Universidade de São Paulo
Instituto de Matemática e Estatística

Prof. Yoshiharu Kohayakawa

Relatório de Estudos - MAC 5701

Construção e Estudos sobre Ontologias de Bioquímica com o uso do PowerLoom.

Orientadora: Renata Wassermann

Anderson Carlos Daniel Sanches
anderson@ime.usp.br

Índice

0.0 Resumo

Palavras-chave: ontologia, bio-ontologia, powerloom, bioquímica, bioinformática.

Na seção 0.1 são demonstradas as motivações deste trabalho e por que ontologias são importantes na área de biologia, em especial bioquímica. Durante o capítulo 1 são debatidos os vários significados do termo ontologia, seguido de descrições e aplicações de bioinformática. No capítulo 2 é feito um estudo de algumas das mais importantes bio-ontologias atuais, como o Gene Ontology, o EcoCyc, o TAMBIS e Ontology for Molecular Biology. Dois assuntos formam a base do capítulo 3, as linguagens de descrição de ontologias, onde são apresentados quatro das mais influentes, OKBC, XML, DAML+OIL e LOOM/PowerLoom, e os editores de ontologia mais promissores que trabalham com cada uma destas linguagens, Protégé, SymOntoX, OILED e OntoSaurus. A seguir, no capítulo 4 há a conclusão e a indicação de como a pesquisa será continuada. Por fim é apresentada a bibliografia.

0.1 Introdução: por que ontologias para biologia?

Biólogos necessitam de conhecimento para realizarem seus trabalhos. Um biólogo freqüentemente irá usar algum tipo de conhecimento pré-existente para fazer inferências sobre o assunto pesquisado. O exemplo mais comum disto na biologia molecular é o uso de comparação de seqüências para inferir a função de uma nova seqüência de proteínas, ou até a descoberta de novos caminhos metabólicos (metabolic pathways) [BOM]. A causa disto é que se uma seqüência de função desconhecida é altamente similar a uma seqüência de função conhecida, então é provável que a nova seqüência também tenha a mesma função. Então, ao invés de usar uma regra, lei ou equação para encontrar a função da proteína, um biólogo usa o conhecimento de que uma seqüência similar tem uma função conhecida, para fazer um julgamento sobre a função da nova seqüência. Esta é a razão de os biólogos dizerem que Biologia é uma disciplina baseada em conhecimento, ao invés das disciplinas “baseadas em axiomas”.

Os biólogos modernos também precisam de conhecimento para comunicação. Biologia é uma disciplina rica em informações, que se apresentam como um banco de conhecimento em que os biólogos adicionam mais conhecimento. Este conhecimento é guardado em centenas de bancos de dados e muitos deles precisam ser usados coordenadamente durante uma pesquisa. Conhecimento é vital em duas partes durante este processo. Quando é usado mais de um repositório de conhecimento ou ferramenta de análise, um biólogo precisa estar seguro de que o conhecimento de um repositório pode ser comparado confiantemente com o outro. Um primeiro exemplo pode ser o diferente uso do termo “gene”, dentro da comunidade de pesquisadores. Em alguns bancos de dados, gene pode ser definido como “a região de codificação do ADN (DNA)”, em outro como “um fragmento do ADN (DNA) que pode ser transcrito e traduzido em uma proteína” ou ainda “região de interesse biológico do ADN (DNA) que possui um nome e que transporta características genéticas ou fenótipos” [SSK]. Estar apto a confrontar uma definição ou razão sobre diferentes definições, para poder comparar os repositórios, seria uma vantagem. A segunda necessidade para o conhecimento é definir e confinar informação em um repositório. Informações biológicas podem ser muito complexas. Não apenas no tipo de dados armazenado, mas também na riqueza e no trabalho de estabelecimento das relações entre estas informações. Quando do projeto de um banco de dados é útil que se possa descrever quais valores podem ser especificados para cada atributo sob certas condições. Isto é o encapsulamento do conhecimento biológico dentro do esquema do banco de dados.

É importante para um único biólogo lidar com todos os domínios do conhecimento. O surgimento do mapeamento de todo o genoma só realça essa situação. Existe, entretanto, uma necessidade de criar sistemas que podem transcrever o conhecimento dos especialistas do domínio em informações biológicas representadas em software. Ainda não está claro se estes sistemas especialistas podem ser melhores para fazer novas descobertas do que os especialistas humanos, entretanto, estes sistemas podem ter um papel crucial na ajuda do processamento de informações e interagir com estes especialistas. A partir disto surgem várias questões em particular o que concerne como o conhecimento pode ser guardado de forma que seja utilizável por aplicativos.

Uma das respostas a esta questão é o uso de ontologias para descrever este conhecimento, tornado-o utilizável por pessoas e máquinas. A premissa da necessidade de ontologias em bioinformática é a necessidade

de disponibilizar o conhecimento para a comunidade e para aplicativos. No capítulo 1, será definido com mais detalhe o termo “ontologia”.

1.0 Definições de ontologia

Neste capítulo serão explicados termos e definições de ontologia e palavras relacionadas. Ao final as ontologias serão comentadas do ponto de vista da bioinformática. O estudo de algumas destas bio-ontologias é o assunto do capítulo 2.

1.1 Ontologias: o que são e pra que servem?

Definição da palavra

A palavra “ontologia” dia a dia ganha mais popularidade, principalmente entre os que estudam o compartilhamento de conhecimento. Entretanto, seu significado tende a permanecer um pouco vago, já que o termo é usado em diferentes significações. Limitando o escopo às acepções mais comumente encontradas nas comunidades que estudam o compartilhamento de conhecimento, pode-se isolar sete interpretações possíveis para o termo “ontologia” [G&G]. São elas:

1. Ontologia como uma disciplina da Filosofia;
2. Ontologia como um sistema conceitual informal;
3. Ontologia como uma proposta semântica formal;
4. Ontologia como uma especificação de uma “conceituação”;
5. Ontologia como uma representação de um sistema conceitual através de uma teoria lógica:

5.1 caracterizada por propriedades formais ou
5.2 caracterizada apenas para propósitos específicos;

6. Ontologia como um vocabulário usado por uma teoria lógica;
7. Ontologia como um meta-nível de especificação de uma teoria lógica.

A interpretação 1 é bastante diferente de todas as outras, pois trata de toda uma disciplina. As discussões deste capítulo focalizarão as interpretações 2 ~ 7. As interpretações 2 e 3 concebem uma ontologia como entidade conceitual semântica, formal ou informal, enquanto nas interpretações 5 ~ 7 uma ontologia é objeto sintático específico. A interpretação 4, que pode ser definida como a interpretação mais próxima da definição de o que é uma ontologia para a comunidade de Inteligência Artificial, é a que mais desperta interesse. Ela pode ser classificada como sintática, mas seu significado preciso depende do entendimento dos termos “especificação” e “conceituação”.

De acordo com a interpretação 2, uma ontologia é um sistema conceitual (não especificado) que pode-se assumir para basear uma base de conhecimento particular. Na interpretação 3, ao contrário, a ontologia que suporta uma base de conhecimento é expressa em termos de estruturas formais satisfatórias no nível semântico. Em ambos os casos, pode-se dizer que “a ontologia da base de conhecimento 1 é diferente daquela da base de conhecimento 2”.

Na interpretação 5, uma ontologia não é nada mais que uma teoria lógica. A questão é se essa teoria necessita ter uma descrição formal para ser uma ontologia, ou então é suficiente considerar qualquer teoria lógica como uma ontologia. Esta última posição é apoiada por aqueles que defendem que uma ontologia é uma coleção de afirmações formais (formal assertions), anotada e indexada, sobre algo. Somente a coleção das afirmações, na lógica, é chamada de teoria.

De acordo com a interpretação 6, uma ontologia não é vista como uma teoria lógica, ela é apenas o vocabulário usado por uma teoria lógica. Esta interpretação recai na 5.1 quando uma ontologia é pensada como uma especificação de um vocabulário que consiste de uma coleção de definições lógicas. Pode-se dizer que a interpretação 4 também recai na 5.1 se uma conceituação for planejada como um vocabulário. O problema é deixar claro o que significa o termo “conceituação”.

Finalmente, sob a interpretação 7, uma ontologia é vista como um meta-nível de especificação de uma teoria lógica, no sentido que especifica a arquitetura dos componentes (ou primitivas) usadas em uma teoria de domínio particular. Pode-se dizer também que é a ontologia que especifica, para uma teoria que tem a forma de constantes matemáticas, o que é uma constante e como ela difere de uma fórmula de outro tipo. A ontologia seria como uma representação dos componentes e suas possíveis interações, com o propósito de prover uma estrutura para elaborar o resto do sistema.

Como visto, há muitas interpretações do sentido de ontologia. Por hora pode-se resumir o significado, de acordo com o campo da interpretação, desta forma:
-Filosofia: estudo do que existe no mundo.
-Inteligência Artificial: definição das entidades e das relações entre elas, relevantes a um domínio. Um entendimento compartilhado explícito em uma linguagem.

Há autores que definem ontologia como “uma especificação explícita e formal de uma conceituação compartilhada” [GRU]. Porém esta definição não é completa se não contemplar o significado de termos como especificação explícita, formal, conceituação e compartilhada:
-especificação explícita: conceitos, propriedade, relações, funções, constantes e axiomas são definidos explicitamente;
-formal: que pode ser lido por máquina
-conceituação: modelo abstrato de algum fenômeno do mundo
-compartilhada: conhecimento consensual

Voltando às definições de Guarino e Giaretta, para distinguir a Ontologia como disciplina filosófica, ela será escrita “O” (o maiúsculo). Os outros sentidos de ontologia serão analisados como uma maneira de especificar bases de conhecimento ou teorias lógicas, projetadas com o intuito de expressar conhecimento que pode ser compartilhado.

Um ponto de partida nesse sentido será a análise da interpretação 4. O problema principal com tal interpretação é que ela é baseada na noção de conceituação diferente da que comumente imaginada. Uma conceituação é uma coleção de relações extensíveis (extensional relations) descrevendo um estado particular das coisas, enquanto que a noção que comumente se tem é de algo intencional, como uma grade que deve ser preenchida com os vários estados das coisas [G&G].

A primeira importante distinção das interpretações possíveis de ontologia, já foi feita que é a separação da Ontologia como disciplina filosófica de todas as demais. Quando escrito “Ontologia” (com o “O” maiúsculo e sem um artigo indeterminado na frente) a referência é à disciplina filosófica, que é o ramo da Filosofia que trata da organização e da natureza da realidade. A Ontologia é geralmente contrastada com a Epistemologia, que se preocupa com as fontes e a natureza do conhecimento.

Aristóteles definiu Ontologia como a ciência de propósitos especiais. Ao invés de investigar uma classe de coisas e suas determinações, a Ontologia estuda a organização de todas as coisas. Isto é o conceito de Ontologia Geral, em contraste às Ontologias de alguma área, como as da Biologia (objeto de estudos deste trabalho) ou da Química. Esta distinção

pode ser denominada como as distinções entre Ontologia Formal e Ontologia Material. A tarefa da Ontologia Formal é determinar as condições da possibilidade do objeto em geral, e a individualidade dos requisitos que cada constituição de objeto deve satisfazer, não envolvendo nenhuma generalidade.

Algumas vezes, define-se Ontologia Formal como um desenvolvimento sistemático, formal e axiomático da lógica de todas as formas e modalidades de ser. Entretanto, a interpretação do termo “Ontologia Formal” ainda é debatida. Em geral a segunda definição parece ser mais fértil, já que leva em consideração ambos os significados de “formal”: de um lado o sinônimo de rigoroso e de outro relacionado com as formas de ser. Entretanto, com o que a Ontologia Formal se preocupa não é tanto quanto a existência de certos objetos, mas sim com a descrição rigorosa das formas de ser, como as características estruturais. Na prática, Ontologia pode ser entendida como uma teoria de distinções, que pode ser aplicada independentemente do estado do mundo. As distinções podem ser entre as entidades do mundo (objetos físicos, eventos, regiões, quantidades de assunto) ou entre as categorias meta-nível usadas para modelar o mundo (conceito, propriedade, qualidade, estado, regra, parte). Neste sentido, Ontologia Formal, como uma disciplina, pode ser relevante tanto para as áreas de representação do conhecimento quanto para aquisição do conhecimento [G&G].

Na comunidade de inteligência artificial, a palavra ontologia adquire outro significado. Aqui é ela é usada para denotar um objeto em particular em lugar de uma disciplina. Então surge uma possível confusão entre uma ontologia entendida como uma estrutura conceitual particular no nível semântico (interpretações 2 e 3) e uma ontologia entendida como um artefato concreto, no nível sintático, para ser usado em propósitos determinados (interpretações 4 ~ 7). Isto é uma importante distinção, e é evidente que não se pode usar o mesmo termo técnico para denotar ambas as coisas. Na prática corrente, porém, o termo ontologia é usado ambiguamente, com ambos os significados, tanto para se referir a (vários tipos de) artefatos no nível simbólico, ou suas contrapartes semânticas [G&G].

Ao invés de usar somente um significado para o termo, Giaretta e Guarino, propõe a adoção de diferentes condições para se referir explicitamente a cada um dos dois níveis, enquanto tolera-se uma certa ambigüidade do termo “ontologia”. A conceituação é usada para denotar uma estrutura semântica referente a um sistema conceitual em particular (interpretação 3) e teoria de ontologia (ontological theory) para denotar uma teoria lógica que pretende expressar conhecimento ontológico (interpretação 5). A base destas afirmações, que teorias ontológicas são um artefato de projeto, é que bases de conhecimento podem ser lidas, vendidas ou fisicamente compartilhadas. Conceituações, por outro lado, são a contraparte semântica das teorias ontológicas. A mesma teoria ontológica pode servir a diferentes conceituações, como uma mesma conceituação pode se basear em diferentes teorias ontológicas. Então o termo “ontologia” poderá ser usado ambiguamente, tanto como sinônimo de teoria ontológica, como sinônimo de conceituação. Só é preciso ser consistente à escolha que feita dentro do mesmo texto. De 1 ~ 4 o termo “ontologia” tem uma interpretação sintática clara.

Inicialmente o uso do termo “ontologia” (ontology) como relacionado a uma teoria ontológica é compatível com [G&G]:

1. A engenharia de ontologias é um ramo da engenharia de conhecimento que usa Ontologia (da filosofia) para construir ontologias.
2. Ontologias são tipos especiais de bases de conhecimento.
3. Toda ontologia tem sua conceituação subjacente.
4. A mesmo conceituação pode basear diferentes ontologias.
5. Duas bases de conhecimento diferentes podem compartilhar a mesma ontologia.

A definição de Tom Gruber vista anteriormente [GRU], que uma ontologia é como "uma especificação explícita de uma conceituação", onde está um objeto "explícito" que é concreto e nivelado ao símbolo. O problema com definição de Gruber, entretanto, é que ela faz uso de uma extensão da noção de "conceituação" que, ao ser compatível com a caracterização preliminar dada anteriormente, não serve aos propósitos de definir o que uma ontologia é. É necessária uma definição intencional de "conceituação" que satisfaça a esta necessidade.

Tomando o exemplo clássico do mundo dos blocos, pode-se levar em conta uma situação onde duas pilhas de blocos estejam dispostas sobre uma mesa. Uma conceituação possível desta cena é dada pela seguinte estrutura:

({a, b, c, d, e }, { em, sobre, limpo, mesa })

onde {a, b, c, d, e} é um conjunto chamado universo do discurso, consistindo de 5 blocos, e {em, sobre, limpo, mesa} é um conjunto de possíveis relações entre os blocos. Os dois primeiros, em e sobre, são relações binárias e os outros dois, limpo e mesa, são relações unitárias (unary relations). Fica claro que objetos e relações são entidades que podem ser estendidas. Por exemplo, a relação mesa, que é entendida como “suporte de um bloco” se e somente se um bloco está sobre a mesa, é apenas o conjunto {c, e}. É exatamente esta extensão da interpretação que origina problemas.

Usar termos da linguagem natural (como sobre, abaixo) na escolha da meta-linguagem para descrever uma conceituação, pode parecer nada mais que um dispositivo didático, mas estes termos lingüísticos conservam informações essenciais para se entender o critério usado para se considerar alguns conjuntos de tuplas de relações relevantes. Esta informação extra não pode ser contabilizada pela conceituação por si só.

Referindo-se ao exemplo dado, considerando agora um arranjo diferente dos blocos, onde c não está no topo de d, enquanto a e b juntos formam uma pilha separada, sobre a mesa. A estrutura correspondente seria diferente da anterior, gerando portanto uma conceituação diferente. É claro que não há nada de errado com este ponto de vista, se forem consideradas visões isoladas do mundo dos blocos. Mas o significado dos termos usados denota que as relações relevantes continuam as mesmas, desde que são invariantes a respeito das possíveis configurações dos blocos. De fato, na meta-linguagem adotada, são usados os mesmos termos (em, sobre, limpo, mesa) para demonstrar uma nova conceituação. Seria preferível dizer neste caso que o estados das coisas é diferente, mas a conceituação é a mesma. A estrutura proposta assim parece estar mais apta a representar um estado das coisas do que uma conceituação.

Para se capturar estas intuições, os termos lingüísticos que são usados para denotar relações relevantes não podem ser tratados como meros

comentários ou informação extra. Ao invés disso, a estrutura formal usada para uma conceituação deve levar em conta seu significado. Como a literatura filosófico-lógica ensina, este significado não pode coincidir com uma relação extensível [G&G].

Usando-se um arcabouço teórico-formal, um jeito comum de aproximar esse significado é conceber uma relação intencional (intencional relation). Isto significa que uma única relação extensível é sempre relativa a um possível estado do mundo. Formalmente, uma relação intencional de cardinalidade n em um domínio D é uma função de um conjunto W de todos os possíveis mundos de um conjunto 2^D^n de todas as possíveis relações de cardinalidade n em D. Esta função especifica um conjunto de extensões admissíveis relativas a um domínio e um conjunto de possíveis mundos considerados. Isto significa que não estão sendo considerandos apenas uma extensão do mundo, mas também estas relações com os outros possíveis mundos são especificadas. Conseqüentemente é possível representar uma conceituação pela seguinte estrutura intencional:

{ W; D; R }

onde W é um conjunto dos mundos possíveis, D é o domínio dos objetos e R é um conjunto de relações intencionais em D [MAS].

De acordo com essa representação intencional, uma conceituação leva em conta a intenção de significado dos termos usados para denotarem relacionamentos relevantes. Esse significado é supostamente permanece o mesmo se as extensões dos relacionamentos mudarem devido a diferentes estados das coisas. Isto significa que, por exemplo, a extensão da relação sobre, permanece a mesma relação intencional, seja qual for a disposição dos blocos. Intuitivamente, pode-se descrever uma conceituação como sendo dada por um conjunto de regras que confinam estruturas de pedaços da realidade, que um agente usa para isolar e organizar objetos e relacionamentos relevantes: as regras que dizem se um certo bloco esta sobre outro permanecem as mesmas independentemente de um arranjo particular destes blocos. Essas regras podem ser vistas como ligações conceituais que põem juntas extensões diferentes que pertençam ao mesmo relacionamento intencional.

Dado um conjunto de relacionamentos especificados por termos lingüísticos como os exemplos citados, existirão muitas conceituações de diversas formas que satisfarão ao significado de tais expressões. Uma teoria modal conveniente poderia ser usada para uma caracterização aproximada de tal significado pretendido, com o objetivo de excluir extensões diversas. Por exemplo, pode-se expressar esse confinamento intuitivo que uma tupla como <a; a> nunca deve pertencer à extensão dos relacionamentos especificados pela palavra sobre, por indicação.

para todo x, ~sobre(x, x)

Um outro confinamento interessante que pode ser útil para caracterizar relações unitárias como bloco é aquela relação que nunca pode ser perdida para suas instâncias, isto é, se é uma relação do objeto, o é em todos os mundos possíveis.

(para todo x, bloco(x) está contido em bloco(x))

Esse confinamento é chamado “rigidez ontológica”, e é usado para descriminar entre várias categorias ontológicas de relações unitárias [MAS].

Um conjunto de confinamentos formais como aqueles acima, expressos em uma linguagem modal apropriada, pode conseqüentemente ser usado para (parcialmente) caracterizar uma conceituação, no sentido da exclusão de extensões não desejadas dos relacionamentos relevantes, até para possíveis palavras diferentes daquela considerada. Em geral não se pode identificar uma única conceituação pelo significado de um conjunto de confinamentos formais, desde que este conjunto tenha muitos modelos. O conjunto destes modelos é exatamente o que foi definido como compromisso ontológico (ontological commitment) [G&G].

De acordo com essas considerações, uma teoria particular não é uma especificação de uma conceituação, já que conceituação pode ser caracterizada apenas parcialmente. O que pode-se especificar é um conjunto de conceituações, isto é um compromisso ontológico.

Um exemplo simples para ver como a notação semântica pode ser relacionada a objetos sintáticos como teorias lógicas:

Considere a seguinte teoria lógica [MAS].

T1:

para todo x, maça(x) está contido em fruta(x).
para todo x, pêra(x) está contido em fruta(x).

maça(a1).
vermelha(a1)

Se fosse preciso isolar o conteúdo ontológico dessa teoria, a tentativa seria a de individualizar, entre seus axiomas, aqueles considerados mais estritamente relacionados ao significado intrínseco pretendido dos predicados usados na linguagem. Por exemplo, os seguintes axiomas (que são geralmente relacionados no que é chamado de Tbox [G&G]) devem ser pretendidos como captura de parte do significado de maçã, pêra e fruta:

T2:

para todo x, maça(x) está contido em fruta(x).
para todo x, pêra(x) está contido em fruta(x).

Um conjunto destes axiomas recebe o nome de teoria ontológica (ontological theory). Uma teoria ontológica contém fórmulas que são consideradas serem sempre verdadeiras (e então compartilháveis entre múltiplos agentes), independentemente de um estado das coisas particular. Formalmente, pode-se dizer que esta fórmula deve ser verdadeira em todos os mundos possíveis.

Uma teoria ontológica como a anterior caracteriza muito aproximadamente o conteúdo ontológico da teoria de onde é extraída. Para melhorar o foco nesse contexto, deve-se olhar para a conceituação pretendida que baseia ambas T1 e T2, o que modela de uma forma melhor os aspectos ontologicamente relevantes da linguagem usada na nossa teoria inicial. De acordo com as seções anteriores, essa conceituação pode ser caracterizada

(de uma forma aproximada) por uma teoria modal apropriada T3. As fórmulas (teoremas) de T2 serão verdadeiras em todos os possíveis mundos pertencentes a essa conceituação, e necessariamente aparecerão como fórmulas em T3. Ainda mais, T3 deve conter outras fórmulas capturando fatos necessários não capturados por T2. Para o exemplo presente, tem-se uma teoria muito simples, como a seguinte [MAS]:

T3:

(para todo x, maçã(x) está contido em fruta(x)).
(para todo x, pêra(x) está contido em fruta(x)).
(para todo x, maçã(x) está contido em maça(x)).
(para todo x, pêra(x) está contido em pêra(x)).
(para todo x, fruta(x) está contido em fruta(x)).
~(para todo x, vermelho(x) está contido em vermelho(x)).

Essa teoria expressa alguns confinamentos no significado dos predicados, o fato de que maça, pêra e fruta formam a hierarquia, e que eles são “rígidos”, diferentes de vermelho. T3 é a especificação do compromisso ontológico de T1.

A mesma informação trazida por T3 pode ser expressa por uma teoria meta-nível, cujo domínio é dado pelos símbolos não-lógicos usados em T1. Por exemplo:

T4:

mação <= fruta.
pêra <= fruta.
rígido(maça).
rígido(pêra).
rígido(fruta).
~rígido(vermelho).

Essa teoria pode ser útil para ser adotada como uma especificação alternativa de um compromisso ontológico, assumindo é claro, que o significado de predicados como “<=” e rígido são de forma que T4 pode ser imediatamente convertido em T3 por meios de regras de transição apropriadas.

Mas o que é uma ontologia? Os principais componentes de uma ontologia são conceitos, relacionamentos, instâncias e axiomas.
-Conceito representa um conjunto ou classe de entidades em um domínio. Por exemplo proteína é um conceito dentro do domínio biologia molecular;
-Relacionamentos descrevem interações entre conceitos ou propriedades dos conceitos. Por exemplo, enzima é um tipo de proteína;
-Instâncias são as especializações ou exemplos dos conceitos. Por exemplo, albumina é uma instância do conceito de proteína;
-Axiomas são sentenças consideradas verdadeiras, sem necessidade de prova. Axiomas podem ser usados para restringir valores das classes e suas instâncias, ou ainda para incluir regras mais genéricas, tais como monossacarídeo são moléculas de açúcar simples que não podem ser decompostos em açúcares menores.

Voltando para a discussão inicial da interpretação do significado de “ontologia”. Guarino e Giaretta propuseram uma escolha entre as interpretações 2~7, dando um senso preciso e ao menos algumas indicações

dos sentidos das frases numerada nos tipos de ontologia. Restringe-se então, a escolha a três possíveis sentidos da palavra “ontologia”.

No primeiro sentido, ontologia é um sinônimo de teoria ontológica. Neste caso, as afirmações 1~4 têm uma única interpretação, enquanto que 5 significa que duas bases de conhecimento devem ter uma subteoria comum, que é a teoria ontológica. Esta escolha é consistente com a interpretação 5. Como já discutido, uma teoria ontológica difere de um teoria lógica arbitrária (ou uma base de conhecimento) pela sua semântica, desde que todos os axiomas sejam verdadeiros em todos os possíveis mundos que baseiam essa conceituação. Isto significa que enquanto uma teoria lógica arbitrária (contendo, por exemplo, uma afirmação como maça(a) e pêra(a), expressando incerteza sobre o objeto (a) deve representar um estado particular epistêmico, uma teoria ontológica pode ser usada para representar conhecimento comum independentemente de estados epistêmicos singulares. Devido a essa diferença entre uma teoria ontológica e uma teoria lógica arbitrária, a segunda interpretação 5. é descartada em favor da primeira. T2 é uma ontologia de acordo com essa interpretação [G&G].

Num segundo sentido, “ontologia” é um sinônimo de especificação de um compromisso ontológico. Esta escolha continua consistente com a primeira interpretação 5. Neste caso as afirmações 1~4 continuam tendo um único significado, enquanto 5 não faz sentido, e deve ser substituído por: “O compromisso ontológico de duas bases de conhecimento diferentes deve ser especificado pela mesma teoria”. T3 não é uma ontologia de acordo com essa interpretação. A linguagem usada por T4 é geralmente mais rica que a usada por T1: os propósitos são diferentes, já que o propósito de T3 é fazer saber o significado usando-se uma linguagem bastante expressiva, enquanto que a linguagem de T1 é o resultado de um balanceamento entre expressividade e eficiência computacional. T3 é uma teoria ontológica como T2, já que as fórmulas são sempre verdadeiras.

O terceiro sentido de ontologia é como sinônimo de conceituação. Esta escolha é consistente com a interpretação 3. Neste caso as afirmações 1~4 não tem sentido, enquanto a ocorrência de ontologia na afirmação 5 ganha uma interpretação semântica. Neste caso, a afirmação 5 é equivalente a “Duas bases de conhecimento diferente devem ter a mesma conceituação”. Nenhuma das teorias mostradas anteriormente são ontologias de acordo com essa escolha.

Mas qual o significado que a definição de Gruber: “uma ontologia é a especificação de uma conceituação” deve ter? Antes de tudo, é evidente que a terceira interpretação é incompatível com essa definição. Já que há boas razões para manter a última, é melhor que não seja usado o termo “ontologia” em um senso semântico sem que isto esteja claro no contexto.

Considerando os sentidos primeiro e segundo, que atribuem o rótulo ontologia a T2 e T3, respectivamente. Falando estritamente, nenhuma das duas pode ser considerada como uma especificação de uma conceituação, e aqui a definição de Gruber não pode ser aplicada. Se forem mantidas as intuições iniciais, deve-se enfraquecer a definição de Gruber, dizendo que uma ontologia é apenas uma contabilização parcial de uma conceituação. De acordo com essa escolha, ambos T2 e T3 devem ser chamados ontologias.

De fato, essa definição enfraquecida deixa espaço para ambos os sentidos, primeiro e segundo, e isto é exatamente o que deseja-se: o nível de especificação de uma conceituação que baseia a linguagem usada varia de acordo com o propósito: uma ontologia do segundo tipo chega perto a especificar a conceituação pretendida (e talvez seja usada para estabelecer consenso sobre a utilidade do compartilhamento de uma base de conhecimento particular), mas isto paga o preço de uma linguagem rica (e quase sempre indecidível e ineficiente). Uma ontologia do primeiro tipo pelo outro lado, é desenvolvida com um exemplo particular em mente, projetado para ser compartilhado entre usuários que já concordam com a base da conceituação [G&G].

Ainda existem alguns outros sentidos de ontologia entre os reportados nas definições, como o 6 e o 7. A abordagem que parece adotar essas definições é aquela seguida no projeto KAKTUS, onde uma ontologia é definida como um “ponto de vista meta-nível de um conjunto de possíveis teorias de domínio”. Em geral, este ponto de vista é um conjunto de definições meta-nível das categorias semânticas usadas em uma base de conhecimento. A forma dessas definições não está bem definida. O que é interessante é que a descrição de um conhecimento particular de acordo com essa categoria de meta-níveis deve ter a forma de uma teoria como T4. Existe entretanto um importante diferença: T4 usa categorias de meta-nível semântico definidas na linguagem de T3, enquanto pode-se querer evitar qualquer noção semântica.

O que não é uma ontologia?

Uma ontologia não é uma coleção de fatos advindos de uma situação específica, mas ela provê todas as entidades semânticas (ex.: classes) para descrever essa situação. Uma descrição concreta de uma situação usa esses conceitos para criar exemplos e anotá-los com seus predicados.

Uma ontologia não é um modelo de um domínio de aplicação, mas sim um compêndio de todos os blocos de construção com seus modos válidos de combinação requeridos para expressar uma teoria. Um modelo completo de um domínio de aplicação (ex.: química enzimática) seria um conjunto de hipóteses (possivelmente verificado) ou uma teoria.

Uma ontologia não é um esquema de banco de dados que descreve categorias e tipos de dados e organização no banco de dados, mas não necessariamente as relações entre as entidades atuais e a representação do mundo real a qual elas representam. Um esquema de banco de dados pode ser derivado de uma ontologia pela adição da informação do tipo de dados e traduzindo o formalismo da representação do conhecimento em um paradigma de gerenciamento do banco de dados, como o relacional ou o orientado a objetos. Vice-versa, um esquema de banco de dados pode ser usado inicialmente para estabelecer conceitos para povoar uma ontologia.

Uma ontologia não é uma base de conhecimento que obtém conhecimento sobre objetos individuais atuais, eventos, situações, experimentos etc, mas ela armazena uma coleção de tipos de objetos, eventos etc, usados para especificar aqueles objetos em sua situação atual. Alternativamente, poderia ser dito que uma ontologia é uma base de conhecimento particular, preenchida com conhecimento sobre conceitos e suas relações ontológicas.

Uma ontologia não é uma taxonomia que conhece apenas relações de super classes e sub-classes, onde uma ontologia é aberta em muitos tipos de relacionamentos entre conceitos (ex.: topológicos, compositivas).

Uma ontologia não é um vocabulário ou dicionário já que as palavras no dicionário não descrevem necessariamente a hierarquia e a relação entre cada conceito e não são organizadas de forma que suportem inferência computacional. Numa ontologia pode-se seguir um caminho de qualquer conceito para outro através das relações “é um” por exemplo, ou outras relações.

Uma ontologia não é uma rede semântica, que é mais um formalismo de representação geral que pode ser usado para implementar uma ontologia, mas não é a única escolha para isto.

Pode-se exemplificar as distinções ontológicas através do seguinte exemplo: ADN (DNA) pode significar várias entidades diferentes. Primeiro, existe uma substância, que é física e pode “cair no seu pé”. Segundo, ADN (DNA) pode referir-se a uma classe particular de substâncias químicas, que incluem características gerais comuns a todas as moléculas de ADN (DNA) e é usada por exemplo em modelagem molecular. Terceiro, ADN (DNA) pode significar um certo tipo de seqüência ou string que é um conceito matemático abstrato, pode ser objeto de certas operações matemáticas, mas não pode “cair no seu pé”. Quarto, ADN (DNA) é também usado no laboratório para referir-se a uma instância particular de uma seqüência, ex.: a seqüência do ADN (DNA) da E. coli K 12, que pode ser armazenada num banco de dados e precisa de um transmissor (memória, chip, papel) para existir. Existem ainda outras conotações de ADN (DNA) não citadas aqui [SSK].

Glossário

Será feita agora uma pequena síntese das definições e interpretações dos termos tratados neste capítulo.

- conceituação: uma estrutura semântica intencional que codifica as regras implícitas que contém pedaços da realidade.
- Ontologia Formal: a o desenvolvimento sistemático, formal e axiomático da lógica de todas as formas e modos de ser.
- engenharia ontológica: o ramo da engenharia do conhecimento que explora os princípios da Ontologia (Formal) para construir ontologias.
- teoria ontológica: um conjunto de formulas que pretende-se sempre verdadeiras de acordo com uma certa conceituação.
- Ontologia: o ramo da Filosofia que trata da organização e natureza da realidade.
- ontologia: (sentido 1) uma teoria lógica que dá uma contabilização explícita e parcial de uma conceituação; (sendido 2) sinônimo de conceituação.

1.1 Ontologia do ponto de vista da bioinformática

Como já foi dito, Ontologia é o estudo sobre que tipos de coisas existem – quais entidades e 'coisas' que existem no universo. A visão da Ciência da Computação sobre ontologia é um tanto mais estreita, onde uma ontologia é um modelo de trabalho de entidades e interações genéricas quaisquer (exemplo a ontologia Cyc) ou mais especificamente em um domínio particular de uma área do conhecimento, como biologia molecular ou bioinformática. A seguinte definição pode ser dada:

"Uma ontologia pode assumir uma variedade de formas, mas necessariamente inclui um vocabulário de termos, e alguma especificação dos seus significados. Isto inclui definições e indicações de como os conceitos são inter-relacionados, o que impões coletivamente uma estrutura ao domínio e limita as possibilidades de interpretação dos termos" [GRU].

Gruber define uma ontologia como uma "especificação de uma conceituação, usada para permitir que programas e pessoas compartilhem conhecimento". A conceituação é uma coleção de conhecimento sobre o mundo nos termos das entidades (coisas, as relações que carregam e o significado de seus termos). A especificação é a representação desta conceituação de uma forma concreta. Um passo nesta especificação é a codificação da conceituação numa linguagem de representação do conhecimento. O objetivo é criar um vocabulário coerente e uma estrutura semântica para trocar informação sobre aquele domínio.

Os principais componentes de uma ontologia são conceitos, relacionamentos, instâncias e axiomas. Um conceito representa um conjunto ou classe de entidades ou "coisas" dentro de um domínio. Proteína é um conceito dentro do domínio da biologia molecular. Conceitos são classificados em dois tipos:

1. Conceitos primitivos são aqueles que possuem apenas as condições necessárias (em termos de suas propriedades) para serem membros da classe. Por exemplo, uma proteína globular é um tipo de proteína com um núcleo hidrofóbico, então todas as proteínas globulares devem ter um núcleo hidrofóbico, mas podem existir outras coisas que também tenham um núcleo hidrofóbico que não sejam proteínas globulares.
2. Conceitos definidos são aqueles em que sua descrição é ao mesmo tempo necessária e suficiente para que algo seja um membro da classe. Por exemplo, células eucarióticas são tipos de células que têm um núcleo. Não apenas todas as células eucarióticas têm um núcleo, cada núcleo que contém a célula é eucariótico.

Relações descrevem as interações entre conceitos ou entre propriedades dos conceitos. Relações também caem em dois tipos:

1. Taxonomias que organizam conceitos dentro estrutura de árvores de sub e super conceitos. As formas mais comuns delas são:
    - Relações de especialização, comumente conhecidas como relações 'é um tipo de'. Por exemplo, uma enzima é um tipo de proteína, que por sua vez é um tipo de macromolécula.
    - Relações de parte descrevem conceitos que são partes de outros conceitos – proteínas têm o componente local da modificação.

2. Relacionamentos associativos que relacionam conceitos entre partes da estrutura de árvore. Exemplos comuns incluem os seguintes:

    - Relacionamentos nominais descrevem os nomes dos conceitos, como proteína tem um número de acesso chamado NumeroAcesso (no contexto da bioinformática) e gene tem um nome, no campo NomeGene.

    - Relacionamentos de posicionamento descrevem a localização de um conceito com respeito a outro – Cromossomos têm uma localização sub-celular, núcleo.

    - Relacionamentos associativos representam, por exemplo, as funções, que informam o que um conceito tem ou se está envolvido em algo, e outras propriedades do conceito – proteínas tem função receptora, proteínas são associadas com o processo transcrição e proteínas têm classificação no organismo como espécies (Protein hasFunction Receptor, Protein isAssociatedWithProcess Transcription, Protein hasOrganismClassification Species).

    - Existem muitos outros tipos de relacionamentos, como os de causa.

As relações, como os conceitos, podem ser organizadas em taxonomias. Por exemplo, o campo nome pode ser subdividido em nome do gene, nome da proteína e nome da doença. Relações também têm propriedades que capturam mais conhecimento sobre as relações entre conceitos. Isto inclui, mas não são restritas a:

- É necessário que um relacionamento guarde um conceito. Por exemplo, quando se descreve um banco de dados de proteínas, poder-se-ia dizer que o número de acesso da proteína (Protein hasAccessionNumber AccessionNumber) é uma propriedade de todas as proteínas.

- Se o relacionamento pode opcionalmente guardar um conceito, por exemplo, descrever que uma enzima tem a possibilidade de ter um cofator qualquer (Enzyme hascofactor Cofactor), mas nem todas as enzimas têm um cofator.

- Se o conceito ligado a um relacionamento é restrito a certos tipos particulares de conceitos. Por exemplo, proteína tem_função receptora (Protein hasFunction Receptor) restringe a relação hasFunction apenas para conceitos que são do tipo receptores. A relação proteína tem_função diz que uma proteína tem uma função mas não restringe a que tipo de conceito deve ser.

- A cardinalidade do relacionamento. Por exemplo, um número de acesso particular se refere a apenas uma proteína, mas um cromossomo tem vários genes.
- Se o relacionamento é transitivo, por exemplo se “proteína é_associada_com_processo transcrição” (Protein isAssociatedWithProcess Transcription) e transcrição é_associada_com_processo expressão_gênica (Transcription isAssociatedWithProcess GeneExpression), então proteína é_associada_com_processo expressão_gência (Protein isAssociatedWithProcess GeneExpression). As relações de taxonomia sempre têm esta propriedade.

Quando esta conceituação estiver terminada, uma ontologia foi produzida.

Instâncias são ‘coisas’ representadas por um conceito – colágeno é uma instância do conceito proteína. Estritamente falando, uma ontologia não deve conter nenhuma instância, porque ela é supostamente uma conceituação do domínio. A combinação de uma ontologia com instâncias associadas é o que se chama base de conhecimento (knowledge base). Entretanto, decidir quando algo é um conceito de uma instância é difícil, e freqüentemente depende da aplicação. Por exemplo, átomo é um conceito e potássio é uma instância deste conceito. Pode ser argumentado que potássio é um conceito que representa as diferentes instâncias de isótopos de potássio. Esta é uma questão bem conhecida e ainda aberta dentro da área de pesquisa de representação de conhecimento [SSK].

Finalmente, axiomas são usados para confinar valores para classes ou instâncias. Neste sentido as propriedades dos relacionamentos são tipos de axiomas. Axiomas também, entretanto, incluem mais regras gerais, como os ácidos nucléicos com menos de 20 resíduos são oligonucleotídeos.

1.2 Aplicações e tipos de bio-ontologias

Ontologias podem prover a um software muito do senso comum e da experiência que os especialistas humanos usam. Entretanto sua faixa de aplicação é ainda maior. Dois exemplos, integração e anotação de dados serão explicados.

Integração está envolvida com o problema da heterogeneidade semântica e sintática. Enquanto os problemas sintáticos são fáceis de serem resolvidos por programas de reconhecimento de padrões, a heterogeneidade semântica precisa de um repositório semântico unificado para ser resolvida. Cada banco de dados, por exemplo, deve ser alinhado com a estrutura e o conteúdo de cada um dos outros bancos envolvidos. Já que o significado não é necessariamente simétrico quando mapeado para outro banco, no caso de n bancos de dados teria-se n * n possibilidades de integração. Entretanto, se conseguirmos colocar uma ontologia “no meio” desses n bancos de dados, o esforço de integração seria reduzido para n apenas, já que cada banco de dados deveria ser mapeado para essa ontologia. Um algoritmo poderia inferir os conceitos similares de qualquer banco para a ontologia e da ontologia para qualquer outro banco [SSK] [SEM].

Para anotação de dados, em princípio, não é necessário possuir-se uma ontologia completa como descrita anteriormente. É necessário somente um vocabulário controlado, já que o propósito principal é prover pontos de referência únicos e constantes. Um vocabulário deste tipo é o que está sendo desenvolvido no Gene Ontology (GO). O objetivo do GO é prover os GO ID, ou identificadores GO, o que significa que novos conceitos ganham novos GO ID, velhos conceitos permanecem com os seus números, mesmo que forem para outras localizações da hierarquia, e os números dos conceitos que forem apagados não serão reutilizados.

Um ideal comum para uma ontologia é que ela deve ser reutilizável. Esta ambição distingue uma ontologia de um banco de dados, mesmo ambos sendo conceituações. Por exemplo, um banco de dados é feito com o intuito de satisfazer apenas uma aplicação, mas uma ontologia pode ser reutilizada em muitas aplicações. Entretanto, uma ontologia só é reutilizável quando for empregada para os mesmos propósitos originais a que foi desenvolvida. Nem todas as ontologias têm a mesma intenção e podem haver partes que sejam reutilizáveis e partes que não sejam. Elas podem variar também em sua cobertura e nível de detalhe.

Pode-se dividir as ontologias em três tipos:

1. Orientadas ao domínio, que são tanto de domínios específicos (exemplo de um domínio: bactéria E. coli) como de generalização de um domínio (ex.: função do gene);

2. Orientadas a tarefas, onde há também tarefas específicas (como análise de anotação) ou generalização de tarefas (como resolução de problemas);

3. Genéricas, que capturam conceitos comuns de alto nível, como físico, abstrato, estrutura e substância. Isto pode ser especialmente útil quando da reutilização de uma ontologia, já que isso permite que os conceitos sejam corretamente ou confiantemente colocados. Elas podem ser importantes também quando da geração ou análise da linguagem natural

usando uma ontologia. Ontologias genéricas são também conhecidas como “ontologias superiores” (upper ontologies), ou de referência.

A maioria das bio-ontologias tem uma mistura de todos estes três tipos de ontologia. Uma ontologia bem escrita é construída de uma forma modular usando uma mistura de domínio genérico, tarefas genéricas e ontologias de aplicação. Suas partes são claramente definidas de forma que ela possa ser reutilizada. Uma ontologia não tão bem escrita tem estas distinções borradas, fazendo o reuso e as modificações mais difíceis. A medida de quão bem as dependências em uma ontologia foram separadas é conhecida como compromisso ontológico (ontological commitment). Outras medidas para a qualidade de uma ontologia incluem clareza, consistência, completude e concisão.

Ontologias são usadas em uma larga área de aplicações:

1. Uma referência comunitária de autoria neutra. O conhecimento é descrito em uma única linguagem, e convertido em diferentes formas para ser usado em múltiplos sistemas alvo. Benefícios incluem o reuso do conhecimento, ótima manutenibilidade e retenção de conhecimento de longo termo;

2. Definindo um esquema da base de dados ou definindo um vocabulário comum para anotação da base de dados – ontologia como especificação. Descrever uma entrada de proteína como “ADN mitocondrial ligando proteínas em helicoidal dupla” garantirá que um vocabulário comum esteja disponível para descrição, compartilhamento e elaboração de questões (ver item 4 desta lista). Benefícios incluem a documentação, manutenção, confiabilidade, compartilhamento e reuso do conhecimento.

3. Prover um acesso comum a informação. A informação deve ser compartilhada, mas é expressa com o uso de um vocabulário informal. A ontologia facilita a compreensão por prover um entendimento compartilhado dos termos ou mapeamentos entre os termos. Benefícios incluem interoperabilidade e um uso e reuso mais efetivo dos recursos do conhecimento.

4. Busca baseada em ontologia por perguntas (queries) em bancos de dados. Uma ontologia é usada para procurar num repositório de informações. Por exemplo, quando da busca em um bancos de dados por “ADN mitocondrial ligando proteínas em helicoidal dupla”, apenas estas proteínas serão encontradas, pois termos exatos para procura foram usados. Apesar do usuário dos termos estar certo sobre seu significado, o conhecimento pode ter sido representado de outras formas, não favorecendo a pesquisas por termos exatos.

Perguntas (queries) às bases de dados podem ser refinadas acompanhando relacionamentos dentro da ontologia, por exemplo, seguindo relacionamentos para encontrar aqueles processos nos quais proteínas de certas funções agem, e recolhendo informações sobre quais as proteínas são associadas. Mover para cima e para baixo a hierarquia “é um tipo de” dentro da ontologia também pode ser usado para refinar perguntas. Por exemplo, especializando “ADN ligando proteínas” para “helicoidal simples de DNA ligando proteínas” por mover para baixo a hierarquia quando a busca anterior trouxer respostas em demasia. Benefícios incluem um acesso mais eficaz e a partir deste, um melhor reuso dos recursos de conhecimento.

5. Entendendo a anotação da base de dados e literatura técnica. Essas ontologias são projetadas para suportar o processamento de linguagem natural, que não apenas liga conhecimentos do domínio, mas também conhecimentos relacionados com a estrutura lingüística como gramáticas e léxicos.

Embora algumas metodologias que comparem a estrutura e as regras de várias ontologias comecem a surgir, nenhuma delas compara o conteúdo de uma ontologia com outra para um domínio específico.

2.0 Estudo de algumas bio-ontologias

O uso de ontologias dentro da bioinformática é relativamente recente, conseqüentemente não há um grande número em existência. Nesta parte, uma amostra representativa de algumas bio-ontologias existente será mostrada. Este estudo foi restrito as ontologias mais pertinentes às tendências correntes em bioinformática e biologia molecular, ao invés de um largo campo da Biologia. A Biologia é rica em termos de taxonomias, como as classificações das enzimas e das espécies. Sendo taxonomias, elas apenas usam uma hierarquia simples. As ontologias estudadas aqui tendem a ser ricas no uso de seus relacionamentos, daqui sua inclusão, mas isto não é para denegrir a utilidade das taxonomias em muitas aplicações. As ontologias estudadas são as seguintes:

2.1 Gene Ontology (GO) http://www.geneontology.org/

2.2 EcoCyc ontology http://www.ecocyc.org/

2.3 TAMBIS Ontology (TaO) http://img.cs.man.ac.uk/tambis

2.4 RiboWeb ontology http://smi-web.stanford.edu/projects/helix/riboweb.html

Existem dois fatos importantes no estudo destas ontologias. O primeiro é que ontologias estão sendo usadas pelas comunidades de pesquisa, para prover entrada de conhecimento a bancos de dados e aplicações. O segundo é que todas essas ontologias são diferentes e específicas para seu uso pretendido. TaO é uma ontologia de tarefas de bioinformática, então ela contém conceitos como Identidade da Proteína (ProteinId), que não são parte da biologia molecular. O TaO não pode ser substituído pela ontologia EcoCyc. Go é uma ontologia de funções da produção genética, e RiboWeb representa o conhecimento das sub-unidades das estruturas Ribossômicas, dados e metodologias. Como o GO é usado para anotação de banco de dados (isto significa que ele funciona como uma referência de termos) ele possui um nível fino de detalhamento onde o TaO é bastante raso, mas ganha-se precisão no TaO durante a formulação de perguntas, unindo-se conceitos. Mesmo que uma única ontologia fosse desenvolvida, as aplicações continuariam a usar um subconjunto da ontologia, conduzindo a uma exigência de ontologias altamente modulares com dependências e suposições minimizadas entre elas. O uso da ontologia influencia o conteúdo e a natureza do conhecimento capturado, e isto não é uma contradição das capacidades de armazenar conhecimento das ontologias. Não apenas o propósito determina o escopo e a granularidade para qual o mesmo conhecimento é representado em diferentes ontologias, mas também as conceituações do mesmo domínio podem ser diferentes sem estarem erradas.

Por exemplo, o TaO descreve que o ADN (DNA) deve ser traduzido para proteína. Isto está errado em termos de biologia molecular, mas é uma característica da bioinformática, então conceituações do mesmo domínio podem ser diferentes. Algumas vezes um confinamento é necessário para uma aplicação e em outras não é. Isto apenas muda qual conhecimento é capturado ou como ele é capturado, isto não muda o próprio conhecimento [SGB].

O conteúdo, em termos de escopo, conceitos e relacionamentos, bem como o uso de cada ontologia será apresentado. Haverá um breve resumo a respeito da organização, estrutura, propósitos e conteúdo de cada uma.

2.1 Gene Ontology

O Gene Ontology (GO) tem a anotação da base de dados como seu objetivo principal. O GO tem crescido bastante e é um grupo de bancos de dados. O escopo do projeto GO é limitado. Ao invés de tentar descrever todo o conhecimento de biologia molecular compreendido pelos bancos de dados comunitários, GO procura armazenar informação sobre regras de produção gênica dos organismos. A classificação das funções do gene por Riley tem um escopo similar, mas é apenas para a bactéria E. coli. GO foi inicialmente criado para refletir as funções genéticas na drosófila, através do banco de dados Flybase, mas foi expandido para abranger o banco de dados da expressão gênica de ratos, moscas, leveduras e minhocas, e espera-se que continue expandindo. Assim, o objetivo principal do GO é um vocabulário controlado, para anotações conceituais, dos processos e da localização da função de produção gênica, em bancos de dados.

Sintetizando, o Gene Ontology trata de quatro organismos: S. cerevisiae, D. melanogaster, M. musculus e C. elegans. Destes três organismos pode-se procurar por conceitos em três classificações: função molecular, componente celular e processo biológico.

GO não possui nenhuma ontologia de organização de alto-nível. GO é essencialmente composto de três hierarquias, representando a função da produção gênica e o processo no qual ela ocorre, além da localização celular e estrutura. GO contém um vasto campo de conceitos, e provê um rico nível de detalhe nas suas três hierarquias. São usados relacionamentos “é um tipo de” e “é uma parte de” para descrever as regras da produção genética. Já há mais de 5.000 conceitos na ontologia.

GO define um nível fino, de detalhe conceitual, por exemplo: ADN mitocondrial ligando proteínas em helicoidal dupla, fatores de transcrição, proteína de músculo motor, memória e aprendizado, coagulação sanguínea, morfogenética, formação do padrão ventral e muitos caminhos [GLI], transportes e sinais em vários sistemas. GO utiliza herança múltipla na relação “é um tipo de” para formar alguns de seus conceitos e há algum uso do relacionamento “é uma parte de”. Muitas das relações capturadas pelos conceitos, entretanto, continuam implícitas no GO. Por exemplo, o conceito “succinate (cytosol) to fumarate (mitochondrion) transporter” implicitamente carrega propriedade sobre o local e orientação na membrana mitocondrial [SGB].

Outro importante papel do Gene Ontology Consortium é o de estabelecer vocabulários controlados. Eles permitem que um software explore as bases de dados genéticas e vinculem genes relacionados entre si utilizando palavras que descrevem sistematicamente suas funções, independentemente de como se chamam os genes. Porém, alcançar um consenso sobre toda a variedade de organismos estudados nos laboratórios pode ser muito difícil. Acostumados a trabalhar em relativo isolamento, os cientistas que estudam espécies diferentes se habituam a suas respectivas tradições de denominação genética.

Os geneticistas da mosca do vinagre, por exemplo, deram nomes inusitados a alguns genes: “hedgehog” (roda dentada) que produz uma proteína sinalizadora que participa de diversos processos de desenvolvimento, “lost in space” (perdido no espaço) guia o crescimento das células nervosas [ELP]. Devido à característica de um mesmo gene poder ter

funções diferentes, uma nomenclatura por função que estaria correta hoje, pode não ser a mais adequada no futuro, quando da descoberta de outras funções de tal gene. Alguns pesquisadores defendem que não deve haver nomes, apenas números para se referir a um determinado gene.

Mais do que tentar impor um sistema homologado de denominações ou números para os genes, os membros do consorcio GO estão estabelecendo palavras aceitáveis para descrever as funções moleculares, os processos biológicos e os componentes celulares. Mediante esses termos, pode-se vincular genes relacionados, independentemente da nomenclatura.

Os pesquisadores do GO determinam termos GO a cada gene e a seus produtos, e a informação é enviada a base central do GO, na Universidade de Stanford. O objetivo é construir uma base de dados plenamente consultável, que explique a função dos genes em todos os organismos. Os glossários são dinâmicos.

O GO também é suficientemente flexível ao aceitar sinônimos para os casos em que os pesquisadores utilizam-se habitualmente de vários termos para o mesmo processo, como divisão celular e citoquinese (citoquinesis). Devido a estas características, o GO está ganhando popularidade com rapidez. Em muitas ontologias estudadas durante este trabalho foram encontrados termos referindo o GO.

Entretanto, os princípios de projeto do GO não previnem todos os problemas. As principais relações “é um” e “é uma parte de” não são sempre usadas de forma consistente. Por exemplo, “é um” pode ser uma subclasse ou “instância de”, isto é, não há uma distinção entre entidades genéricas e individuais no GO que restrinja claramente a capacidade de expressão. Similarmente, “é uma parte de” é encontrado em alguns lugares com significados diversos, como “feito de”, “pertence a”, “é uma parte física de”, “é uma parte conceitual de”, “sub-processo de”, “controla”, “causa”, “ativa”, “inibe", "incluído por" e "ligado a" [SSK].

O GO não auxilia muito na compreensão dos seus vários conceitos, já que na maior parte deles não há uma definição. Não existe um princípio de projeto claro no GO. Não há uma orientação clara de qual caminho deve-se seguir para encontrar um determinado conceito. É quase impossível descobrir porque o GO é como ele é hoje, já que não há uma orientação de em que classe colocar cada conceito. Se alguém novo no GO quiser agregar algo a ele, não há outra forma senão a de examinar todos os ramos e conceitos e tentar imaginar se o que está sendo colocado faz ou não sentido de estar neste determinado ramo da árvore. Isto parece ser feito por intuição, já que não há qualquer critério de subclassificação específico para guiar a inclusão de novos termos. Não há também nenhum confinamento da integridade que garanta a consistência e a corretude do GO após cada adição de novos conceitos. Nem ao menos há regras ou gramáticas que expliquem como relacionar ou usar combinações de conceitos.

As três distinções do GO funcionam não como uma ontologia, mas sim como três ontologias, já que estas hierarquias não são ligadas entre si, e aparentam não estar relacionadas. Atualmente, o GO ainda é mais uma nomenclatura controlada para biologia molecular do que uma ontologia genética completa.

2.2 EcoCyc

Ecocyc, definida pelos autores, é uma “enciclopédia dos genes e metabolismo da Escherichia coli”. Na EcoCyc, uma ontologia é usada para descrever a riqueza e complexidade de um domínio e os confinamentos que agem com este domínio (os genes do E. coli) para especificar um esquema de banco de dados. A apresentação usa uma metáfora de enciclopédia. Ela cobre os genes, o metabolismo e caminhos metabólicos, regulação e transdução de sinais, por exemplo, nos quais os biólogos podem explorar, visualizando as informações. O banco de dados descreve atualmente 4402 genes, 944 enzimas codificadas por subconjuntos destes genes, 990 reações enzimáticas e 177 reações de transporte, organizadas em 173 caminhos metabólicos. EcoCyc usa a classificação das funções do gene por Riley [MRI] como parte de sua descrição. Os cientistas podem visualizar os genes dos cromossomos do E. coli, ou uma reação bioquímica individual, ou ainda um caminho metabólico com as estruturas dos componentes.

A visualização [SBS] e comparação de caminhos metabólicos são especialmente importantes na biologia, pois através da observação de caminhos metabólicos de seres semelhantes, os biólogos podem inferir as reações bioquímicas dos organismos que estão sendo pesquisados. Essa comparação se dá por observação ou é feita com o uso de ferramentas para isto, como o BioMiner [BOM], que atualmente só utiliza o KEGG [KEG].

O EcoCyc usa uma ontologia para definir um esquema de banco de dados com a vantagem da expressividade e da habilidade para levar em conta mudanças necessárias para acomodar a informação biológica. Desta forma o uso da ontologia é transparente para o usuário, exceto que os confinamentos expressos quando da captura do significado do conhecimento, mantenham seu significado preciso. No EcoCyc, por exemplo, o conceito de gene é representado por uma classe com vários atributo, ligados a outros conceitos como: produto polipeptídico, nome do gene, sinônimos e identificadores usados em outros repositórios, como o GO. O sistema de representação pode ser usado para impor confinamentos nestes conceitos e instâncias podem aparecer em lugares descritos pelo sistema [SSK].

2.3 TAMBIS

O objetivo do projeto TAMBIS é ajudar pesquisadores de ciências biológicas provendo um único local para acesso de informações biológicas. Apesar de acessar dados de diferentes repositórios, que os tratam de forma diferente, as respostas são processadas de uma forma consistente, para que o usuário tenha a impressão de estar consultando apenas uma ontologia. As questões dos usuários são procuradas nas fontes apropriadas e depois de processadas são devolvidos com detalhes sobre as fontes de informação.

TAMBIS tenta prover consultas transparentes e filtragem de informações de cunho biológico. Para fazer isto, existe uma camada de homogeneização, no topo das fontes. Esta camada usa mediadores e envoltórios para criar a ilusão de uma única fonte de dados. O projeto TAMBIS foi criado para ajudar a tarefa de obter e comparar seqüências e outras informações heterogêneas, já que são processos que demandam grande esforço manual, retardando o progresso das pesquisas.

Existem mais de 200 fontes de informações biológicas, cada uma delas repletas de dados, mas que só são úteis para seus próprios mecanismos. Existem três tipos principais dessas fontes:
- bancos de dados
- serviços on-line
- arquivos

O usuário deve fazer questões para cada tipo de sistema, como SQL para os bancos de dados e buscas nos arquivos. O usuário deve também interpretar as respostas recebidas das diferentes fontes, e ainda assim não tem uma visão geral. Isto significa que muito tempo é gasto selecionando-se a fonte apropriada, elaborando a questão e interpretando as respostas, levando à sub-utilização destes recursos. É nessa parte que o uso do TAMBIS ajuda os pesquisadores, por funcionar como uma única interface para todos os recursos e por dar a ilusão de um único recurso [TAM].

Quando o usuário faz uma questão, o TAMBIS encontra as fontes de informação, elabora as questões, integra os resultados e reporta-os de uma forma consistente. Isto torna mais efetivo o uso de múltiplas fontes de informação biológica.

Esse funcionamento se dá através de uma ontologia de referência, chamada BioCon. Essa ontologia é usada para: descrever meta-dados que baseiam as fontes de consultas, formular questões na linguagem de modelagem, dirigir uma interface gráfica para a formulação das questões, já que os usuários não estão preparados para escrever questões complexas e intermediar as várias fontes para tradução do modelo do mediador para as o modelo das fontes. A mediação explora a BioCon para ajudar a identificar e resolver equivalências ou quase equivalências nas fontes de informação.

Como citado anteriormente, há um mediador, que é um corretor de informações, e vários envoltórios. O mediador envia as consultas realizadas nesta ontologia para cada envoltório de cada ontologia utilizada nas bases de consulta, e depois recebe as respostas e os resultados obtidos são transcritos de volta nesta ontologia de referência.

2.4 RiboWeb

O objetivo do RiboWeb é a construção de modelos tridimensionais dos componentes do ribossomos e a comparação dos resultados com os estudos existentes. O conhecimento usado pelo RiboWeb para realizar suas tarefas é capturado em quatro ontologias:
-Coisa-física (Physical-thing)
-Informações (Data)
-Publicações (Publication)
-Métodos (Methods)

A ontologia das coisas físicas descreve os componentes e associações físicas dos ribossomos. Há aqui três princípios de conceituação: moléculas, moléculas-construídas e partes-de-molécula. A primeira descreve moléculas ligadas por covalência e inclui a maior parte das macromoléculas biológicas. A segunda captura as coleções de moléculas ligadas por outras formas, como complexos enzimáticos. O terceiro guarda conhecimento sobre regiões de moléculas que não existem independentemente, mas precisam ser conhecidas por conterem certas propriedades específicas. Este último tipo inclui cadeias de amino-ácido e os finais de algumas moléculas de ácido nucléico. A ontologia de Informações captura conhecimento sobre detalhes experimentais bem como dados da estrutura das coisas físicas. A ontologia de métodos contém informação sobre técnicas para analisar a informação. Ela guarda conhecimento de qual a técnica que pode ser aplicada em qual informação, bem como as entradas e saídas de cada método.

Instâncias são adicionadas ao RiboWeb na correspondência destes conceitos. Por exemplo, um artigo de uma publicação descreve uma estrutura tridimensional de 30 subunidades ribossômicas. Isto significa que instâncias precisam ser criadas e ligadas nas ontologias de publicação, informação e coisas-físicas. Um usuário pode querer visualizar se essa estrutura é consistente com as outras pertinentes ao RiboWeb. RiboWeb pode então mostrar conflitos com o conhecimento corrente do biólogo [SSK].

3.0 Ferramentas para descrever ontologias

Neste capítulo serão estudadas algumas das principais linguagens e ferramentas para se descrever ontologias. A escolha da linguagem em que será descrita a ontologia é um dos passos mais importantes de sua construção, já que o poder de expressão de uma linguagem determina que conhecimentos pode-se capturar em cada confinamento. Como as ontologias evoluem ao longo do tempo, de acordo com as novas descobertas e novos conhecimentos, é útil o uso de ferramentas para ajudar o mapeamento das transformações do conhecimento.

A linguagem de especificação é especialmente importante, pois as ontologias não são todas construídas da mesma maneira. Um grande número de linguagens pode ser usado, incluindo linguagens de programação em lógica (general logic programming languages) como o Prolog. Entretanto é mais comum que se use linguagens desenvolvidas especificamente para a construção de ontologias. O modelo “Open Knowledge Base Connectivity” (OKBC) e linguagens como “KIF” ou “Common Logic” são exemplos que se tornaram bases para a construção de outras linguagens. Existem também muitas linguagens baseadas em uma forma de lógica de forma especialmente computável conhecida como lógica de descrição (description logics). Exemplos incluem o Loom / PowerLoom e o DAML+OIL. Este último coopera com o padrão OWL (Web Ontology Language). A grande quantidade de informações disponíveis na Internet têm também dado incentivo à pesquisa de ontologias para organizar conteúdo. O World Wide Web Consortium tem proposto algumas tecnologias para isto, como os esquemas RDF como uma camada de linguagem e o XML para tipos de dados.

Quando da comparação de linguagens de descrição de ontologias, o que se deixa para trás por computabilidade e simplicidade é geralmente a expressividade da linguagem, o que geralmente é um mau negócio. Uma linguagem de descrição de conhecimento precisa ser rica e expressiva tanto quanto as nuanças e as características do conhecimento que ela captura [OBT].

Nas próximas seções serão estudadas quatro linguagens:
-OKBC
-XML
-DAML+OIL
-LOOM e PowerLoom

Editores de ontologias

Apesar de haverem numerosas abordagens para o desenvolvimento de ontologias de forma semi-automática, a maioria das ontologias atuais é criada manualmente com o uso de editores de ontologia.

Então quando inicia-se um projeto de construção de ontologias, um dos primeiros passos é encontrar um editor de ontologias apropriado [MOD]. Atualmente há mais de 50 editores disponíveis (em http://www.xml.com/2002/11/06/Ontology_Editor_Survey.html há uma lista comparativa de 52 deles), talvez devido a pouca idade do campo de pesquisas, que ainda não consagrou somente alguns poucos. Estas ferramentas são úteis para construirmos esquemas de ontologias sozinhos ou com suas instâncias.

Editores de ontologia são ferramentas que proporcionam navegação, codificação e modificações de forma a facilitar as tarefas de construção e manutenção de ontologias. Os editores existentes hoje são diferentes em muitos aspectos. Eles variam, por exemplo, na complexidade subjacente do modelo de conhecimento, na usabilidade e na escalabilidade. Entretanto, o desenvolvimento de ontologias é um processo necessariamente interativo e dinâmico. Porém todos eles provêm suporte suficiente para o desenvolvimento inicial de uma ontologia.

É praticamente impossível que a primeira ontologia escrita seja a melhor, e que continue sendo usada ao longo dos anos sem nenhuma modificação. As causas da mudança são inerentes à complexidade da realidade e a habilidade humana limitada para lidar com esta complexidade. Então, as ontologias devem estar aptas à evolução e a modificação, por muitas razões: ontologias freqüentemente têm erros de projeto, e algumas vezes não cumprem os requisitos dos usuários imediatamente; o ambiente no qual as ontologias operam pode mudar de forma imprevisível, levando a invalidar algumas afirmações que foram feitas quando a ontologia foi construída; os requisitos dos usuários podem mudar depois da construção inicial da ontologia, necessitando que a ontologia original evolua para cumprir os novos requisitos.

A necessidade de efetuar estas mudanças é advinda também de muitas aplicações ligadas ao mundo real, já que estas tipicamente operam em ambientes mutáveis. Um típico exemplo é o MEDLINE, um banco de dados com 11 milhões de referências a artigos de 4.600 revistas científicas em ciências biológicas ou o UNSPSC que a cada duas semanas faz mudanças em seus 16 mil conceitos, mudanças estas que alteram geralmente entre 100 e 600 conceitos. Isto causa sérios problemas a empresas de comércio eletrônico que a utilizam para classificar seus produtos [EVO].

A evolução de uma ontologia é a adaptação gradual da ontologia, bem como a propagação consistente destas alterações. Como a modificação de um conceito pode gerar inconsistências em outras partes de uma ontologia, principalmente se ela possuir muitos conceitos, fica evidente que o uso de ferramentas de software é bastante adequando e muitas vezes se faz necessário.

Além disto, a evolução de uma ontologia está se tornando mais importante a cada dia. O principal motivo é o grande número de ontologias que vem surgindo e o os altos custos associados com suas adaptações para requisitos que se modificam ao longo do tempo. O desenvolvimento de ontologias e suas aplicações são caros, mas modificá-los é ainda mais custoso. Entretanto, mesmo sendo a evolução de ontologias um requisito essencial ao longo do tempo, ferramentas e estratégias apropriadas ainda faltam. Este nível de gerenciamento de ontologias é necessário não apenas para o desenvolvimento inicial e a manutenção das ontologias, mas é essencial durante a distribuição, quando a escalabilidade, a disponibilidade, a confiabilidade e o desempenho são absolutamente críticos.

Nas seções seguintes, serão apresentados quatro editores:
-Protégé, que usa OKBC
-SysmOntoX, que usa XML
-OilEd, que usa DAM+OIL
-OntoSaurus, que usa LOOM

Há ainda outros tipos de ferramentas que cobrem outros aspectos do desenvolvimento de uma ontologia. Enquanto os editores de ontologias são úteis durante as etapas de construção, outros tipos de ferramentas para construção de ontologias são necessários ao longo do caminho.

Alguns projetos desenvolvem soluções que usam várias outras ontologias como fontes externas bem como ontologias recentemente desenvolvidas para completar estes projetos. As ontologias de outras fontes progredirão em uma série de versões. Ao final, a gerência cuidadosa destas coleções de ontologias heterogêneas se torna necessária para conhecer suas mudanças. Ferramentas podem ajudar neste processo, realizando mapeamentos, ligações e comparações entre elas, além de reconciliamentos e validações (após as mudanças acontecerem), bem como combinações e converções para outros formatos. As ontologias podem ser transformadas em outros formatos como esquemas XML W3C (W3C XML Schemas), esquemas de bancos de dados ou UML, para serem associadas em aplicações [OBT].

Exemplos destas outras ferramentas incluem, mas não estão limitadas a: Chimaera, FCA-Merge, PROMPT, ODEMerge (para combinação de ontologias); OntoAnalyser, OntoGenerator, OntoClean in WebODE, ONE-T (para geração de ontologias); AeroDAML, COHSE, MnM, OntoAnnotate, OntoMat-Annotizer, SHOE Knowledge Annotator (para anotação de ontologias); ICS-FORTH RQL, ILRT SquishQL, Intellidimension RDFQL, RDFPath, VERSA RDF Query Language, TRIPLE, DAML+OIL Query Language, Topic Maps Query Language, Ontopia Tolog (para a realização de consultas); ICS-FORTH RDFSuite, Sesame, Inkling, rdfDB, RDFStore, EOR, Redland, Jena, RDF Gateway, TRIPLE, KAON Tool Suíte, Cerebra, Emplis K42, Ontopia Knowledge Suite (para armazenamento e recuperação) [OBT] [OCE].

Outras ferramentas podem ainda ajudar a adquirir, capturar e visualizar os conhecimentos do domínio, antes e durante a construção da ontologia.

3.1 Linguagens para descrição de ontologias

Antes da apresentação das linguagens para descrição de ontologias, é necessário o conhecimento prévio de dois termos: KIF e Lógica de descrição.

KIF é o acrônimo de formato de troca de conhecimento (knowledge Interchange format) e foi uma das primeiras linguagens de representação de conhecimento. É uma linguagem formal, que contém semântica declarativa (o significado da na expressão representação pode ser entendido sem o uso de um interpretador para manipular esta expressão). KIF é logicamente compreensível, isto é, ele provê para a expressão de sentenças arbitrárias um cálculo de predicado em lógica de primeira ordem. Ele provê regras de raciocínio não monotônicas para a representação e ele provê definição de objetos, funções e relações [MOT].

Lógicas de descrição (description logics) descrevem conhecimento em termos de conceitos e regras de restrições que podem ser usadas para automaticamente derivar hierarquias de classificação. Lógicas de descrição permitem a definição de classes em termos da descrição que especifica as propriedades satisfazíveis pelos objetos pertencentes ao conceito. Lógicas deste topo, em geral, provêm uma faixa de formadores de operadores de conceitos que podem ser usados nessas descrições, incluindo conjunção, disjunção, negação e várias formas de regras de quantificação. Um aspecto chave das lógicas de descrição é sua semântica formal e suporte ao raciocínio. Lógicas de descrição definem fragmentos de lógica de primeira ordem que em geral tem alta expressividade mas que ainda assim são decidíveis e eficientes em procedimentos de inferência.

Lógicas de descrição tem uma interação difícil se forem tratadas diretamente. No passado, elas foram concebidas como grandes sistemas monolíticos que requeriam que o usuário construísse seus modelos diretamente em sua sintaxe [OIE].

Nas seções seguintes, serão mostradas as linguagens OKBC, SymOntos, DAML+OIL e LOOM/PoweLoom.

3.1.1 OKBC

OKBC é um acrônimo de Open Knowledge Base Connectivity, conhecido previamente como Protocolo de Quadro Genérico (Generic Frame Protocol). Ele especifica um protocolo, não uma linguagem. O protocolo faz suposições sobre o sistema subjacente do KR (quadros), e é complementar às especificações da língua desenvolvidas para suportar o compartilhamento do conhecimento.

O modelo de conhecimento GFP que é o formalismo de representação implícita no qual se baseia OKBC, provê uma representação do conhecimento centrada no objeto e suporta uma coleção de construtores de representação comumente encontrados em sistemas de representação de quadros: constantes, quadros, aberturas, facetas, classes etc. Ele também define uma interface completa do tipo conte&pergunte (tell&ask) para as bases de conhecimento acessadas usando-se o protocolo OKBC e procedimentos (com uma sintaxe parecida com a do Lisp) para descrever operações complexas a fim de prover acesso as bases de conhecimento quando acessadas através de uma rede. Ultimamente o OKBC-Ontology for Ontolingua tem ganhado destaque na comunidade OKBC [RDM].

3.1.2 XML

XML é o acrônimo de eXtended Markup Language, derivado do SGML, Standard General Markup Language. Ele foi desenvolvido pelo XML Working Group do World Wide Web Consortium, o W3C. XML está próximo de se tornar um padrão.

Como outras linguagens para a Web, suas principais vantagem são a facilidade de análise gramatical sintática por computador já que sua sintaxe é bem definida e a possibilidade de ser lida por pessoas. Existem muitas ferramentas de software para análise sintática e manipulação de XML. XML permite aos usuários a definição de suas próprias etiquetas (tags) e atributos, definição das estruturas de dados (aninhando-os), extração de dados de documentos e desenvolvimento de aplicações que testam a validade estrutural de um documento XML.

Quando XLM é usado como a base para a especificação da linguagem da ontologia, tem-se as seguintes características: a definição de uma especificação sintática comum por meio de um DTD (Document Type Definition); a informação codificada em XML é facilmente legível por humanos; pode ser usada para representar conhecimento distribuído através de várias páginas web, já que pode estar embutido nelas. Porém há também algumas desvantagens, que influenciam a especificação da ontologia: a falta de informações dentro dos rótulos torna difícil a localização de componentes dentro de um documento, e ferramentas padrão são disponíveis para a análise sintática e a manipulação de documentos XML, mas não para se fazer inferências. Estas ferramentas devem ser criadas para que se possa fazer inferências com linguagens que são baseadas em XML.

XML propriamente dito não possui características especiais para a especificação de ontologias, já que ele apenas oferece um simples, mas potente modo de se especificar uma sintaxe para uma linguagem de especificação de ontologias. Entretanto, pode ser usado para cobrir necessidades de compartilhamento de ontologias, explorando as facilidades de comunicação da world wide web [RDM].

3.1.3 DAML+OIL

Inicialmente DAML e OIL surgiram como duas linguagens distintas. A linguagem OIL (Ontology Inference Layer) é uma proposta para um esforço conjunto para descrição e compartilhamento de ontologias. Ela é uma linguagem baseada em quadros, como OKBC, XOL e RDF. O projeto é financiado pela Comunidade Européia através da IST do projeto Ibrow (Intelligent Brokering Service for Kowledge-Component Reuse on de World Wide Web), e do On-to-Knowledge.

O OIL unifica três importantes aspectos providos por diferentes comunidades: semântica formal e eficiente suporte a raciocínio (reasoning suport) como provido em lógicas de descrição, primitivas ricamente modeladas como provido pela comunidade das linguagens baseadas em quadros e uma proposta padrão para intercâmbio de anotações sintáticas como provido pela comunidade Web. Nesta linguagem, uma ontologia é descrita em três camadas:
-Nível de objetos, onde instâncias concretas da ontologia são descritas;
-Primeiro meta-nível, onde a ontologia pode ser definida;
-Segundo meta-nível ou recipiente da ontologia, onde meta-informações da ontologia são descritas como o nome do autor, o assunto etc.

Esta linguagem propõe uma descrição de uma ontologia nos elementos básicos de uma descrição baseada em quadros: classes, aberturas, facetas, e ainda oferece a possibilidade de declarar axiomas genéricos predefinidos, como classes disjuntas, cobertas ou disjuntas e cobertas e classes equivalentes. Isto também permite a adição de propriedades particulares em aberturas, como transitivo, simétrico ou funcional [MOT].

O projeto DAML (Darpa Agent Markup Language) propôs uma primeira versão de uma linguagem para descrição de ontologias chamada DAML-Ont. Depois de discussões entre esta linguagem e a proposição, os dois projetos se uniram para propor o DAML+OIL. DAML+OIL é uma linguagem baseada em um RDF e em um esquema RDF com rica modelagem de primitivas. Ela é muito similar a especificação original do OIL, mas possui algumas diferenças [OIE]. Em particular, DAML+OIL abandonou os ideais originais baseados em quadro do OIL, em um senso muito mais forte que o do OIL, uma sintaxe alternativa para uma lógica de descrição.

DAML+OIL provê uma modelagem de primitivas comumente encontrada em linguagens baseadas em quadros. A linguagem tem uma semântica clara e bem defina, baseada em lógica de descrição.

Uma ontologia DAML+OIL consiste de zero ou mais cabeçalhos, seguidos por zero ou mais elementos de classe, elementos de propriedade, axiomas e instâncias. Ela tem a vantagem do modelo RDF assim como URI e XML, para o intercâmbio e expressividade da proposição OIL. Um problema desta linguagem é que ela vai se tornando cada vez mais complexa para ser lida e também interpretada, tornando difícil saber se uma ontologia pode ou não ser reutilizada [MOT].

Afirmações em uma ontologia DAML+OIL como a superclasse ou confinamentos de aberturas aplicados a classes, residem em termos de axiomas gerais. A idéia de quadro, um lugar único no qual os fatos sobre a classe são recolhidos, está perdida, ou ao menos não é inerente à linguagem [OIE].

3.1.4 LOOM e PowerLoom

LOOM é uma linguagem de programação de alto nível e um ambiente, par uso na construção de sistemas especialistas e outros programas de aplicação inteligentes. Ele é um descendente da família KL-ONE e é baseado em descrição lógica, obtendo uma grande integração entre os paradigmas baseado em regras (rule based) e baseado em quadros (frame based).

LOOM suporta uma linguagem de descrição para modelagem de objetos e seus relacionamentos, e uma linguagem de afirmações (assertion) para a especificação de constantes ou conceitos e relacionamentos, e para afirmar fatos sobre indivíduos. Programação procedimental é suportada através de métodos direcionados a padrões, enquanto as capacidades de inferência baseadas em produção e classificação suportam um poderoso mecanismo de raciocínio, na forma de um motor de inferência, o classificador.

É importante focar na abordagem da lógica de descrição para modelagem de ontologias, que se diferencia da abordagem baseada em quadros. Definições escritas usando esta abordagem tentam explorar a existência de um poderoso classificador na linguagem, especificando conceitos pelo uso de um conjunto de restrições [RDM].

PowerLoom é o sucessor da linguagem de representação de conhecimento LOOM. Ele provê uma linguagem e um ambiente para a construção de aplicações inteligentes. PowerLoom usa uma linguagem de representação baseada em lógica bastante expressiva (uma variante do KIF), e usa como motor de inferência um estilo de dedução natural com encadeamento direto e reverso (forward and backward chainer). O motor de inferência não é um provador de teoremas completo de primeira ordem, mas ele pode manusear regras complexas, negações, raciocínio de igualdade (equality reasoning), e formulários restritos do raciocínio de ordem mais elevada. PowerLoom tem um classificador que pode classificar descrições expressas em lógicas de cálculo de predicados de primeira ordem. PowerLoom usa módulos como dispositivos estruturais para bases de conhecimento, e mundos (ambientes) leves para suportar raciocínio hipotético [PWL].

Na implementação do PowerLoom, uma nova linguagem de programação chamada STELLA (Strongly Typed, Lisp-like Language) foi criada. Ela pode ser traduzida em Lisp, C++ e Java. PowerLoom é escrito em STELLA e está disponível em versões Lisp, C++ e Java. PowerLoom está sendo desenvolvido na Divisão de Sistemas Inteligentes da USC Information Sciences Institute.

A título de exemplo da linguagem LOOM, será mostrada agora uma pequena ontologia sobre o vírus da hepatite A, proposta em [SRE]. Na próxima parte da pesquisa, a linguagem LOOM/PowerLoom será comparada com outras, em conjunto com um breve tutorial, e será proposta uma ontologia da área de bioquímica.

```
(define-class viral-hepatitis-a (?vh)
      "the inflammation process of liver caused by virus A; it has an
      incubation of 15 to 50 days and is accompanied by jaundice"
      :def (and (inflammation-process ?vh)
            (exists ?vir
                  (and (has-a-cause ?vh ?vir) (virus-a ?vir)))
            (exists ?liv
```

```
                  (and (is-embodied-in ?vh ?liv)
                        (and (liver ?liv)
                              (exists ?pat
                                    (and (part ?liv ?pat) (*patient
                                    ?pat))))))
            (exists ?inc
                  (and (is-constitutive-phase-of ?inc ?vh)
                        (and (incubation ?inc )
                              (= (temporal-value ?inc) ?n)
                              (>= ?n 15) (=< ?n 50))))
            (forall (?jau ?pat)
                  (=> (and (jaundice ?jau) (*patient ?pat) (is-embodied-
                  in ?jau ?pat))
                        (occurs-in ?jau ?vh))))
      :issues ((:see-also "in SNOMED-II the code is D-0521" "in ICD9-CM
      the code is 070.1")))

(defconcept viral-hepatitis-a :context infectious-diseases :is-primitive
      (:and inflammation-process
            (:some has-a-cause virus-a)
            (:some is-embodied-in (:and liver (:some part *patient)))
            (:some has-constitutive-phase (:and incubation
                  (:the temporal-value (:and day (:through 15 50)))))
            (:all has-occurrence-of (:and jaundice (:some is-embodied-in
            *patient))))
      :annotations
      ((documentation
      "the inflammation process of liver caused by virus A; it has an
      incubation of 15 to 50 days and is accompanied by jaundice")))

(define-class incubation (?i)
      "the kind of temporal context for the initial phases of infectious
      diseases, producing no evident medical signs or symptoms"
      :def (and (time-span ?i)
            (exists ?id
                  (and (infectious-disease ?id)
                        (exists ?ph
                              (and (has-constitutive-phase-of ?id ?ph)
                              (is-started-by ?id ?ph)
                              (is-context-of ?i ?ph)))
                        (exists ?pat
                              (and (*patient ?pat)
                                    (is-embodied-in ?id ?pat)
                                    (not (exists ?ms
                                    (and (or (evident-medical-sign ?ms)
                                          (symptom ?ms))
                                          (forall ?ph
                                                (=> (embodies ?pat ?ms)
                                                      (occurs-in
                                                      ?ms ?ph)))))))))))


(define-relation occurs-in (?te1 ?te2)
      "the relation of occurrence between temporal entities. For any
      temporal entities p and q such that occurs-in(p, q) holds, there
      can be a part of p in which q does not occur. Temporal entities
```

```
      include processes, contexts like situations and time spans, and
      signs representing an underlying process"
:def (and (or (process ?te1)
            (*sign ?te1)
            (context ?te1))
      (or (process ?te2)
            (*sign ?te2)
            (context ?te2))
      (exists (?s1 ?s2)
            (and (situation ?s1) (situation ?s2)
            (is-context-of ?s1 ?te1)
            (is-context-of ?s1 ?te2)
            (is-context-of ?s2 ?te2)
            (ist ?s1 "(occurs-in ?te1 ?te2)")
            (ist ?s2 "(not (occurs-in ?te1 ?te2)"))))
      :issues ((:see-also "the definition of 'during' in theory: time")))

(define-theory infectious-diseases (diseases micro-organisms))
```

3.2 Editores de Ontologias

Como já foi dito, editores de ontologia são ferramentas que proporcionam navegação, codificação e modificações de forma a facilitar as tarefas de construção e manutenção de ontologias.

Há vários tipos de editores, desde os gerais, para a construção de ontologias de qualquer domínio até os específicos, como por exemplo o OpenKnoMe, que suporta a referência GALEON de termos da área médica, para a construção de ontologias de medicina. Alguns editores são feitos especificamente para a construção de ontologias de nível superior, como o Cyc, ou o OpenCyc, que é a iniciativa de tornar público o ambiente Cyc.

Alguns editores de ontologias oferecem muitas outras coisas, como ferramentas de integração de informações, classificação lingüística e análise estocástica, para auxiliar a extração de informações de fontes com conteúdo não estruturado. A informação extraída pode se tornar instâncias ou estender a própria ontologia, como no editor Applied Semantic's.

Quando as tecnologias de ontologias apareceram nos anos 90, o foco era principalmente a aquisição do conhecimento, e isto influenciou algumas características dos editores de ontologias de então. Muitos deles adotaram o então popular método para desenvolvimento de bases de conhecimento conhecido como KADS. Esta orientação não é evidente nas ferramentas de hoje em dia. Umas poucas exceções incluem o "Ontology Works' IODE" e o "WebODE" da Universidade Técnica de Madri, ambos com suporte para abordagens específicas na organização das ontologias. Eles também incluem suporte para ontologias de nível superior, como o WordNet, o Cyc e outros [OBT].

A construção de ontologias é hoje uma prática fragmentada. A situação, em parte, é um resultado da proliferação das linguagens de lógica e modelos de informação que são combinados podendo criar ainda mais formatos de ontologias e ambientes de edição. Essas ferramentas e metodologias, junto com as ontologias criadas com elas, geralmente existem sem uma prova de interoperabilidade adequada. Este é um dos desafios frente aos modelos estabelecidos para integrar componentes das ontologias com outros padrões e sistemas de informação.

Ontologias existem para serem compartilhadas. Sua proposta de uso é servir como um ponto consensual de troca e interpretação de informações. Quanto mais aplicações e outras ontologias utilizarem uma determinada ontologia, maior é a utilidade desta e das ontologias inter-relacionadas. Isto requer compatibilidade formal em níveis sintáticos, bem como semânticos. Uma consideração, por exemplo, é a habilidade de uma ontologia de domínio em acomodar linguagens XML especializadas e vocabulários controlados que estão sendo adotados como padrão em inúmeros campos, como o GO. Nenhum editor atualmente tem capacidades como estas, entretanto alguns, como Modulant e Unicorn, já estão se movendo nesta direção.

A interoperabilidade, ao invés disto, está sendo implementada simplesmente pela habilidade do editor em importar e exportar ontologias em diferentes linguagens/serializações. Algumas ferramentas como o Ontolingua do Stanford Knowledge System Lab, oferecem um grande escopo de conversões, porém a maioria tem limitações. Importar ou exportar

ontologias em linguagens como o DAML+OIL e OWL geralmente significam traduções parciais e alguma expressividade perdida. Uns poucos editores como o Web ODE também oferecem capacidades de combinação entre ontologias homogêneas.

Em adição as características já mencionadas, os editores de ontologia variam consideravelmente em sua apresentação geral ao usuário. Em termos de variedade de características, especialmente as relativas a interfaces com outros sistemas de informação, o Protégé é um dos mais completos, além de contar com *plugins*. De um ponto de vista estritamente da linguagem, o Ontolingua e o OpenCyc oferecerão ambientes de desenvolvimento com grande expressão e especificação completa das ontologias. OpenCyc também provê acesso nativo a uma das maiores e mais completas ontologias disponíveis, a Cyc. OILED oferece um forte suporte para composição de expressões de descrição lógica.

As habilidades de organizar e gerenciar uma ontologia emergente são pontos chave de usabilidade em um editor de ontologias. A apresentação e manipulação conveniente e intuitiva dos conceitos e relações de uma ontologia são essenciais. Já que muitos modelos de ontologias suportam herança múltipla em suas hierarquias de conceito, e em suas hierarquia de relacionamentos, manter as associações visíveis é um desafio. A abordagem padrão é o uso de múltiplas visualizações em árvore, com níveis expansíveis e retráteis. Uma representação em grafo é menos comum, apesar de poder ser bastante útil para edição de ontologias que modificam conceitos e relacionamentos. Os grafos podem ser ainda mais efetivos, provendo uma amplificação local para facilitar a navegação pela ontologia, sendo ela de qualquer tamanho.

O “visualizador hiperbólico” incluído com o produto Applied Semantics, por exemplo, foca-se no grafo os conceitos, sem rotular as relações. Outras abordagens como o *plugin* Jambalaya para Protégé exibe um tipo de ‘zoom’ gráfico que aninha sub-conceitos dentro de seus super-conceitos e permite ao usuário seguir relações por saltos entre conceitos relacionados. Em ontologias complexas, a visualização gráfica ainda é um pouco confusa. É por isso que alguns outros editores, como o GALEN, por exemplo, não possuem nenhum visualizador gráfico.

Finalmente, vale a pena considerar o suporte a inferência contido nos editores de ontologia. Enquanto ontologias podem ser tratadas como especificações únicas, elas geralmente são usadas para responder perguntas sobre um corpo de informações. Alguns editores incorporam a habilidade de adicionar axiomas e regras de dedução a ontologia para avaliação de determinados objetivos no ambiente de desenvolvimento [OBT].

Neste capítulo, serão demonstrados quatro editores de ontologias, que funcionam respectivamente com as quatro linguagens já estudadas nas seções anteriores deste capítulo:
-Protégé, que usa OKBC
-SysmOntoX, que usa XML
-OilEd, que usa DAM+OIL
-Ontossauros, que usa LOOM.

3.2.1 Protégé

O Protégé [PTG] é uma ferramenta construída pela Seção de Informática Médica da Universidade de Stanford. Ele possui um ambiente de edição de bases de conhecimento e uma arquitetura extensível para a criação de outras ferramentas, ou seja, ele também tem uma API. Esta API é dedicada a desenvolvedores de software que desejem implementar novas linguagens e características que eles gostariam de suportar em suas aplicações. Protégé está disponível em diferentes plataformas, como Linux, Unix, Solaris, Mac OS e Windows. Também é possível a instalação de *plugins* para estender as capacidades do software [MOT].

A ferramenta tem ganhado muitos adeptos por todo o mundo que a utilizam para modelar um largo escopo de domínios, como protocolos para tratamento de câncer e estações de energia nuclear. Protégé está disponível gratuitamente através da licença Mozilla de fonte aberta.

O Protégé conta com um ambiente gráfico e interativo para o projeto de ontologias e bases de conhecimento. Isto ajuda os especialistas no domínio a realizarem suas tarefas. Os desenvolvedores de ontologias podem acessar informações importantes quando necessário, e pode-se usar manipulação direta para se administrar uma ontologia. Controles em árvore permitem uma navegação simples e rápida através da hierarquia de classes. O Protégé usa formulários como sua interface para preenchimento dos valores das aberturas (slots). O modelo de conhecimento do Protégé é compatível com o OKBC. Ele inclui suporte para classes e hierarquia de classes, com herança múltipla, modelos e aberturas (slots) próprias, especificações de facetas (facets) pré-definidas ou arbitrárias para aberturas, que incluem valores permitidos (allowed values), restrições de cardinalidade, abertura inversas (inverse slots), meta-classes e hierarquia de meta-classes.

Em adição a interface com boa usabilidade, pode-se destacar ainda duas qualidades: escalabilidade e extensibilidade. Desenvolvedores têm empregado o Protégé com sucesso na construção de ontologias que consistem de 150.000 quadros (frames). Para que se suporte bases de conhecimento com centenas de milhares de quadros são necessários, entre outros, dois componentes: um banco de dados para armazenar e consultar informações e um mecanismo de cachê para ler os quadros na memória conforme a necessidade, já que seu número total excede o limite da memória [OCE].

Uma das maiores vantagens da arquitetura do Protégé é que o sistema é construído em uma forma aberta e modular. Esta arquitetura baseada em componentes permite aos construtores do sistema a adição de novas funcionalidades com a criação de *plugins*. Desta forma é possível a conversão da ontologia em outras linguagens, como o CLIPS por exemplo.

A maioria dos *plugins* pode ser classificada em uma destas três categorias: *backends* que permitem ao usuário a importação e a exportação das ontologias em vários formatos diferentes, como esquemas RDF, arquivos XML com um DTD, arquivos de esquemas XML, OIL e DAML+OIL; *slot widgets* que são usados para mostrar e editar valores das aberturas ou suas combinações em tarefas específicas de domínios específicos – alguns incluem interfaces com imagens, vídeo e áudio e outros permitem que se crie os elementos da base de conhecimento através da manipulação de uma diagrama colorido - e *tab plugins* que são aplicações baseadas em

conhecimento estritamente conectadas com as bases de conhecimento Protégé.

Estes últimos são o tipo mais popular e incluem *plugins* de visualização, combinação e administração de versões. O OntoViz e o Jambalaya, são exemplos de visualizadores gráficos da base de conhecimento. O Jambalaya permite navegação interativa, focar-se em um determinado elemento da estrutura e diferentes apresentações dos nós em um grafo para demonstrar as conexões entre os agrupamentos de informação. O Flora e o Jess provêm acesso a vários motores de raciocínio (reasoning engines). O *plugin* PROMPT provê um ambiente para a administração de múltiplas ontologias. Seus componentes incluem ferramentas para combinação de ontologias, que auxiliam o usuário a encontrar similaridades entre as fontes das ontologias para combiná-las, ferramentas para controle de versões, que automaticamente encontram diferenças estruturais entre versões de uma mesma ontologia e ferramentas para extração semântica de partes de uma ontologia e rearranjo dos quadros entre diferentes ontologias ligadas [OCE].

3.2.2 SymOntoX

O SymOntoX (http://www.symontox.org) (Symbolic Ontology Manager XML) é um protótipo de software para a gerência de domínios de ontologias. Ele foi desenvolvido pelo LEKS (Laboratory for Entreprise Knowledge and Systems) na IASI-CNR. Há também um conjunto de API Java para interoperabilidade e integração com outros sistemas.

No SymOntoX conceitos do domínio e relações são modelados de acordo com OPAL (Object, Process and Actor modelling Language) uma metodologia para representação de ontologias. Um validador de consistência verifica se a ontologia está de acordo com os axiomas propostos no OPAL. SymOntoX é o sucessor do SymOntos. De acordo com OPAL, conceitos são organizados por significado destes três idéias de modelagem primárias: Ator, Processo e Objeto. Mais precisamente, há:
-Ator (actor): qualquer entidade relevante do domínio que está apta para ativar ou executar um processo, como um vírus ou um hormônio;
-Objeto (object): uma entidade passiva no qual o processo opera, como uma enzima.
-Processo (process): uma atividade com a intenção de satisfazer um objetivo (goal), como glicólise;
Além destas idéias preliminares de modelagem, a especificação OPAL propôs as seguintes modelagens complementares:
-Componente de Informação (Information Component): um grupo de informações pertencentes à estrutura da informação de um ator ou um objeto, como nome para o vírus e fórmula para a enzima;
-Elemento de Informação (Information Element): elemento de informação atômica que é parte de um Componente de Informação, como as moléculas da fórmula da enzima;
-Ação (Action): atividade que representa um componente do processo, que pode ser futuramente decomposto;
-Ação Elementar (Elementary Action): Atividade que representa um componente do processo que não pode mais ser decomposto;
-Objetivo (Goal): um estado desejado das coisas que um ator procura alcançar, como moléculas de ATP.
-Estado (State): um padrão característico de valores que instanciam as variáveis que uma entidade pode assumir, como proteína_não_solúvel;
-Regra (rule): é uma expressão que tem por objetivo restringir os possíveis valores das instâncias, uma regra de confinamento (constraint rule) ou uma de que se deriva novas informações, uma regra de produção (production rule), como em “quebra do ADP produz energia”.

Essas idéias são necessárias para se definir conceitos unitários. De acordo com OPAL, conceitos são ligados entre si por meio das relações ontologias, que podem ser: especialização, decomposição, predicação, similaridade e relacionado-com.

SymOntoX foi concebido para ser um serviço disponível na Internet, acessível através de um navegador comum. Ele é baseado em XML, todas as informações são guardadas em um banco de dados XML, e usa Java, para garantir uma boa flexibilidade, interoperabilidade e independência de plataforma. O sistema tem três distinções quanto aos usuários, que podem ser registrados como “user”, com direito apenas de leitura; “superuser’ com direito de leitura e de propor novos conceitos e “ontology master”, que é o responsável pela aceitação ou não dos novos conceitos propostos. O sistema pode ainda ser usado de diferentes formas: como um glossário, onde apenas o nome e a descrição em linguagem natural dos conceitos é

mostrado; ou como uma enciclopédia, onde também são mostradas as relações de hierarquia e a similaridade entre os conceitos; como um sistema de ontologias, onde todos os relacionamentos são mostrados e como uma base de conhecimento, onde incluem-se todas as instâncias dos conceitos.

SymOntoX suporta um formulário gráfico para interface com o usuário, na edição e visualização da ontologia. Existe também uma funcionalidade de diagramação para navegação no conteúdo da ontologia [OCE].

3.2.3 OILED

OILED (http://oiled.man.ac.uk) é um simples editor de ontologias gráfico desenvolvido pela Universidade de Manchester que permite ao usuário construir ontologias usando a linguagem DAML+OIL. Inicialmente ele foi projetado como uma demonstração das possibilidades e benefícios do uso como mecanismo para classificar ontologias, mas ganhou algum sucesso como editor de ontologias.

O modelo do conhecimento do OILED é baseado no do DAML+OIL, mas é estendido pelo uso de representações baseadas em quadros para a modelagem. Então, OILED oferece um ambiente familiar baseado no paradigma de quadros para modelagem, enquanto ainda suporta a rica expressividade do DAML+OIL quando requerido. Isto foi uma das causas para o sucesso já que esta abordagem é vital para o exercício de modelagem, especialmente para biólogos [OIE]. Ferramentas como o Protégé (que influenciou bastante o OILED) e OntoEdit também usam o paradigma baseado em quadros. Classes são definidas em termos de suas superclasses e restrições de propriedades, como axiomas adicionais capturando adicionalmente relacionamentos como a disjunção. O modelo de conhecimento expressivo permite o uso de composições de descrição complexas como regras de filtragem. Isto está em contraste com muitos editores baseados em quadros, onde estes quadros anônimos devem ser nomeados antes que sejam usados como modelos.

A tarefa principal que é o alvo do OILED é a edição de ontologias ou esquemas, em oposição à aquisição do conhecimento ou a construção de grandes bases de conhecimento. Então funcionalidades são providas para permitir a definição de indivíduos, isto é a primeira intenção para o desenvolvimento de nominais, que são usados no construtor um-de (one-of) do DAML+OIL [OCE].

Um aspecto chave do comportamento do OILED é o uso do mecanismo de raciocínio FaCT para classificar ontologias e verificar consistência através de uma tradução de DAML+OIL para a linguagem de descrição lógica SHIQ. Isto permite ao usuário descrever suas classes de ontologias e ver o mecanismo de raciocínio determinar o lugar apropriado na hierarquia para a definição. Algumas vezes uma definição de conceito equivocada pode ser determinada como insatisfazível.

O esquema RDF do DAML+OIL é usado para carregar e guardar ontologias. Em adição, a ferramenta lê e escreve hierarquia de conceitos em RDF puro e converte definições de ontologias em HTML para navegação, e em SHIQ para classificação posterior pelo FaCT através de sua interface CORBA. A hierarquia de conceitos pode também ser convertida em um formato legível pela ferramenta Dotty da AT&T [OCE].

O OILED é implementado em Java e está disponível gratuitamente (um registro no entanto é necessário) em: http://oiled.man.ac.uk.

3.2.4 OntoSaurus

Ontossauros foi desenvolvido pelo Information Sciences Institute (ISI) na USC (University of South California). Ele consiste de dois módulos: um servidor de ontologias, que usa LOOM como sistema de representação de conhecimento e um servidor de navegação de ontologias que dinamicamente cria páginas HTML, incluindo imagens e documentação textual, que demonstram a hierarquia da ontologia. A ontologia pode ser editada em formulários HTML, e pode ser convertida de LOOM para Ontolingua, KIF, KRSS e C++. OntoSaurus está disponível em http://www.isi.edu/isd/ontosaurus.html. Consultas por exemplo, e diversas visualizações são possíveis no software.

A tela do navegador é dividida em cinco quadros: barra de ferramentas, janela de referência, marcações, controle de marcações e janela de conteúdo. Na barra de ferramentas, pode-se selecionar um entre os oito exemplos pré-escritos na página, onde é possível consultar os módulos para cada ontologia. Há os botões ‘show’, para carregar o módulo, ‘view’ para ver as descrições do módulo, ‘hold window’ para mover a janela de conteúdo para a janela de referência, ‘options’ para configurar diversas opções do OntoSaurus. Há ainda uma caixa de busca, com opções boleanas. Na janela de referência, pode-se consultar as informações sobre os conceitos da ontologia atual e compara-los com os conceitos da janela de conteúdo. A parte de marcações exibe os itens previamente marcados pelo quadro de controle de marcações, onde é possível manuseá-los.

4.0 Conclusão

Até aqui foi dada uma visão geral do panorama atual da bioinformática, na área de ontologias para biologia.

Como visto no capítulo 0.1, informações são cruciais para os biólogos realizarem seus trabalhos. Devido a isto, o estímulo para o uso de ontologias para o desenvolvimento da biologia será cada vez mais intenso. Ontologias como o GO, TAMBIS etc crescem a cada dia, e são citadas em cada vez mais trabalhos. No entanto, estas ontologias não fazem uso de mecanismos de encadeamento das linguagens de descrição de ontologias, para realizarem novas descobertas. As poucas que utilizam, apenas fazem classificação automática dentro da hierarquia.

Na próxima fase do trabalho, será proposta uma pequena bio-ontologia, na área de bioquímica, descrita em PowerLoom, que faz uso ativo das características da linguagem e de seus mecanismos de inferência, demonstrando novos exemplos de descobertas e consultas através desta abordagem.

4.1 Continuação

Este trabalho será continuado com uma proposta de uma ontologia para a área de bioquímica. O capítulo inicialmente abordará princípios de projeto de construção de ontologias, e em especial, bio-ontologias. Após esta parte será demonstrado o porquê da escolha de PowerLoom como linguagem de representação, através de comparações com outras linguagens. Haverá também um pequeno tutorial de PowerLoom. O capítulo segue com mais duas seções, onde serão apresentados os problemas de representação de conhecimento encontrados, e exemplos de novas funcionalidades e consultas da bio-ontologia proposta, com o uso do PowerLoom. O estudo sobre essa nova ontologia proposta está sendo realizado com ajuda de uma especialista no domínio, Beth Koo.

4.0 Construindo uma bio-ontologia
4.1 Como construir uma bio-ontologia?
4.2 Por que PowerLoom?
4.3 Breve tutorial de PowerLoom
4.4 Problemas de Representação de Conhecimento encontrados
4.5 Exemplos de novas funcionalidades/consultas usando PowerLoom
4.6 Conclusões

5.0 Bibliografia

[BOM] BioMiner – modeling, analyzing and visualizing biochemical pathways and networks.
M. Sirava, T. Shafer, M Eiglsperger, M. Kaufmann, O. Kohlbacher, E. Bornberg-Bauer e H. P. Lenhof.
http://www.brc.dcs.gla.ac.uk/~yde/Papers/sirava2002.pdf

[SSK] Ontologies for molecular biology and bioinformatics.
Steffen Schulze-Kremer.
http://www.ep.liu.se/ea/cis/2001/021/cis01021.ps

[G&G] Ontologies and Knowledge Bases. Towards a Terminological Clarification.
Nicola Guarino e Pierdaniele Giaretta.
http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/KBKS95.pdf

[GRU] Toward Principles for the Design of Ontologies Used for Knowledge Sharing.
Thomas R. Gruber.
http://www-ksl.stanford.edu/knowledge-sharing/papers/onto-design.ps

[MAS] An Ontology of Meta-Level Categories
Nicola Guarino, Massimiliano Carrara, Pierdaniele Giaretta
www.ladseb.pd.cnr.it/infor/Ontology/Papers/KR94.pdf

[SEM] The Semantic Metadatabase (SEMEDA): Ontology based integration of federated molecular biological data sources.
Jacob Köhler and Steffen Schulze-Kremer.
http://www.bioinfo.de/isb/2002/02/0021/

[SGB] Ontology-based Knowledge Representation for Bioinformatics.
Robert Stevens, Carole A. Goble, Sean Bechhofer.
http://www.cs.man.ac.uk/~stevensr/onto/

[ELP] El Pais (Espanha) - La catarata de nuevos genes pone en evidencia la anarquia en sus nombre (11/07/2001).
Helen Pearson.
http://www.elpais.es/suplementos/futuro/20010711/24gentes.html

[MRI] Functions of the gene products of Escherichia coli - *Microbiological Reviews*, 57:862-952, 1993.
M. Riley.

[KEG] Kioto Encyclopaedia of Genes and Genomes.
http://www.genome.ad.jp/kegg/

[TAM] TAMBIS – Escola de Ciências Biológicas, Information Management Group, Universidade de Manchester, Inglaterra.
http://imgproj.cs.man.ac.uk/tambis/index.html

[EVO] Ontology Evolution within Ontology Editors.
Stojanovic e B. Motik
http://km.aifb.uni-karlsruhe.de/eon2002/EON2002_Stojanovic.pdf

[OBT] Ontology Building: A Survey of Editing Tools.
Michael Denny.
http://www.xml.com/pub/a/2002/11/06/ontologies.html

[OCE] OntoWeb Ontology-based information exchange for knowledge management and electronic commerce.
http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D13_v1-0.zip

[MOT] Ontology Overview. Motorola Labs.
Myriam Ribière e Patricia Charlton.
http://www.fipa.org/docs/input/f-in-00045/f-in-00045.pdf

[OIE] DAML+OIL is not Enough.
Sean Bechhofer, Carole Goble, Ian Horrocks.
http://www.semanticweb.org/SWWS/program/full/paper40.pdf

[RDM] A Roadmap to Ontology Specification Languages.
Oscar Corcho e Asunción Gómez-Pérez.
http://delicias.dia.fi.upm.es/articulos/ocorcho/ekaw2000-corcho.pdf

[PWL] PowerLoom Knowledge Representation System.
Hans Chalupsky e Thomas A. Russ.
http://www.isi.edu/isd/LOOM/PowerLoom/

[SRE] Some Requirements and Experiences in Engineering Terminological Ontologies over the WWW.
Aldo Gangemi, Domenico M. Pisanelli, Gerardo Steve.
http://saussure.irmkant.rm.cnr.it/onto/publ/kaw98/kaw98.pdf

[PTG] Protégé-2000. The Protégé Ontology Editor and Knowledge Acquisition System.
Stanford Medical Informatics.
http://protege.stanford.edu/

[GLI] Metabolic Pathways of Biochemistry – Glycolysis
http://www.gwu.edu/~mpb/glycolysis.htm

[SBS] Step by Step Glycolisys
Jon Maber
http://www.jonmaber.demon.co.uk/glysteps/

[MOD] Methodologies for Ontology Development
Dean Jones, Trevor Bench-Capon, Pepijn Visser
http://cweb.inria.fr/Resources/ONTOLOGIES/methodo-for-onto-dev.pdf