não se enquadrem neste paradigma.

É importante que o profissional bibliotecário tenha clareza da aplicabilidade e das limitações deste tipo de pacote. Sem dúvida é o preferido dos profissionais de processamento de dados, devido a

facilidade de desenvolvimento de aplicações. Não se preconiza aqui um tipo de ortodoxia que condene o uso deste tipo de pacote. Ao contrário, cada caso é um caso, principalmente num país do 3º Mundo como o nosso, com tantas dificuldades e carências com um mercado em que as opções de pacotes bibliográficos são ainda tão limitados, com profissionais de informática tão mal preparados. Pode haver casos em que, no cômputo dos prós e contras, o uso de um SGBD comercial seja razoável. O esquema apresentado aqui se propõe a ser somente mais um dos parâmetros de avaliação.

Automation of library functions and software packages: characteristics and applicability

Application programs used for library automation in Brazil have been developed with software tools (data base management systems, languages) not always adequate to handle with library information and procedures. The work proposes an analysis of the problem of library functions representation and its complexities and to relate such characteristics with the functionalities of software package available for library automation.

REFERÊNCIAS BIBLIOGRÁFICAS

1. CHEN, P. P. The entity-relationship model: towards an unified view of data. **ACM Trans. Database Sys.** v. 4, n. 1, mar, 1979.
2. CODD, E. F. A relational model of data for large shared data banks. **Comm. of AM**, v. 13, n. 6, june 1970.
3. COM OS LIVROS em ordem. **Exame Informática**, v. 6, n. 8, p. 24-26, ago. 1991.
4. GANE, C, SARSON, T. **Análise estruturada de sistemas**. Rio de Janeiro: LTC, 1983. 257 p.
5. MARCONDES, C. H. A solicitação do software: um obstáculo na comunicação bibliotecário/pessoal de sistemas. In: SEMINÁRIO DE AUTOMAÇÃO DE BIBLIOTECAS - SEAB, 3, Águas de Lindóia, 1989. **Anais...** Águas de Lindóia, 1989, p. 29-34.
6. MEY, E. S. A. Bibliotecários e analistas de sistemas: a convivência necessária. **Rev. de Bibl. de Brasília**, v. 16, n. 1, p. 75-81, jan./jun. 1988.
7. SAYÃO, L. F. et. al. Relatório do Projeto "Avaliação dos Processos de Automação em Bibliotecas Universitárias". In: SEMINÁRIO NACIONAL DE BIBLIOTECAS UNIVERSITÁRIAS, 7, 1989, Belém. **Anais...** Belém: MEC/ SESU, 1990, v. 1, p. 223-245.
8. TAKAHASHI, T. **O paradigma dos objetos: introdução e tendências**. Uberlândia: UFU, 1989. 95 p. (Curso ministrado durante o Congresso da SBC, 9, 1989, Uberlândia. Jornada de Atualização em Informática, 8.)