Ketib – Um processo de representação
de informações para textos complexos

Eduardo Santos Kerr

Trabalho Final de Mestrado Profissional

Instituto de Computação
Universidade Estadual de Campinas

Ketib – Um processo de representação de informações para textos complexos

Eduardo Santos Kerr

Julho de 2003

Banca Examinadora:

- Prof. Dr. Paulo Lício de Geus (Orientador)
  Instituto de Computação - UNICAMP
- Rev. Dr. Rudi Zimmer
  Escola Superior de Teologia do Instituto Concórdia
- Prof.ª Dr.ª Ariadne Maria Brito Rizzoni Carvalho
  Instituto de Computação - UNICAMP
- Prof. Dr.Jacques Wainer (Suplente)
  Instituto de Computação – UNICAMP

Co-orientador

- Prof. Fernando Antônio Vanini
  Instituto de Computação - UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Kerr, Eduardo Santos

K461k Ketib – Um processo de representação de informações para textos complexos / Eduardo Santos Kerr – Campinas, [S.P. :s.n.], 2003

Orientadores: Paulo Lício de Geus; Fernando Vanini
Trabalho final (mestrado profissional) – Universidade Estadual de Campinas, Instituto de Computação.

1. Recuperação da informação. 2. Estrutura de dados. 3. Processamento eletrônico de dados. I. Geus, Paulo Lício. II. Vanini, Fernando. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Titulo

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 30 de julho de 2003, pela Banca Examinadora composta pelos Professores Doutores:

Prof. Dr. Rudi Zimmer
EST - Instituto Concordia

Profª. Drª. Ariadne M. B. R. Carvalho
IC - UNICAMP

Prof. Dr. Paulo Lício de Geus
IC - UNICAMP

Ketib – Um Processo de representação de informações para textos complexos

Este exemplar corresponde à redação final do Trabalho Final devidamente corrigida e defendida por Eduardo Santos Kerr e aprovada pela Banca Examinadora

Campinas, 30 julho de 2003.

Prof. Dr. Paulo Lício de Geus
(Orientador)

Trabalho Final apresentado ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Computação na área de Engenharia de Computação.

Dedicatória

Esse trabalho é dedicado à minha querida família, minha esposa Denise, minha filha Ana Luisa e meu filho Eduardo, pela importância da compreensão e colaboração durante o tempo que dediquei para essa atividade acadêmica.

Aos meus pais Lysias e Norma, pelo exemplo de vida, de fé, pelo incentivo, mas principalmente pelo amor e carinho que tenho tido o privilégio de conviver em todos esses anos, e que estão entre as coisas mais preciosas que poderia receber.

Dedico também à memória do meu avô, Rev. William Kerr, que durante décadas ensinou o hebraico a uma geração de alunos que vieram a se tornar mestres dedicados. Durante anos William se empenhou na composição e elaboração da primeira gramática de português-hebraico no Brasil, publicada em 1940.

Sero sed serio.

Agradecimentos

Agradeço a valiosa ajuda dos Rev. Paulo Teixeira da SBB, Dr. Waldir Luz da Unicamp, Dr. Deomar Roos do Instituto Concórdia e do Dr. Paulo Benício do Instituto Mackenzie, pelo tempo dedicado ao esclarecimento de um obra tão complexa como a Biblia Hebraica Stuttgartensia e pela importante orientação sobre a bibliografia na área de pesquisa.

Agradeço a Sociedade Bíblica do Brasil, pelo incentivo profissional, suporte e apoio dado aos estudos e pesquisa do trabalho final do curso.

Por fim, agradeço a Deus, pelas formas inescrutáveis pelas quais transmitiu, inspirou e preservou o conteúdo das Sagradas Escrituras durante milhares de anos, para orientar o homem durante sua breve existência na terra.

Índice

Ketib – Um Processo para Representação de Informação em Textos Complexos

Abstract:

The phrase "complex text" applies to texts that possess transtextual relations within the source text and that relate to other supplementary texts, either for aggregating the different kinds of critiques and analyses, or just for better understanding. This work proposes a process for the representation of information in complex texts using the XML language and some related technologies. As part of the representation process, the model defined was one called Ketib, and it is applied, as a case study, to the Biblia Hebraica Sttutgartensia. The reason for selecting this text was its high degree of complexity, making it possible to validate the potential of the model when treating the representation and the codification of information of a multidimensional nature, containing different reference systems. The results are not restricted to the case study and demonstrate the possibilities of this process in texts in other domains.

Resumo:

O termo "texto complexo" é utilizado para classificar um texto que possua relações transtextuais dentro do texto de origem e com textos complementares, que sirvam para agregar os diferentes tipos de análise e crítica, ou simplesmente para permitir uma melhor compreensão. Nesse trabalho, é proposto um processo para representação de informação em textos complexos utilizando a meta linguagem XML e algumas tecnologias relacionadas. Como parte do processo de representação, foi definido um modelo denominado Ketib, que é aplicado, a título de estudo de caso, à obra Biblia Hebraica Stuttgartensia. Este texto foi escolhido por apresentar um alto grau de complexidade, tornando possível demonstrar o potencial do modelo no tratamento da representação e da codificação de informações com natureza multidimensional que contenham diferentes sistemas de referência. Os resultados não estão restritos ao estudo de caso e demonstram a viabilidade desse processo em textos de outros domínios.

Capítulo 1

Introdução

A representação de informação de texto em formato eletrônico tem sido objeto de pesquisas multidisciplinares. A busca de um modelo que atenda diferentes tipos de texto aponta para a utilização de um conjunto de tecnologias baseadas em uma meta-linguagem, conhecida como XML[Durusau2002].

As dificuldades variam de acordo com a complexidade do texto e com o volume de informações que serão representadas em formato eletrônico e da forma em que essas informações poderão ser recuperadas, processadas e manipuladas no grau de detalhes que for pretendido.

Este trabalho descreve um processo adotado para representar diferentes visões da informação contida em textos complexos.

O texto escolhido para implementar o processo de representação das informações foi a Biblia Hebraica Stuttgartensia, 5ª edição[BHS1997]. Essa obra possui um alto grau de complexidade, em que a informação pode ser explorada a partir de visões distintas tanto na forma como no conteúdo. Não existe até o momento uma representação que supra os requisitos mencionados.

Para ser possível a representação da informação, é necessário ter definida a estrutura do domínio a ser utilizada, bem como suas subcategorias relevantes e propriedades essenciais.

Além da solução do problema, é desejável que o tipo de representação utilizada possibilite apresentar os mesmos resultados, independente do software empregado para ler e processar a representação das informações. Essa propriedade é conhecida como interoperabilidade.

Embora o texto tratado neste trabalho seja relativo a um domínio específico, com características históricas, lingüísticas e teológicas próprias, o processo descrito pode ser aplicado a textos de outros domínios, tais como obras de conteúdo didático nas áreas de ciências exatas, humanas e biológicas.

O capítulo 2 apresenta conceitos importantes para definição do modelo Ketib[1], como informação, conhecimento, formas de representação, análise de domínio, os tipos de ontologia, a estrutura da BHS, as visões e relações transtextuais da BHS e as propriedades.

O capítulo 3 aborda as principais metodologias em codificação de texto no domínio definido para aplicação, incluindo a proposta do Ketib.

No capítulo 4, são relatadas as principais dificuldades encontradas na realização do trabalho e é feita uma comparação do Ketib com outros métodos já existentes.

O capítulo 5 é reservado para as conclusões e proposta de trabalhos futuros que podem ser desenvolvidos, partindo do modelo de representação descrito neste texto.

O nome do processo de representação apresentado neste trabalho, Ketib, é uma homenagem aos massoretas. Por várias gerações, esses judeus eruditos foram responsáveis pelas cópias das Escrituras e pelo ressurgimento de um idioma mantido apenas pelas tradições orais. A palavra ***ketib*** é associada a um tipo tradicional de anotação "*Ketib-Qere*", presente nas cópias do Antigo Testamento feitas através dos séculos.

---

[1]Em hebraico = כְּתִיב, significa: está escrito

Capítulo 2

Conceitos do Problema

Em geral, a representação de informação nos aplicativos é feita através de estruturas de dados específicas criadas pelos programadores, ou através das estruturas pré-definidas nos bancos de dados utilizados em conjunto com esses aplicativos. Os dois casos requerem modificações freqüentes, de acordo com o domínio do problema a ser estudado.

A falta de um modelo de representação padrão exige um grande esforço de programação na construção de novas máquinas de inferência e mecanismos de busca. Quando é possível utilizar as estruturas pré-definidas de bancos de dados, o preço da pouca flexibilidade e adequação aos vários tipos de problemas é uma barreira a ser superada. No final, essas dificuldades se resumem não só ao custo financeiro, mas também ao tempo de desenvolvimento, que pode ser fator limitante nas novas implantações, extensões e modificações nos sistemas de informação.

No trabalho aqui apresentando, ficará claro que não foi encontrado ainda um tipo de representação que suporte as informações em textos complexos. As opções existentes apresentam soluções parciais ou direcionadas a tipos específicos de domínio. Para compreender melhor essas dificuldades, considero pertinente um comentário feito por Marvin Minsky, publicado em um estudo sobre representação e recuperação de informação, no AI Magazine 1991, intitulado "Logical x Analogical or Symbolic x Connectionist or Neat x Scruffy", no qual ele comenta:

> "Nas décadas de 1960 e 1970, os estudantes perguntavam freqüentemente, 'que tipo de representação é a melhor?' e eu respondia geralmente que nós necessitaríamos mais pesquisa antes de responder a isso. Mas agora eu daria uma resposta diferente: 'para resolver problemas realmente complexos, nós teremos que usar diversas representações diferentes'" [Minsky1991].

Baseado na observação de Minsky, o trabalho foi desenvolvido tendo como princípio o fato de que: um modelo que possa representar textos complexos deve comportar diferentes tipos de representação.

Como resultado das pesquisas e dos trabalhos de representação de informação e conhecimento nos últimos 20 anos, tem havido um envolvimento multidisciplinar dos desenvolvedores e têm crescido as aplicações de sistemas de informação nas áreas de ciências humanas e biológicas.

O modelo ideal que está sendo buscado enfrenta outros desafios, como por exemplo, não criar mais uma linguagem proprietária, permitir interoperabilidade, flexibilidade de extensão e reusabilidade. Nessa busca por um modelo mais adequado é preciso investir no estudo de ontologias e procurar adotar padrões abertos de tecnologia.

2.1 SGML

Em 1986, após alguns anos de trabalho, a linguagem de marcação SGML (*Standart General Markup Language*), atingiu o status de padrão internacional. Essa linguagem teve suas origens na linguagem GML proprietária da IBM, e que representou um grande avanço no campo da marcação e codificação de texto. Os recursos e mecanismos apresentados no padrão SGML são extremamente flexíveis, e os autores classificam SGML como uma metalinguagem. Por definição uma metalinguagem tem propriedades que permitem definir outras linguagens.

O lado negativo dessa linguagem é a complexidade e o alto custo para criar produtos de software que implementem o padrão definido.

As duas linguagens que tiveram papel fundamental para o desenvolvimento da Web são dialetos da SGML:

- HTML (Hyper Text Markup Language), que era simples na sua implementação e alcançou popularização de forma muito rápida. A falta de recursos mais poderosos logo se tornou um obstáculo para os anseios dos usuários,

- XML (eXtended Markup Language), que manteve a característica de meta-linguagem, porém menos complexa que SGML. Com boa aceitação, passa a ser a base de muitos dos padrões definidos a partir de 1999.

A proposta descrita é baseada nas tecnologias desenvolvidas pelo Consórcio World Wide Web (W3C) que tem como princípios à interoperabilidade, utilização de padrões abertos e comprometimento com a evolução técnica [www.w3c.org/Consortium]. A linguagem XML é a base do conjunto de tecnologias adotadas nesta proposta, que são: XML-S, RDF, RDF-S e Xlink. No capítulo 3 serão apresentados mais detalhes sobre essas tecnologias.

2.2 A Informação

Na obra BHS[2], impressa, a quantidade de informações que os editores apresentam exige um complexo conjunto de referências, símbolos, marcação de texto e tabelas suplementares que servem como fonte de consulta para compreensão da codificação.

Não é objetivo deste trabalho formalizar o conceito de *informação*. Na verdade, na literatura da tecnologia da informação não existe consenso sobre o que é *informação*. Nesse trabalho é utilizada uma versão simplificada da definição de *dado*, *informação e conhecimento*. O subconjunto de conceitos usados é extraído dos trabalhos de Setzer, Davenport e Devlin. Esses autores apresentam divergências entre si quando analisados integralmente; porém, os conceitos usados aqui no processo de representação e codificação não apresentam divergências.

[2] Maiores detalhes sobre a obra BHS são fornecidos no item 2.5

Definições:

- Dado: "É uma seqüência de símbolos quantificados ou quantificáveis. Portanto, um texto é um dado. De fato, as letras são símbolos quantificados, já que o alfabeto, sendo um conjunto finito, pode por si só constituir uma base numérica... qualquer texto constitui um dado ou uma seqüência de dados." [Setzer2001]

- Informação: "É o dado acompanhado de semântica". [Davenport2002 e Setzer2001]

Embora não seja tratado diretamente a representação de conhecimento nesse trabalho, é oportuno definir representação de conhecimento:

- Representação de conhecimento: É uma forma de codificar a informação apreendida e/ou estruturada com o objetivo de manipular esse conhecimento.[Devlin2001]

2.2.1 Representação de informação

As técnicas propostas para representar o conhecimento podem ser aplicadas de forma simplificada para representar informação. Os recursos necessários para codificar conhecimento apresentam maior complexidade na sua formulação.

Na representação de conhecimento, as estruturas criadas para organizar e manipular esse conhecimento necessitam de elementos adicionais às informações codificadas que permitam explicitar propriedades como correlação, associação, regras condicionais, coeficiente de certeza etc. Tais estruturas são determinantes para o sucesso no mapeamento do processo dedutivo, necessário às bases de conhecimento.

Como conseqüência, a manipulação da informação codificada é uma tarefa de menor complexidade do que manipular conhecimento, uma vez que a necessidade do complexo processo de tomada de decisão numa máquina de inferência é substituído por algoritmos que se baseiam na manipulação de dados.

No escopo deste trabalho são utilizados conceitos de representação do conhecimento, muito embora não seja objetivo do trabalho criar uma base de conhecimento ou um sistema especialista para interpretações teológicas ou literárias.

Com o método adotado, é possível obter benefícios indiretos tais como uma maior facilidade de efetuar transformações de formato, criar filtros e consultas sobre as diferentes visões do conteúdo codificado.

2.2.2 Modelos de Representação para Ketib

Alguns dos trabalhos desenvolvidos para representação simbólica do conhecimento nas últimas décadas formam a base conceitual do modelo utilizado neste trabalho, o Ketib. Os pontos iniciais nas pesquisas realizadas apresentam conceitos aplicáveis para solucionar os problemas na representação e codificação de informação em textos complexos.

Entre os textos pesquisados estão os trabalhos na área de representação de informação e conhecimento e a área de codificação e marcação de textos propriamente dita. Começando com os trabalhos na área de representação de informação, coloco em posição de destaque:

> Frames[Minsky1974]- Procura agrupar de forma estruturada elementos e atributos que ajudem a representar o estado de um cenário ou a descrição de uma cena. Essa estrutura é conhecida como um *frame*. Os *frames* podem ser agrupados em vários conjuntos e podem manter um outro tipo de frame de mais alto nível com diversas informações desses conjunto. Minsky explora minuciosamente o potencial desse tipo de estrutura, onde algumas destas informações tratam de como manipular os *frames*, algumas tratam do que pode acontecer em seguida na cena, outras, sobre o que fazer se estas expectativas não forem confirmadas.
>
> Nesse trabalho, é possível identificar conceitos de meta-frame, frames hierárquicos e frames com heranças múltiplas, frames com regras de transformação definidas e frames com propriedades dinâmicas.

O artigo pode ser considerado como um dos fundamentos importantes para as áreas de inteligência artificial e análise orientada a objetos.

Semantic Net[Quillian1968] – Contemporâneo de Minsky, Quillian propôs um modelo de representação de conhecimento que serviu de base a muitas das pesquisas na área de sistemas especialistas e base de conhecimento.

O termo "rede semântica" (*semantic net*) surge na tese de doutorado de Ross Quillian, que o introduziu primeiramente como uma maneira de falar sobre a organização da memória semântica humana, ou a memória para conceitos da palavra.

A idéia de uma rede semântica, isto é, uma rede de conceitos associativos, é muito mais antiga, de acordo com Anderson e Bower[Anderson1973]. Eles demonstram que é possível encontrar exemplos dessa metodologia desde a época de Aristóteles.

As redes semânticas foram concebidas especificamente como uma forma de representação que permitia armazenar o significado das palavras. A proposta de Quillian, bem como a grande maioria dos trabalhos que se seguiram, baseada nesses conceitos, visavam a armazenar a parte "não-emocional" do significado, ou seja, as propriedades objetivas das coisas, ao invés de armazenar que forma/sentimento se poderia ter a respeito dessas coisas.

Web Semantic[Berners-Lee2001] – No início da década de 90 do séc. XX, Berners-Lee percebeu que o crescimento da Internet teria uma projeção exponencial em poucos anos e, conseqüentemente, os mecanismos de busca de informação na rede se tornariam inviáveis. Berners-Lee apresentou o conceito de Web Semantic, em que seria necessário adotar semântica nas informações que eram colocadas na Web, permitindo aos mecanismos de busca escolher o conjunto de páginas mais adequado ao pedido enviado.

Os trabalhos evoluíram, e com a adoção e padronização da linguagem XML pelo W3C como uma linguagem mais poderosa e flexível que o HTML na codificação do conteúdo para internet, novos recursos passaram a ser oferecidos. Nos últimos dois anos um conjunto de novas linguagens e padrões baseado no XML vem sendo adotado e incorporado, aceleradamente, nos aplicativos para comércio eletrônico e em todos os setores, público e privado. Particularmente, para a Web Semantic, Berners-Lee destaca a importância de modelos de representação para metadados, como RDF[3] e RDFS[4][Berners-Lee2001], ambos baseados em XML.

No campo da codificação de texto os trabalhos mais importantes que serviram de fonte de consulta e referência foram:

What Text really is?[Durand1990]. Trabalho que apresentou a tese "Ordered Hierarchy of Content Objects" (OHCO) para codificação e processamento de textos eletrônicos, propondo uma rígida estrutura hierárquica para representação/codificação de texto na área das ciências humanas.

Refining our Notion of What Text Really Is: The Problem of Overlapping Hierarchies[Durand1996] – Revisão da tese OHCO pelos seus autores, devido às dificuldades de representar a sobreposição hierárquica de textos com visões múltiplas da informação contida. Durand propõe o modelo OHCO-3, que define hierarquias de perspectivas e sub-perspectivas; contudo deixa algumas classes de problemas de codificação de textos em aberto, visto que mesmo o modelo OHCO-3 não apresenta soluções práticas.

Guidelines for Text Encode Initiative (TEI)[TEI2002]. O trabalho mais completo e genérico no campo da codificação de texto é utilizado como referência e padrão na maioria das iniciativas comerciais, acadêmicas e governamentais. Teve início em 1987 e foi baseado nos padrões da linguagem SGML. Em julho/2002 foi publicada a versão P4, com revisões e adaptações para XML.

3

4 RDF e RDFS estão definidos no item 3.2

O modelo Ketib aplica os conceitos de *Web Semantic* que atualmente vem sendo usados pelo W3C com a utilização de *Resource Description Framework* (RDF). No capítulo 4 serão comentados os detalhes desse recurso.

2.3 Análise de domínio – Ontologia do domínio

O termo *Análise de Domínio* foi introduzido por Neighbors com a seguinte definição: "A Análise de Domínio é uma tentativa de identificar os objetos, operações e relações entre o que peritos em um determinado domínio consideram importante". [Neighbors1981]

Apesar desta definição informal do termo contribuir com uma boa idéia inicial, uma definição mais rigorosa se faz necessária como base para as discussões que aparecem no decorrer do capítulo. A seguir são descritos alguns termos necessários à construção desta definição.

Arango entende que, intuitivamente, a análise de domínio pode ser considerada equivalente à atividade de análise de requisitos convencionais na engenharia de software[Arango1994]. Estendendo esses conceitos para um meta-nível, ao invés de explorar requisitos de uma aplicação específica, os requisitos explorados dizem respeito a uma família de aplicações de uma determinada área. Arango define dois conceitos adicionais :

a) Domínio do problema: O domínio do problema representa um conjunto de itens de informação presentes em um certo contexto do mundo real, inter-relacionados de forma bastante coesa, e que despertam o interesse de uma certa comunidade. Esta definição cobre duas perspectivas:

- Domínio do problema como um conjunto de problemas correlatos para os quais existe conhecimento suficiente capaz de produzir soluções;

- Domínio do problema como uma taxonomia de componentes que torna explícitas as partes comuns de aplicações presentes e futuras identificadas como similares.

É importante mencionar que essa caracterização de *Domínio do problema* é dependente da comunidade que o aborda, ou seja, diferentes grupos podem ter diferentes visões do que seria, por exemplo, o domínio de *gerência de recursos*.

Outro aspecto importante dessa caracterização diz respeito à necessidade de existência de conhecimento suficiente que pode ser aplicado para a resolução dos problemas, o que abre possibilidade para realização de processos baseados em conhecimento para a identificação e aquisição de informações[Prieto-Díaz1991].

b) Modelo do Domínio: Pode ser descrito como um sistema formal de termos, relações entre termos e regras de composição de termos, regras para raciocínio que utiliza estes termos e regras para mapeamento de itens do domínio do problema para expressões neste modelo e vice-versa.

Resumindo, Modelo do Domínio define entidades, operações, eventos e relações que abstraem similaridades e regularidades em um determinado domínio, formando uma arquitetura de componentes comuns às aplicações analisadas e também cria modelos que tornam possível identificar, explicar e prever fatos difíceis de serem observados diretamente.

Depois de pronto, este modelo é útil para auxiliar na discussão e solução de problemas que apresentem ambigüidades e exigem tomada de decisão. Funciona como um repositório de conhecimento comum, auxiliando de forma direta a comunicação. Além disso, permite o aprendizado e reuso em um nível mais alto de abstração [Arango1994].

2.3.1 Ontologia - Definição

Ontologia é um tema que tem sido estudado em diversas áreas, como por exemplo: Filosofia, Linguagem e Cognição, Ciência da Informação e Ciência da Computação.

Dentro de uma mesma área podem ser encontradas diferentes definições e classificações de ontologia. A seguir serão apresentadas algumas das definições de ontologia.

- Na área de Filosofia:
  - Em 1647 Johannes Clauberg utiliza o termo ontologia no seu trabalho "*Elementa Philosophiae sive Ontosophiae*", quando afirma que assim como a ciência que trata sobre Deus chama-se Teosofia ou Teologia, seria apropriado chamar de Ontosofia ou Ontologia a ciência que trata de seres em geral, seus nomes e propriedades[Gilson1952].

- Na área de Linguagem e Cognição:
  - Ontologia refere-se a tudo que existe no mundo composto por objetos, mudanças e relações entre eles. Ontologia pode ser baseada no mundo, na mente/intelecto, na cultura ou na linguagem [Dahlgren1995].

- Na área de Engenharia do Conhecimento
  - Guarino[Guarino1997] define ontologia como uma caracterização axiomática do significado do vocabulário lógico; para Sowa[2000] a ontologia define os tipos de coisas que existem no domínio de uma aplicação .
  - Swatout & Tate[Tate1999] definem ontologia como um conjunto de conceitos e termos que podem ser usados para descrever alguma área do conhecimento ou construir uma representação para o conhecimento.
  - Chandrasekaran[Chandrasekaran1999] define que ontologias são teorias de conteúdo sobre os tipos de objetos, propriedades de objetos e relacionamentos entre objetos que são possíveis em um domínio de conhecimento específico.

Os esforços empregados na formalização de análise do domínio vieram ao encontro dos anseios de pesquisadores das áreas de inteligência artificial, engenharia de conhecimento e sistemas de informação.

Em 1993, Clancey defendeu a necessidade de mudança no foco do desenvolvimento dos novos sistemas especialistas. Ela argumentava que: "a Engenharia de Conhecimento deve ser voltada para a modelagem de sistemas, e não para tentar reproduzir a maneira como os especialistas raciocinam"[Clancey1993]. Essa visão passou a ter grande aceitação e como conseqüência ofereceu um novo conceito na estruturação da informação. As bases de conhecimento passaram a ser vistas como um produto de uma atividade de modelagem e não como um repositório de conhecimento especializado.

Essa mudança de foco levou os pesquisadores a recorrerem aos conceitos e a teorias no campo da filosofia. Alguns desses conceitos, provavelmente da época de Aristóteles ou mesmo antes, vieram do estudo da ontologia, e outros, do séc. XVII, estudados por Locke, vieram do estudo da epistemologia. Esses conceitos foram adaptados à ciência da computação e tecnologias de informação no final do séc. XX.

Os dois principais pesquisadores que trabalharam na formalização da Ontologia, como disciplina incorporada à área de representação de conhecimento, foram Thomas Gruber, da Universidade de Stanford, EUA, e Nicola Guarino, do Ladseb-CNR, Itália. Em geral, os artigos dessa área fazem referência a definições feitas por Gruber e aos aperfeiçoamentos na formalização concisa dos conceitos que Guarino desenvolveu [McGuinness2001].

Noy e McGuinness, professoras da Universidade de Stanford, um dos núcleos de pesquisa em desenvolvimento de bases de conhecimentos mais consagrados nessa área, destacam cinco motivos para incorporar o estudo de ontologias aos novos sistemas:

- Formalizar e explicitar o domínio a ser estudado. Estruturar as informações na área que estiver sendo estudada de forma a eliminar inconsistência e ambigüidade. A

utilização de uma notação formal pode facilitar a verificação e validação automática da especificação.

- Compartilhar e compreender a informação estruturada entre pessoas e agentes de software. Contribui na obtenção de consenso dos especialistas e elicitação de conhecimento de diversas fontes.
- Possibilitar a reutilização do conhecimento do domínio. Obtendo um vocabulário de consenso e permitindo que o conhecimento na camada do domínio possa ser especializado em diferentes aplicações, servindo a diferentes propósitos, por diferentes equipes em diferentes pontos no tempo e espaço.
- Separar dois tipos de conhecimento: conhecimento do domínio e conhecimento operacional. Noy e McGuinness descrevem a configuração de um produto com seus componentes de acordo com uma especificação (conhecimento do domínio) e a implementação um algoritmo para configurar um pedido do produto (conhecimento operacional). Por exemplo, em uma aplicação com dois domínios distintos, *computador* e *elevador*, e usar o mesmo algoritmo para configurar elevador ou computador.
- Análise do conhecimento. Permite que a partir da especificação de um domínio disponível, haja uma maior facilidade para análise de reusabilidade, extensão e atualizações desse domínio.

É importante acrescentar que o uso de ontologias para definição de domínios pode apresentar dificuldades. O'Leary[O'Leary1997], por exemplo, identificou os seguintes problemas:

- A escolha de uma ontologia é um processo político, já que nenhuma ontologia pode ser totalmente adequada a todos os indivíduos ou grupos.
- Ontologias não são necessariamente estacionárias, isto é, necessitam evoluir. Poucos trabalhos têm enfocado a evolução de ontologias.
- Estender ontologias não é um processo direto. Ontologias são, geralmente, estruturadas de maneira precisa e, como resultado, são particularmente vulneráveis a questões de extensão, dado o forte relacionamento entre complexidade e precisão das definições.

- A noção de bibliotecas de ontologias sugere uma relativa independência entre diferentes ontologias. A interface entre elas constitui, portanto, um impedimento, especialmente porque cada uma delas é desenvolvida no contexto de um processo político.

Os tipos de ontologias, segundo Guarino[Guarino1997, 1998], podem ser classificadas, com base em seu conteúdo, nas seguintes categorias:

- *ontologias genéricas*: descrevem conceitos bastante gerais, tais como, espaço, tempo, matéria, objeto, evento, ação, etc., que são independentes de um problema ou domínio particular;
- *ontologias de domínio*: expressam conceituações de domínios particulares, descrevendo o vocabulário relacionado a um domínio genérico, tal como Medicina e Direito.
- *ontologias de tarefas*: expressam conceituações sobre a resolução de problemas, independentemente do domínio em que ocorram, isto é, descrevem o vocabulário relacionado a uma atividade ou tarefa genérica, tal como, diagnose ou vendas;
- *ontologias de aplicação*: descrevem conceitos dependentes do domínio e da tarefa particulares. Estes conceitos freqüentemente correspondem a papéis desempenhados por entidades do domínio quando da realização de uma certa atividade;
- *ontologias de representação*: explicam as conceituações que fundamentam os formalismos de representação de conhecimento.

A Figura 2-1 mostra a relação entre as categorias de ontologias classificadas por Guarino.

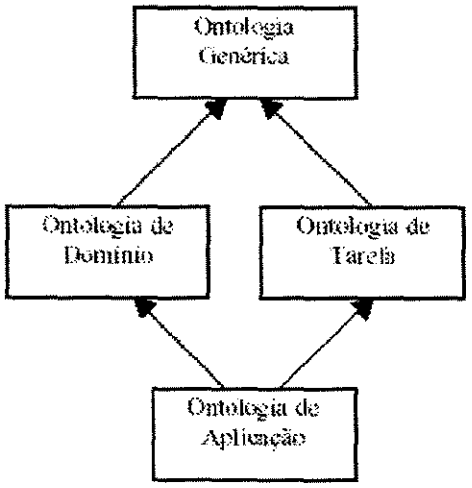

Fig. 2-1 - relação entre as categorias de ontologias.

A construção de ontologias genéricas apresenta três preocupações principais:

- A definição do objeto
- Como o objeto se comporta no contexto
- Como os objetos se relacionam

As discussões sobre a criação de uma ontologia genérica estão fora do escopo deste trabalho; no entanto para trabalhos futuros, principalmente no campo da lingüística, será preciso uma avaliação mais profunda das estruturas que serão necessárias e quais são suas relações.

Guarino tem atuado na organização e formalização dos conceitos de ontologia relacionados à área de sistemas de informação. Ele propõe que as ontologias sejam construídas segundo seu nível de generalidade.

Os conceitos de uma ontologia de domínio ou de tarefa devem ser especializações dos termos introduzidos por uma ontologia genérica. Os conceitos de uma ontologia de aplicação, por sua vez, devem ser especializações dos termos das ontologias de domínio e de tarefa correspondentes[Guarino1998].

As ontologias de domínio são construídas para serem utilizadas na solução de uma classe definida de problemas. Essas ontologias são as mais comuns de serem desenvolvidas, sendo que diversos trabalhos são encontrados na literatura, enfocando áreas como:

- química[Gómez-Pérez1996],
- modelagem de empreendimento - TOVE (Toronto Virtual Enterprise),
- codificação de texto (TEI) [TEI2002],
- identificação de conteúdo DC (Dublin Core),
- representação de codificação genética – GO (Gene Ontology Consortium),
- distribuição de energia – CIM (Common Information Model)[PICA2001],
- PRISM (Publishing Requirements for Industry Standard Metadata).

As quatro últimas ontologias mencionadas (**DC, GO, CIM e PRISM**) têm sua descrição baseada em RDF, e estão em uso em diferentes setores, a saber:

DC - Documentação em Geral: criado originalmente no Metadata Workshop, em Dublin, Ohio/EUA, como um conjunto de elementos para descrever propriedades dos documentos. Pode ser implementado com diversas estruturas (HTML/meta tags, XML/DTD, XML-S). Em 2002, o *Dublin Core Metadata Initiative* definiu como representação padrão dos elementos do DC, o modelo RDF implantado em XML[www.dublincore.org].

GO - Genética: com o objetivo de padronizar o vocabulário para descrição de produtos genéticos, permitindo que as anotações das bases de dados possam ser compartilhadas, oferecendo as evidências documentadas nas pesquisas já realizadas. Mantida pelo Gene Ontology Consortium [www.geneontology.org]

CIM - Energia Elétrica: com objetivo de definir uma semântica comum para sistemas geradores de energia, especificar seus atributos e relacionamentos. Essa representação permite a troca de informações dos modelos na indústria de energia elétrica. Esse modelo é mantido pela organização Electric Power Research Institute (EPRI)[deVos2001].

PRISM - Informações Editoriais: é uma especificação de metadados voltada às necessidades das editoras. Criado e mantido pela IDEAlliance (International Digital Enterprise Alliance) que busca soluções e padronizações para as indústrias na área editorial[www.prismstandard.org].

2.3.2 Delimitação do domínio

Tradicionalmente, quando um domínio de representação de um texto genérico é reduzido para codificar uma obra específica, ocorre uma simplificação da representação e do modelo usado. Esse modelo reduzido é adequado apenas às obras pertencentes ao domínio do texto específico. No entanto, devido à grande complexidade do conteúdo da obra BHS e à necessidade de representar uma grande variedade de elementos literários, se for aplicado o modelo na BHS, esse mesmo modelo pode ser utilizado para representar textos em outros domínios.

O conteúdo da BHS não entra no campo da analise lingüística, o que pode simplificar a ontologia e o modelo de representação, pois a necessidade de prover um conjunto de visões de forma a atender ao estudo lingüístico dos idiomas hebraico e aramaico apresenta um grau adicional de complexidade na codificação. Por exemplo, passa a ser importante a marcação de cada letra das palavras e até mesmo o posicionamento dos acentos e pontuações esta sujeito a variações (mais à esquerda, à direita ou ao centro de uma letra). Mesmo com essas necessidades, em trabalhos futuros, seria possível estender as estruturas do processo Ketib para incorporar as novas classes, objetos e propriedades específicas da lingüística.

O ponto de partida na escolha da ontologia mais adequada ao texto escolhido reside no trabalho do TEI[TEI2002]. Esse trabalho é classificado como **Ontologia de Domínio**.

Seguindo a proposta de Guarino, o próximo passo na criação de uma estrutura para representação das informações seria definir uma especialização da ontologia de domínio, criando dessa forma uma **Ontologia de Aplicação**. Para definir essa especialização foi adotado o trabalho desenvolvido pelo Bible Technologies Group (BTG), na criação do

padrão OSIS[5] (Open Scripture Information Standard), baseado em XML-S. Algumas modificações são propostas para utilização do OSIS e a camada de metadados é introduzida para configurar o Ketib.

2.3.3 TEI como ontologia de domínio

Esse grupo de estudo tem empenhado esforços nos últimos 15 anos na definição e formalização da codificação de textos. As diretrizes definidas por esse grupo estão contidas em *Guidelines for Text Encode Initiative*[TEI2002]. Para chegar à lista final dos elementos e atributos definidos por esse grupo de trabalho, foi elaborada uma ontologia para os mais variados tipos de textos. Inicialmente, os elementos e atributos descritos eram baseados na linguagem SGML para construção das diversas estruturas de representação dos textos. Recentemente as estruturas definidas foram atualizadas e expandidas para contemplar o padrão XML 1.0. A última versão, P4, foi publicada em julho de 2002.

O documento da versão P4 serve como base para a ontologia de textos em quaisquer que sejam suas áreas de expressão. O custo dessa abrangência é alto, pois como conseqüência não existe linguagem que implemente e utilize todas as definições. Além da complexidade já existente, o documento desta versão deixa para futuras versões a solução de alguns problemas, entre eles a representação de texto com múltiplas hierarquias sobrepostas. No final do capítulo 31, do documento da TEI, versão P4, é feita a seguinte observação:

> "Esse capítulo será intensamente revisado e expandido para futura versão".

As alternativas para tratar hierarquias múltiplas de forma concorrentes, apresentadas a seguir, são:

1. A diretiva *Concur*, presente somente na linguagem SGML.
2. *Milestones*, elemento de marcação de texto que não possui comprimento.

[5] O padrão OSIS é comentado no item 2.3.4

3. Fragmentação, divide logicamente um elemento em duas ou mais partes, eliminando a sobreposição.
4. União Virtual, monta os elementos na ordem necessária a satisfazer cada visão.
5. Codificação Redundante, duplica a descrição das estruturas que existem em mais de uma visão.

A diretiva *Concur* do SGML, o recurso de "união virtual" e a codificação redundante, permitem a codificação de visões simultâneas; a fragmentação é eficiente na solução de problemas que apresentem uma sobreposição de hierarquia; e o uso de *milestones* apenas introduz uma "marca de fronteira" dessas visões no texto, não permitindo a manipulação das estruturas.

As soluções apresentadas no capítulo 31 oferecem diferentes formas na solução dos problemas em textos complexos. Essas soluções, contudo, têm sido criticadas por serem incompletas, difíceis de implantar ou ineficientes[Durusau2002].

Outros pesquisadores propuseram soluções alternativas ao TEI e as mais conhecidas são:

- *Mecs, Multi-Elements Code System*, de 1993, que evolui para Texmecs, Trivially Extended MECS, em 2001. Desenvolvida na Universidade de Bergen, Noruega, por Claus Huitfeldt. Essa solução cria uma nova linguagem para representação de documentos complexos.
- *Stand-off Markup*, de 1997, proposta por Thompson e McKelvie, da Universidade de Edimburgo. Baseada em hiperlinks, implementa a codificação de marcação externa ao texto.
- *JITT, Just in Time Tree*, de 2002, proposta por Durusau e O'Donnel. Baseado na quebra da sintaxe do XML, extraindo a visão desejada numa etapa de pré-processamento, gerando cada uma das visões em XML válido, somente quando pedido.

A evolução dessas soluções alternativas tem sido lenta, e quase a totalidade das propostas de codificação e representação de textos, nas mais variadas áreas, ainda tomam como base o documento de diretrizes da TEI e evitam textos com hierarquias múltiplas.

2.3.4 OSIS como ontologia de aplicação

Apresentado em 2001, a primeira versão do OSIS não contemplava elementos estruturados que permitissem a marcação de capítulos e versículos, sendo, por isso, de pouca aceitação.

Atualmente a versão 1.1 é formada por 63 elementos, e se propõe a oferecer recursos para codificação de qualquer texto teológico produzido pelas Sociedades Bíblicas Unidas, tendo como objetivo explícito "um formato comum para muitas visões"[OSIS v1.1].

A versão atual incorpora elementos adicionais que permitem a codificação da estrutura de capítulos e versículos e utiliza a solução de fragmentação para tratar as ocorrências de sobreposição de hierarquias. Utiliza o XML-S para definição dos elementos, atributos e tipos no padrão OSIS, incorpora os elementos do padrão internacional conhecido como Dublin Core (DC). O padrão DC é usados para informações de catalogação da obra, tais como, editor, autor, editora, ISBN, etc.

2.4 As relações transtextuais

Além da tarefa de definir a ontologia, que deve ser feita sobre o domínio do texto e das definições das dimensões, é importante definir quais os tipos de relação que estão presentes entre as dimensões do texto. Dependendo dessas relações, a forma de representar e codificar as informações pode implicar numa maior complexidade da estrutura definida.

Para examinar as relações textuais, foram utilizados os conceitos do trabalho que Gérard Genette publicou, em 1982, que é considerado um dos mais importantes no campo da

crítica textual, "Os Palimpsestos[6]", onde ele analisa e classifica as relações que podem surgir em um texto. Genette chama essas relações de "transtextuais"[Genette1982].

Segundo Genette, cinco são os tipos de relações transtextuais:

1. Intertextualidade, considerada como a presença efetiva de um texto em outro. É a co-presença entre dois ou vários textos. Recurso utilizado no *aparato crítico*[7].

2. Paratextualidade, representada pelo título, subtítulo, prefácio, posfácio, notas marginais, epígrafes, ilustrações... Este campo de relações é muito vasto e inclui as notas marginais, as notas de rodapé, as notas finais, advertências, e tantos outros sinais que cercam o texto, como a própria formação da palavra indica. Presentes na *Massorá Magna*[8] e no texto, podendo ser consideradas ainda algumas ocorrências no cólofon e no aparato crítico.

3. Metatextualidade, vista como a relação crítica, por excelência. É a relação de comentário que une um texto a outro texto. Presente na *Massorá Magna e Parva*[9], e no aparato crítico.

4. Arquitextualidade, que estabelece uma relação do texto com o estatuto a que pertence – incluídos aqui os tipos de discurso, os modos de enunciação, os gêneros literários etc, em que o texto se inclui e que tornam cada texto único. Presentes na visão canônica com o sistema de referência, na visão da estrutura textual, na representação de poesia e na visão histórica, com as perícopes.

5. Hipertextualidade. Toda relação que une um texto (texto B – hipertexto) a outro texto (texto A – hipotexto). Presentes nas massorás.

Na época da publicação do trabalho de Genette o termo hipertexto não tinha o significado que se popularizou com o surgimento dos navegadores de Internet.

---

[6] Termo usado para um texto que foi sobrescrito por um novo conteúdo. Ver glossário. Exemplo na fig. 2-2
[7] Aparato crítico – Ver item 2.5.1
[8] Massorá Magna – Ver item 2.5.1

Para a representação das relações no Ketib será utilizada a tecnologia *Xlink*, que é um subconjunto do XML. Xlink foi padronizado em 2001 e simplifica a representação do tratamento das relações transtextuais definidas por Genette.

2.5 Texto complexo - BHS

A obra BHS, foi impressa pela primeira vez em 1967, resultado de 40 anos de pesquisas de um grupo seleto de estudiosos. O resultado desse trabalho foi concentrado em um só livro, a somatória de quatro obras distintas: o texto original proveniente dos papiros, ainda na forma consonantal com o trabalho dos *massoretas* na pontuação do texto; as anotações marginais, a *Massorá Parva*; as referências do texto de uma obra paralela, a *Massorá Magna*; e por fim o *aparato crítico*, elaborado por diversos autores durante a preparação do texto impresso.

As conseqüências de unificar, de forma concorrente, conteúdos gerados por diferentes autores, de diferentes culturas, numa linha do tempo que abrange pelo menos 3500 anos (aproximadamente, do séc. XV a.C ao séc. XX d.C), com anotações de naturezas distintas, justifica o uso do termo "texto complexo".

A partir do séc. IV d.C, os livros cristãos passaram a ser escritos em códices (do latim codex), palavra derivada de *caudex*, que era uma pequena tábua coberta de cera na qual se escrevia com um estilete metálico (*stylus*). "Reunidos por um cordão que passava por orifícios feitos no alto dos exemplares, os códices ficavam em forma de livro, portando bem mais prático de serem manuseados que os rolos de papiros" [Almeida, A. e Costa, M. 1992].

Muitos papiros antigos foram encadernados no formato de codex. As Figuras 2-2 e 2- mostram um dos códices mais importantes para os estudiosos do Antigo Testament escrito em grego, denominado *Codex Sinaiticus*. A Figura 2-2 traz um exemplo de palimpsestos, que se refere ao texto original que foi sobrescrito posteriormente.

---

[9] Massorá Parva – Ver item 2.5.1

Num primeiro instante foi pensado em uma simplificação na representação do conteúdo da BHS, reduzindo de quatro para dois o número de obras distintas. Essa simplificação teria como o texto principal o *Codex* Leningradense[10], também chamado *Codex* L (ver figs. 2-4 e 2-5), e o aparato crítico seria considerado apenas como notas dos editores. Ainda assim o texto continuaria sendo altamente complexo, porque:

- As notas que compõem o aparato crítico possuem várias relações transtextuais, o que torna complexa a representação das informações. Além de necessitar uma grande quantidade de símbolos, abreviações e estrutura de referência numérica para minimizar ao máximo a ambigüidade.
- *Codex* L tem uma composição de elementos que possui um alto grau de complexidade, pois na época não havia divisão de capítulos, e as notas da *Massorá Parva* não eram numeradas, apenas posicionadas à margem de cada linha que possuía comentários
- Quando havia questões sobre a exatidão das anotações, outras fontes de referência eram utilizadas. As mais confiáveis eram os códices, que nem sempre estavam em perfeito estado de conservação devido ao primitivo processamento manual utilizado.

[10] Codex Lenigradense – Biblioteca Nacional da Russia, em St. Petersburgo, datado de 1008 d.C

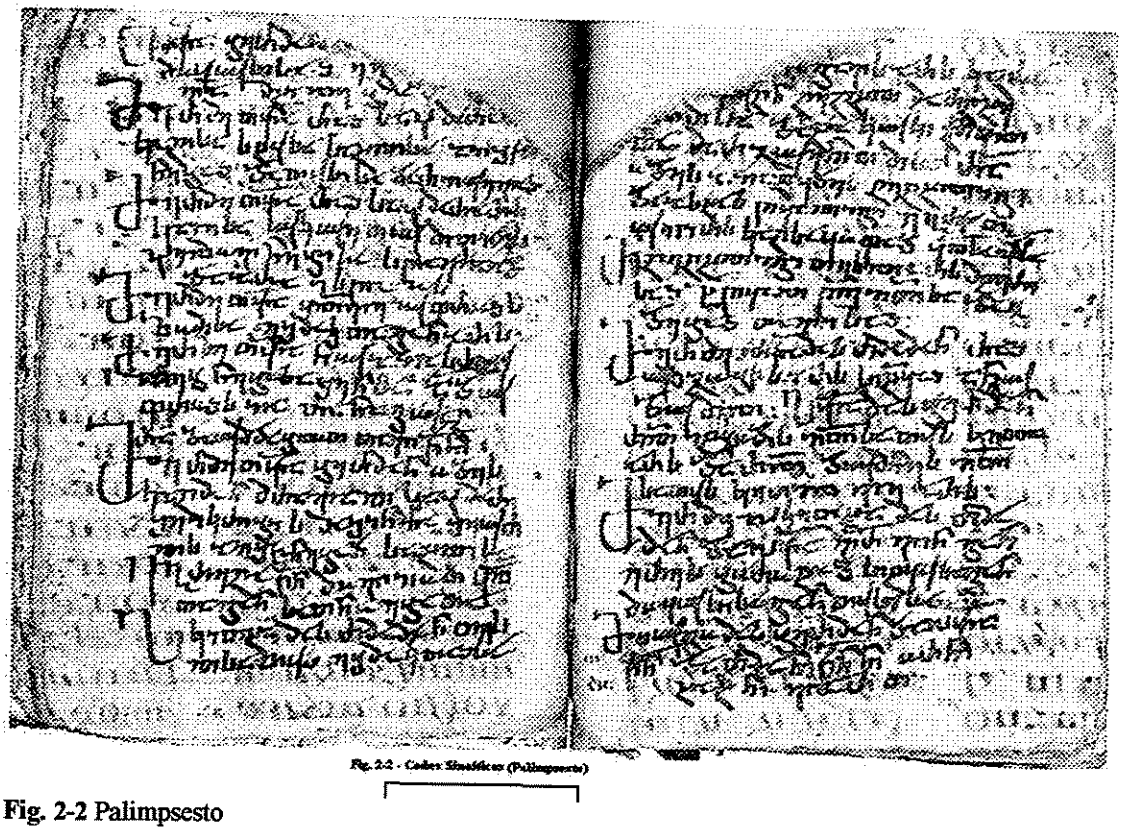

Fig. 2-2 Palimpsesto

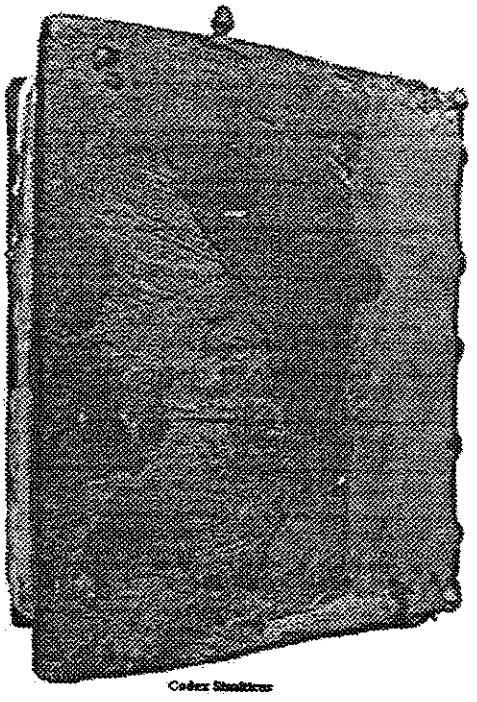

Fig. 2-3 Codex Sinaiticus

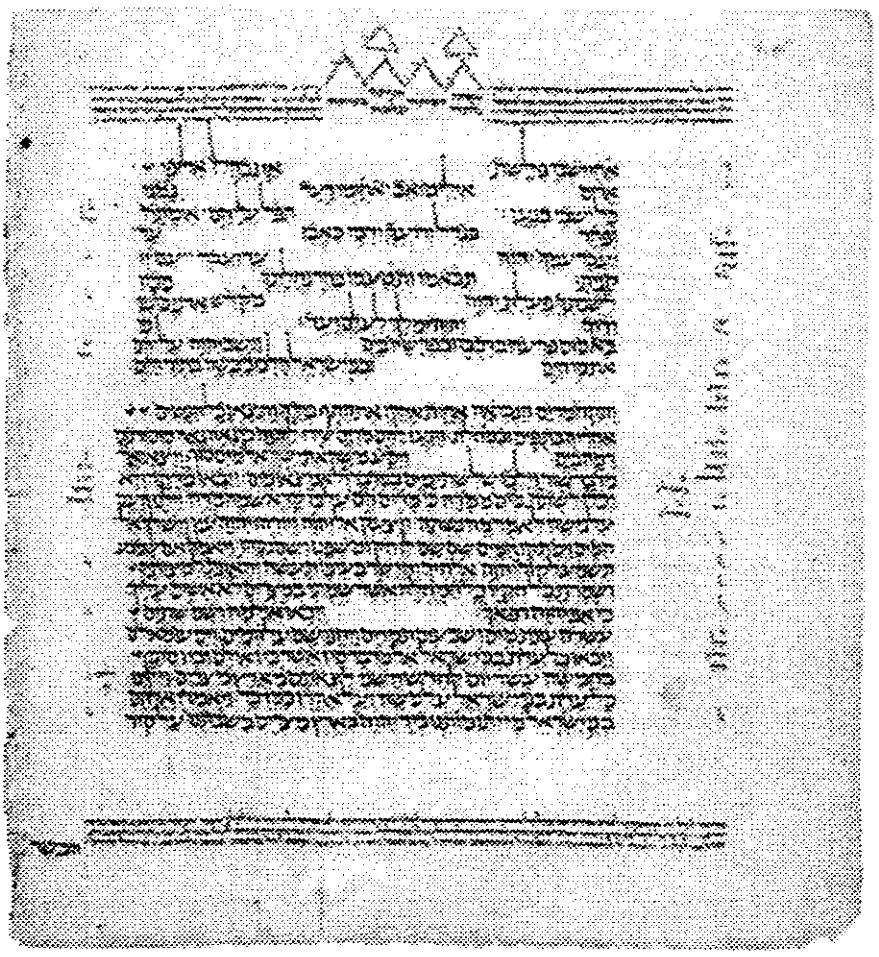

Fig. 2-4 – Codex L - Êxodo 15.22 – 16.3

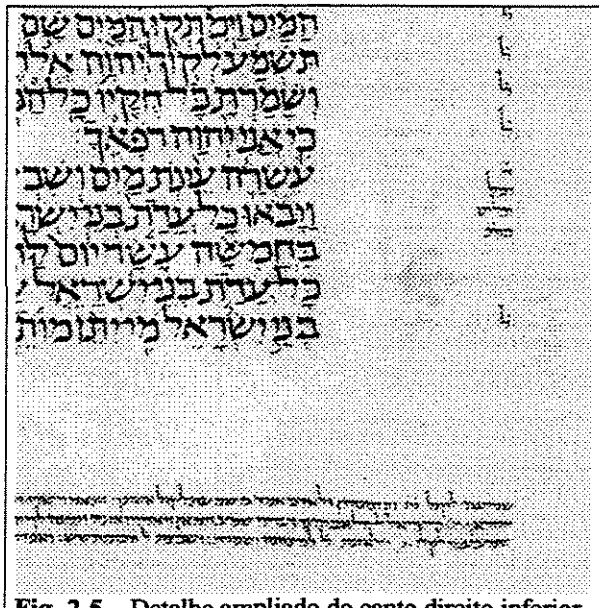

Fig. 2-5 – Detalhe ampliado do canto direito inferior da fig. 2-4

Codex L

Copiado no Cairo, entre 1008 e 1010 d.C.

Contém, aprox., 60.000 notas da Mp e 4271 notas da Mm

Composta de aprox. 1000 páginas.

As páginas, até o séc X d.C, eram feitas com o couro de animais (*parchment)*

Em especial, o Codex L é o mais conservado de todos códices (em hebraico) do Antigo Testamento.

O texto anterior à criação dos códices estava escrito na linguagem consonantal, isto é, o conteúdo escrito não tinha no alfabeto símbolos que representassem as vogais. A linguagem consonantal não é uma característica exclusiva do hebraico; outras escritas antigas também usavam a forma de escrita consonantal.

Os textos contidos nos códices são constituídos pelo texto modificado pelos massoretas. Foram os massoretas que introduziram a pontuação vocálica no idioma hebraico. Esse mecanismo foi responsável pelo renascimento do idioma de um povo que estava disperso na face da terra, pois eles não possuíam território geográfico nem governo formal, e certamente foi um passo definitivo na definição da atual escrita hebraica.

A palavra hebraica Ketib, na escrita consonantal KTB[11], quer dizer "**o que está escrito**". O nome do processo de representação apresentado neste trabalho é uma homenagem aos massoretas, pelas formas engenhosas que foram utilizadas através dos séculos para manter a fidelidade aos textos originais, dos papiros aos códices e, finalmente, ao livro impresso. Segundo a tradição, a prática de verificação formal das cópias dos textos das Escrituras Sagradas, desenvolvida pelos massoretas, existia seis séculos antes de Cristo, e foi iniciada pelo sacerdote Esdras, sendo esse autor conhecido como "escriba da lei de Deus". (Mencionado no livro de Esdras 7.10-12).

Em muitos casos, essa pontuação vocálica provocava uma escrita e (ou) pronúncia de duas ou mais palavras semelhantes. Nesses casos os massoretas não modificavam o original, mas adicionavam uma nota marginal com a nova forma de pronúncia; essa modificação é conhecida como *Qere*, קְרֵא que em hebraico quer dizer "**leia-se**".

O texto bíblico da BHS registra o que está escrito *(Ketib)*, a *Massorá Parva* registra a anotação do massoreta *(Qere)*, de como o texto deveria ser lido. Esse tipo de anotação, eventualmente, foi usado também para documentar os pontos que apresentavam divergência de ordem gramatical no processo de cópia, ou como forma de proteger o nome divino.

Outra forma engenhosa, que ajudou a dar consistência às muitas cópias que eram produzidas, foi a utilização de estatísticas de ocorrências de palavras ou expressões, anotadas na *Massorá Parva* e a introdução de *cólofons*, em geral ao final dos livros, que continham indicações do número de letras e seções que foram escritas. Sem exagero, pode-se afirmar que os recursos das estatísticas auxiliaram na tarefa de consistência e integridade de conteúdo durante os séculos em que foram produzidas as cópias do texto original. Esse recurso de verificação dos massoretas utiliza conceitos semelhantes aos que vieram a ser utilizados centenas de anos depois na área de processamento de dados, o *checksum*.

Além dos métodos práticos mencionados acima, havia rígidos rituais que deviam ser observados pelo seleto grupo de pessoas que realizavam as cópias. Contudo, os que estudam e observam as Escrituras Sagradas, acreditam que sem a ação providencial de Deus na preservação destas obras, não seria possível, nos dias de hoje, o acesso, à leitura e estudo de seu conteúdo.

A descoberta de vários pergaminhos antigos no deserto da Judéia, na região do Mar Morto, a partir de 1947 (até 1954), dentro das cavernas de *Qumran*, pôde comprovar que, mesmo através dos séculos e das inúmeras cópias que foram feitas, o texto conhecido das mais antigas e melhores fontes dos textos bíblicos, quando comparado com os descobertos recentemente, não apresentaram variantes significativas.

Os papiros descobertos em *Qumran* são as cópias mais antigas dos textos bíblicos até hoje conhecidas, em torno de 200 a.C (ver Figuras 2-6 e 2-7). Infelizmente, esses documentos não estão completos, representando fragmentos da obra do Antigo Testamento. Apenas o livro do profeta Isaías está completo e em bom estado de conservação. As primeiras obras completas de crítica textual específica, baseadas nesses pergaminhos, começam a ser publicadas em 2003, quase 50 anos após terem sido encontradas.

---

[11] em hebraico consonantal=כתב

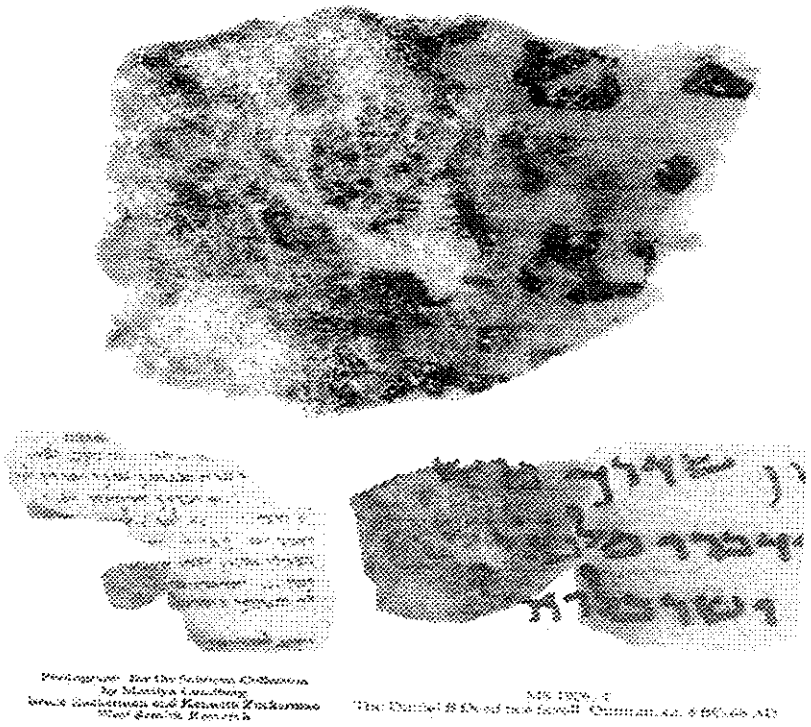

Fig. 2-6 Fragmento do livro de Daniel encontrado em Quramn

Fig. 2-7 Isaías cap. 1 – Livro completo encontrado em Qumran

O trabalho aqui proposto tem o objetivo de representar o que está escrito no texto, sem tentar introduzir um nível adicional de interpretação do texto e a efetiva codificação dessa interpretação.

Para aplicação deste modelo é necessário ter a definição da ontologia do domínio do texto que está sendo usado, conhecer as dimensões, também chamadas de visões do texto, que serão necessárias representar e de que forma essas visões se relacionam.

2.5.1 BHS - Estrutura do texto

O conteúdo da BHS foi dividido em subconjuntos bem definidos (ver figura 2-8), sendo que cada um dos conjuntos apresentam propriedades e relações distintas:

- Texto bíblico: Esse texto foi gerado a partir dos papiros copiados através dos séculos, tendo como conteúdo o texto hebraico puramente consonantal e acrescidos de acentuações, pontuações, cantilenas e cólofons. O texto bíblico era copiado por uma classe especial de pessoas conhecidas como *sopherim*, e posteriormente eram pontuadas gerando os atuais caracteres diacríticos do atual idioma hebraico; essa tarefa era atribuída às pessoas conhecidas como *nakdanim*. Além da representação das vogais, muitas vezes a pontuação apresenta significados sintáticos, lingüísticos e litúrgicos. Um mesmo sinal pode representar duas funções diferentes. Essa transformação no texto foi executada ao longo de aproximadamente 500 anos.
- *Massorá Parva*: Composta das anotações dos massoretas com referências ao *Qere*, e outras referências à *Massorá Magna*. Possui muitas notas estatísticas, e algumas notas explicativas. Esse subconjunto de anotações é conhecido também com *Massorá Marginalis*.

Referências à *Massorá Magna*: Esse subconjunto é formado por anotações em cada página da BHS com referências a um texto, que não está incluído no livro, conhecido como *Massorá Magna*. Essa obra foi composta com base na análise das diferentes fontes dos papiros, do conhecimento e tradição oral do povo judeu e de comentários teológicos

dos estudiosos que se dedicavam exclusivamente a essa tarefa. O seu conteúdo é um volume superior a todo o conteúdo da BHS. Essas referências são mencionadas pela *Massorá Parva* e pelo *aparato crítico.*

Fig. 2-8 – Gênesis 1.1-16 (BHS)

Além do TEXTO COPIADO, a figura destaca a **Mp** (*Massorá Parva*), as referências da **Mm** (*Massorá Magna*) e o **Ap** (*Aparato Crítico*).
Os sistemas de numeração das notas foram criados pelos editores da BHS .

- *Aparato Crítico*: Conteúdo com anotações coordenadas pelo primeiro editor da obra, Dr. Kittel, e pelos outros colaboradores que trabalharam durante quatro décadas na análise e codificação do conteúdo para o formato impresso. Possui referências internas à BHS e externas, a outras fontes de consulta em formato de códices ou livros, além das anotações de crítica textual. Inclui um complexo conjunto de símbolos e referências cruzadas ao texto, que documenta e justifica decisões dos editores ao optarem por trechos que apresentavam divergências. Na composição dos símbolos, são utilizados o alfabeto ocidental, o grego e o hebraico, além de um conjunto de mnemônicos listados em tabelas suplementares.

O livro contém ainda seções com tabelas de símbolos, notas introdutórias e textos que não apresentam dificuldades na representação, servindo como documentação do processo de compilação.

2.5.2 BHS – Visões do texto

Para aplicação da proposta deste trabalho, o método Ketib, foram definidas para a obra BHS as seguintes divisões do livro, suas dimensões e representação simbólica:

- *Canônico* – Apresenta os livros que formam o cânon do Antigo Testamento e divide o texto nas unidades criadas no séc. XVI, gerando um sistema de referência composto de capítulos e versículos. A divisão adotada pela BHS, embora não seja a única, é aceita como a mais adequada e mais utilizada; em alguns casos essa divisão é adotada com ligeiras variações por outros editores. As subdivisões são:
    - **L**, livro (ex: Gn, Ap, Mt),
    - **R**, referência numérica(ex: 1.1, 149.2) que identifica o capítulo e verso.

  Estas dimensões formam as principais unidades de referência deste trabalho. São vitais para esse estudo, pois o processo primário de identificação do texto é utilizado nas outras camadas de informação. Pode-se dizer que sua importância é

análoga ao sistema de coordenadas de longitude e latitude utilizada para navegação marítima.

- *Estrutura textual* – Criada pelos massoretas entre o séc. V e X, divide o texto em unidades devocionais (chamadas perícopes) e sentenças indivisíveis. As perícopes são subdivididas em 3 unidades menores. Para representar essas estruturas, foi adotada a nomenclatura:
  - **P** (perícope)
    - **Pa**
    - **Pb**
    - **Pc**
  - **S** (sentença indivisível).
  - **PO** (estrutura de poesia)

As perícopes são numeradas seqüencialmente dentro de cada livro. As sentenças que são indivisíveis não possuem identificação e possuem um valor mais histórico do que lingüístico, com exceção da ocorrência na poesia.

No caso específico do conteúdo poético, é necessário recorrer a elementos adicionais na sua estrutura de representação. Alguns sinais usados no texto poético possuem outros significados no texto normal. A importância desses sinais se torna ainda mais evidente em virtude do idioma hebraico não possuir o conceito de rimas na poesia. A poesia era formada com "rimas de idéias". Por isso a correta pontuação de sentenças assume importância fundamental na correta interpretação.

- *Histórico* – Para efeito de implementação do Ketib, essa dimensão é subdividida em 4.
  - **Mp** (Massorá Parva)
  - **Mm** (Massorá Magna)
  - **Cs**, os cólofons de verificação de integridade adicionados pelos massoretas,
  - **Ap**, o aparato crítico criado pelos editores do livro no séc. XX

A *Massorá Parva* foi dividida em três categorias pelos editores da BHS, de forma a facilitar a identificação para os estudiosos que teriam acesso ao texto impresso (ver fig. 2-8), pois originalmente os massoretas não indexavam a Mp (ver fig. 2-4 e 2-5), e a obra só era completamente compreendida por um grupo pequeno de pessoas, sendo esse conhecimento passado de geração em geração. A classificação dos tipos de notas da Mp usada na BHS é:

- Sem índex
- Com índex relacionado *à Massorá Magna*
- Com índex sem relação com as notas da Mm

Nessa categoria foram incluídos também os cólofons, que são notas que em geral aparecem no fim dos livros, mas podem estar associados a algumas referências de capítulo e versículo, na margem oposta às anotações da *Massorá Parva.*

Neste trabalho estão sendo utilizados os termos dimensão e visão indistintamente. Essas visões auxiliam no processo de análise e interpretação da obra. As visões que o texto da BHS apresenta contêm elementos que apresentam sobreposição entre as visões e tornam sua distinção uma tarefa complexa. Isso inviabiliza a adoção direta de XML-S devido às restrições da linguagem.

Para definir as relações encontradas no texto e suas conseqüências no processo de representação, foi utilizada a representação simbólica L, R, P, S, PO Mp, Mm, Ap e Cs.

2.5.3 BHS - Relações transtextuais

De acordo com a notação definida no item anterior, foi elaborada a seguinte lista de dimensões e suas relações:
L – Cada um dos livros da BHS, por exemplo: Gênesis, Êxodo, Salmos, Isaías.

Historicamente os livros foram agrupados de acordo com os assuntos, livros poéticos, livros dos reis, livros dos profetas etc; mas esses agrupamentos não influem diretamente nos sistemas de codificação.

1ª relação: Todas as dimensões estão contidas em L

R- Sistema de divisão de um livro (L) em unidades menores, os capítulos e versículos.

Diferentes grupos religiosos adotam algumas variações na quantidade de capítulos e versículos em alguns livro; as divergências mais conhecidas são entre os judeus, cristãos e cristãos ortodoxos.

2ª relação: As dimensões L e R, são hierárquicas e formam o principal sistema de referência do Ketib.

P- As perícopes são importantes para a liturgia e para o estudo de crítica textual. Foi o primeiro sistema de divisão do texto criado.

3ª relação: P e R apresentam sobreposições.

S- As sentenças estão contidas nas perícopes.

Mp e Mm- *Massorá Parva e Massorá Magna*. São duas obras distintas. Compostas de notas e comentários do texto contido em L.

4ª relação. Mp e Mm possuem relação de metatextualidade e hipertextualidade.

Ap- Aparato Crítico, é o texto mais recente incorporado à BHS no séc. XX.

5ª relação. Ap possui relação de metatextualidade e hipertextualidade com Mp, Mm e L-R, além de outras referências a obras externas.

Cs- Os cólofons. Informações introduzidas pelos copistas, em geral ao fim de cada livro. Não fazem parte do texto. Têm grande valor para estudos na crítica literária, e precisam ser representados na obra final.

6ª relação. Cs é resultado direto do conteúdo em L.

As relações transtextuais presentes no texto, tanto na quantidade como na diversidade, tornam a representação do conteúdo da BHS uma tarefa complexa.

2.6 Propriedades Essenciais

As experiências com as linguagens utilizadas na área de editoração ensinaram uma demorada e dispendiosa lição. Durante anos têm sido usadas linguagens sem flexibilidade, específicas para uma classe rígida de problemas e, na maioria das vezes, fora de padrões de mercado. Como conseqüência existe pouca mão de obra disponível para trabalhar em ferramentas de software e manutenção. A evolução tem sido lenta, a compatibilidade e portabilidade dos sistemas deixam a desejar e adoção de recursos tecnológicos modernos tem sido visto como "uma dor de cabeça" a ser postergada nos centros de produção.

Na formulação de um novo padrão de linguagem para marcação de texto, é preciso considerar as dificuldades do passado e evitar a repetição dos erros. Baseado nos conceitos de programação orientada a objeto, quatro propriedades são colocadas como objetivos na formulação de um novo padrão a ser proposto.

2.6.1 Reusabilidade

A complexidade das pesquisas e das definições de padrões de representação torna essa atividade cara, e lenta. O sucesso dos resultados pode depender de apoio político, de empresas ou organizações, na escolha do padrão a ser adotado.

Quando há possibilidade de reutilizar estruturas de representação para diferentes obras, reduzindo os custos e o tempo de produção, a reusabilidade se torna um forte atrativo para sua ampla aceitação.

Na busca de soluções que preencham os requisitos técnicos de um novo padrão, o alto grau de reusabilidade é tido como obrigatório. Essa necessidade fica ainda mais clara em setores que não dispõem de abundantes recursos materiais e financeiros.

Não há originalidade nem novidade nesses anseios, pois na engenharia de software esse problema já é conhecido.

2.6.2 Interoperabilidade

Essa propriedade vem assumindo sua importância na medida em que a diversidade de plataformas de hardware e software e suas inúmeras configurações distintas não mostram tendências de convergirem. A implantação de novas linguagens vem empregando esforços para alcançar o status de interoperável.

Essa preocupação deve ser estendida aos mecanismos de definição das novas linguagens de forma a que não haja ambigüidade na interpretação da codificação.

2.6.3 Padrão aberto

Devido às características do problema de codificação de texto serem de grande complexidade, existe uma forte tendência a deduzir que será necessário uma linguagem nova ou um padrão proprietário para resolver o problema, o que pode ser uma repetição de erros do passado.

Deve haver todo empenho possível pela busca de soluções que façam o uso de padrões abertos. Padrões proprietários tendem a reduzir o número de participantes no desenvolvimento e na manutenção de ferramentas. A antiga lei da "oferta e procura" pode tornar essa tarefa economicamente inviável.

2.6.4 Flexibilidade

A tecnologia que for adotada precisa ser flexível, para incorporar novas estruturas de representação.

A linguagem HTML, por exemplo, apresentava um bom nível de reusabilidade, era um padrão aberto, e apresentava um grau de interoperabilidade, pelo menos enquanto os interesses pessoais dos principais desenvolvedores de browsers não adotaram a criação de padrões separados. No entanto, a falta de flexibilidade da linguagem foi um grande obstáculo na evolução do HTML. Algumas tentativas de incrementar a linguagem para oferecer recursos mais sofisticados acabaram gerando falta de padrão, problemas de segurança e perda da interoperabilidade.

Não foi surpresa a migração de HTML para o XML, que vem sendo registrada pelos desenvolvedores de *sites* Web nos últimos meses.

Capítulo 3

Modelos de Representação

Esse capítulo apresenta as duas formas de codificação de texto no domínio que estão sendo tratado.

A primeira forma está baseada em XML-S como linguagem que modela a ontologia da aplicação. A segunda usa RDF-S, a proposta desse trabalho.

Para melhor compreensão do Ketib, será feita uma breve introdução de XML-S, RDF e RDFS

3.1 XML-S

XML-S é a sigla do *XML Schema*. Possibilita a criação de modelos para documentos escritos em XML. O XML-S é uma evolução do DTD que a linguagem XML oferecia inicialmente, permitindo uma maior flexibilidade na definição de tipos de estruturas.

> **DTD** - *Document Type Definition* - define as regras de formatação para uma dada classe de documentos. Pode definir elementos, atributos e entidades válidos num documento.

O XML-S é a primeira tentativa de substituir o DTD por algo melhor. Na verdade, o DTD é um mecanismo herdado do SGML e na época parecia ser uma solução muito prática. Os DTDs estão sendo substituídos pelos seguintes motivos:

O primeiro problema é referente à sintaxe. Os documentos em XML possuem uma sintaxe diferente da que é usada no DTD; sendo assim, as ferramentas existentes de XML não podem ser usadas para verificar a validade da sintaxe do DTD.

O segundo, que tem se tornado crítico, é referente à semântica. DTDs permitem apenas uma forma limitada de representar informação semântica sobre os documentos. Isso ocorre porque DTDs empregam poucos tipos de dados e fornecem pouca flexibilidade na especificação dos tipos definidos pelos usuários. Por exemplo, a declaração de um elemento que represente um mês do ano, embora correta na forma, pode ter um conteúdo inválido. Com DTD o máximo que se pode dizer sobre tal elemento é que deve ser um conjunto de caracteres.

```
<!ELEMENT MES #PCDATA>
```

Dessa forma o seguinte mês seria válido:

```
<MES> Domingo</MES>
```

Eventualmente, é sempre possível pensar em alguma codificação que pudesse representar melhor tais elementos, mas isso implica em definições bem mais complexas.

Um terceiro problema com o DTD é em relação a reusabilidade. Nos DTDs ela é feita somente através do uso de mecanismos de macros parametrizadas. Isso significa que a estrutura sintática de tais entidades deve ser previamente conhecida e ao reutilizar os DTDs não é possível usar os mesmos nomes sintáticos dos parâmetros.

Em resumo, o DTD não foi projetado para ser reutilizável nem para ser aplicado em documentos distribuídos, não possui sintaxe compatível com a da linguagem XML e tem baixo potencial na codificação de semântica.

O *XML Schema* estende e generaliza o uso de DTD na linguagem XML. Um *Schema* é um modelo de descrição de uma estrutura de informação e às vezes, da sua semântica.

O XML-S soluciona as debilidades presentes no DTD. Os documentos em XML-S são antes de tudo documentos XML. Isso significa que usam os elementos e atributos para expressar a estrutura e semântica dos documentos, além de poderem ser editados e processados com as mesmas ferramentas usadas para processar outros documentos em

XML. Os documentos XML-S são válidos somente se estiverem de acordo com a estrutura descrita em XML-S; assim, fica resolvido também o problema de ter mais de uma operação de análise sintática para um único documento em XML.

O vocabulário de um documento em XML-S é formado por aproximadamente trinta elementos e atributos, além de possibilitar o uso de *namespaces*[12] num documento, facilitando a reusabilidade. O poder de definição de modelos em XML-S pode ficar evidente na tabela I, que compara recursos do DTD com XML-S.

Os modelos que utilizam o XML-S são: OSIS, XSEM, JITT e BUVH.
A Tabela 1 apresenta um resumo dos principais recursos que são desejados para criação de ontologias de aplicação e suas disponibilidades no caso de DTD, XML-S.

	DTD	XML-S
Tipo de dados	N	S
Cardinalidade	S	S
Restrição de Intervalo	N	S
Reusabilidade	N	S
Classes	N	N
Herança Múltipla	N	N
Reificação[13]	N	N
Negação, união e interseção de classes	N	N
Inferência: Transitiva e inversa	N	N

Legenda

N: Não disponível

S: Disponível

Tabela 1 - Recursos para criação de ontologias de aplicação: DTD x XML-S

[12]Em XML, namespace é um conjunto de nomes, identificado por uma URI, os quais podem ser usados em documentos XML como tipos e atributos. Esses nomes possuem uma estrutura interna e, de acordo com a definição matemática, não é um conjunto

[13] Palavra derivada do inglês *reify*= tornar real.
Esse termo não possui tradução padronizada em português

Tabela I- Recursos para ontologia de aplicação:

Tipo de dados: possibilitar definição de vários tipos de dados
Cardinalidade: definir cardinalidade de um elemento
Restrição de intervalo: definir regras de restrição de valores
Reusabilidade: possibilidade de definir escopo
Classes: definir classes e sub-classes
Herança múltipla: semelhante ao recurso em linguagem orientada a objeto
Reificação: traduzida do inglês *reification*. Propriedade que permite predicados de primeira ordem numa operação
Negação, união e interseção de classes: operações básicas com classes
Inferência transitiva e inversa: propriedades que podem ser deduzidas automaticamente na afirmação de um predicado.

Como será visto no próximo tópico, algumas propriedades essenciais não são contempladas pela criação de esquemas com o uso de XML-S (seção 3.2 RDF-S).

3.1.1 Modelos que usam XML-S

Entre os trabalhos publicados recentemente sobre a representação de informação e marcação de texto, destacam-se os seguintes modelos:

OSIS - Este é o padrão que vem sendo definido por iniciativa das Sociedades Bíblicas Unidas, de forma a reduzir custos, unificar bases de texto, ter flexibilidade e agilidade na produção de novos produtos que usam como base o texto bíblico. Sua última versão é de setembro/2002.

É o padrão mais completo até o momento para representação de textos bíblicos. Utiliza artifícios para alguns problemas de sobreposição de visões. Ainda não contempla visões mais complexas como por exemplo, lingüística e paginação editorial. O **Anexo A** mostra um exemplo de codificação com OSIS.

Este modelo não resolve de forma genérica os problemas de sobreposição de visão em um texto. A cada nova ocorrência será necessário o uso de artifícios para permitir a codificação desses casos.

Existe duplicação de elementos, como por exemplo, os elementos do Dublin Core estão quase todos repetidos.

Por fim, é utilizado um tipo de dado, "*annotation*", para armazenar anotações ao texto. Pode-se classificar essas anotações como metadados. Essa decisão cria na verdade uma discriminação entre metadados com um conjunto de elementos definidos, por exemplo informações catalográficas, e outros metadados que tenham que ser "acomodados" nos elementos definidos pelo tipo de dado "*annotation*". Estão previstas modificações no padrão OSIS para incorporar novos tipos de dados, mas enquanto esses novos tipos de dados não forem implementados, os elementos terão que ser registrados como tipo *annotation.*

JITTs - Esta é a solução proposta por pesquisadores da Society of Biblical Literature em setembro de 2002, para possibilitar a representação de textos com hierarquias de visões sobrepostas.

Os autores utilizam uma etapa de pré-processamento feito por linguagem de programação, por exemplo *scripts* em PERL, que filtram as visões em etapas distintas e passam resultados para o visualizador de XML. A descrição como um todo quebra as regras de sintaxe do XML, sendo portanto um arquivo de conteúdo inválido, se analisado integralmente por um *parser* de XML.

BUVH - Apresentada em março/2002 pelos pesquisadores da Society of Biblical Literature, trata todo o conteúdo do texto no nível do elemento mais básico, a palavra, e

cria estruturas de *links* para representar os diferentes conjuntos de informação, através do uso de Xpointer e Xpath[14].

XSEM - Produzido pelo Summer Institute of Linguistics (SIL), com o propósito de substituir a linguagem *Standard Format for Markup* (SFM) que vem sendo usada nos últimos 20 anos na produção de obras no campo teológico e linguistíco nos mais diferentes formatos e idiomas. Visa representar somente o layout de texto, sendo que a última versão publicada é de julho/2000.

Uma vez produzido o arquivo em XML, alguns aplicativos são utilizados para transformarem o texto de entrada em diferentes formatos de saída. No caso do XSEM, o SIL produziu saída para pdf, html, wml e E-book[15]. O aplicativo que efetuou as transformações foi o XEP, desenvolvido pela Renderx (www.renderx.com), com suporte do aplicativo MSXSL, da Microsoft. No **Anexo B** é apresentado um exemplo de codificação na estrutura do XSEM.

3.2 RDF-S

O Resource Description Framework (RDF), uma recomendação do *World Wide Web Consortium* - W3C, constitui-se em uma arquitetura genérica de metadados. Permite descrever recursos no contexto Web, sendo um dos pilares para construção de *Web Semantic*, através da adoção de padrões de metadados[Lassila1999].

A proposta do RDF é permitir a formulação de vocabulários que possam ser processados por máquinas e ainda legíveis por seres humanos, impulsionando o intercâmbio, o uso e a extensão da semântica de metadados entre comunidades das mais diferentes áreas do conhecimento.

---

[14] Linguagens baseadas em XML que permite endereçamento, referência e recursos para definir links estruturado com controle e semântica associada.

[15] E-book é a codificação de dados em texto e imagens em formato digital. Um E-book pode ser uma réplica exata de um livro impresso, ou pode ter conteúdo especificamente preparado para o formato de E-book format. Existe vários formatos de E-book.

O RDF busca resolver um dos principais desafios encontrados pelas diferentes comunidades de descrição de recursos: prover interoperabilidade entre os diversos padrões de metadados. Para tanto, RDF define um mecanismo para descrição de recursos independente de um domínio particular de interesse, porém com as primitivas de modelagem necessárias para descrição de recursos sob qualquer domínio de aplicação, independente de plataforma computacional.

A tecnologia RDF representa uma convergência de influências de diversas áreas da tecnologia da informação. As principais influências vêm da comunidade de padronização da Web, o W3C, na forma de metadados em HTML e há influência da linguagem PICS (*Platform for Internet Content Selection*) [www.w3c.org/PICS], que define camadas de metadados que permitem criar políticas de acesso ao conteúdo da Internet.

A proposta do *Dublin Core* de utilizar RDF para representar estruturas de metadados em documentos da Web uniu os interesses dos pesquisadores na codificação de documentos com SGML/XML, e dos pesquisadores na área de representação do conhecimento. As principais contribuições foram a criação de um formato análogo ao de redes semânticas e a utilização do conceito de reificação.

O modelo *RDF Schema* (RDF-S), baseado no modelo RDF básico, é fortemente influenciado por conceitos de orientação a objetos e de linguagens de especificação de bancos de dados, como o modelo conceitual NIAM (Nijssen Analysis Method) [Brickley2000].

As áreas de aplicação que podem se beneficiar das potencialidades da tecnologia RDF são inúmeras. Entre elas destacam-se os contextos de:

- descoberta de recursos, em que o uso do RDF, possibilita a implementação de mecanismos de busca mais eficientes;
- de catalogação, onde o RDF pode ser utilizado para descrever recursos de informação disponíveis em um *Web site*;

- em uma página ou em uma biblioteca digital; em que o RDF pode facilitar a descrição e o compartilhamento do conhecimento entre agentes inteligentes.

Em função da sua flexibilidade e capacidade de representação de informação em estruturas com a criação de classes e tirando proveito das propriedades dos predicados de primeira ordem, o RDF tem se mostrado uma solução atraente para resolução de problemas de interoperabilidade, desde conflitos de esquemas em bancos de dados relacionais até a integração com outros tipos de recursos.

3.2.1 O modelo RDF básico

A especificação da tecnologia RDF destaca-se pela simplicidade com que busca estruturar o conteúdo contido na Web. Tecnicamente, RDF não é uma linguagem, mas um modelo de dados para descrição de recursos com mais semântica, através da adoção de metadados. O modelo de dados RDF é muito simples, baseando-se em quatro tipos de objetos, descritos a seguir:

Resources: representam o universo de objetos que podem ser descritos. Todo recurso necessita de um *Uniform Resource Identifier* (*URI*) associado. São exemplos de recursos: uma página Web, parte de uma página Web, uma coleção de páginas Web e objetos fora da Web, como por exemplo um livro impresso.

Literals: representam os tipos de dados que o valor de uma propriedade pode assumir. Os tipos mais usuais de literais são os do tipo *string*.

Properties: representam os aspectos do recurso a serem descritos. Podem ser visualizadas como atributos de recursos e neste sentido correspondem a pares de atributo-valor. Propriedades também são utilizadas para descrever relacionamentos entre recursos. Neste sentido, o modelo de dados RDF se assemelha ao modelo Entidade-Relacionamento. Cada propriedade tem um significado específico, define seus valores permitidos, os tipos de recursos que podem descrever, e seus relacionamentos com outras propriedades.

Statements: representam a relação entre um recurso, uma de suas propriedades e o valor que essa propriedade pode assumir.

Os *statements* correspondem à construção básica que estabelece o modelo de dados em RDF. Um *statement* é chamado de *declaração ou predicado*: define uma relação binária, envolvendo uma propriedade e um par de atributo-valor. Usando a notação em forma de tripla, a propriedade e o par atributo-valor é formada por: *subject* (recurso) e *objetct* (valor de uma propriedade). Por exemplo, a tripla formada pela expressão

flutua(oleo, agua),

representa uma relação entre óleo e água. RDF pode também conter uma variável como elemento da tripla,

flutua(?x, agua),

é o predicado que representa que *?x* tem a propriedade de flutuar na água.

A notação utilizada para representação dessa tripla,

(predicate,[subject],[object]),

é particularmente proveitosa, uma vez que permite que recursos e valores sejam misturados, ou seja, qualquer recurso pode atuar no papel de valor, o que garante maior flexibilidade ao modelo na representação de estruturas mais complexas.

Algumas operações booleanas podem ser usadas para representar fatos mais complexos. Por exemplo, ponto_vapor(agua, 100C, 1atm), poder ser representada pelo seguinte conjunto de relações binárias:

vaporizacao(?y, agua) E temp(?y, 100C) E (pressao_atm(?y, 1atm).

As triplas do RDF correspondem a um sub-conjunto das operações de lógica de primeira ordem, pois definem a operação de conjunção (AND) mas não implementa, no modelo básico, as operações de negação (NOT) e disjunção (OR)[Sowa2000].

As operações de negação e disjunção são implementadas nas extensões de RDF que estão em processo de padronização. Está sendo proposta uma nova linguagem a ser padronizada em 2003, pelo W3C, com o nome de OWL (*Ontology Web Language*). [www.w3c.org/TR/2002/WD-owl-ref-20021112].

Além do grande poder de expressão do modelo RDF, que possibilita utilizar predicados de lógica de primeira ordem, este modelo permite a representação através da reificação de relações geralmente associadas a lógicas de ordem superior, como por exemplo:

Tipo(flutuar, propriedade_fisica) E flutuar(oleo, agua)

onde concluí-se, através de lógica de ordem superior, que o fato de óleo flutuar na água é uma propriedade física. No entanto, no modelo RDF, a expressão é representada na semântica de lógica de primeira ordem [www.w3.org/TR/rdf-mt].

A representação em RDF pode utilizar com muita flexibilidade os vocabulários definidos através das URIs, conhecidos como *namespaces*. Desta forma as afirmativas em relação às propriedades dos elementos podem ter um número arbitrário de predicados. Pode ser construida uma URI para qualquer conjunto de fatos que seja apresentado, e incorporar os namespaces nessa estrutura de representação [www.w3.org/TR/rdf-concepts].

Além do formato de tripla, o modelo RDF também pode ser visualizado na forma de um grafo, que consiste de um conjunto de nós conectados por arcos rotulados, em que os nós representam os recursos Web e os arcos representam as propriedades destes recursos. Ainda na representação de grafos convencionada pelo W3C, literais são representados por retângulos [Lassila1999].

3.2.2 XML para representação de RDF

Um dos principais aspectos que tem contribuído para o sucesso da tecnologia RDF no contexto Web é a possibilidade de representar e trocar modelos RDF via XML [BRAY2001]. Como já mencionado, o RDF não é uma linguagem, mas sim um modelo de dados que provê uma estrutura *(framework)* conceitual e abstrata para definição e uso de metadados no contexto Web. Para tanto, faz-se necessário o uso de uma linguagem

que consiga expressar este modelo. A linguagem de marcação XML é uma das possíveis formas de representação das instâncias dos modelos RDF. Dentre os motivos que levaram à escolha da XML, destacam-se os seguintes:

- Uma sintaxe baseada em XML certamente facilitará a tarefa de tornar o RDF o padrão de metadado para descrição de recursos no contexto Web.
- XML é hoje um padrão amplamente aceito no contexto de interoperabilidade sintática de informações via rede, haja vista o grande número de ferramentas disponíveis no mercado, e a preocupação cada vez maior dos fornecedores em desenvolver produtos que incorporem as características do XML.
- XML fornece o mecanismo *Namespaces*, através do qual a arquitetura RDF consegue misturar diferentes padrões de metadados para compor descrições de recursos dentro de um mesmo documento.

Duas sintaxes em XML são propostas para expressar os modelos RDF: *serializada*, que expressa toda a potencialidade do modelo RDF; e *abreviada*, que inclui construtores adicionais para expressar de forma mais compacta o modelo RDF.

As Figuras 3-1 e 3-2 apresentam um exemplo na forma serializada e na forma abreviada.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
         xmlns:ex=http://local/elementos.estrutura/
         xml:base="http://local/departamento/secao/produto">
   <ex:barraca rdf:ID="10245">
        <ex:modelo>Campestre 1</ex:modelo>
        <ex:ocupantes>2</ex:ocupantes>
        <ex:peso>2400</ex:peso>
        <ex:tamanho>14x56</ex:tamanho>
   </ex:barraca>
</rdf:RDF>
```

Fig. 3-1 - RDF na forma serializada

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
         xmlns:ex=http://local/elementos.estrutura/
         xml:base="http://local/departamento/secao/produto">
   <ex:Tent rdf:ID="10245"
    ex:modelo="Campestre 1"
    ex:ocupantes="2"
    ex:peso="2400"
    ex:tamanho="14x56"/>
</rdf:RDF>
```

Fig. 3-2 - RDF na forma abreviada, os elementos são transformados em atributos

3.2.3 Repositórios

RDF permite definir uma propriedade para um conjunto de fatos, dupla de objetos/valores. Esses fatos podem ser encapsulados em repositórios, "*containers*".

Containers podem ser enumeráveis ou sem identificação numérica. Podem representar propriedades alternativas de um objeto, variantes ou um subconjunto de fatos que represente uma relação de especialização.

- **Bag**: Conjunto de fatos que não apresentam ordem específica de identificação.
- **Seq**: Conjunto de fatos ordenados segundo um critério escolhido e indexado de forma a recuperar precisamente o enésimo valor/predicado desejado.
- **Alt**: Fatos que são utilizados em substituição ao originalmente descrito.

Com os repositórios e o mecanismo de reificação (ver 3.2.4), uma estrutura em RDF não fica limitada a representação de relacionamento binário, um par atributo-valor, pois o valor pode ser um objeto do tipo *container*, com vários valores adicionais.

Containers podem ainda ser particurlamente úteis no registro de variações de texto, pontos de vistas divergentes, codificação de traduções em diferentes idiomas ou utilização de um conjunto de caracteres fora do padrão. Emanuel Tov é autor de um livro altamente conceituado no campo da crítica textual do texto bíblico em hebraico, *The textual criticism of Hebrew Bible*[Tov2001]. No capítulo 9 ele defende que no estudo do texto hebraico é preciso registrar as diferentes opiniões acerca do trecho em que os

estudiosos divergem; mesmo que na sua posição pessoal não haja dúvida, o registro permite que outros pesquisadores fundamentem suas teses.

3.2.4 O mecanismo de reificação

Uma importante característica do modelo de dados RDF é a descrição de *statements*. Isso é possível através do mecanismo de reificação que permite considerar qualquer *statement* RDF como um recurso. Desta forma é possível aninhar descrições obtendo assim descrição sobre descrição, requisito fundamental em gerência de metadado. Descrições deste tipo são denominadas *descrições em lógica de ordem superior*, uma vez que utilizam o mesmo modelo, porém em um nível maior de abstração.

Formalmente, a reificação em RDF significa expressar um *statement* como um recurso que contenha quatro propriedades. Estas quatro propriedades são definidas pelo modelo de dados RDF e são listadas abaixo:

subject : identifica o recurso sendo descrito pelo statement modelado.

predicate: identifica a propriedade original no statement modelado.

object : identifica o valor da propriedade no statement modelado.

type: descreve o tipo do novo recurso. Todos os statements reificados são instâncias de *rdf:statement*.

A figura 3-3 mostra um exemplo de classe com hierarquias simples.

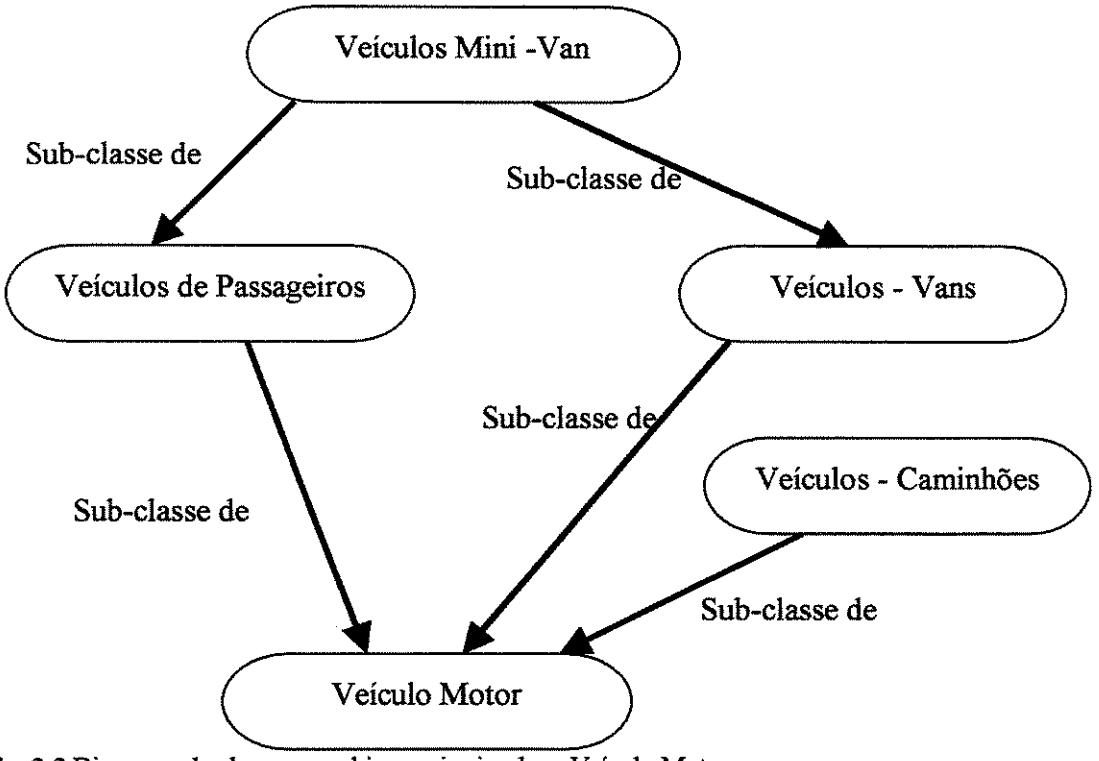

Fig. 3-3 Diagrama de classes para hierarquia simples - Veículo Motor

O anexo C implementa a descrição em RDF-S da Figura 3-3

As vantagens do RDF-S para XML-S são resumidas na tabela a seguir:

	XML-S	RDF-S
Tipo de dados	S	S
Cardinalidade	S	N
Restrição de Intervalo	S	S
Reusabilidade	S	S
Classes	N	S
Herança Múltipla	N	S
Reificação	N	S
Negação, união e intersecção de classes	N	N
Inferência: transitiva e inversa	N	N

Legenda

S: Sim possui o recurso

N: Não, recurso indisponível

Tabela 2 - Recurso para criação de ontologia de aplicação: XML-S x RDF-S

3.3 Processo de representação - Modelo Ketib

Após uma análise das ontologias e soluções existentes, as observações a seguir são pertinentes na definição do processo de representação:

1. Em qualquer texto é possível adotar um sistema de referência. É necessária uma atenção especial quando o sistema de referência do conjunto primário do texto for escolhido.
2. Na representação em um texto complexo, eleger um conjunto primário, em torno do qual os outros conjuntos interagem direta ou indiretamente é uma decisão fundamental. A escolha desse conjunto deve ter o maior apoio possível da comunidade que trabalha com o texto. Sempre haverá vozes discordantes. Decisões políticas podem comprometer a evolução técnica.
3. As múltiplas visões, presentes em textos complexos, e as relações entre seus conjuntos de elementos, podem ser classificadas como metadados ou meta-informação do conjunto primário do texto.
4. Os metadados devem ser representados por estruturas distintas (*frames*), que permitam codificar as anotações, comentários, alternativas e variações que um texto pode apresentar em camadas diferentes da que contém o texto primário.
5. Os modelos analisados nesses trabalhos buscaram representar diferentes níves de informação em uma mesma camada de dados. Essa decisão gera dificuldades de codificação atuais e barreiras técnicas para ampliar o modelo no futuro.

Para tornar possível um método que atendesse aos requisitos mencionados no capítulo 2, foram definidas as seguintes medidas:

- Utilizar uma representação do texto principal, considerando como um conjunto de dados.
- Utilizar camadas de metadados para representar anotações, aparato crítico, e dimensões paralelas do texto.
- Definir uma estrutura de metadados que resolva sobreposição eventual de duas camadas dentro de um mesmo conjunto.
- Criar um modelo flexível que possa definir formas distintas e alternativas de representação.

A Figura 3-4 apresenta o diagrama do modelo Ketib usado na represenção.

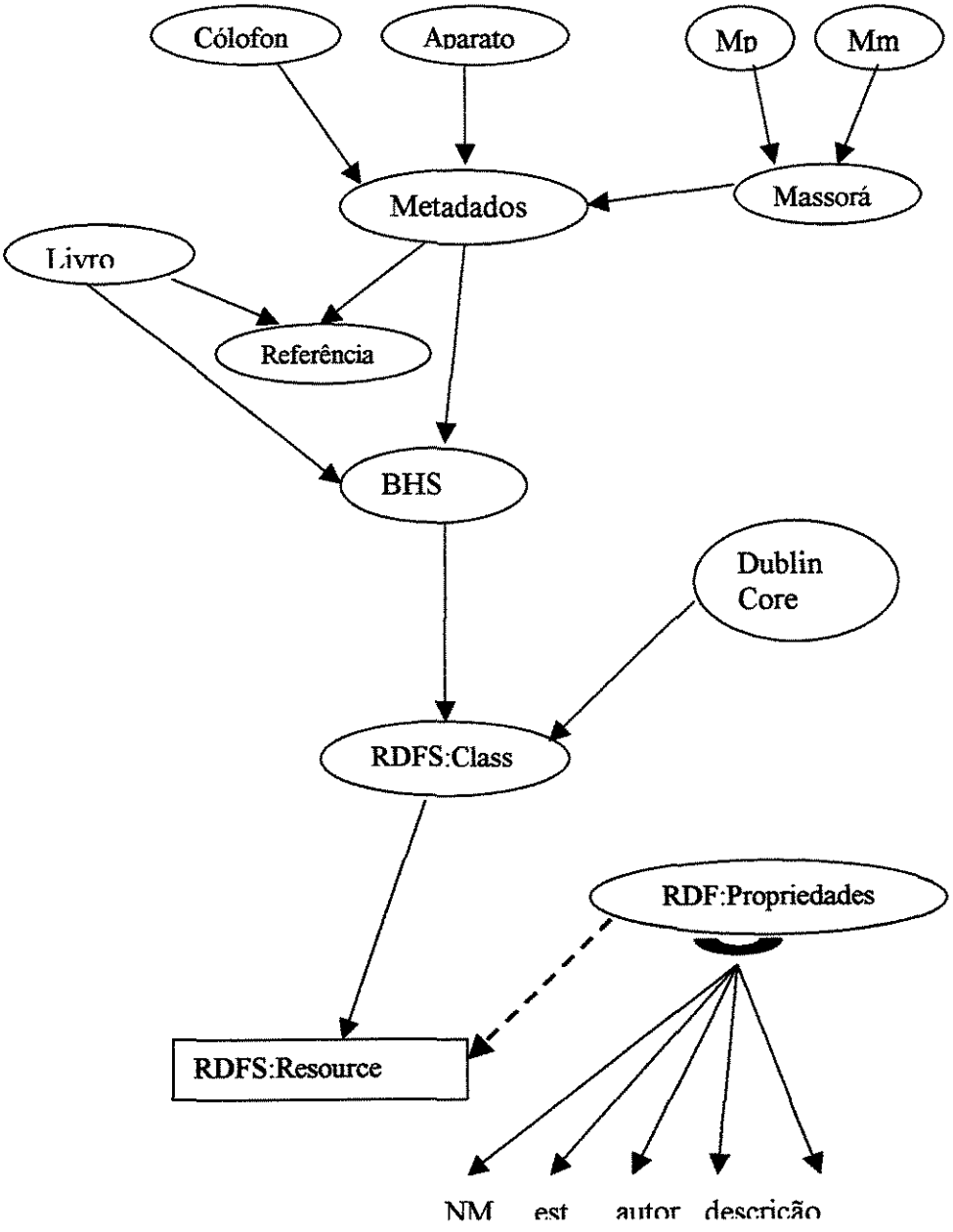

Fig. 3-4 -Modelo Ketib na representação RDF Schema

A descrição em RDF/XML do digrama da Figura 3-4 está no Anexo D.

Cada dimensão possui uma estrutura hierárquica bem definida. As dimensões que não apresentam sobreposições podem ser representadas na mesma camada de dados que o texto principal. A diferença para os outros modelos é a adição de uma estrutura de metadados que contém uma riqueza muito maior de informações e que podem ser facilmente recuperadas.

Quando existe sobreposição que pertence a duas dimensões distintas, foi criado uma camada virtual, que define uma nova estrutura de informação que acomoda as sobreposições. Essa solução utiliza uma combinação de dois recursos, o uso de *milestones* e *stand-off markup*. O *milestone* é colocado no início da ocorrência dentro do texto primário, mas ao invés de servir apenas como "marca", contém uma estrutura de metadados que descreve as propriedades. Para evitar duplicações do texto, dentro da estrutura iniciada pelo "*milestone*", foi usado a codificação proposta pelos pesquisadores da Universidade de Edimburgo, *stand-off markup*, apenas com a melhoria de utilizar Xlink como forma de referência ao texto principal. O Xlink[16], por ser um mecanismo muito mais poderoso e estar apoiado em estrutura de metadados na sua padronização definida pelo W3C, atende mais adequadamente as necessidades do Ketib.

A decisão de criar uma nova camada é análoga ao recurso presente em algumas das linguagens orientadas a objeto que implementam herança múltipla. Quando é preciso modelar uma classe que utiliza métodos de duas ou mais classes distintas, utiliza-se uma classe com herança múltipla.

[16] Linguagem que permite referenciar elementos em um documento XML, criar e descrever *links* entre *resources*. Usa a sintaxe de XML para definir estruturas de *links* sofisticados, além do tradicional *link* unidirecional do HTML.

Da mesma forma que na análise orientada a objeto, algumas vezes é possível eliminar a herança múltipla, alterando o modelo de classes. O mesmo pode ser tentado no Ketib, com a redefinição das dimensões iniciais. Essa etapa é semelhante ao proposto por Renear no modelo OHCO-3.

Do ponto de vista de implantação para a BHS, foram criados arquivos em XML/RDF, baseado em templates que descrevem a ontologia de aplicação, utilizando RDF-S. Os arquivos definidos codificam os seguintes conteúdos:

1. Texto bíblico com codificação de Livro (L), Capítulo.Versículo (R) e com as estruturas e *milestones* dos metadados. Incluem sobreposição, cólofons, cantilenas e possíveis anotações lingüísticas;
2. *Massorá Parva* (Mp) e *Massorá Magna* (Mm);
3. Aparato Crítico (Ap).
4. Níveis sobrepostos, codificados em XML com referências em Xlink.

A opção adotada foi conveniente devido às seguintes propriedades:

- L e R são visões hierárquicas.
- Mp e Mm são visões sem interseções e com conteúdo distinto.
- Ap é uma dimensão totalmente subordinada às visões L e R. Seu conteúdo pode ser incluído no mesmo arquivo que contém a codificação de L e R, mas para melhor clareza e simplificação da estrutura de XML-S, foi escolhido usar um arquivo independente.
- P, S e Cs são representadas dentro do texto principal com recursos combinados do *milestone* e *stand-off markup*, sendo devidamente documentada, se necessário com metadados específicos. Os problemas de sobreposição de P e S com R são facilmente resolvidos.
- Cs não possuem sobreposição com L, R, P ou S e têm sua representação anexada ao texto principal com a devida camada de metadados.

Para integrar todos os elementos da BHS (texto e metadados), de forma estruturada, podem ser usados tantos arquivos quantos forem desejados, de forma a encapsular as

visões do texto a serem exploradas. A criação desses arquivos pode ser opcional, pois a partir do arquivo que contém o texto bíblico é possível navegar por todo o conteúdo, incluindo as visões P, Ap, Mp e Cs. A Figura 3-5 traz alguns exemplos da BHS.

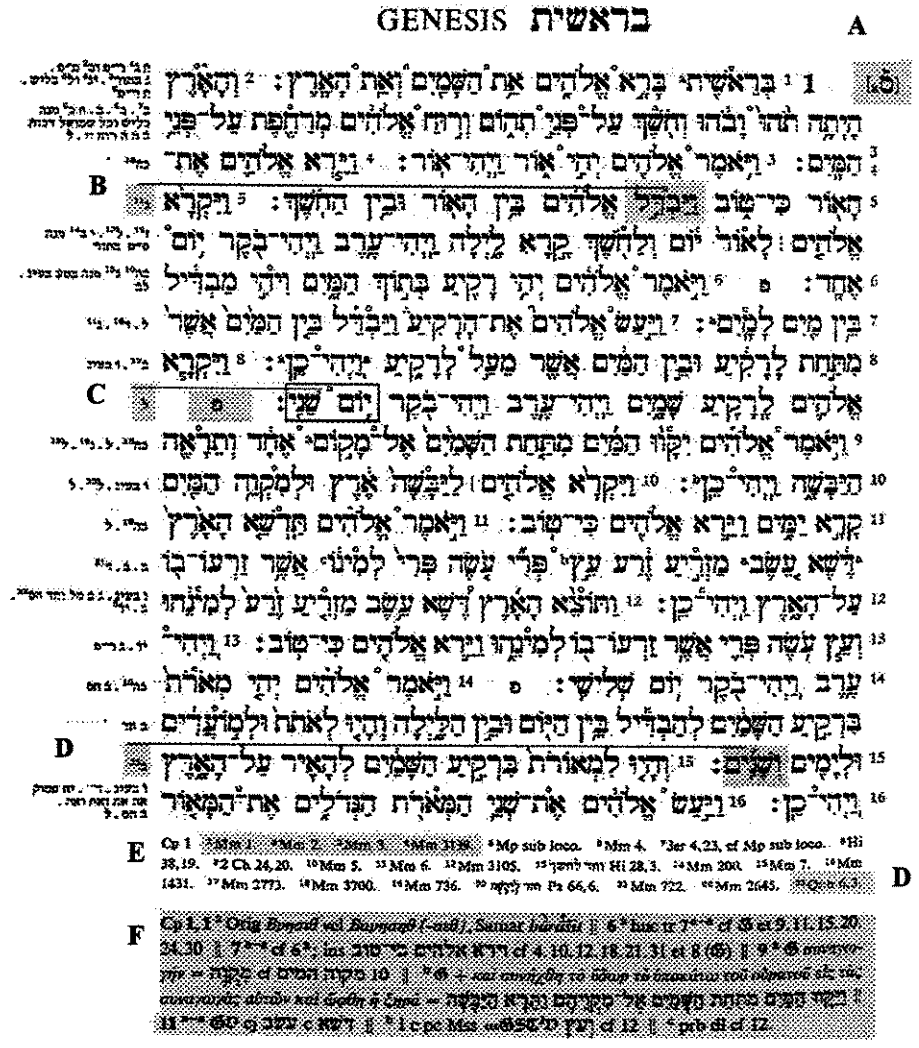

Fig. 3-5 – Exemplo do texto e metadados da BHS (Gênesis 1.1–16)

Legenda da Figura 3-5:

A- Colófon e marca de perícope.
B- Nota estatística da Massorá Parva que indica a ocorrência da palavra ou expressão por três vezes. Indica que a nota número 11 da Massorá Magna traz comentário sobre a palavra marcada.
C- Nota estatística da Massorá Parva que indica a ocorrência da palavra somente aqui e em nenhum outro lugar no Antigo Testamento.
D- Nota da Massorá Parva remete a uma referência na Massorá Magna que indica a ocorrência de duas vezes no Antigo Testamento. Sendo que a segunda é em Eclesiastes 6.3 (Qoh 6.3).
E- Notas da Mm referenciadas no verso 1 pelas notas da Mp. Sendo que as notas da Mm são 1, 2, 3 e 3139
F- Aparato crítico, escrito no séc. XX

- No texto dos massoretas não havia numeração nas notas acrescentadas, apenas as divisões (perícopes). O sistema de numeração das notas foi adotado em 1927.

- As notas da Mp que fazem referências à Mm são sempre numeradas dentro do mesmo capítulo, a cada novo capítulo essa numeração é reiniciada.

- As notas no Aparato Crítico são marcadas com letras dentro de cada verso.

Capítulo 4

Ketib para BHS

O processo aplicado ao caso de estudo, o texto da BHS, teve as seguintes etapas:

4.1 Roteiro do modelo Ketib

- Estudo do texto de aplicação e suas propriedades
- Entrevista com especialistas
  - Revisão de conceitos
  - Definição da ontologia
- Validação e complementação com especialistas
  - Análise das dimensões do texto e relações transtextuais presentes
  - Soluções para as sobreposições de dimensões que ferem a estrutura hierárquica e definição dos elementos usados na camada de metadados
- Construção do modelo RDF/XML

4.2 Problemas encontrados

1. Áreas de pesquisas envolvidas

O objeto em estudo, representação de informações em texto complexo, pertence à área de aplicações para ciências humanas, sendo necessário pesquisas de conceitos de inteligência artificial, padrões em definição pelo W3C e análise da Bíblia Hebraica.

2. Orientação e artigos de referência

O conjunto de áreas envolvidas oferece um reduzido número de trabalhos publicados e especialistas que possam dominar conceitos básicos.

3. Novo padrão e expectativas

Na proposta de criação de novo padrão deve existir a preocupação de oferecer um modelo que atenda aos atuais requisitos e permita extensão para futuras aplicações. Pela experiência com especialistas da área de ciências humanas, foi notado que a instabilidade da definição dos requisitos e propriedades apresentadas tende a ser maior do que as tratadas na área de ciências exatas.

4. Atrito inercial

A apresentação de novas propostas não é recebida com credibilidade, se não estiver acompanhada de muitos exemplos e ferramentas de suporte. Por outro lado, a codificação de um bom volume de exemplos e oferta de ferramentas, em geral, leva tempo e requer uma boa aceitação dos usuários. Uma iniciativa fica esperando pela outra, sendo necessário que algo seja feito. Esse tipo de dificuldade é semelhante ao conceito da Física conhecido como atrito inercial.

4.3 Soluções adotadas

Para superar os obstáculos mencionados no item anterior, as decisões tomadas foram:

1. Consultar um grupo multidisciplinar de especialistas, realizando uma etapa de análise de requisitos e extração de conhecimento. Como resultado, foram comparados os conceitos e preferências mais importantes na definição do modelo.
2. Foi necessário analisar periodicamente o material publicado pelo W3C, devido à velocidade das modificações que ocorreram durante o ano de 2002.
3. Fazer uso de padrões abertos, que apresentassem flexibilidade. Evitar elementos que não estivessem padronizados e modelos de representação que utilizassem "artifícios" para superar as dificuldades estruturais presentes no texto.
4. O objetivo desse trabalho não incluiu o desenvolvimento de ferramentas, mas para tratar do "atrito inercial" mencionado acima, é sugerido reutilizar códigos fontes de várias ferramentas disponíveis na Web, em que é possível encontrar módulos que implementam editores com recursos para XML e Xlink,

analisadores de estruturas RDF e outros recursos que podem reduzir o tempo de desenvolvimento de ferramentas.

4.4 Exemplo da codificação Ketib

Trechos do capítulo 1 do livro Gênesis, codificado utilizando o modelo Ketib.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs=="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xml:base="http://dcc.unicamp.br/~800271/n1">
 <rdf:Description rdf:ID="Gênesis">
  <rdfs:subClassOf rdf:resource="#Livro"/>

  <dc:title>Gênesis</dc:title>
  <dc:description>bereis</dc:description>
  <dc:publisher>Württemberg Bible Society</dc:publisher>
  <dc:date>1997</dc:date>
  <dc:type>Livro impresso</dc:type>
  <dc:format>text/html</dc:format>
  <dcterms:isPartOf rdf:resource="BHS"/>
  <div type="testament">
    <div type="book" osisID="Gen">
      <div type="chapter" osisID="Gen.1">

<N:col rdf:resource="col/gn1/1"/>
<verse osisID="Gen.1.1"><N:nm rdf:resource="gn1/nm/1"/>berê'shiyth bârâ'
<N:mp rdf:resource="gn1/mp/2"/> 'elohiym 'êth <mmp/> hashâmayim
<N:mp rdf:resource="gn1/mp/3"/> ve'êth <mmp/> hâ'ârets</verse>

<verse osisID="Gen.1.2"> <N:mp rdf:resource="gn1/mp/4"/>vehâ'ârets <mmp/>
  hâyethâh thohu vâbhohu vechoshekh `al-penêy thehom veruach'elohiym merachepheth `al-penêy
  hammâyim </verse>

<verse osisID="Gen.1.3"> vayyo'mer 'elohiymyehiy 'or vayhiy-'or </verse>

      </div>
      <div type="chapter" osisID="Gen.2">

      </div>
    </div>
  </div>
 </rdf:Description>
</rdf:RDF>
```

* O Anexo D traz a descrição do metadado nm e mp, além do template do Ketib em RDF.

Exemplos que representação em RDF-S pode oferecer na recuperação do conteúdo:

- Predicados criados com a representação em RDF:

  subClass(livro, Genesis)
  Ispartof(Genesis, BHS)
  Date(Genesis, 1997)

  :- O livro de Gênesis, é parte da BHS. O conteúdo codificado foi impresso em 1997.

  Aparato(gn1.1, a)
  Tipo(a, variante)
  Variante(a, bag1)
  Bag1(autor, Origenes)
  Bag1(text, "....")

  :- O texto "...." é uma variante de Orígenes, registrada no aparato crítico da BHS, no livro de Gênesis, referente ao verso gn1.1

- Ações que poderiam ser efetuadas por programas que implementam a representação em RFD-S:

  - Listar todas a variantes de texto dos capítulos 1 e 2

  - Listar todas a variantes de Orígenes, registrada no Ap (aparato crítico) em Gênesis

  - Listar todos os versículos que possuem notas da *Massorá Parva*

  - Listar todas a palavras ou expressões que ocorrem uma única vez em todo Antigo Testamento

4.5 Comparação dos modelos

A comparação dos modelos se restringe somente aos objetivos alcançados, pois o Ketib trabalha em um nível de estrutura superior aos outros modelos.

- Ketib x OSIS

O modelo OSIS está totalmente implantado em XML-S, o que obriga a adaptações nesse modelo para atender às necessidades na representação.

O modelo Ketib usa um conjunto simplificado e reduzido dos elementos do OSIS em XML-S. Com esses elementos é feita a validação e codificação das estruturas básicas do texto, L e R.

O XML-S do padrão OSIS tenta representar todos os elementos numa única camada de dados. O Ketib utiliza o RDF-S para representar camadas de metadados. Dessa forma a representação da sobreposição das visões é feita nas diferentes camadas.

Além dessas vantagens diretas, as futuras extensões que o padrão pretende incorporar como elementos para publicação, direitos autorais, notas editoriais, exegese e lingüística, seriam mais facilmente implementadas nas camadas de RDF-S, com possibilidade de herança múltipla, sem necessidade de inchar o padrão básico com novos elementos.

- Ketib x XSEM

O modelo XSEM foi o primeiro protótipo colocado em operação, demonstrando o potencial das operações com XLS e XLST. No entanto, não contém um conjunto definido que atenda representações mais complexas como no caso da BHS. No entanto, para o domínio e propósitos definidos pelos seus autores, o modelo atingiu seus objetivos.

– Ketib x JITT

Os autores do JITT conseguiram resultados notáveis de desempenho na aplicação do seu modelo com estilo de pré-processamento através de scripts.

No campo teórico, JITT demonstra que não é viável o tratamento de marcação de visões concorrentes em uma única camada, ou com uma única raiz para todo o documento, sem ferir os fundamentos da linguagem XML.

No modelo JITT, os autores optaram por contornar a restrição da linguagem, e criaram um arquivo de descrição que é inválido para os softwares que implementam as regras de XML. No Ketib, o conteúdo dos arquivos está de acordo com os padrões da W3C.

No campo prático, requer uma etapa de pré-processamento a cada operação que requisitar uma nova visão para ser trabalhada. Mantém as desvantagens comentadas no modelo OSIS, de utilizar uma estrutura que não disponha de herança múltipla, inchando o modelo a cada novo conjunto de elementos que forem utilizados.

– Ketib x BUVH

O modelo BUVH, embora tenha apresentado conceitos fundamentais utilizados na formulação do Ketib, utiliza a unidade mínima para indexação de referência, a palavra, deixando de usar a potencialidade oferecida pelos elementos de intervalos (*range*) no Xlink, que foi proposta no Ketib e que reduz a complexidade da codificação, sem perder contudo a flexibilidade de lançar mão dos recursos previstos para o Xpath e Xpointer.

Na análise da representação do BUVH, Patrick Durasau, em artigo publicado em 2002, observa que os editores para esse modelo não estão implementados e seria complexo e lento o seu desenvolvimento. O autor do modelo adota a tese de que uma linguagem hierárquica pode acomodar dados e metadados estruturados, numa mesma camada de representação.

Capítulo 5

Conclusões

5.1 Argumentos finais

O processo Ketib que é descrito é flexível e está baseado em padrões abertos, RDF/XML. O uso de RDF/XML vem sendo adotado por diferentes setores das industrias.

Para criação de ontologia, a estrutura descrita em RDF é superior aos outros modelos, está padronizada pelo organismo internacional W3C e serve como base da linguagem OWL, que está em fase de padronização e oferece recursos ainda mais avançados. Existe trabalhos já desenvolvidos com esses padrões que estão disponíves ao público, inclusive com o código fonte, entre eles: editores gráficos, analisadores sintáticos e máquinas de inferência.

Com a solução apresentada é possível codificar as informações de obras com texto complexo, além de permitir adicionar ao modelo camadas de informação com reaproveitamento de todo trabalho já realizado. Uma vez que uma obra tenha sua representação no modelo Ketib, o fácil acesso, a pesquisa e recuperação de informações, a formatação de saída para diferentes periféricos e a adição de notas pessoais, são algumas das facilidades que podem ser exploradas pelos usuários.

Comparado à solução da representação proposta com os principais modelos conhecidos, o Ketib soluciona problemas tradicionais, que os outros modelos não contemplam ou resolvem de forma parcial.

O modelo faz uso de camadas de metadados para armazenar informações complementares, anotações, possíveis variações do texto e estruturas que apresentam sobreposição. Com a utilização desses recursos é possível evitar a duplicação de conteúdo no processo de representação e dissociação do código de *markup* com o conteúdo da obra.

Esse modelo não tem o propósito de codificar ou deduzir conhecimento teológico. Seu objetivo é, tal como os objetivos dos massoretas, registrar o que contém o texto, quais as escolhas feitas pelos estudiosos em crítica textual. Através de uma estrutura especial o processo de captura e recuperação do conteúdo codificado utiliza lógica de primeira ordem. O modelo comporta as informações necessárias para permitir aos teólogos e estudiosos a elaboração de suas teses. As estruturas de metadados podem suportar de forma mais adequada o complexo conteúdo de obras como a BHS.

O processo apresentado permite agregar outras camadas de informação de forma padronizada, facilitando o compartilhamento de segmentos específicos da informação. Na fase inicial, onde o volume de codificação é grande, esse modelo é adequado para fragmentar o conteúdo da obra e distribuir as tarefas entre vários colaboradores, ou ainda dividir o trabalho de marcação entre grupos especialistas em uma visão determinada.

Esse processo pode ser estendido em aplicações de outros domínios que tenham necessidade de integrar dados e metadados. Os princípios em que foram baseados esses modelos são os mesmos que influenciam a evolução da Web. Construir aplicativos e codificar conteúdo para a *Web Semantic*, parece ser inevitável.

5.2 Trabalhos futuros

5.2.1 Frame de controle

A proposta de criar um conjunto de *frames* que possa controlar o comportamento dos frames que codifiquem as informações e as relações transtextuais, também faz parte da teoria de *frames* apresentada por Minsky, embora, como mencionado no seu trabalho, tenha sido proposta por um de seus alunos, Scott Fahlman:

> "Eu imagino uma base de dados em que conjuntos dos fatos e agentes (*demons*) relacionados são agrupados em pacotes. Dentro desses pacotes quaisquer números de fatos e/ou agentes podem ser ativados ou disponibilizados para o acesso. Um pacote pode ativar outros pacotes (recursivamente); se um pacote, que relaciona um lista de pacotes, for ativado, os pacotes listados serão ativados também, e quaisquer dados deles tornam-se disponíveis a menos que sejam especificamente modificados ou cancelados. Assim, ativando alguns pacotes apropriados, o sistema pode criar um ambiente sob medida para execução que contém somente a parcela relevante de seu conhecimento global e de um conjunto apropriado de agentes. Eventualmente, teremos que adicionar pacotes novos específicos ao conjunto ativo a fim de tratar de alguma

> situação especial, mas esta inconveniência será de longe mais conveniente do que a complexidade de constantemente tropeçar sobre conhecimento não desejado ou de disparar agentes indesejáveis."[Minsky1974].

Berners-Lee fala sobre o mesmo conceito no artigo da Scientific American, maio/2001, chamando-o de *Composite Capability/Preference Profile* (CC/PP). A W3C está em fase de padronização do recurso CC/PP, onde são definidos os elementos básicos e atributos que permitem sua ampla utilização na indústria móvel de comunicação, além de ser um dos principais elementos da Web Semantic. A padronização que está sendo proposta utiliza a representação descrita com RDF-S.

Nos sistemas de *E-learning* essa função é responsável por definir que caminhos um usuário vai percorrer, dependendo da interação com o tutor virtual. Esse conceito é conhecido como AHS (*Adaptive Hypertext System).*

Em uma eventual utilização na representação da BHS, poderia ser implantada como um frame que receberia valores para seus atributos de forma interativa com o usuário, criando visões seletivas de forma dinâmica ou produzindo diferente formatação, conforme o tipo do periférico de saída detectado.

Embora de importância clara na implementação de filtros dinâmicos, ou adaptivos sobre uma base de informação codificada, nesse trabalho essa etapa não é obrigatória, podendo ser implantado futuramente, quando o modelo tiver atingido maturidade.

5.2.2 Outros domínios

A solução adotada pode ser aplicada amplamente com suas funcionalidades em obras didáticas de ensino, por exemplo, na área de direito, medicina, literatura e mecânica. Essas áreas possuem, em geral, um texto base, com uma série de anotações, comentários ou conteúdo relacionado com trechos do texto base. Esses elementos adicionais podem ser encarados como metadados e de tal forma podem ser codificados com os recursos em XML/RDF.

O processo de busca pela informação será facilitado pela estrutura de representação do RDF. Os textos voltados para a área de ensino, em geral, possuem uma estrutura bem definida por seus autores, o que facilita a organização das diferentes visões e dimensões que precisam ser codificadas. As ontologias, uma vez definidas, possuem um alto grau de reusabilidade para obras de um mesmo domínio.

Outro domínio que pode ser atendido pelo modelo presente é a codificação de "obras antigas", pois as características dessas obras em geral incluem preocupação com qualidade do original, comparações das cópias existentes, identificação de estilo do autor ou do copista, originalidade, anotações posteriores, ausência de um sistema de referência e relações transtextuais. O Ketib oferece uma forma de representação que pode ser facilmente navegada, flexível para ser recomposta em diferentes unidades e que permitam adicionar meta níveis de informação. Com esses recursos, o trabalho dos peritos encarregados de diferentes tipos de análises é facilitado. A representação com RDF-S é a estrutura mais adequada para atender a essa lista de requisitos, sendo as instâncias descritas com XML/RDF.

Particularmente, no domínio de trabalhos de crítica textual é possível aplicar esse processo para representar as informações dos textos associados ao conteúdo das análises de um ou mais pesquisadores. Nesses trabalhos existe uma identificação clara da necessidade de estruturas de metadados.

5.2.3 Redefinição da ontologia de aplicação

Na expansão do modelo sugerido neste trabalho, poderá ser fortemente conveniente a tarefa de reescrever o padrão OSIS na linguagem *Ontology Web Language* (OWL), que está em fase de padronização pela W3C.

A linguagem OWL é resultado da fusão das linguagens DAML e OIL, que por sua vez são baseadas em RDF-S. Os recursos presentes nessas linguagens são construídos especialmente para definir ontologias. Além dos mecanismos presentes na linguagem RDF-S, a OWL oferece recursos adicionais como: operações de negação, conjunção e

disjunção com classes, propriedades de transitividade, inversões e restrições qualificadas. Esse conjunto presente na OWL supera a somatória dos benefícios de XML-S e RDF-S e poderia ser criado um novo modelo Ketib baseado também na OWL.

O novo modelo que seria criado teria recursos para consultas e criação de filtros complexos. Permitiria por exemplo, que uma propriedade definida tivesse predicados com lógica de ordem superior. Esse recurso é muito apreciado por usuários no estudo e análise de texto.

Atualmente, o Xlink permite a inferência inversa; entretanto ainda há falta de editores e *browsers* que suportem a padronização desse recurso aos usuários. Está previsto para o primeiro semestre de 2003 a padronização pelo W3C da estrutura de Xpointer, que vem complementar as referências que a tecnologia Xlink define. Tão logo as ferramentas processem esse novo padrão, seria muito interessante sua utilização em conjunto com Xlink.

Algumas ferramentas, como por exemplo, uns dos mais conhecidos softwares de análise de textos teológicos do mercado, implementam parcialmente essas funcionalidades, no entanto não estão baseado em textos descritos em XML, e sim em estruturas de dados proprietárias.

O nível de maturidade que a OWL deve atingir para ser usado regularmente pode levar vários meses. Enquanto isso, o investimento de criar uma definição em RDF-S não representaria desperdício de esforços, pois a representação seria integralmente aproveitada na linguagem OWL, apenas podendo ser otimizada.

Bibliografia

[Ahmed2001]K., et.al; *Professional XML Metadata.* Wrox Press. 2001.

[Almeida, A. e Costa, J. M. 1992] Material complementar da Bíblia de Thompson

[Anderson1973]J. e Bower, G; *Human Associative Memory.*

[Arango1994] *Domain Analysis Concepts and Research Directions.* Workshop on Software Architecture, 1994, USC Center for Software Engineering, Los Angeles.

[Berners-Lee2001] J.Hendler and O.Lassila; *The Semantic Web*: Scientific American

[BHS1997] *Biblia Hebraica Stuttgartensia*, Editio Funditus Renovata, 1997

[Bray2001]T.; *What is RDF?*, http://www.xml.com, Jan, 24, 2001

[Brickley2000]D., GUHA R; *Resource Description Framework (RDF) Schema Specification* 1.0. W3C

[Chandrasekaran1999]B, et al. *What Are Ontologies, and Why do We Need Them*?, IEEE Inteligent Systems & their applications, vl 14 n. 1, jan/fev 1999.

[Clancey1993]W. J; *The knowledge level reinterpreted: modelling socio-technical systems*, International Journal of Intelligent Systems

[Dahlgren1995]K. A; *Linguistic Ontology.* International Journal of Human-Computer Studies

[Davenport2002]K.; *Library Journal,* Northeast Iowa Regional Library System, May

DC (Dublin Core). *The Dublin Core Home Page.*
http://purl.oclc.org/metadata/dublin_core.

[Devlin1999]K.; *Infoscience: Turning Information into Knowledge.* New York: W.H. Freeman

[DeVos2001]A., Widergren S.; *XML for CIM Model Exchange* - Power Industry Computer Application (IEEE)-PICA2001]

[Durand1990]D., Allen Renear; *What text realy is?*

[Durand1996]D., Allen Renear , Elli Mylona; *Refining our Notion of What Text Really Is: The Problem of Overlapping Hierarchies*; Research in Humanities Computing, Oxford University Press

[Durusau2002]P., Matthews O'Donnell; *Coming down from the trees Next step in the evolution of markup?* - Conference Extreme Markup 2002

[Fahlman1973]Scott; *Frame Verification*, MIT Press

[Genette1982]G.; *Palimpsestes*. Paris: Seuil

[Gilson1952]E.; *Being and Some Philosophers*. Pontificial Institute of Mediaeval Studies. Toronto

[Gómez-Perez1996]A.; *Towards a Method to Conceptualize Domain Ontologies*. ECAI'96 - Workshop on Ontological Engineering, Budapest.

[Guarino1997]N.; *Understanding, building and using ontologies*. International Journal Human-Computer Studies, v. 45, n. 2/3, fev./mar. 1997.

[Guarino1998]N.; *Formal Ontologies and Information Systems*, First International Conference (FOIS), Trento, Itália.

[Janssen2000] W, Koolwaaij J., Stefanova M; *XML Hype or Hope*, TI/RS/2000/038 Telematica Instituut

[Lassila1999]O.; *Resource Description Framework (RDF)*. W3C

[Lassila2001]O., Deborah McGuinness; *The Role of Frame Based Representation on the Semantic Web*, KSL Tech Report Number KSL-01-02. January, 2001. Knowledge Systems Laboratory, Stanford University

[McGuinness2001]D., Noy N.; *Ontology Development 101: a Guide to Creating Your First Ontology*, Stanford University

[Minsky1974]M.; *Frames*, MIT-AI Laboratory Memo 306, Cambridge - EUA

[Minsky1991]M.; *"Logical x Analogical or Symbolic x Connectionist or Neat x Scruffy"*, AI Magazine vol1.

[Neighbors1981] *Software Construction Using Components*. 1981. Tese (Doutorado) - Universidade da Califórnia, Irvine

[O'Leary1997] *Impediments in the use of explicit ontologies for KBS development*. International Journal Human-Computer Studies, v. 46, n. 2/3, 1997.

[OSIS2002]*Open Scripture Information Standard* – htttp://www.bibletechnologies.net

[PICA2001]*Power Industry Computer Applications Conference*, Sydney

[Prieto-Diaz1991]R.; *Domain Analysis and Software Systems Modeling*. Los Alamitos, CA: IEEE Computer Society Press

[Quillian1968]R.; *Semantic Information Processing*, MIT press, Cambridge Mass.

RDF Prime – Web Consortium 2002 – http://www.w3c.org/rdf

[Robinson2001]D, Levy E.; *The Masoretes and the Punctuation of Biblical Hebrew*, Bible Society in Israel

[Scott1999]W. R; *A Simplified Guide to BHS*, Bibal Press

[Setzer2001]V.; Meios Eletrônicos e a educação: Dado, Informação, Conhecimento e Competência, Editora Escrituras/SP

[Sowa2000]J.; *Knowledge Representation: logical, philosophical and computational foundations*, Brookes/Cole

[TEI2002]*Text Encode Initiative* P4, 2002

[Tov2001] Emanuel, *Textual Criticism of the Hebrew Bible*, Second Revised Edition

Glossário

Aparato Crítico: Comentários dos editores da obra que contêm explicações, variações ou observações sobre trechos específicos. Geralmente estão em formato simbólico extremamente compactado.

Cantilena: Acentos colocados no texto que indicavam a musicalidade da leitura do texto hebraico. Em alguns casos é útil na análise de discurso.

Caractere diacrítico: Sinais adicionados a um caractere com a função de modificar o seu significado.

Cólofons: Informações acrescentadas pelos escribas que dão detalhes sobre os massoretas, as fontes usadas, a data da conclusão, número de letras, sentenças ou outras seções mais amplas do texto.

Exegeta: Estudioso das Escrituras Sagradas com a função de interpretar e explicar.

Massoretas: Nome dado aos homens que escreviam as notas marginais no texto da Bíblia Hebraica.

Nakdanim: Em hebraico, significa "pontuadores". Nome dado àqueles que realizavam o trabalho de escrever no texto consonantal os pontos vocálicos e acentos.

Palimpsesto: O uso de escrever-se em pergaminhos fez com que o couro de animais utilizado para a escrita fosse, muitas vezes, reaproveitado, apagando-se a escrita antiga para, sobre ela, colocar-se a nova escritura. Era o palimpsesto, no qual a nova escritura, recobrindo a escritura anterior, deixava entrever os traços da primeira. O conteúdo de maior valor é o que foi sobrescrito.

Perícope: Divisões criadas pelos massoretas. Podem ser de seções ou de conjunto de frases.

Pontuação: Adicionado ao texto consonantal, com o propósito de representar vogais e documentar a forma correta de leitura do texto em hebraico.

Sopherim: Em hebraico, significa "aqueles que contam". Ficaram conhecidos como "escribas". Responsáveis pela cópia das Escrituras Sagradas.

Anexo A

```
<?xml version="1.0" encoding="UTF-8" ?>
<osis xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
 xsi:noNamespaceSchemaLocation="http://www.bibletechnologies.net/osisCore.1.1.1.xsd">
 <osisText osisIDWork="Codex L" osisRefWork="defaultReferenceScheme">
 <header>
  <work osisWork="Codex L">
    <title>Codex L</title>
    <identifier type="OSIS">Bible.BHS</identifier>
    <language>he</language>
    <refSystem>Bible.BHS</refSystem>
  </work>
  <work osisWork="defaultReferenceScheme">
    <refSystem>Bible.BHS</refSystem>
  </work>
 </header>
 <div type="testament">
  <div type="book" osisID="Gen">
   <div type="chapter" osisID="Gen.1">
     <verse osisID="Gen.1.1"> berê'shiyth bârâ' 'elohiym 'êth hashâmayim ve'êth
                              hâ'ârets</verse>
     <verse osisID="Gen.1.2"> vehâ'ârets hâyethâh thohu vâbhohu vechoshekh `al-penêy
                              thehom veruach'elohiym merachepheth `al-penêy hammâyim
     </verse>
     <verse osisID="Gen.1.3"> vayyo'mer 'elohiymyehiy 'or vayhiy-'or </verse>
   </div>
  </div>
 </div>
</osisText>
</osis>
```

Anexo B – Exemplo de codificação OSIS v1.1

```
<book id="BCV-MRK" value="MRK">
<title>
<part type="main">Sit amet Consectetuer</part>
</title>
<text>
<div> <chapter value="1" />
<head>Adipiscing elit Sed diem nonummy nibh</head>
<parallelPassage>
 <sourceRef book="MRK" chapter="1" verse="2" verseEnd="8" />
 <targetRef book="MAT" chapter="3" verse="1" verseEnd="11" />
 <targetRef book="LUK" chapter="3" verse="2" verseEnd="16" />
 </parallelPassage>
<p continued="no">
<verse id="BCV-MRK-1.1" value="1" />
 Euismod tincidunt
 <keyWord type="glossary">erat</keyWord>
  ut  <keyWord type="glossary">Volutpat</keyWord>
 , lacreet Dolore magna Aliguam.
<note type="variant">Ut. wisis enim:
<refText>ad Minim veniam Quis</refText> .
</note>
<verseEnd id="BCV-MRK-1.1-END" />
</p>
<p continued="no">
 <verse id="BCV-MRK-1.2" value="2" />
  Nostrud exerci tution ullamcorper suscipit lobortis nisl Ut:
</p>
<lineGroup type="stanza">
<lineGroup>
 <line>
  <q to="Q-BCV-MRK-1.2-000-END" id="Q-BCV-MRK-1.2-000"
     direct="unspecified" />
     Aliquip ex ea commodo consequat duis te
     feugifacilisi duis autem, </line>
 <line>
     dolor in hendrerit in vulputate.
<qEnd from="Q-BCV-MRK-1.2-000"
          id="Q-BCV-MRK-1.2-000-END" />
<note type="xref">
  <canonRef book="MAL" chapter="3" verse="1" />
</note>
<verseEnd id="BCV-MRK-1.2-END" />
</line>
</lineGroup>
```

Submetendo a descrição em xml do trecho anterior ao aplicativo XEP, é produzido o seguinte conteúdo no formato 'pdf':

Sit amet Consectetuer
Adipiscing elit Sed diem nonummy nibh
1:2–8 — Mt 3:1–11; Lu 3:2–16

1 Euismod tincidunt *erat ut *Volutpat,
lacreet Dolore magna Aliguam.[a]

2 Nostrud exerci tution ullamcorper
suscipit lobortis nisl Ut:
«Aliquip ex ea commodo consequat
duis te feugifacilisi duis autem,
dolor in hendrerit in vulputate.»[b]

Anexo C – Exemplo de codificação em RDFS

```
<?xml version="1.0"?>
<rdf:RDF
        xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
        xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#
        xml:base="http://example.org/schemas/vehicles">
<rdf:Description rdf:ID="MotorVehicle">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
<rdf:Description rdf:ID="PassengerVehicle">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="Truck">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="Van">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="MiniVan">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#Van"/>
        <rdfs:subClassOf rdf:resource="#PassengerVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="Person">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.w3.org/2001/XMLSchema#integer">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Datatype"/>
</rdf:Description>
<rdf:Description rdf:ID="registeredTo">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:domain rdf:resource="#MotorVehicle"/>
        <rdfs:range rdf:resource="#Person"/>
</rdf:Description>
<rdf:Description rdf:ID="rearSeatLegRoom">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:domain rdf:resource="#PassengerVehicle"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
</rdf:Description>
<rdf:Description rdf:ID="driver">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:domain rdf:resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="primaryDriver">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Description>
</rdf:RDF>
```

```
<?xml version="1.0"?>
<rdf:RDF
        xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
        xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#
        xml:base="http://dcc.unicamp.br/Livro">

<rdf:Description rdf:ID="BHS">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>
<rdf:Description rdf:ID="R">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#BHS"/>
</rdf:Description>
<rdf:Description rdf:ID="N">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#BHS"/>
</rdf:Description>
<rdf:Description rdf:ID="mp">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#N"/>
</rdf:Description>
<rdf:Description rdf:ID="ap">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#N"/>
</rdf:Description>
<rdf:Description rdf:ID="col">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#N"/>
</rdf:Description><rdf:Description rdf:ID="ap">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#N"/>
</rdf:Description>
<rdf:Description rdf:ID="TextoComplexo">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
        <rdfs:subClassOf rdf:resource="#N"/>
        <rdfs:subClassOf rdf:resource="#R"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.w3.org/2001/XMLSchema#integer">
        <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Datatype"/>
</rdf:Description>
<rdf:Description rdf:ID="est">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:domain rdf:resource="#N"/>
        <rdfs:range rdf:resource="#mp"/>
</rdf:Description>
<rdf:Description rdf:ID="ref">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:domain rdf:resource="#N"/>
        <rdfs:range rdf:resource="#mp"/>
</rdf:Description>
```

```
<rdf:Description rdf:ID="mm">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:domain rdf:resource="#N"/>
        <rdfs:range rdf:resource="#mp"/>
</rdf:Description>

<rdf:Description rdf:ID="rearSeatLegRoom">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:domain rdf:resource="#PassengerVehicle"/>
        <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#integer"/>
</rdf:Description>
<rdf:Description rdf:ID="driver">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:domain rdf:resource="#MotorVehicle"/>
</rdf:Description>
<rdf:Description rdf:ID="primaryDriver">
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
        <rdfs:subPropertyOf rdf:resource="#driver"/>
</rdf:Description>
</rdf:RDF>
```

Anexo D – Exemplo de codificação Ketib

1) Schema do modelo Ketib

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:rdfs="http://www.w3.org/2000/01/rdfschema#"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:dcterms="http://purl.org/dc/terms/">

<rdf:Description rdf:ID="BHS">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Description>

<rdf:Description rdf:ID="R">
   <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdf:type rdf:resource=http://www.bibletechnologies.net/osisCore>
   <rdfs:label>Referência</rdfs:label>
   <rdfs:comment>Classe baseada na simplificação dos elementos definidos no padrão
                 OSIS.</rdfs:comment>
</rdf:Description>

<rdf:Description rdf:ID="Livro">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#BHS"/>
    <rdfs:subClassOf rdf:resource="#R"/>
</rdf:Description>

<rdf:Description rdf:ID="N">
   <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:subClassOf rdf:resource="#BHS"/>
   <rdfs:subClassOf rdf:resource="#R"/>
   <rdfs:label>Metadados</rdfs:label>
   <rdfs:comment>Camada de metadados que acomoda todas as dimensões
                 não pertencente ao texto básico.</rdfs:comment>

</rdf:Description>

<rdf:Description rdf:ID="col">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#N"/>
</rdf:Description>

<rdf:Description rdf:ID="ap">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#N"/>
</rdf:Description>

<rdf:Description rdf:ID="massora">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#N"/>
</rdf:Description>
```

```
<rdf:Description rdf:ID="mp">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#massora"/>
</rdf:Description>

<rdf:Description rdf:ID="ms">
  <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:subClassOf rdf:resource="#massora"/>
</rdf:Description>

<rdf:Description rdf:ID="nm">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="#n"/>
</rdf:Description>

<rdf:Description rdf:ID="est">
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="#mp"/>
  <rdfs:label>Estatística</rdfs:label>
  <rdfs:comment>Contém dado estatístico de ocorrência de palavra ou, expressão marcado
                pelos massoretas </rdfs:comment>
</rdf:Description>
</rdf:RDF>
```

2) Instância do metadado NM

```
<rdf:RDF
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   xmlns:dcterms="http://purl.org/dc/terms/">
   xmlns:N="http:// dcc.unicamp.br/~800271/N-ketibSchema#/">
   xml:base="http://dcc.unicamp.br/~800271/n1">

   <rdf:Description rdf:about="http://dcc.unicamp.br/~800271/n1/gn1">

      <dc:description>Notas massoréticas de Gn 1</dc:description>

      <N:nm>
         <rdf:Bag>
            <rdf:li> rdf:resource="/gn1/mp/1"/>
            <rdf:li> rdf:resource="/gn1/ap1#a"/>
         </rdf:Bag>
      </N:nm >

   </rdf:Description>

   <rdf:Description rdf:about="http://dcc.unicamp.br/~800271/n1/gn2">

      <dc:description>Notas massoréticas de Gn 2</dc:description>

   </rdf:Description>
</rdf:RDF>
```