

UM ESTUDO SOBRE O TRATAMENTO DA ESTRUTURA LOGICA DE DOCUMENTOS

José Valdeni De Lima¹

Université Joseph Fourier-IMAG-LGI

ABSTRACT

Este trabalho tem como objetivo principal a especificação de alguns critérios para o armazenamento e o acesso de documentos, visando sua estrutura lógica, dentro de um Sistema de Gerência de Banco de Dados. Os documentos aqui considerados são documentos generalizados: compostos de imagens, textos, gráficos, voz, etc. Na bibliografia americana, este tipo de documento é conhecido pelo nome de "Multimedia". O tratamento da estrutura lógica de documentos facilita a troca de documentos entre duas estações de trabalho, facilita a tarefa de edição e, finalmente, permite consultas mais inteligentes aos documentos armazenados. Por exemplo: <<Qual o nome do remetente da carta que trata sobre o aumento dos preços dos produtos plásticos?>>. O tratamento da estrutura lógica aqui apresentado é função de dois parâmetros: as restrições declaradas e o mecanismo adotado de conformidade a estas restrições. O mecanismo de conformidade dita a aceitação ou não aceitação do documento pelo Banco de Dados.

1 Professor da Universidade Federal do Rio Grande do Sul (Brasil) e bolsista CAPES, afastado para Doutorado na Université Joseph Fourier.

1. Introdução

Os documentos desempenham um papel importante na troca de informações. Por esta razão, muito esforço foi dispensado pelos comitês internacionais de padronização para criar "standards" capazes de trocar documentos entre sistemas abertos. Como exemplos destes esforços podem ser citados: CCITT T.6 (Photographic Information Facsimile Group 4), CCITT T.73 (Restricted Layout Structure - T.61 et T.6) et ECMA 101 (ODA-Office Document Architecture). Este último documento, criado pelo grupo TC29 da ECMA (European Computers Manufacturers Association) [ECM85], define o padrão ODA/ODIF, que permite a troca de documentos considerando-os como unidades contendo suas estruturas lógicas e/ou de formatos.

Segundo Horak [Hor85] a compreensão de uma estrutura comum é fundamental quando da troca de documentos entre estações de trabalho cooperantes dentro de um ambiente de escritório. Para certas aplicações de escritório, é marcante a necessidade de tratamento da estrutura lógica dos documentos. Por exemplo, o consideremos a manipulação de cartas compostas semanticamente de três partes: cabeçalho, corpo e conclusão e, o cabeçalho contendo os nomes de remetente e destinatário. É evidente que este aspecto é indispensável para responder a consultas tais como: <<Qual é o nome do remetente da carta que trata do preço dos computadores?>>.

Além disso, na área de pesquisa inteligente de informações em documentos bibliográficos, é perfeitamente desejável o conhecimento de pelo menos alguns elementos da estrutura lógica. Por exemplo, a pesquisa de um artigo, por autor, dentro dos anais de uma conferência, necessita conhecer pelo menos a lista destes autores para que seja possível responder consultas do tipo: <<Quais são os artigos escritos por Jose?>>.

Tendo em vista estes motivos, nosso trabalho começa citando algumas propriedades que podem influenciar no armazenamento e na recuperação dos documentos. Estas propriedades são, na verdade, restrições sobre a estrutura lógica, que são declaradas na especificação do tipo de um documento. Finalmente são identificados vários mecanismos de tratamento de estrutura lógica, em função das restrições declaradas e

estratégia utilizada para garantir a conformidade destas restrições.

2. Restrições sobre a estrutura lógica dos documentos

Com relação às restrições sobre a estrutura lógica de um documento, são identificados três casos:

- Restrições de existência de elementos;
- Restrições restritas sobre a hierarquia dos elementos;
- Ausência de restrição.

2.1 Restrições de existência de elementos

Na verdade, as restrições quanto à existência (facultatividade, obrigatoriedade...) podem ajudar na escolha de uma estrutura de armazenamento, garantir a integridade dos elementos correspondentes aos diferentes tipos de dados e facilitar a recuperação dos documentos armazenados.

Existem muitas restrições de existência e agora serão descritas algumas delas:

- a. Lista de elementos possíveis de pertencer à estrutura lógica. Esta restrição é contraditória com qualquer outra restrição sobre a hierarquia dos elementos. Ela simplesmente informa a possibilidade de existência dos elementos, sem nenhuma informação sobre a posição hierárquica dos mesmos. Ela deve ser usada conjuntamente com outras restrições de existência. Por exemplo, as propriedades dos elementos tais como: obrigatório, facultativo, tipo de base, cardinalidade, etc são consideradas restrições de existência.
- b. Restrição de existência obrigatória. Esta propriedade é importante pois ela é utilizada como restrição de aceitação dos documentos assegurando uma certa padronização dos documentos armazenados. Com relação à estrutura interna de armazenamento das ocorrências dos documentos contendo elementos com esta restrição, é facilitada a alocação de espaço. Esta maneira flexível de se declarar restrições é extremamente interessante.

Normalmente trabalhamos com documentos onde sua estrutura não é totalmente conhecida e, além disso, a recuperação dos documentos armazenados é feita levando em conta somente certos elementos. Rabitti em [Rab85] afirma que uma das maiores ênfases que se poderia dar aos modelos de documentos generalizados é a definição de tipo flexível. Esta definição flexível é possível através de especificação de restrições como as descritas neste trabalho.

Todos os exemplos de declarações de restrições sobre a estrutura lógica apresentados no presente trabalho utilizarão uma sintaxe próxima à sintaxe da linguagem FMQL especificada em [OIS87a]. Por exemplo a declaração de obrigatoriedade de presença do elemento "Destinatário" de uma "Carta" pode ser expressa da seguinte maneira:

[Carta,Content:ODA(*Destinatário)]

- c. Restrição de existência facultativa. Esta restrição torna a declaração de um tipo de documento mais flexível. Por exemplo, a declaração do fato de uma carta possuir facultativamente anexos, ou que seu remetente seja facultativo, permite a aceitação de cartas anônimas com ou sem anexo. A declaração deste fato pode ser expressa da seguinte maneira:

[Carta,Content:ODA(remetente,anexo)]

- d. Restrição do tipo de base de um elemento. Este tipo de restrição é importante no sentido de que pode manter a integridade semântica dos elementos correspondentes aos diferentes tipos de dados que podem compor o documento. Na realidade, parece interessante manter a possibilidade de especificar que um elemento é de um tipo determinado. Por exemplo, "abstract" é do tipo Texto ou "foto" é do tipo Imagem. Outro exemplo, mais simples, é a declaração de cartas, contendo o elemento "remetente" do tipo "string". A restrição de tipo "string" é importante pelas seguintes razões:

- A estrutura interna de armazenamento deve conter campos, identificados por "remetente", capazes de armazenarem valores do tipo "string".
- Todos os valores armazenados devem ser obrigatoriamente do tipo "string".
- A recuperação do documento pode utilizar explicitamente o elemento "remetente".

Os tipos de dados dos elementos de um documento são os tipos de base convencionais mais os dados do tipo Texto, Imagem, Gráfico e Voz.

- e. Restrição de cardinalidade. Esta restrição normalmente permite informar ao sistema que o elemento tem um número mínimo e um número máximo de valores. A especificação desta restrição pode ser feita através de dois limites: inferior e superior. Um exemplo, utilizando a sintaxe próxima da linguagem FMQL, é dado abaixo e especifica que um documento pode ter zero ou vários autores:

[Document:Document,Content:ODA(autor(0,*))]

A indicação de zero como o número mínimo de valores para um elemento é interpretada pelo servidor OIS que este elemento pode não existir em certas ocorrências de um tipo de documento. Informações sobre o servidor OIS podem ser encontradas em [OIS86a,OIS86b,OIS86c,OIS87b]. Esta interpretação pode ser contraditória com a restrição de obrigatoriedade. Na verdade, as diferentes interpretações existentes apresentam uma analogia com os valores nulos referenciados por [AG87,GZC87]. A interpretação de zero valores, para um elemento que existe, não é contraditória com a restrição de obrigatoriedade. Desta forma, se o valor do elemento não existe, é possível modelá-lo em 3 (três) situações distintas:

- O elemento lógico existe sem nenhum valor. Por exemplo, um livro de um autor desconhecido tem um autor, mas ninguém conhece seu nome.

- O elemento lógico existe, mas seu valor não é conhecido neste momento. Por exemplo, o livro cujo título é "Le Parfum" sabe-se que tem um autor, mas no momento não se conhece seu nome.
- O elemento lógico não existe. Por exemplo, um manual técnico não tem autor.

No caso de existência do valor do elemento, são duas as possibilidades de modelagem:

- O elemento lógico existe com um único valor. Por exemplo, o livro "Introdução à Linguagem PASCAL" tem uma autora: V.L.S. de LIMA.
 - O elemento lógico existe com vários valores. Por exemplo, o livro "Bases de Données et Systèmes Relationnels" tem dois autores: C. DELOBEL e M. ADIBA.
- f. Restrição de proibição. Esta restrição pode, talvez, ser desconsiderada. Ela poderia ser útil quando da especialização de um documento. Por exemplo, "cartas-sem-assinaturas" contendo todas as cartas sem o elemento "assinaturas". Na verdade, este problema poderia ser resolvido de outra maneira, via programação.
- g. Restrição de elementos alternativos. Esta restrição é útil se utilizada para especificar a estrutura lógica dos documentos do tipo formulário, onde são conhecidos os elementos alternativos segundo o seu preenchimento. Por exemplo, o elemento "nome-de-solteira" existe para as pessoas do sexo feminino não casadas; jamais existe para as pessoas do sexo masculino. Em compensação, o elemento "situação militar" só existe para as pessoas do sexo masculino. A restrição de elementos alternativos também é útil no caso de sua utilização juntamente com as restrições sobre a hierarquia dos elementos. Isto implica a possibilidade de especificar estruturas genéricas mais gerais, que podem aceitar documentos com estruturas variadas. Por exemplo, o artigo de um congresso pode ser um artigo completo ou um resumo e, como consequência, com estruturas diferentes.

2.2 Restrições sobre a hierarquia dos elementos

- a. Restrição restrita sobre a hierarquia dos elementos. Esta restrição é mais rica no sentido de que ela reflete um conhecimento hierárquico da estrutura lógica do documento. Por exemplo, o elemento autor é composto de nome, prenome e endereço, que por sua vez é composto de nome da rua, número e bairro.
- b. Restrição sobre o nível dos elementos. Esta restrição é importante para o aumento de performance quando da pesquisa do elemento dentro da estrutura hierárquica. Entretanto, na prática, é difícil de dispor desta informação, pois as estruturas dos documentos variam muito. Por exemplo, o capítulo pode estar no primeiro nível de uma estrutura de um livro mas, dentro de um outro, organizado em partes I e II, os capítulos estão no segundo nível.
- c. Restrição de recursividade. Um elemento recursivo acarreta um ciclo sobre o modelo conceitual. Isto significa que o elemento é composto dele mesmo ou de um outro que o compõe. Por exemplo, seção composta de seção que pode, por sua vez, ser composta de novas seções. Para o tratamento de documentos dentro de um servidor de informações, a restrição de recursividade é vista como dispensável, principalmente se levado em conta o modelo ODA. Esta constatação se deve ao fato do modelo ODA quando da representação de um elemento recursivo, notar explicitamente as ocorrências do mesmo através de numeração progressiva. Por exemplo: seção 1, seção 1.1,... Tanto o servidor de informação MULTOS [CYT86] como o OIS [OIS87c] levam em conta o modelo ODA.

2.3 Ausência de restrição

A ausência de restrição implica a estratégia de sistema aberto. A ausência de restrição será descrita no decorrer deste trabalho, mais precisamente, na seção sobre a ausência de mecanismo de tipo.

3. Mecanismos de conformidade às restrições

São identificados três mecanismos de conformidade às restrições:

- Mecanismo de tipo forte.
- Mecanismo de tipo fraco.
- Ausência de mecanismo de tipo.

3.1 Mecanismo de tipo forte

A utilização deste mecanismo de tipo acarreta uma estratégia de sistema fechado. Isto significa que o sistema só aceita documentos onde todos os elementos que compõem a estrutura lógica sejam declarados. Este mecanismo é interessante quando se trabalha com documentos onde, pelo menos, são conhecidos todos os elementos de sua estrutura lógica. Observe que não é obrigatoriamente necessário conhecer a estrutura hierárquica entre os elementos, se não for declarada restrição sobre a mesma. Este mecanismo se adapta perfeitamente ao tratamento de documentos onde sua estrutura lógica é fixa e determinada (como, por exemplo, os formulários).

3.2 Mecanismo de tipo fraco

Este mecanismo é o mais próximo da realidade pois ele aceita documentos de uma maneira flexível segundo a vontade do usuário. Este mecanismo permite a aplicação de uma estratégia parcialmente fechada acarretando um tratamento flexível da estrutura lógica dos documentos manipulados pelo sistema. Aqui um documento é aceito mesmo se todos os elementos de sua estrutura lógica não são conhecidos. O importante aqui é o reconhecimento dos elementos declarados como obrigatórios.

3.3 Ausência de mecanismo de tipo

Não se pode utilizar um mecanismo de tipo quando nenhuma restrição sobre a estrutura lógica é declarada. Desta forma a única estratégia possível de se aplicar é a estratégia de sistema aberto. Esta estratégia aceita todos os tipos de documentos, sem se preocupar com a estrutura lógica dos mesmos. A principal desvantagem desta estratégia é a falta de padronização dos documentos aceitos pelo sistema. Como consequência, não é possível responder a consultas do tipo: <<Quais são os

documentos cujas estruturas lógicas contenham "introdução" e "conclusão"?>>.

A utilização destes mecanismos acarreta vantagens e inconvenientes, dependendo das restrições declaradas sobre a estrutura lógica dos documentos. As próximas seções deste trabalho analisarão a aplicação destes mecanismos quando da utilização de restrições sobre os elementos da estrutura lógica considerando ou não a hierarquia entre eles.

4. tipo forte e lista de elementos

O mecanismo de tipo forte tem como principal inconveniente a falta de flexibilidade para aceitar documentos onde não são conhecidos todos os elementos de sua estrutura lógica. Este mecanismo funciona muito bem para os documentos que possuem estruturas lógicas conhecidas e regulares (por exemplo, formulários). No entanto, para os documentos onde as estruturas lógicas não são regulares e definidas, este mecanismo pode causar problemas. Segundo Rabitti [Rab84] o conceito de tipo poderia ser utilizado no modelo documento, se ele permite um alto grau de flexibilidade com relação à estrutura lógica dos documentos.

A declaração da estrutura lógica como uma lista de elementos é simples, no sentido de que ela não leva em conta a estrutura hierárquica entre estes elementos. Todas as restrições citadas anteriormente podem ser utilizadas sobre os elementos da lista.

Com este mecanismo de tipo, um documento pode ser recusado pelo sistema se contém pelo menos um elemento não especificado na lista. Suponhamos que uma carta tenha sido declarada composta obrigatoriamente dos elementos "destinatário", "remetente" e "assinatura" e, opcionalmente, dos elementos "introdução" e "conclusão". Neste caso, a carta especificada abaixo é recusada pelo sistema pelo simples fato da presença do elemento "corpo", que não tinha sido declarado.

Carta(destinatário,remetente,assinaturas,corpo)

Outro motivo que pode causar a recusa de um documento é a falta de

um elemento declarado como obrigatório. Por exemplo, as cartas abaixo são recusadas:

Carta(destinatário,remetente,conclusão)

Carta(destinatário,assinatura)

VANTAGENS:

- a. O conhecimento da hierarquia dos elementos da estrutura lógica é dispensado.
- b. As necessidades de aplicações que manipulam documentos padrões tais como certos tipos de cartas, formulários, etc são satisfeitas plenamente.
- c. A análise semântica dos comandos de manipulação é feita, de uma maneira mais completa, pois todos os elementos possíveis de serem referenciados são declarados e, portanto, conhecidos.

INCONVENIENTES:

- a. O conhecimento de todos os elementos que compõem a estrutura lógica é exigido. Isto não significa conhecer a posição dos elementos dentro da estrutura hierárquica ou melhor a hierarquia entre estes elementos.
- b. A estrutura hierárquica entre os elementos não é conhecida. Como consequência, não podem ser respondidas consultas do tipo: <<Quais são os títulos dos capítulos do livro x?>>. Além disso, é impossível saber se um elemento título declarado é de um livro ou de um capítulo.

5. tipo fraco e lista de elementos

O mecanismo de tipo fraco é mais flexível que o mecanismo de tipo forte, e parece mais próximo das situações reais. Na verdade, os documentos encontrados no mundo real têm estruturas frequentemente complexas e difíceis de serem completamente identificadas. Por exemplo, é fácil deduzir que uma "Carta" tem, normalmente, "destinatário" e "remetente". No entanto, outros elementos tais como "logotipo", "introdução" ou "conclusão" não são evidentes. Existem elementos que dependem da aplicação ou do tipo de documento. Por exemplo, uma carta informando o preço de um produto pode conter uma tabela com os preços dos elementos que compõem o produto, bem como gráfico contendo a evolução dos preços dos produtos concorrentes nos últimos seis meses.

A principal diferença entre os mecanismos de tipo forte e tipo fraco diz respeito à declaração dos elementos: o primeiro exige a declaração de todos os elementos da estrutura lógica, enquanto, o segundo, aceita documentos contendo elementos não declarados. De forma semelhante ao mecanismo de tipo forte são também recusados os documentos nos quais faltam elementos declarados como obrigatórios. Por exemplo, se uma carta é declarada como composta, obrigatoriamente, dos elementos "remetente" e "destinatário", pelos mesmos motivos já citados as seguintes cartas são recusadas:

Carta(destinatário,assinaturas)

Carta(destinatário,corpo)

VANTAGENS:

- a. O conhecimento da estrutura hierárquica dos elementos é dispensado.
- b. Não é necessário conhecer todos os elementos componentes da estrutura lógica do documento manipulado.
- c. Funciona bem se o usuário conhecê os elementos sujeitos a recuperação futuras.

INCONVENIENTES:

- a. A estrutura hierárquica entre os elementos não é conhecida. Como consequência, não podem ser feitas consultas do tipo: <<Quais os títulos dos capítulos e das seções do livro x?>>.
- b. A análise semântica dos elementos não declarados é impossível de ser feita.

6. Tipo forte e restrições sobre a hierarquia entre elementos

As restrições sobre a hierarquia entre elementos da estrutura lógica de documentos são mais ricas, no sentido de que dão um conhecimento hierárquico da referida estrutura. A utilização destas restrições com o mecanismo de tipo forte exige o conhecimento antecipado de todos os elementos da estrutura lógica dos documentos que serão manipulados pelo sistema, bem como, suas posições dentro da referida estrutura.

VANTAGENS:

- a. A estrutura arborescente dos elementos é conhecida completamente pelo sistema.
- b. A análise semântica dos comandos de acesso pode ser feita de uma maneira mais clara e completa.
- c. Funciona bem para os tipos e documentos onde as estruturas lógicas são totalmente conhecidas.

INCONVENIENTES:

- a. É obrigatório declarar todos os elementos e a hierarquia entre eles. Este nível de detalhe nem sempre é necessário.

7. Tipo fraco e restrições sobre a estrutura hierárquica

A principal vantagem desta combinação é a flexibilidade oferecida para especificação da estrutura lógica dos tipos de documentos que serão manipulados. Desta maneira, não é necessário o conhecimento completo dos elementos, nem da hierarquia entre seus componentes.

VANTAGENS:

- a. Permite a especificação parcial da estrutura lógica dos documentos que serão armazenados e manipulados posteriormente. Como consequência, o usuário pode especificar somente os elementos que lhe interessam, tornando o sistema flexível e sob medida.
- b. Permite a especificação parcial da arborescência da estrutura lógica.

INCONVENIENTES:

- a. A análise semântica dos elementos não especificados é impossível de ser feita.

8. Ausência de mecanismo de tipo e de restrição

Na verdade, o mecanismo de tipo e as restrições sobre os elementos da estrutura lógica estão intimamente ligados. Isto quer dizer que é impossível a aplicação de um mecanismo de tipo se nenhuma restrição é declarada. Este caso caracteriza a estratégia de um sistema aberto onde todos os documentos são aceitos independentemente de suas estruturas lógicas.

Esta estratégia abre várias possibilidades de tratamento da estrutura lógica, as quais exigiriam uma análise mais detalhada.

9. Conclusão

Neste artigo foi abordado a modelagem flexível da estrutura lógica dos documentos. Como visto, esta modelagem poderia levar em conta os elementos independentemente de sua posição dentro da hierarquia. Neste caso, uma lista simples dos elementos seria suficiente. Mas, também se poderia declará-los conforme a estrutura hierárquica. Neste caso, as restrições sobre a hierarquia dos elementos seriam necessárias. Além disso, foram descritas várias outras restrições que poderiam ser aplicadas a cada elemento declarado da estrutura lógica.

Foram também apresentados os mecanismos de conformidade a estas restrições, que ditam, na realidade, a estratégia de aceitação dos documentos. A estratégia de sistema fechado é resultado da aplicação de um mecanismo de tipo forte, enquanto que a estratégia de sistema parcialmente fechado é resultado da aplicação de um mecanismo de tipo fraco. A impossibilidade da aplicação de um dos dois mecanismos, devido, por exemplo, à falta de declaração de restrições, acarreta uma estratégia de sistema aberto.

Como conclusão, pode-se considerar que o mecanismo de tipo fraco é o mais conveniente e próximo da realidade, pois permite uma aceitação flexível dos documentos pelo sistema, sem o conhecimento prévio de suas estruturas lógicas. Além disso, esse mecanismo permite a especificação de alguns elementos, contribuindo desta maneira para uma certa padronização dos documentos armazenados dentro do servidor de informações.

Agradecimentos

O autor agradece a todas as pessoas do projeto DOEOIS que contribuíram para realização deste trabalho em especial aos Srs. Mauricio Lopez e Jean-Pierre Martin.

10. Bibliografia

- [AG87] ABITEBOUL, Serge et GRUMBACH, Stéphane, Bases de données et objets structurés. Technique et Science Informatiques v.6 n.5, v.6 1987.

- [CYT86] CHRISTODOULAKIS S. et al., Office Document Retrieval in MULTOS, ETW 1986.
- [ECM85] ECMA/TC29/85 Standard ECMA-101 Office Document Architecture. September 1985.
- [GZC87] GUTING Ralf Hartmut et al., An Algebre for Structered Office Documents. Research Report of Almaden Research Center, San Jose, California, March 1987, 77 pag.
- [Hor85] HORAK W. Office Document Architecture and Office Document Interchange Formats: Current Status of International Standardization. IEEE Computer, october 1985, pag. 50 - 60.
- [OIS87a] DOEOIS Data Definition and Data Manipulation Services. OIS-DH-01.01, april 1987.
- [Rab85] RABITTI F., A Model for Multimedia Documents. In: D. Tsichristzis (ed.), Office Automation, Springer 1985,226-249.
- [OIS87b] DOEOIS Future Office Information Server Functionality. 3A3/3A5 Deliverable, March 1987.
- [OIS86a] DOEOIS Bibliographical Annex Bibliographical Informations and Standards in OIS: Overview. 1B1 Deliverable, January 1986.
- [OIS86b] DOEOIS Review of Data Models and Data Management Systems. 1B2 Deliverable, January 1986.
- [OIS86c] DOEOIS External Functional Interface for an Office Information Server. A Description. 2B1 Deliverable, September 1986.
- [OIS87c] DOEOIS Document Representation in an Office Information Server. 2B3 Deliverable, 1987.