

# O papel das linguagens de marcação para a Ciência da Informação

## *The role of marking languages in Information Science*

Sérgio FURGERI<sup>1</sup>

### RESUMO

Este artigo apresenta um panorama sobre algumas linguagens de marcação que possuem ligação direta com a Ciência da Informação. São apresentadas as características principais dessas linguagens, seus pontos fortes e suas limitações, tendo em vista sua relação com a Ciência da Informação. Numa abordagem comparativa, o artigo, além de determinar o papel de cada linguagem na disseminação, troca e recuperação de informações, sugere maneiras de criar, tanto estruturas de representação de informação por meio da XML, como vocabulários, por meio da linguagem RDF e *Namespaces*; a discussão apresenta-se sem uma preocupação técnica, apesar das relações do assunto com a área da computação. O artigo procura ainda apontar as oportunidades emergentes do uso das linguagens de marcação para o campo da Ciência da Informação.

**Palavras-chave:** linguagem de marcação; HTML; XML; RDF; *schema*; metadados; semântica; Ciência da Informação.

### ABSTRACT

*This article presents a general view on some marking languages that have a direct connection with Information Science. The main characteristics, strong points and limitations of these languages are presented, from the point of view of their relationship with Information Science. In a comparative approach, the article determines the role of each language in the dissemination, exchange and retrieval of information. It also suggests how to create information representation structures through XML and vocabularies through RDF and Namespaces language, these subjects are discussed without much technical concern, despite being related to the computer science area. The article also attempts to indicate emergent opportunities for the use of marking languages in the field of Information Science.*

**Key words:** Marking languages; HTML; XML; RDF; *schema*; metadata; semantics; Information Science.

### INTRODUÇÃO

A Ciência da Informação, afirma Almeida (2002), “encontra boas oportunidades para estudo e discussão das linguagens de marcação”. Este artigo acredita nessa afirmação e pretende apresentar

algumas aplicações das linguagens de marcação (LM) no âmbito da Ciência da Informação.

É apresentado um panorama sobre diversas LM, em especial a HTML (*HyperText Markup Language*), XML (*eXtensible Markup Language*) e a RDF (*Resource Description Framework*), seus aspectos

<sup>1</sup> Mestrando, Programa de Pós-Graduação em Ciência da Informação, Centro de Ciências Sociais Aplicadas, Pontifícia Universidade Católica de Campinas. Rua Marechal Deodoro, 1099, Centro, 13020-904, Campinas, SP, Brasil. E-mail: <furgerii.idt@terra.com.br>. Recebido em 7/6/2005 e aceito para publicação em 5/4/2006.

em comum, suas principais diferenças e limitações, além de situar cada uma dentro do escopo da Ciência da Informação.

De certa forma, o artigo procura complementar as discussões feitas por Bax (2001) e Almeida (2002). Para isso, inicia-se com uma visão geral sobre as LM; a seguir são apresentados alguns pontos importantes de contribuição dessas linguagens para a disseminação da informação; mais à frente são discutidas as relações existentes entre HTML, XML e RDF no campo da Ciência da Informação, seus benefícios e limitações e finalmente são apresentadas as possíveis aplicações dessas linguagens na recuperação da informação e suas contribuições para a Web Semântica.

## VISÃO GERAL DAS LINGUAGENS DE MARCAÇÃO

Diversos autores apresentam diferentes definições para LM. Para Davies (2004), a linguagem de marcação possui “uma forma de descrever a estrutura lógica ou semântica de um documento e fornecer instruções a computadores sobre como apresentar o conteúdo de um arquivo”. Essa definição apresenta uma visão que interessa diretamente aos estudos da Ciência da Informação: a possibilidade de descrever o conteúdo semântico de um texto.

Existem diversos tipos de LM, cada uma contém usos e finalidades específicas. Editores de texto, por exemplo, podem manter marcas internas para controlar diversos atributos de um texto, tais como cor, tamanho, formatação, etc. Aplicativos para navegação na Internet apresentam na tela documentos criados a partir de uma LM, documentos compostos por inúmeras marcações. Um celular comum também pode utilizar uma LM específica para navegar pela Internet. Enfim, existem muitas aplicações para as LM.

Como o próprio nome sugere, as LM têm como característica principal criar marcas (*tags*) para delimitar um texto. Num primeiro momento, essas marcas eram usadas apenas para definir a forma como um texto seria apresentado. Mais tarde, com a evolução das linguagens, tornou-se possível usar marcas para fornecer significado ao texto. Uma marca é um tipo de código que envolve uma palavra ou um trecho de um texto. Por exemplo, um trecho delimitado pelas marcas **<B>** e **</B>** aparecerá em negrito

(*bold*), um trecho delimitado pelas marcas **<U>** e **</U>** aparecerá sublinhado (*underline*), e um trecho delimitado pelas marcas **<I>** e **</I>** será apresentado em itálico.

Todas LM possuem um conjunto de convenções utilizadas na definição de suas marcas, isto é, contém diversas marcas, cada qual com um significado próprio, estipulado previamente. Conforme citado no exemplo anterior, em HTML a *tag* **<B>** é usada para delimitar um texto que será apresentado em negrito. Cada *tag* possui uma função específica, seja simplesmente para marcar um texto, seja para definir uma estrutura mais complexa como, por exemplo, uma tabela, um formulário, etc. Ao se delimitar um texto por marcas, é possível estabelecer um conteúdo semântico que pode ser tratado e manipulado por programas de computador.

Desde 1994 existe um comitê internacional chamado *World Wide Web Consortium (W3C)*, responsável por desenvolver e manter padrões para a Web. O W3C é dirigido por Tim Berners-Lee, o inventor da *World Wide Web (W3 CONSORTION, 2005a)*. Dentre esses padrões encontram-se dezenas de linguagens de marcação. É necessária a existência da padronização para que as LM possam ser compartilhadas e utilizadas mundialmente. Para um fabricante desenvolver um *browser (software para navegação na Internet)*, por exemplo, é necessário que ele reconheça que a *tag* **<B>** indica negrito, e assim por diante.

As LM possuem um ancestral comum chamado SGML (*Standard Generalized Markup Language*), um padrão internacional independente de sistemas e máquinas para a definição de métodos de representação de textos em formato eletrônico. Trata-se de uma linguagem complexa a partir da qual surgiram diversas outras, como a HTML e a XML. Todas as LM devem ser capazes de diferenciar as *tags* (que compõem a estrutura do documento) do texto propriamente dito (que constitui o conteúdo do documento). Tendo essa característica, uma ferramenta como um *browser*, por exemplo, tem condições de identificar as marcas contidas e apresentar ao usuário somente o conteúdo de texto do documento, sem as *tags*.

## LINGUAGENS DE MARCAÇÃO E DISSEMINAÇÃO DA INFORMAÇÃO

As LM permitem criar documentos com uma estrutura de representação que seja compreendida

por diversos sistemas de *software*, independentemente da máquina onde estão sendo executados. Na prática isso quer dizer que um mesmo documento pode ser manipulado em um computador pessoal ou em um *mainframe* (computador de grande porte), sem que haja dependência de um determinado sistema operacional ou sistema proprietário. Os documentos se tornam independentes do sistema ou *software* onde são visualizados, contribuindo fortemente para a disseminação da informação em ambientes heterogêneos.

As LM contribuíram para tornar a comunicação livre de formatos proprietários, uma vez que elas representam padrões abertos e podem ser usadas livremente. Apesar de existir o órgão regulamentador W3C, ninguém é proprietário de uma LM, não sendo necessário pagar direitos para elaborar um documento baseado em LM.

As características citadas nos parágrafos anteriores - a independência de plataforma e o uso livre da linguagem - foram essenciais para o grande sucesso da linguagem HTML, fato que ocasionou o surgimento de bilhões de documentos para a *Web*. Essa grande quantidade de documentos espalhados pelo globo contribuiu enormemente para a disseminação de informações. Pesquisadores e estudantes espalhados em diversas regiões do planeta podem, agora, compartilhar informações e conhecimentos, acelerando o desenvolvimento da sociedade. Apesar disso, é importante destacar que essa “avalanche” de documentos estabeleceu certo “caos” na Internet, principalmente no que diz respeito à recuperação de informações relevantes, ainda que se utilizem mecanismos de busca como o *Google*.

Como a evolução tecnológica é contínua e crescente, nos últimos anos apareceu uma nova maneira de disseminar informação; os documentos criados a partir das LM tornaram-se acessíveis por meio de dispositivos móveis, como celulares, PDAs (*Personal Digital Assistance*) ou *Palms*. O protocolo *Wireless Application Protocol* (WAP) tem possibilidade que celulares e outros dispositivos acessem páginas *Web* criadas a partir da linguagem *Wireless Markup Language* (WML).

A tecnologia favorece o aparecimento de novas aplicações no campo da Ciência da Informação tanto no que se refere à disseminação quanto no que se refere ao armazenamento e à recuperação da informação. Uma pessoa, por exemplo, pode

consultar um artigo armazenado na Internet enquanto faz uma viagem; o usuário de uma biblioteca pode consultar o acervo para verificar se uma obra se encontra disponível, etc.

Outra aplicação no campo da disseminação da informação refere-se ao surgimento do padrão RSS (*RDF Site Summary*), um padrão desenvolvido a partir das linguagens de marcação e definido pelo W3C. O RSS cria um ambiente adequado para compartilhar informações entre *sites* e aplicações, facilitando a integração entre diferentes sistemas; foi criado a partir da linguagem XML, citada mais à frente na seção “XML e a Ciência da Informação”, com o objetivo de agilizar o processo de troca e coleta de informações. Muitos *sites* já disponibilizam esse serviço gratuitamente, principalmente os de notícia, como jornais, revistas e portais. Para identificar a disponibilidade de um serviço RSS, os *sites* inserem em seus documentos a figura de um botão laranja, geralmente com a palavra XML, apesar de existirem outras variações, tais como RSS e ATOM. Esse botão aponta para o endereço da Internet que disponibiliza o serviço RSS.

Para melhor compreensão do princípio de funcionamento do RSS, será apresentado um pequeno exemplo prático. Um *site* de notícias disponibiliza (em um endereço da Internet) um arquivo contendo um resumo de uma notícia. Esse resumo, criado por meio de uma LM, contém dados como: título da notícia, data de publicação, breve descrição da notícia, a língua em que a notícia foi escrita, entre outras informações (DORNFEST, 2000).

Um usuário instala em sua máquina um *software* leitor de RSS, conhecido como leitor de *feeds*, e cria um *link* apontando para o endereço do *site* de notícias onde está localizado o serviço RSS. Todas as vezes que o *site* publicar novas notícias, o usuário as visualiza em seu *software feed*, sem que seja necessário acessar o *site*. Se o usuário realizar o mesmo processo para outros *sites*, ele terá um resumo de notícias de diversos *sites* diferentes, sem necessitar visitar nenhum deles. Os resumos das notícias são transferidos automaticamente para o *software* do usuário. Quando o usuário tiver interesse em ler alguma notícia, basta clicar sobre o *link* e o *software* se encarrega de mostrá-la na tela.

Rucavina (2005) desenvolveu um serviço baseado em RSS para avisar aos usuários de uma

biblioteca a data de devolução dos livros emprestados.

Observa-se a grande importância das LM para o campo da Ciência da Informação. Inúmeras possibilidades e oportunidades vêm surgindo tendo como base as LM. As seções seguintes apresentam algumas das linguagens que possuem relação direta com a Ciência da Informação.

## HTML E CIÊNCIA DA INFORMAÇÃO

Apesar de a HTML ser uma “velha conhecida” da Ciência da Informação, torna-se necessário descrever alguns de seus aspectos visando à comparação com a XML.

Bush (1945) descreve em seu artigo “*As we may think*”, a visão de um computador dotado de um sistema de hipertexto denominado Memex. Outros visionários, como *Engelbart*, também deixaram importantes contribuições nessa área. A idéia de Bush foi finalmente concretizada por Tim Berners-Lee, ao criar a Internet e a linguagem HTML. A proposta de Bush, de criar termos associados que pudessem ser consultados rapidamente, foi concretizada por meio da criação de *links* em HTML. Documentos pequenos, ou mesmo documentos grandes, podem ser divididos e interligados por *links* que realizam associação entre termos.

As LM devem fornecer recursos para criar documentos com três componentes distintos e independentes: 1) estrutura: define a organização da informação, 2) conteúdo: a informação propriamente dita; e 3) estilo (ou formatação): define o visual de apresentação das informações ao usuário. Diversos benefícios podem ser alcançados quando esses três elementos são mantidos e manipulados de forma separada. O conteúdo poderá ser aproveitado em diversos contextos e apresentado de maneira mais adequada para cada tipo de aplicação. Esses três elementos são usados de forma mais explícita na linguagem XML, mas também são comuns em HTML.

Normalmente, o estilo de apresentação dos documentos HTML é armazenado em um arquivo externo conhecido pelo nome de “folhas de estilo”, ou CSS (*Cascading Style Sheets*), uma linguagem complementar usada para melhorar a qualidade de

apresentação dos documentos, uma grande deficiência da HTML.

As *tags* determinam o início e o fim do texto marcado como uma unidade ou elemento de informação. Por exemplo, o texto delimitado pelo marcador `<TITLE>` se refere ao título do documento, assim como o texto delimitado por `<P>` corresponde a um parágrafo. Os mecanismos de busca podem analisar o conteúdo delimitado pela *tag* `<TITLE>` a fim de verificar a ocorrência de uma palavra-chave pesquisada. É extremamente importante que o título de um documento HTML sempre utilize palavras-chave, isto é, termos que digam respeito ao conteúdo do documento. Além disso, é importante que o título esteja bem descrito, pois ele será exibido nos resultados de um mecanismo de busca. Empresas especializadas em criação de *Web sites* apontam que “mais da metade dos usuários da *Web* no mundo usa, todos os dias, um mecanismo de busca”. (SYMANTEC, 2005). Como os mecanismos de busca analisam o conteúdo do título, ele possui relação direta com a facilidade de recuperação do documento.

Outro aspecto muito importante se refere à utilização da *tag* `<META>`. Trata-se de uma *tag* que possibilita realizar uma série de tarefas, tais como redirecionar páginas para outro *site*, recarregar uma mesma página de tempos em tempos, produzir efeitos de animação durante a transição de uma página para outra e, o que nos interessa diretamente, a possibilidade de adicionar informações sobre o documento, isto é, de criar metadados. A *tag* `<META>`, como o próprio nome sugere, permite definir metadados por meio de um par de atributos chamados “*name*” e “*content*”. O atributo “*name*” armazena um parâmetro, enquanto o atributo “*content*” armazena o conteúdo do parâmetro. Observe a sintaxe seguinte: `<META name=“nome do Parâmetro” content=“conteúdo da informação”>`. Com isso, torna-se possível adicionar metainformações para que mecanismos de busca possam identificar melhor o conteúdo do documento.

Apesar da enorme importância da HTML para a disseminação da informação por meio da *Web*, ela é muito limitada no que diz respeito à semântica dos termos. Somente as *tags* `<TITLE>` e `<META>` podem ser usadas para fornecer algum significado ao conteúdo do documento. Essas marcações se

restringem ao cabeçalho do documento, permitindo a definição de metadados muito parciais, tal qual um sistema de fichas quando apresenta apenas alguns dados sobre uma obra. Não é possível fornecer significado a um termo como, por exemplo, “campo”. Para um leitor humano é simples distinguir seu significado dependendo do contexto onde o termo foi usado. Esse termo pode se referir ao “campo” da Ciência da Informação, ou a um “campo” magnético ou ainda a um “campo” de futebol. Essa interpretação, apesar de simples para humanos, é muito complexa para um mecanismo de *software*.

A HTML torna-se limitada para representar o significado das informações presentes no documento, pois ela não foi concebida para esse fim. A HTML foi desenvolvida para definir como a informação deve ser apresentada e não o que ela significa. O conjunto fixo de *tags* e atributos não permite que representações mais aprimoradas sejam criadas. Esse aspecto, além de dificultar o trabalho humano, praticamente impossibilita que computadores troquem informações entre si de maneira “inteligente”. Essa falta de flexibilidade da HTML impede que os diversos tipos de comunidades e organizações possam trocar documentos e informações de maneira mais efetiva por meio da *Web*, fator que culminou com a elaboração da linguagem XML, descrita a seguir.

## XML E CIÊNCIA DA INFORMAÇÃO

Antes de abordar efetivamente os aspectos essenciais da XML que possuem relação direta com a Ciência da Informação, será fornecida uma visão geral sobre a linguagem, assim como seus principais pontos comuns e divergentes em relação a HTML.

A linguagem XML foi criada por Jon Bosac da Sun, uma das principais empresas da área de informática. Assim como HTML, XML foi definida como um padrão de marcação para ser utilizado na Internet, constituindo-se uma versão simplificada da SGML, cujo objetivo principal foi fornecer aos desenvolvedores da *Web* uma maneira de definir e criar seus próprios marcadores e atributos em vez de estarem restritos ao esquema de marcação da HTML. O próprio significado da XML sugere essa característica, pois é uma linguagem de marcação extensível.

Da mesma forma que HTML, XML é um padrão aberto e independente de plataforma, entretanto, enquanto a HTML apenas trata de especificar a formatação de uma palavra ou um trecho de texto, a XML trata de criar estruturas para representar seu significado. Enquanto a HTML indica como algo deve ser exibido, a XML procura indicar o que a informação significa. Pode-se considerar que a XML é uma evolução da HTML (FURGERI, 2001). Apesar disso, tem ocorrido uma “convivência pacífica” entre HTML e XML. No princípio do aparecimento da XML muito se questionava se ela seria a substituta da HTML, fato que ainda não ocorreu. Conforme citado anteriormente, o que o W3C tem incentivado é que desenvolvedores em HTML criem seus documentos de acordo com a especificação da XHTML, atualmente em sua versão 2.0.

A XML pode ser usada em conjunto com a HTML para a elaboração de documentos para a Internet. O mais freqüente é inserir trechos de código XML em uma página HTML, processo conhecido pelo nome de *Data Islands* (SILVA, 2001). Isso permite que um mesmo documento escrito em XML possa ser reutilizado em diversos outros documentos.

A listagem do Quadro 1 apresenta um exemplo de estrutura criada pela XML. A estrutura criada pela XML apresentada no Quadro 1 é facilmente legível por seres humanos e máquinas. É fácil identificar que se trata de um catálogo com informações sobre livros. Cada livro possui um número de isbn, título, autor, editora, ano, preço e disponibilidade (poderiam existir muitos outros dados). É perfeitamente possível para um *software* reconhecer que o número 2001 corresponde ao ano de lançamento do livro, o

**Quadro 1.** Trecho de um documento XML para descrição de livros.

---

```
<livros>
  <livro isbn="85-719-4797-X">
    <titulo>Ensino Didático da Linguagem XML </titulo>
    <autor>Sérgio Furgeri</autor>
    <editora>Érica</editora>
    <ano>2001</ano>
    <preco>20,00</preco>
    <disponivel>sim</disponivel>
  </livro>
</livros>
```

---

que não aconteceria com um mesmo documento feito em HTML caso uma máquina o interpretasse. Na XML cada unidade de informação está devidamente delimitada por uma *tag* que fornece significado.

Pelos marcadores usados no Quadro 1 é possível detalhar as informações sobre os livros, criando dados sobre eles, isto é, metadados. Os livros passam a possuir diversas propriedades identificáveis e distintas; o mesmo poderia ocorrer com outro marcador. Por exemplo, a *tag* editora poderia conter outros elementos filhos - como nome, endereço, estado -, detalhando, assim, ainda mais a estrutura. Com esse pequeno exemplo é possível identificar que a XML é um importante recurso para a criação de metadados, que, por sua vez, são recursos de vital importância para a recuperação de informações (RODRIGUEUZ, 2002).

A XML permite representar o documento como uma estrutura padronizada, em forma de árvore. Dessa forma, torna-se perfeitamente possível para um *software* realizar uma varredura no catálogo com a finalidade de recuperar informações, que podem ser do tipo: quais os livros do catálogo que foram publicados pela editora Érica, ou quais os títulos dos livros que possuem a palavra XML em seu conteúdo, e assim por diante. A estrutura criada em forma de árvore, além de proporcionar procedimentos de pesquisa, torna possível a manipulação dos elementos do catálogo, permitindo inserção ou remoção de elementos. Existem diversas linguagens que atuam sobre a XML para realizar essas tarefas, tais como a XSLT (*eXtensible Style Language Transformation*) e a XQL (*eXtensible Query Language*), cuja descrição não é abordada por este artigo.

Pelo fato de XML ser uma linguagem que permite ao criador elaborar suas próprias *tags*, torna-se possível desenvolver outras linguagens de marcação a partir da XML. Diversos padrões para marcação de documentos podem ser construídos baseados na estrutura que a XML oferece. Já existem dezenas, talvez centenas, de outras linguagens que foram desenvolvidas a partir da XML nas mais diversas áreas do conhecimento. Isso possibilita que comunidades formem vocabulários próprios, fornecendo significado a termos específicos e utilizados em conjunto. A criação de novas linguagens é possível por meio da utilização do DTD (*Document Type Definition*) ou XML *Schema*, que é uma evolução daquele. Apesar de o DTD ser ainda bastante utilizado, esse artigo procura

apresentar os aspectos relevantes do XML *Schema* para o campo da Ciência da Informação.

Apesar de a XML permitir que se criem marcações na medida da necessidade, torna-se necessário considerar que as coisas não são “totalmente livres”. Para que um documento XML seja criado e possa ser compreendido em diferentes contextos é necessário que esteja de acordo com certas regras, definidas em outro documento chamado de XML *Schema*. Por meio dele se definem padrões que o autor deve obedecer para que o documento seja considerado válido. O Quadro 2 apresenta um documento XML completo e o Quadro 3 um possível XML *Schema* para realizar sua validação. O catálogo da biblioteca apresentado no Quadro 2 foi elaborado de acordo com as regras definidas pelo documento “biblioteca.xsd” apresentado em seguida no Quadro 3.

Vamos analisar as regras definidas pelo *Schema* (GRAVES, 2003) sem nos preocupar com questões muito técnicas:

- As *tags* do arquivo XML são declaradas no XML *Schema* por meio da palavra *element*,

- A linha 5 inicia as declarações do elemento de nome “biblioteca”. Ele deve possuir de um (minOccurs=”1” na linha 8) a mil (maxOccurs=”1000” na linha 8) elementos filhos de nome “livro”; as declarações correspondentes ao elemento “biblioteca” se encerram na linha 46. A linha 44 define que o elemento “biblioteca” deve conter um atributo obrigatório (*required*) de nome “universidade”, que, por sua vez, deve conter um conteúdo do tipo texto (*string*);

- As linhas 8 a 42 declaram um elemento de nome “livro” que deve conter a seguinte seqüência de elementos filhos:

- Linha 11: um elemento com o nome “titulo” de conteúdo tipo texto (*string*);

- Linha 12: de um a dez elementos com o nome “autor” de conteúdo tipo texto;

- Linha 13: um elemento com o nome “keywords” que, por sua vez, pode possuir de três a dez elementos com o nome “key” de conteúdo texto (linha 16);

- Linha 20: um elemento com o nome “editora” de conteúdo tipo texto (*string*);

- Linha 21: um elemento com o nome “ano” de conteúdo tipo inteiro (*integer*), com valores entre 1700 (linha 24) e 2010 (linha 25);

Quadro 2. Catálogo de livros em XML.

```

<?xml version="1.0" encoding="iso-8859-1"?>
  <biblioteca xmlns="http://www.w3schools.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3schools.com biblioteca.xsd"
  universidade="PUC">
    <livro isbn="85-7194-797-X">
      <titulo>Ensino didático da Linguagem XML</titulo>
      <autor>Sérgio Furgeri</autor>
      <keywords>
        <key>XML</key>
        <key>HTML</key>
        <key>DTD</key>
        <key>CSS</key>
        <key>XSL</key>
        <key>XSLT</key>
      </keywords>
      <editora>Érica</editora>
      <ano>2001</ano>
      <preco moeda="R$">20</preco>
      <disponivel>sim</disponivel>
    </livro>
  </biblioteca>

```

- Linha 29: um elemento com o nome “preco” de conteúdo numérico decimal (linha 32) que contenha um atributo obrigatório (required), com o nome “moeda” (linha 33) e de conteúdo tipo texto (*string*);

- Linha 38: um elemento com o nome “disponivel” de conteúdo tipo texto (*string*);

- Linha 40: o elemento de nome “livro” deve conter um atributo obrigatório de nome “isbn” de conteúdo tipo texto (*string*);

- Todos os elementos devem, obrigatoriamente, ser usados no documento XML na mesma ordem em que foram declarados no *Schema*.

Observa-se que o XML *Schema* contém regras rígidas que determinam a estrutura que o documento XML deverá assumir. Existem apenas alguns pontos flexíveis, mas que também estão sujeitos a determinadas regras. Por exemplo, o elemento “preço” da linha 29 é opcional (minOccurs=“0”), entretanto, quando for usado, ele deve estar imediatamente depois do elemento “ano” e imediatamente antes do elemento “disponível”. Podem existir de três a dez elementos “key”, mas não fora dessa faixa. Por esses

motivos, a linguagem XML permite a criação de documentos estruturados de forma quase rígida, porém ligeiramente flexíveis, diferentemente do exposto em Almeida (2002), onde XML é considerada uma linguagem semi-estruturada pelo fato de ser flexível. Nossa proposta é que a linguagem XML deva ser usada para a criação de documentos estruturados, porém com alguma flexibilidade.

De forma resumida, as principais definições do XML *Schema* que permitem criar documentos padronizados são:

- Definir os elementos permitidos e seu conteúdo (filhos, texto, vazio). Exemplos: o elemento <biblioteca> contém o filho <livro>; o elemento <editora> contém um texto.

- Definir quais são os elementos filhos, sua quantidade, e a ordem em que aparecem. Exemplo: o elemento <livro> contém o filho <titulo>, seguido de um ou mais filhos <autor>, seguido dos filhos <keywords>, <editora>, <ano>, seguidos talvez pelo filho <preco> e seguido pelo filho <disponivel>.

**Quadro 3.** Schema para o catálogo de livros (numeração de linhas não faz parte Schema).

```
1.      <?xml version="1.0"?>
2.      <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3.      targetNamespace="http://www.w3schools.com"
4.      xmlns="http://www.w3schools.com" elementFormDefault="qualified">
5.      <xs:element name="biblioteca">
6.      <xs:complexType>
7.      <xs:sequence>
8.      <xs:element name="livro" minOccurs="1" maxOccurs="1000">
9.      <xs:complexType>
10.     <xs:sequence>
11.     <xs:element name="titulo" type="xs:string"/>
12.     <xs:element name="autor" type="xs:string" minOccurs="1" maxOccurs="10"/>
13.     <xs:element name="keywords">
14.     <xs:complexType>
15.     <xs:sequence>
16.     <xs:element name="key" type="xs:string" minOccurs="3" max
Occurs="10"/>
17.     </xs:sequence>
18.     </xs:complexType>
19.     </xs:element>
20.     <xs:element name="editora" type="xs:string"/>
21.     <xs:element name="ano">
22.     <xs:simpleType>
23.     <xs:restriction base="xs:integer">
24.     <xs:minInclusive value="1700"/>
25.     <xs:maxInclusive value="2010"/>
26.     </xs:restriction>
27.     </xs:simpleType>
28.     </xs:element>
29.     <xs:element name="preco" minOccurs="0">
30.     <xs:complexType>
31.     <xs:simpleContent>
32.     <xs:extension base="xs:decimal">
33.     <xs:attribute name="moeda" type="xs:string" use="required"/>
34.     </xs:extension>
35.     </xs:simpleContent>
36.     </xs:complexType>
37.     </xs:element>
38.     <xs:element name="disponivel" type="xs:string"/>
39.     </xs:sequence>
40.     <xs:attribute name="isbn" type="xs:string" use="required"/>
41.     </xs:complexType>
42.     </xs:element>
43.     </xs:sequence>
44.     <xs:attribute name="universidade" type="xs:string" use="required"/>
45.     </xs:complexType>
46.     </xs:element>
47.     </xs:schema>
```



- Definir quais são os atributos de um elemento. Exemplo: O elemento <livro> contém o atributo “isbn”.

- Definir tipos de dados para elementos e atributos. Exemplos: o elemento <ano> contém um número inteiro entre 1700 a 2010; o atributo “isbn” contém um texto.

- Definir valores-padrão e fixos para elementos e atributos. Exemplo: o atributo “moeda” do elemento <preco> contém o texto “R\$” como padrão, apesar de outros tipos de moeda serem permitidos.

Para eliminar dúvidas vamos usar algumas analogias. Imagine o *Schema* como um molde a partir do qual objetos possam ser arranjados ou armazenados. Por exemplo, os livros se encaixam no *Schema* prateleira, o bolo se encaixa no *Schema* forma. Se você possui familiaridade com algum tipo de banco de dados (*Microsoft Access*, por exemplo), já deve ter percebido que é impossível inserir um dado do tipo texto, como, por exemplo, o nome de uma pessoa (o objeto), em um campo definido como tipo numérico (o *Schema*). As coisas “não se encaixam”. De forma semelhante, só é possível usar elementos XML quando declarados e definidos em *Schemas*. O uso de *Schemas* permite a definição de padrões para descrição de informações. Uma última analogia: se você precisa desenhar uma figura qualquer e só dispõe das formas geométricas quadrado, triângulo e círculo, não será possível usar uma elipse, pois o esquema não permite isso. Só é possível utilizar aquilo que é permitido.

Uma característica adotada por alguns *browsers* é a de apresentar o documento XML com a estrutura em forma de árvore, com todas as *tags*, diferente do que ocorre com os documentos HTML, nos quais as *tags* são suprimidas. Para que as *tags* sejam suprimidas, isto é, para que apenas o conteúdo seja apresentado na tela, deve ser usado outro arquivo, criado a partir de outra linguagem responsável por fornecer o estilo da formatação. A forma como o documento aparecerá na tela pode ser definida pela CSS, citada anteriormente, ou ainda por outras linguagens. De qualquer modo, a XML não tem a função de definir como as informações serão mostradas, ela se concentra em definir uma estrutura de representação, algo que forneça significado às informações.

## APLICAÇÕES DA XML EM CIÊNCIA DA INFORMAÇÃO

A parte inicial deste artigo já apresentou uma importante aplicação da XML no campo da disseminação da informação: o uso do RSS para o compartilhamento de informações entre *sites* e aplicações. Existem diversas outras aplicações nas quais a XML pode ser usada na área da Ciência da Informação; os parágrafos seguintes apresentam algumas delas.

Segundo Almeida (2002), “A definição de *tags* próprias confere à linguagem XML habilidades semânticas que possibilitam melhorias significativas em processos de recuperação e disseminação da informação”. Esses são os aspectos mais importantes da XML que interessam diretamente à Ciência da Informação; eles possibilitam a criação de metadados, isto é, estruturas de representação da informação. Por meio da elaboração de estruturas padronizadas, torna-se perfeitamente possível que diversas comunidades consigam fazer intercâmbio e comparação de dados, além de otimizar a recuperação de informações.

Um documento escrito em XML traz consigo o conhecimento de sua própria estrutura e significado, o que faz com que sua utilização seja mais adaptada às necessidades de automatização entre comunidades quaisquer: uma empresa pode compartilhar documentos com outras empresas, automatizando o processo de troca de dados, conceito conhecido no mercado como B2B (*Business to Business*). O mesmo pode ocorrer com bibliotecas ou qualquer outra instituição. Baseado no XML *Schema* do Quadro 6, diferentes bibliotecas podem elaborar seus catálogos, de forma padronizada, para permitir que uma ferramenta de *software* desenvolvida para a Internet possa realizar buscas no acervo de todas elas e trazer um resultado ao usuário. Muitas são as aplicações que podem ser desenvolvidas nessa área baseadas em XML.

Harris (1977) cita que o custo envolvido na digitalização e indexação de documentos é muito alto, uma vez que os documentos são elaborados sem nenhuma estrutura. O armazenamento digital de documentos sem estrutura, isto é, não plausível de recuperação, transforma-se num mero estoque difícil de ser recuperado. A XML permite que uma obra qualquer, como este artigo, por exemplo, possa ser

armazenado de forma estruturada e, portanto, pronto para ser recuperado. Dessa forma, é possível afirmar que a elaboração de um artigo em XML pode trazer benefícios tanto para quem o produz, quanto para quem o utiliza. O custo de indexação seria praticamente nulo e a recuperação muito mais eficaz. Em vez de elaborar artigos no formato “doc” ou “pdf”, seria mais eficaz se os artigos fossem armazenados diretamente em XML, muito mais passíveis de recuperação. Considere a estrutura do Quadro 8 que poderia ser utilizada para a elaboração de um artigo.

Se todos os artigos de uma base de dados fossem elaborados a partir da estrutura do Quadro 4 e fossem disponibilizados na *Web*, uma ferramenta poderia fazer dezenas e talvez centenas de pesquisas diferentes para recuperar a informação. A seguir são listadas algumas buscas possíveis:

- Procure no cabeçalho a ocorrência da palavra “marcação”;
- Procure nas palavras-chave a ocorrência da palavra XML;
- Procure no resumo e na introdução a ocorrência da palavra HTML;
- Procure nos títulos de seção pela palavra Ciência;
- Retorne todos os artigos que contêm o autor “Bush” nas referências.

A XML pode ser usada na melhoria da recuperação da informação no ambiente *Web*. Os usuários têm dificuldade em localizar informações relevantes na Internet dada a imensa quantidade de páginas existentes, criadas a partir de uma linguagem semi-estruturada como a HTML, que não fornece um modo adequado para descrever os conteúdos do texto, o significado fica perdido e os resultados de busca relativamente “burros”. Utilizando a XML, é possível tornar o serviço de busca mais “inteligente”, evitando, ou pelo menos minimizando, páginas irrelevantes, de acordo com o exemplo do Quadro 4.

A XML pode ser usada na criação de ontologias entre comunidades. Cada tipo de documento pode possuir termos que representam conceitos específicos às necessidades de informação da área. Para que um documento possa expressar as idéias do autor aos seus leitores, as partes envolvidas devem estar de acordo com o significado dos termos utili-

zados, isto é, devem ser elaborados a partir do mesmo *Schema*. Por exemplo, para a área de Biblioteconomia, pode ser essencial que metadados sobre livros contenham informações sobre sua localização física; outra área pode considerar isso irrelevante.

## LIMITAÇÕES DA LINGUAGEM XML

Apesar do exposto até aqui, e de todas as vantagens da XML, há de se considerar suas limitações quando, por exemplo, é necessário analisar dois documentos diferentes que seguem padrões diferentes e utilizam a mesma *tag*, pois *tags* com nomes iguais podem possuir significados e conteúdos diferentes. Como fazer se, por exemplo, a *tag* <titulo> é usada em dois documentos diferentes, representando informações diferentes? Uma alternativa a esse problema se refere ao uso de *namespaces*.

Segundo Marchal (2000), “*namespace* pode ser definido como um mecanismo para identificar os elementos da XML”. Trata-se de um prefixo associado à *tag* que permite torná-la exclusiva. Por exemplo, um documento elaborado para a área da Ciência da Informação poderia ter a *tag* <ci:titulo>, já para a área de computação poderia ser <co:titulo>, uma maneira de diferenciar, e tornar exclusiva, uma *tag* qualquer. O uso de *namespaces* torna possível a criação de um vocabulário controlado para descrever algum domínio do conhecimento.

A estrutura definida pela XML permite representar a mesma informação de maneiras diferentes. Por exemplo, suponha que seja necessário representar a seguinte informação: o autor Bush escreveu um artigo de título “*As We May Think*”. Qual é a estrutura correta para representar essa informação?

A estrutura do documento XML pode variar em função do *Schema* XML usado para validar o documento. Por causa disso, é possível representar uma mesma informação de maneira diferente. Considerando-se que diferentes comunidades utilizam estruturas diferentes para representar uma mesma informação, é improvável que um computador possa reconhecer o verdadeiro significado dos termos. Uma alternativa para esse problema é proposta em Abdalla (2003).

Quadro 4. Artigo estruturado em XML.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<artigo area="CI">
  <cabecalho>
    <titulo>O Papel das Linguagens de marcação para a Ciência da Informação</titulo>
    <autor>Sérgio Furgeri</autor>
    <publicador>TransInformação</publicador>
    <ano>2005</ano>
    <local>Campinas</local>
    <origem>Brasil</origem>
    <keywords>
      <key>XML</key>
      <key>HTML</key>
      <key>XML Schema</key>
      <key>Metadados</key>
    </keywords>
  </cabecalho>
  <corpo>
    <resumo>
      Texto contendo o resumo do artigo.
    </resumo>
    <introducao>
      Texto contendo a introdução do artigo.
    </introducao>
    <secao titulo="Visão geral das linguagens de marcação">
      <p>    Texto do parágrafo 1.    </p>
      <p>    Texto do parágrafo 2.    </p>
    </secao>
    <conclusao>
      Texto contendo a conclusão do artigo.
    </conclusao>
  </corpo>
  <referencias>
    Referências Bibliográficas.
  </referencias>
</artigo>

```

A XML possui limitação quanto ao tamanho da estrutura do documento, isto é, a ampliação do tamanho da estrutura pode representar um problema quando o número de registros for muito grande. Como representar um catálogo de uma biblioteca contendo milhões de registros? Antes de adicionar novos metadados em um documento (em nosso exemplo considere a inserção da data de publicação), o *Schema* deve ser alterado.

Em XML a estrutura gerada sempre se apresenta em forma de árvore, porém nem sempre as

informações podem ser assim representadas. Uma outra linguagem chamada RDF (*Resource Description Framework*) foi proposta pelo W3C com o propósito de reduzir as limitações da XML.

## A LINGUAGEM RDF E A CIÊNCIA DA INFORMAÇÃO

Assim como foi realizado com a descrição da XML, antes de apresentarmos exemplos de uso da

RDF para o campo da Ciência da Informação, vamos entender seus princípios básicos.

A linguagem RDF é uma terminologia desenvolvida pelo W3C para codificar, trocar e reutilizar metadados na *Web* (W3 CONSORTION, 2005b). Elaborada a partir da XML, tem como objetivo principal prover intercâmbio de informações entre aplicações sem a perda do significado. Para isso, define um vocabulário padrão para descrever coisas ou objetos e permitir a troca de metadados entre aplicações. A RDF permite criar declarações sobre esses objetos por meio de propriedades que representam um relacionamento entre recursos (MEISSNER, 2004). Uma declaração é realizada por meio de triplas do tipo “recurso-propriedade-valor”, em que:

- **Recurso** - é o sujeito de uma declaração. Pode ser um *website* ou parte dele, ou ainda um objeto não acessível via *Web* (livro, CD, etc). Um recurso pode ser acessado e reconhecido de forma única por meio de um URI (*Uniform Resource Identifier*). Um artigo científico é um exemplo de recurso.

- **Propriedade** - é o predicado de uma declaração. Trata-se de um atributo usado para descrever um recurso. Um artigo científico pode conter diversas propriedades: nome do autor, título do artigo, data de publicação, etc.

- **Valor** - é o objeto de uma declaração. Representa o conteúdo das propriedades. Em nosso exemplo, trata-se dos conteúdos referentes ao nome do autor, título do artigo, data de publicação.

Exemplos de declarações podem ser:

**www.noticias.com.br/saude.html** tem um **autor** chamado **Junior**

**www.noticias.com.br/saude.html** tem um **título** chamado **o celular faz mal a saúde**.

Observe que essas declarações contêm o formato: “<sujeito> tem <predicado> <objeto>”. Essa tripla definida pela RDF é representada por meio de um grafo em que o sujeito (recurso) é um nó (node) em forma de elipse, o predicado (propriedade) é um arco com uma seta apontando para o objeto (valor), cuja representação é um retângulo (se for um literal) ou uma nova elipse (se representar um outro recurso). O Quadro 5 apresenta um exemplo de declarações RDF e seu grafo correspondente.

Algumas considerações sobre o código do Quadro 5:

- Linha 1: o documento foi elaborado a partir da versão “1.0” da XML e possui o padrão “ISO-8859-1” para a codificação dos caracteres;

- Linha 2: aponta para um endereço Web que contém um *namespace* chamado “rdf” que define o vocabulário para a descrição de elementos em RDF;

- Linha 3: aponta para um endereço Web que contém um *namespace* chamado “dc” que define um vocabulário padrão de termos em metadados;

- Linha 4: aponta para um endereço fictício da *Web* que contém um *namespace* chamado “property” que define o vocabulário próprio para a descrição de propriedades. Isso foi realizado para demonstrar que vocabulários próprios podem ser criados de acordo com as necessidades;

- O restante das linhas descrevem as triplas declaradas no grafo.

Com relação ao grafo do Quadro 5 é possível interpretar o seguinte:

- O recurso “**www.noticias.com.br/saude.html**”, uma página imaginária da *Web*, tem uma propriedade chamada “**título**”, cujo conteúdo é “**o celular faz mal a saúde**”. Observe que a propriedade “**título**” possui referência a um *namespace* cuja função é justamente definir seu significado.

- O recurso “**www.noticias.com.br/saude.html**” possui também uma propriedade chamada “**autor**”, cujo conteúdo é “**Junior**”. Da mesma forma, a propriedade “autor” possui um significado próprio em uma URI. O objeto “Junior” poderia também possuir diversas propriedades como “*email*”, “*cargo*”, etc., ampliando o grafo e a cadeia de ligações entre recursos e objetos.

- As propriedades podem seguir padrões públicos do mercado, como é o caso da propriedade “título”, em nosso exemplo definida a partir do padrão *Dublin Core* em “http://purl.org/dc/elements/1.1/titulo”, ou seguir padrões próprios, como a propriedade “autor” definida em “www.noticias.com.br/propriedades/autor”.

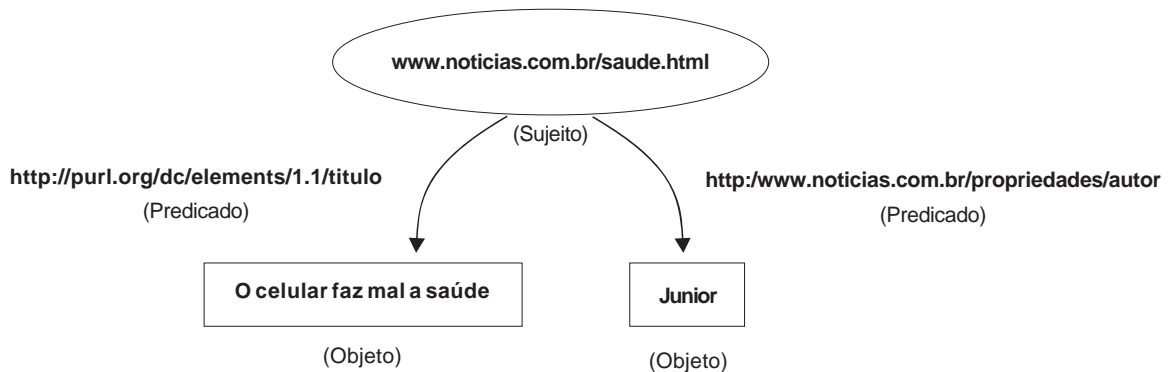
O modelo criado a partir da RDF possui aspectos semânticos que lhe conferem diversas vantagens sobre a XML:

**Quadro 5.** Declarações em RDF e grafo correspondente.

```

1. <?xml version="1.0" encoding="ISO-8859-1"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:dc="http://purl.org/dc/elements/1.1/"
4.     xmlns:property="http://www.noticias.com.br/propriedades/">
5.   <rdf:Description rdf:about="www.noticias.com.br/saude.html">
6.     <property:autor>Junior</property:autor>
7.   </rdf:Description>
8.   <rdf:Description rdf:about="www.noticias.com.br/saude.html">
9.     <dc:titulo>o celular faz mal a saúde</dc:titulo>
10.  </rdf:Description>
11. </rdf:RDF>

```



- A RDF resolve o problema da diversidade na representação da informação que ocorre em XML, criando ligações únicas entre recursos e estabelecendo vocabulários por meio de *namespaces*. A RDF elimina o problema da limitação do tamanho da estrutura enfrentada pela XML, criando ponteiros que unem documentos com estruturas menores.

- A RDF é mais indicada para a criação de metadados, pois um recurso é referenciado a um objeto por meio de um predicado com significado próprio. A ordem com que as declarações são realizadas não importa, diferente do que ocorre na XML.

- A RDF elimina o problema da representação da informação em forma de árvore, criando uma estrutura mais flexível em forma de grafos, possibilitando a formação de uma cadeia de informações e estabelecendo uma rede de conhecimento.

Com RDF é possível criar um vocabulário controlado para descrever um domínio do conhecimento, fato que torna possível a qualquer organização

publicar informações de maneira semântica. Com isso, agentes de *software* podem agir de maneira automática e inteligente sobre recursos *Web*, inferindo sobre o significado dos elementos.

A RDF opera em conjunto com a recomendação RDF *Schema* (RDFS) de Fevereiro de 2004. Trata-se de uma linguagem para descrição de vocabulário RDF (W3 CONSORTION, 2004), responsável pela definição do modelo de dados a ser seguido pelos documentos RDF e que provê mecanismos para descrição dos recursos e propriedades. Um modelo de dados é semelhante ao grafo apresentado no Quadro 5, que contém relações entre os recursos, definidos no documento RDF.

Com a RDFS é possível definir os termos (as triplas) que serão usados nas declarações dos documentos RDF, uma maneira de criar um vocabulário controlado. Esse vocabulário permite estabelecer relações e restrições entre os recursos. Relações tais como: o autor pertence a uma classe chamada pessoa, um *site* pertence a uma classe chamada recurso, um artigo que pertence a uma classe publicações

deve possuir um título, cujo conteúdo deve ser um texto, e assim por diante. São criadas regras e restrições que devem ser seguidas pelos documentos RDF.

De acordo com os conceitos expostos, observa-se que a RDF possui atribuições que lhe conferem diversas possibilidades de uso na área da Ciência da Informação, uma vez que permite criar relações entre documentos e estabelecer significado aos termos por meio de *namespaces*. RDF provê uma estrutura mais flexível que XML, aproximando-se da forma como os seres humanos relacionam informações, isto é, por associações. Um computador, que normalmente mantém dados em estruturas rígidas, passa a poder associar pedaços de informações com estruturas diferentes e de forma automática. Esse é o objetivo principal da *WebSemantica* proposta por Berners-Lee, isto é, dotar os computadores da possibilidade de fazer associações entre informações relacionadas (WEBSEMANTIC, 2005).

Existem ainda outras possibilidades com o uso da RDF. Agüera (2004) propõe a criação de um tesouro baseado em RDF. Outros autores propõem um padrão baseado em RDF chamado SKOS (*Simple Knowledge Organisation Systems*) para representação de tesouros e outros tipos similares de sistemas de organização de conhecimento (MILES, 2005). Como o padrão RDF é escrito em XML, pode-se afirmar que a XML fornece a estrutura fundamental para o desenvolvimento de tesouros.

## CONCLUSÃO

A Ciência da Informação encontra um importante suporte nas LM. Sem elas, talvez o seu campo de atuação fosse mais limitado, visto o grande número de oportunidades por elas proporcionado.

Pode-se dizer que a evolução das LM - da formatação em HTML para a estrutura com XML e a semântica com RDF - estabelece novas fronteiras a serem vencidas para a transformação da *Web*. Enquanto a HTML permitia apenas descrever algumas informações adicionais sobre um documento, por meio das *tags* <TITLE> e <META>, a XML passou a permitir que a própria estrutura do documento se transformasse num metadado. A RDF passa a estabelecer relações entre os documentos, permitindo que

mecanismos de busca realizem operações de inferência.

É preciso estabelecer novas formas de representação da informação por meio dessas linguagens. Os mecanismos de busca estão cada vez mais eficientes, no entanto há muito a evoluir. Espera-se que, num futuro próximo, seja possível fazer pesquisas como: "quais bibliotecas do Brasil possuem o livro X do autor Y disponível para retirada hoje?" Claro que é possível entrar em algumas bibliotecas na Web e fazer a busca de forma individual. Mas tente fazer uma pesquisa complexa desse tipo por meio de um mecanismo de busca. Se você digitar a seqüência "Biblioteca Brasil Titulo-do-Livro Nome-do-autor disponível" entre aspas, o mecanismo de busca provavelmente não retornará nenhum resultado; caso você decida retirar as aspas, o mecanismo pode retornar muitos resultados sem sentido. Bem, se você conseguir um resultado positivo, sorria, o futuro chegou. Brincadeiras à parte, a RDF representa uma nova esperança para o mundo desorganizado da Web.

Num mundo globalizado, preocupado com a questão da inclusão digital, nosso esforço deve se concentrar em desenvolver interfaces de comunicação universais que sejam disponíveis a todos os tipos de pessoas e comunidades, independentemente da linguagem, cultura e capacidades mentais ou físicas. As LM podem ser uma importante ferramenta para auxiliar esse processo.

É importante observar que existem ainda outras linguagens, não abordadas aqui, diretamente ligadas ao projeto da *Web Semântica*, como, por exemplo, a linguagem RQL (*RDF Query Language*) para realizar buscas em documentos RDF e a linguagem OWL (*Web Ontology Language*), usada na criação de ontologias.

Ainda existe pouca literatura brasileira a respeito do impacto das LM para o campo da Ciência da Informação. São relevantes os estudos que verificam os benefícios reais para processos de recuperação da informação a partir das possibilidades da XML e, principalmente, no que se refere aos padrões mais recentes, como o RDF e a OWL, as bases para a criação da *Web Semântica*. Espera-se que autores nacionais estudem e escrevam novos artigos sobre o assunto.

## REFERÊNCIAS

- ABDALLA, K.F. *A Model for semantic interoperability using Xml*. The Technical Resource Connection, Perot Systems. 2003. Available from: <<http://www.sys.virginia.edu/sieds03/proceed2003/proceedings/B102.pdf>>. Access: 20 Jan. 2006.
- ALMEIDA, M.B. Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares. *Ciência da Informação*, Brasília, v.31, n.2, p.5-13, 2002.
- AGÜERA, J.R.P. *Automatización de tesauros y su utilización en la Web Semántica*. 2004. Disponible en: <<http://www.w3.org/2001/sw/Europe/events/200406-esp/trabajo-final-extratesauros/trabajo-final-extratesauros.html>>. Acceso en: 13 May 2005.
- BAX, M.P. Introdução às linguagens de marcas. *Ciência da Informação*, Brasília, v. 30, n.1, p. 32-38, 2001.
- BUSH, V. As we may think. *Athlantic Monthly*, v.176, n.1, p.101-108, 1945.
- DAVIES, J.I. *Glossary the Terms relevant to Mobile Communications*. 2004. Available from: <<http://homepages.nildram.co.uk/~jidlaw/pages/glossary.html>>. Access: 4 May 2005.
- DORNFEST, R. *Writing RSS 1.0*. O'Reilly Publications. 2000. Available from: <[http://www.oreillynet.com/lpt/a/network/2000/08/25/magazine/rss\\_tut.html](http://www.oreillynet.com/lpt/a/network/2000/08/25/magazine/rss_tut.html)>. Access: 23 May 2005.
- FURGERI, S. *Ensino didático de linguagem XML*. São Paulo: Érica, 2001.
- GRAVES, M. *Projeto de banco de dados em XML*. São Paulo: Makron Books, 2003.
- HARRIS, J.L. Terminology change: Effect on Index Vocabularies. *Information Processing & Management*, v.15, p.77-88, 1977.
- MARCHAL, B. *XML: conceitos e aplicações*. São Paulo: Berkeley, 2000.
- MEISSNER, S. *Resource Description Framework*. 2004. Available from: <[http://en.wikibooks.org/wiki/XML:\\_Managing\\_Data\\_Exchange/RDF\\_-\\_Resource\\_Description\\_Framework](http://en.wikibooks.org/wiki/XML:_Managing_Data_Exchange/RDF_-_Resource_Description_Framework)>. Access: 18 Aug. 2005.
- MILES, A.; ROGERS, N.; BECKETT, D. *SKOS-Core 1.0 Guide - An RDF Schema for thesauri and related knowledge organization systems*. Available from: <<http://www.w3c.rl.ac.uk/SWAD/skos/1.0/guide/draft01.html>>. Access: 23 May 2005.
- RODRIGUEUZ, E.M. *Metadatos y recuperación de información: estándares, problemas y aplicabilidad em bibliotecas digitales*. Astúrias: Ediciones Treal, 2002. 429p.
- RUCAVINA, P. *Creating an RSS feed of the books you have checked out of the library*. Available from: <<http://ruk.ca/article/2290>>. Access: 25 May 2005.
- SILVA, O.J. *XML: aplicações práticas*. São Paulo: Érica, 2001.
- SYMANTEC. *Maximize o seu site para mecanismos de busca*. Disponível em: <<http://www.symantec.com/region/br/smallbusiness/howto/max.html>>. Acesso em: 23 Maio 2005.
- W3 CONSORTION. *About the World Wide Web Consortium*. Available from: <<http://www.w3.org/>>. Access: 23 May 2005a.
- W3 CONSORTION. *Resource Description Framework (RDF)*, 2004. Available from: <<http://www.w3.org/RDF/>>. Access: 17 Aug. 2005b.
- W3 CONSORTION. *RDF Vocabulary description language 1.0: RDF Schema*. 2004. Available from: <<http://www.w3.org/TR/rdf-schema/>>. Access: 17 Aug. 2005.
- WEBSEMANTIC, *Web site*. 2005. Available from: <<http://www.semanticweb.org/>>. Access: 26 May 2005.

