Universidade Federal de Pernambuco

Graduação em Engenharia da Computação

Centro de Informática

UMA FERRAMENTA PARA SUMARIZAÇÃO DE ONTOLOGIAS

TRABALHO DE GRADUAÇÃO

Aluno: Victor Bezerra Alencar (vba@cin.ufpe.br)
Orientadora: Ana Carolina Salgado (acs@cin.ufpe.br)
Co-orientador: Carlos Eduardo Santos Pires (cesp@cin.ufpe.br)

Agradecimentos

Dedico este trabalho e agradeço sinceramente aos meus pais, que desde sempre me apoiaram e deram todo suporte quando precisei, fosse na vida acâdemica ou fora dela.

Agradeço a paciência de meu irmão que por tantas vezes se deixou alugar por minhas conversas sobre a graduação ou mesmo sobre este trabalho. Várias vezes foram úteis seus comentários leigos, porém foi graças à ingenuidade desses comentários que outras luzes surgiram e iluminaram melhor os problemas que apareciam.

Outrossim devo agradecer a Carlos Eduardo Pires, que durante todo o desenvolvimento deste trabalho me orientou e sempre esteve à disposição para me ajudar. Valeu Carlos!

Um agradecimento especial faço aos amigos e colegas de graduação, com quem sempre pude dividir os momentos de alegria e tristeza na árdua tarefa de concluir o curso.

“Transportai um punhado de terra todos os dias e farás uma montanha."
Confúcio

Victor Bezerra Alencar.

Resumo

Um esquema é um diagrama que modela um conjunto de conceitos (normalmente relacionados) associados a determinado universo de discurso. Em geral, esquemas que apresentam uma quantidade excessiva de conceitos são considerados esquemas complexos e apenas um número reduzido de usuários que, certamente dedicaram uma quantidade de tempo razoável para compreendê-lo, possui domínio completo sobre um esquema complexo. Este trabalho consiste em desenvolver uma ferramenta que possibilite a sumarização de ontologias. Um resumo ontológico é uma versão reduzida de uma ontologia, contendo os conceitos mais relevantes. Características como centralidade (importância) e freqüência dos conceitos de uma ontologia são exploradas para determinar os conceitos relevantes e, em seguida, gerar seu resumo ontológico.

Palavras-chave: Ontologias, Sumarização, Centralidade.

Sumário

1 Introdução

Um esquema é um diagrama que modela um conjunto de conceitos (normalmente relacionados) associados a determinado universo de discurso. Diferentes formas de representar esquemas foram surgindo ao longo dos anos com o objetivo de melhor representar e abordar novos problemas envolvendo modelagem de bancos de dados e compartilhamento de dados. Assim surgiu os conhecidos modelos ER (Entidade-Relacionamento), OR (Objeto Relacional) e mais recentemente, a fim de melhor representar os dados no contexto da web semântica e dos sistemas de compartilhamento de dados, as ontologias. Estas últimas surgiram como um mecanismo mais natural para representação de esquemas em contextos semânticos, onde conceitos vão além de meras palavras-chave para representar relacionamentos mais genéricos.

Segundo [Gruber, 1993] uma ontologia seria então, como uma especificação de uma conceituação. A Ontologia na web semântica estabelece uma ligação terminológica entre membros de uma comunidade, podendo ser estes membros, agentes humanos ou máquinas.

No contexto de sistemas de compartilhamento de dados, como sistemas P2P (*Peer-to-Peer*), ontologias podem ser utilizadas para representar o conteúdo representado por *peers* enriquecendo-o conceitualmente [Pires al al. 2006].

Em geral, esquemas que apresentam uma quantidade excessiva de conceitos são considerados esquemas complexos [Yu and Jagadish 2006]. Apenas um número reduzido de usuários que, certamente dedicaram uma quantidade de tempo razoável para compreendê-lo, possui domínio completo sobre um esquema complexo.

Um esquema resumido é uma representação sucinta e abstrata de um esquema complexo [Yu and Jagadish 2006]. Dessa forma, sumarização de esquemas pode ser definida como o processo de transformar um esquema complexo em um esquema resumido, conservando da maneira mais fiel possível as características do esquema original.

1.1 Motivação

Em alguns casos, ontologias que tratam de domínios complexos e que passaram por muita evolução ao longo do tempo podem ganhar proporções gigantescas. Nestes casos, a tarefa de análise manual dessas ontologias se torna muito dispendiosa, surgindo então a necessidade de resumir automaticamente tais ontologias a fim de agilizar a análise. Outro exemplo acontece durante a formulação de uma consulta a um esquema, pois o usuário pode ter interesse em acessar apenas uma parte de um esquema (complexo).

Com o aumento do uso de ontologias nas várias áreas da ciência e da indústria, o número de ontologias aumentou consideravelmente. Então, em muitas situações, não é necessário desenvolver uma nova ontologia, apenas reusar uma já existente ou adaptá-la como necessário. Antes de reusar uma ontologia é necessário primeiro compreendê-la. Tal tarefa é comumente

realizada através da análise dos elementos (i.e conceitos e propriedades) da ontologia [Pires, 2008]. Desta forma, resumos do esquema podem ser úteis para facilitar o trabalho deste usuário.

1.2 Objetivos

O processo de resumir ontologias deve se preocupar em preservar no sumário o máximo possível das principais características da ontologia original. Nesse sentido, um processo automático tem de se valer de técnicas que identifiquem os principais conceitos e relacionamentos de uma ontologia a fim de preservá-los ou transformá-los da melhor maneira possível. Sendo assim, uma vez iniciado o funcionamento da ferramenta, esta não deve necessitar da intervenção do usuário, pois o processo de encontrar um resumo deve ser automático.

Este trabalho consiste em especificar e desenvolver uma ferramenta que possibilite a sumarização de ontologias. Características como centralidade (importância) [Gaertler and Wagner 2004] e freqüência dos conceitos de uma ontologia são exploradas para gerar seu resumo ontológico. A ferramenta deve ser parametrizada para permitir a geração de diferentes tipos de resumos de acordo com as necessidades do usuário.

1.3 Estrutura do documento

Além deste capítulo, esta monografia encontra-se organizada em mais 4 (quatro capítulos):

Capítulo 2 – Neste capítulo será descrito o que são ontologias, o contexto em que surgiram, para que servem e uma breve descrição dos padrões de implementação de ontologias existentes hoje.

Capítulo 3 – Neste capítulo serão abordados os tópicos acerca do processo de sumarização: suas etapas desde a ontologia de entrada até a ontologia resumida, bem como critérios de importância de conceitos, critérios de parada e métricas para avaliar a qualidade do resumo

Capítulo 4 – Neste capítulo detalhes de funcionamento e implementação da ferramenta serão apresentados. Alguns exemplos de funcionamento serão demonstrados e avaliados comparativamente.

Capítulo 5 – O último capítulo apresenta as conclusões obtidas pelo trabalho, as dificuldades que surgiram no caminho e possíveis trabalhos futuros.

2 Ontologias

Neste capítulo será apresentado como a idéia de ontologias passou a ser utilizada para ajudar na modelagem de domínios, bem como qual o formalismo necessário para um bom entendimento das aplicações de ontologias e finalmente um padrão muito comum para ontologias será apresentado.

2.1 Definição

Ontologia, como um braço da filosofia, é a ciência que estuda “o que é”, dos tipos e estruturas dos objetos, propriedades, eventos, processos e relações em cada área presente na realidade [Smith, 2008]. Tal conceito de ontologia passou a ser utilizado, com o passar dos anos, em outras áreas além da filosofia, chegando a ser adotada pela engenharia do conhecimento como forma de representação semântica de informação ou conhecimento dentro de um domínio.

Em informática, uma ontologia é um modelo para descrever um mundo que consiste de um conjunto de conceitos, propriedades e relacionamentos entre conceitos. Tal modelo se mostrou extremamente útil como uma forma alternativa de representação semântica de dados na web [Perez & Corcho, 2002]. Nesse contexto, uma ontologia pode ser visa como um grafo, onde cada nó representa um conceito e cada aresta ligando dois nós representa uma propriedade ou relacionamento entre esses dois conceitos [Razmerita & Maedche 2003].

Mais formalmente, segundo [Smith, 2008], ontologias descrevem *individuals* (instâncias), *classes* (conceitos), *attributes* (atributos) e *relations* (relações):

Indivíduos: Instâncias de classes ou objetos.

Classes: Conjuntos, coleções, conceitos, tipos de objetos.

Attributos: Aspectos, propriedades, características ou parâmetros que objetos (ou classes) podem ter.

Relações: Forma como classes e individuals são relacionados uns aos outros.

Restrições: Descrições formais do que deve ser verdade em declarações que podem ser aceitas como entrada para atributos.

Regras: Afirmações na forma de sentenças se-então (antecedente-consequente) que descrevem inferências lógicas.

Axiomas: Declarações (incluindo rules) lógicas que juntas descrevem a teoria geral que a ontologia engloba em seu domínio de aplicação.

Eventos: Representam a mudança de atributos ou relacionamentos.

Ontologias são comumente codificadas por linguagens de ontologias, como por exemplo: Cyc (baseada em lógica de predicados de primeira ordem), Gellish, KIF, RIF, OWL [Ontology Languages, Wikipedia]. O presente trabalho escolheu o padrão OWL para codificação de ontologias por ser aberta e produto de um grande consórcio como W3C.

2.2 Formalização

Em [Pires, 2008] é apresentado um formalismo para ontologias no qual um grafo é utilizado para representação das mesmas. Uma ontologia O é modelada como um grafo direcionado e conectado, cujos nós são conceitos e cada aresta é rotulada com um nome de propriedade que relaciona dois conceitos. Seja O = {C, R}, onde C = $\{c_1,...,c_n\}$ é o conjunto finito dos conceitos; e R = $\{r_1,...,r_n\}$ o conjunto finito dos relacionamentos entre os conceitos.

Assim como em [Pires, 2008], o presente trabalho considera que uma ontologia pode ser obtida como resultado da união de várias ontologias $O_1,\ldots,O_n$. Um conceito $c_n \in C$ pode representar um ou mais conceitos contidos em $O_1,\ldots,O_n$. Formalmente, $\forall\ c_n \in O_1 \cup O_2 \cup ... \cup O_n \Rightarrow \exists\ c_i \in O_1$ or $c_j \in O_2$ ou … ou $c_k \in O_n$. No entanto, diferentemente de [Pires, 2008], nem todo c_n precisa estar contido em $O_1,\ldots,O_n$, ou seja, ser oriundo da união de ontologias prévias, nesses casos, os conceitos podem ser frutos de uma modificação feita diretamente pelo usuário, como uma forma de provocar uma evolução manual da ontologia.

Quando um conceito é procedente de uma das ontologias que participaram do processo de união, este conceito passa a possuir uma freqüência associada, que se refere à quantidade de vezes na qual o conceito aparece nas ontologias unidas.

Um relacionamento $r_k \in R$ é composto de um rótulo que representa uma relação direta entre dois conceitos adjacentes c_i e $c_j \in C$; isto é, r_k = (rótulo(r_k), $c_i \times c_j$). Uma aresta rotulada e direcionada é definida de c_i para c_j se c_i é subconceito direto de c_j. Similarmente, se c_i é uma propriedade de conceito e c_j seu conceito alvo então uma aresta direcionada e rotulada é adicionada de c_i para c_j. O número de conceitos em O indica sua ordem: |C|. Outrossim é assumido que o grafo não conterá arestas com mesma origem e destino, isto é, autorelacionamentos, nem arestas múltiplas de mesma direção entre dois conceitos.

Finalmente, um sumário de ontologia OS é aqui definido como um subgrafo de O, ou seja, OS $\subset$ O. Formalmente, OS = (CS, RS), onde CS $\subset$ C e RS $\subset$ R.

2.3 Padrão OWL

A OWL *(Ontology Web Language*) é uma linguagem para definir e instanciar ontologias na Web [W3C, 2004]. Uma ontologia OWL pode incluir descrições de classes e suas respectivas propriedades e seus relacionamentos. OWL foi projetada para o uso por aplicações que precisam processar o conteúdo da informação ao invés de apenas apresentá-la aos humanos. Ela facilita mais a possibilidade de interpretação por máquinas do conteúdo da Web do que XML (*eXtensible Markup Language*) [W3C XML], RDF (*Resource Description Framework*) [W3C RDF] e RDFS *(RDF Schema*), por fornecer vocabulário adicional com uma semântica formal. A OWL foi baseada nas linguagens OIL (*Ontology Interchange Language*) [OIL] e DAML (*DARPA Agent Markup Language*)+OIL [DAML+OIL], e é hoje uma recomendação da W3C (*World Wide Web Consortium*), isto é, um padrão.

OWL é vista como uma tecnologia importante para a futura implementação da Web semântica. Ela vem ocupando um papel importante em

um número cada vez maior de aplicações. Além disso, vem sendo foco de pesquisa para ferramentas, técnicas de inferências, fundamentos formais e extensões de linguagem.

OWL foi projetada para disponibilizar uma forma comum para o processamento de conteúdo semântico da informação na Web. Ela foi desenvolvida para aumentar a facilidade de expressar semântica (significado) disponível em XML, RDF e RDFS. Conseqüentemente, pode ser considerada uma evolução destas linguagens em termos de sua habilidade de representar conteúdo semântico da Web interpretável por máquinas. Já que a OWL é baseada em XML, a informação pode ser facilmente trocada entre diferentes tipos de computadores usando diferentes sistemas operacionais e linguagens de programação. Por ter sido projetada para ser lida por aplicações computacionais, algumas vezes considera-se que a linguagem não possa ser facilmente lida por humanos, razão pela qual existem ferramentas que facilitam a representação de um arquivo OWL visualmente em grafos ou outros elementos gráficos. OWL vem sendo usada para criar padrões que forneçam um *framework* para gerenciamento de ativos, integração empresarial e compartilhamento de dados na Web.

Uma versão estendida da OWL (algumas vezes chamada OWL 1.1, mas sem ser oficial) foi proposta, incluindo maior expressividade, um modelo de dados mais simples e serialização, assim como uma coleção de sub-linguagens, cada uma com conhecidas propriedades computacionais.

2.4 História da OWL

Um grande número de esforços de pesquisa durante o final da década de 1990 exploraram como a idéia de representação de conhecimento da inteligência artificial poderia ser utilizada na Web. Estes esforços incluíram linguagens baseadas em HTML (chamada SHOE), XML (chamada XOL e, mais tarde, OIL), e várias linguagens baseadas em frames e abordagens de aquisição de conhecimento.

A linguagem OWL é uma revisão baseada em pesquisa [W3C, 2002] da linguagem DAML+OIL. DAML+OIL foi desenvolvida por um grupo chamado "US/UK ad hoc Joint Working Group on Agent Markup Languages", fundado em conjunto pela *US Defense Advanced Research Projects Agency* (DARPA) dentro do DAML program e o projeto IST da União Européia.

A W3C criou o "*Web Ontology Working Group*", que começou a trabalhar em 1 de novembro de 2001, presidido por James Hendler e Guus Shreiber. O primeiro rascunho da sintaxe abstrata, da referência e da sinopse foram publicado em julho de 2002. Os documentos da OWL tornaram-se uma recomendação formal da W3C em 2004, e o grupo de trabalho foi dispensado ainda em 2004 [WebOnt, 2004].

Dentro do grupo de trabalho, o esforço para identificar os objetivos de projeto e os requisitos foi liderado por Jeff Heflin. Alguns requisitos foram contribuição de Deborah McGuinness baseada em mais de uma década de trabalho na construção de sistemas baseados em ontologias. Outros requisitos foram identificados como parte do trabalho do Ph.D. de Heflin na construção de um protótipo de um sistema da Web semântica. Os outros membros do grupo de trabalho contribuíram com mais de 25 casos de uso, que foram depois resumidos na definição de um conjunto de casos de uso [W3C UCs, 2004].

2.5 Sumários de ontologias

Um sumário de ontologia pode ser entendido como um subconjunto (transformado ou não) de uma ontologia, uma subontologia. Tal subconjunto, quando transformado, sofre alterações que convertem um ou mais conceitos em um único conceito, provocando assim a diminuição do tamanho da ontologia. Um resumo decorrente de um subconjunto transformado de uma ontologia necessita de regras semânticas que auxiliem o processo de conversão dos conceitos ou da intervenção de um analista que entenda o domínio ao qual a ontologia pertença.

Como nem sempre regras semânticas estão disponíveis e a intervenção de um analista nem sempre é possível, os sumários decorrentes de um subconjunto não transformado da ontologia é mais propício para gerações automáticas de sumário ontológico. Por essa razão, este trabalho é focado nos sumários como um subconjunto puro de uma ontologia.

O subconjunto interessante para servir de resumo de um ontologia deve ser capaz de preservar o conteúdo mais importante da mesma, bem como permitir um entendimento mais rápido e direto do domínio que a ontologia trata.

2.6 Trabalhos relacionados

Na literatura, é possível encontrar alguns trabalhos que propõem a geração de esquemas resumidos. As técnicas utilizadas em cada trabalho variam de acordo com o tipo de esquema considerado: esquema conceitual (entidade-relacionamento) [Castano et al. 1998], um esquema relacional [Yu and Jagadish 2006] ou uma ontologia [Zhang et al. 2007].

Em [Castano et al. 1998] são propostas um conjunto de técnicas para análise e classificação de esquemas que podem ser utilizadas combinadas ou separadamente, embora tais técnicas permitam derivar propriedades importantes dos esquemas é necessária a intervenção de um analista.

Em [Yu and Jagadish 2006] é proposto uma forma de sumarização de esquemas totalmente automática levando em conta duas propriedades para o sumário, além do tamanho reduzido: conter elementos importantes do esquema e possuir grande cobertura das informações do esquema original.

Em [Zhang et al. 2007] é proposto um modelo de sumarização baseado nas sentenças de um grafo em RDF. Sentenças RDF são extraídas, avaliadas e comparadas de modo a encontrar aquela que melhor possa resumir a ontologia.

O presente trabalho junta algumas das técnicas existentes na literatura para desenvolver uma ferramenta capaz de resumir uma ontologia em OWL de forma automática, necessitando apenas de alguns parâmetros do usuário, que irão permitir maior controle sobre os requisitos do sumário. Em nenhum desses artigos citados a idéia de freqüência havia sido utilizada.

3 Processo de Sumarização

Neste capítulo serão detalhadas as etapas do processo de sumarização, quais devem ser as características do resumo ontológico obtido e de que maneira o processo de sumarização pode ser flexibilizado de modo a atender melhor os requisitos do usuário.

O processo que será apresentado foi inspirado e adaptado do processo proposto por [Pires, 2008].

3.1 Resumo Ontológico

Como ilustrado na Figura 1, a abordagem de sumarização proposta consiste em, dada uma ontologia O, gerar uma versão reduzida de O, denominada sumário. Os conceitos relevantes de O (pintados de cinza na Figura 1) são inicialmente identificados e avaliados quanto ao grau de sua relevância. O_S corresponde à subontologia que concentra o maior número de nós relevantes. Uma vez que conceitos relevantes podem não ser adjacentes em O, conceitos não relevantes (conceitos em branco na Figura 1) podem ser introduzidos em um sumário. Tais conceitos "indesejados" são necessários para manter os relacionamentos originais entre os conceitos. Nesse sentido, O_S também corresponde à subontologia que contém o menor número de conceitos não-relevantes.

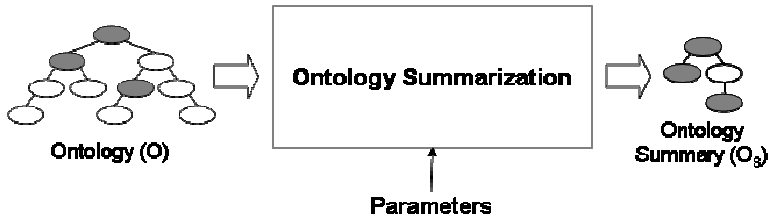

Figura 1. Processo geral de resumir uma ontologia.

De modo a atender diferentes requerimentos, o processo de sumarização deve aceitar diferentes tipos de parâmetros. Dependendo dos valores providos, diferentes sumários podem ser gerados para uma mesma ontologia O. Os parâmetros podem ser classificados de acordo com as seguintes categorias:

1) Medidas de relevância;
2) Tamanho de referência do sumário;
3) Variação do tamanho do sumário;
4) Limite de conceitos relevantes;
5) Peso da média harmônica paramétrica.

Os parâmetros serão melhor explicados ao longo do documento.

3.2 Medidas de relevância

A relevância de um conceito c_n de uma ontologia O é medida considerando os relacionamentos de c_n com outros conceitos da ontologia, sendo este critério denominado centralidade. Além disso, a relevância também pode levar em conta a quantidade de vezes que o conceito aparece nas

ontologias que serviram de base para a união que gerou a ontologia O, tal número indica a freqüência do conceito c_n.

3.2.1 Centralidade

Centralidade é um das mais importantes e utilizadas maneiras de se identificar nós relevantes em um grafo. A noção de relevância é subjetiva uma vez que depende do que é considerado importante para um nó. Desta forma, uma grande variedade de medidas específicas de centralidade têm sido propostas na literatura. Em [Freeman 1979], os autores categorizaram medidas de centralidade em 3 grupos (*degree*, *closeness* e *betweenness*) e apresentaram exemplos típicos para cada uma destas 3 categorias. Como resultado, essas 3 categorias têm dominado a forma como se calcula relevância, juntamente com medidas baseadas em *eigenvector* propostas por [Bonacich 1972].

A centralidade do tipo *degree* é baseada na idéia de que um nó n com um grande número de aresas tem acesso mais rápido e variado para outros nós em um grafo. Em um grafo não direcionado, o *degree* de centralidade de n é medido pelo número de arestas que n possui. Em um grafo direcionado, o *degree* de centralidade é medido pela quantidade de arestas de chegada ao nó n (*degree* de centralidade de chegada) ou pela quantidade de arestas de saída do nó n (*degree* de centralidade de saída). Centralidade baseada em *eigenvector* reconhece que a centralidade de um nó n não depende somente da quantidade de nós adjacentes, mas também do valor de sua centralidade. Desta forma, um nó n é mais central se n pertence a uma relação com nós que são, eles mesmos, centrais.

Os outros dois tipos de centralidade são baseados na noção de caminhos do grafo. Um caminho em um grafo é uma sequência de nós que são conectados por arestas que levam de um nó a outro. Um caminho geodésico é o menor caminho em termos do número de arestas entre um par específico de nós. Caminhos geodésicos não precisam ser únicos. A centralidade baseada em *closeness* de um nó n considera a distância geodésica entre n e todos os seus nós alcançáveis. A centralidade de *closeness* é menor para nós que são mais centrais, pois o caminho é mais curto que a média das distâncias para outros nós. Na prática, o inverso é usado como medida numérica de centralidade baseada em *betweenness*.

O presente trabalho adapta o conceito de *degree* de centralidade de modo a tornar este valor em um número entre zero e um. Desta forma, a centralidade é calculada como a razão entre a quantidade de arestas saindo ou chegando em n e o total de nós do grafo menos um. Seja G um grafo e n um nó de G:

$$\text{centralidade}(G, n) = \frac{\sum \langle \text{arestas que chegam ou saem de n} \rangle}{(\langle \text{total de nós de G} \rangle - 1)}$$

3.2.2 Freqüência

Ontology merging [Natalya and Musen 2000] é o processo no qual duas ontologias de origem são unificadas em uma ontologia alvo. Uma ontologia unificada pode ser descrita segundo arquivos de mapeamento ontológicos, que descrevem os mapeamentos de conceito, propriedades, etc. A Figura 2 representa um exemplo de arquivo de mapeamento ou de correspondência extraído de um PDMS [Pires, 2007]. O conceito *faculty* contido na ontologia unificada (CLO1) é mapeado para os conceitos *phd* e *professor* contidos nas ontologias de origem *LO1* e *LO2*, respectivamente.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CLUSTERONTOLOGY clo="CLO1">
<CLOCLASS>
<LABEL>faculty</LABEL>
<LOCLASS>
<LABEL>phd</LABEL>
<LO>L01</LO>
</LOCLASS>
<LOCLASS>
<LABEL>professor</LABEL>
<LO>L02</LO>
</LOCLASS>
</CLOCLASS>
...
</CLUSTERONTOLOGY>
```

Figura 2. Exemplo de mapeamento de classes. [Pires, 2008]

Até onde se sabe, ao se valer do uso de mapeamentos ontológicos oriundos de ontology merging [Pires, 2008] introduziram um novo critério para o cálculo da relevância, antes nunca usado. A principal explicação para isso é que as soluções existentes não consideram ontologias unificadas (isto é, resultantes de ontology merging) no processo de sumarização. Na abordagem proposta por [Pires, 2008], a freqüência de um conceito c_n pertencente à ontologia O é a razão entre a quantidade mapeamentos envolvendo c_n e a quantidade de ontologias de origem (isto é, que participaram do *ontology merging* que gerou O). Ambas as informações podem ser obtidas do arquivo de mapeamento ontológico dos conceitos. Assim, se dois conceitos c_i e c_j estão contidos na ontologia unificada $O = O_1 \cup ... \cup O_i \cup ... \cup O_n$, pode-se afirmar que c_i é mais freqüente que c_j se existem mais mapeamentos conceituais associados a c_i que c_j.

$$frequency(c_n) = \frac{|mappings(c_n)|}{|O|}$$ [Pires, 2008]

onde, $frequency(c_n) \in [0,1]$, $|O|$= quantidade de ontologias na união $O = O_1 \cup ... \cup O_i \cup ... \cup O_n$ e $|mappings(c_n)|$ = quantidade de mapeamentos de c_n.

Este trabalho faz uma pequena modificação no conceito de freqüência proposto por [Pires, 2008], sendo mantida a mesma propriedade (*frequency*(c_n) $\in$ [0,1]):

$$frequency(c_n) = \frac{|mappings(c_n)|}{\lceil M \rceil}$$

Onde, M = {|*mappings*(c_1)|, ..., |*mappings*(c_i)|, ..., |*mappings*(c_n)|} e o denominador representa o teto de M, isto é, a maior quantidade de mapeamentos para um único conceito de O. Em outras palavras, o denominador pode ser visto como a quantidade de mapeamentos que o conceito mais freqüente possui.

Como bem observado por [Pires, 2008], considerar a freqüência como medida para identificar conceitos relevantes em uma ontologia pode ser especialmente importante em ontologias que evoluem. Por exemplo, considerando um sistema integrado de dados onde o esquema é representado por uma ontologia global, cada vez que uma nova fonte de dados é adicionada ou removida, a ontologia global evolui. Supondo uma situação na qual uma um sumário atualizado da ontologia global esteja disponível a cada mudança, se a freqüência for considerada durante a identificação dos conceitos relevantes, talvez não seja necessário refazer o sumário a cada pequena mudança, pois em certos casos, ainda que as freqüências sejam atualizadas, não é suficiente para alterar a ordenação dos conceitos relevantes, evitando assim, ter de atualizar o resumo.

3.3 Descrição geral do processo

Inspirando-se na proposta de [Pires, 2008], dada uma ontologia O, o primeiro passo no processo de sumarização consiste em calcular a relevância de cada conceito considerando as medidas de centralidade e freqüência. Feito isso, os conceitos mais relevantes são determinados. Cada grupo de 2 ou mais conceitos relevantes que são adjacentes na ontologia em questão é então agrupado de modo a formar um superconceito (conceito que engloba outros). Desta forma, se apenas um superconceito é encontrado e este superconceito contém todos os outros conceitos relevantes, então o processo de sumarização termina e o sumário corresponde a subontologia contendo todos os nós relevantes. Caso contrário, se existe algum conceito relevante não adjacente, o processo de sumarização identifica todos os caminhos (de tamanhos definidos por dois parâmetros) entre os conceitos ou superconceitos relevantes. Cada caminho corresponde a uma subontologia em O e pode conter conceitos não-relevantes. Métricas usadas comumente na área de recuperação de informação são aplicadas para computar a qualidade de cada caminho. Finalmente, o melhor caminho, cujo é tamanho especificado por parâmetro é selecionado como o sumário da ontologia.

Resumidamente, os principais passos do processo de sumarização são:

i) Calcular as relevâncias dos conceitos da ontologia;
ii) Determinar os conceitos relevantes;
iii) Agrupar os conceitos relevantes adjacentes;
iv) Identificar os caminhos entre os conceitos relevantes;
v) Calcular a qualidade dos caminhos encontrados;
vi) Escolher o caminho com tamanho desejado de melhor qualidade .

A Figura 3 corresponde ao diagrama de atividades que descreve todos os passos envolvidos no processo de sumarização. A seguir cada passo será detalhado.

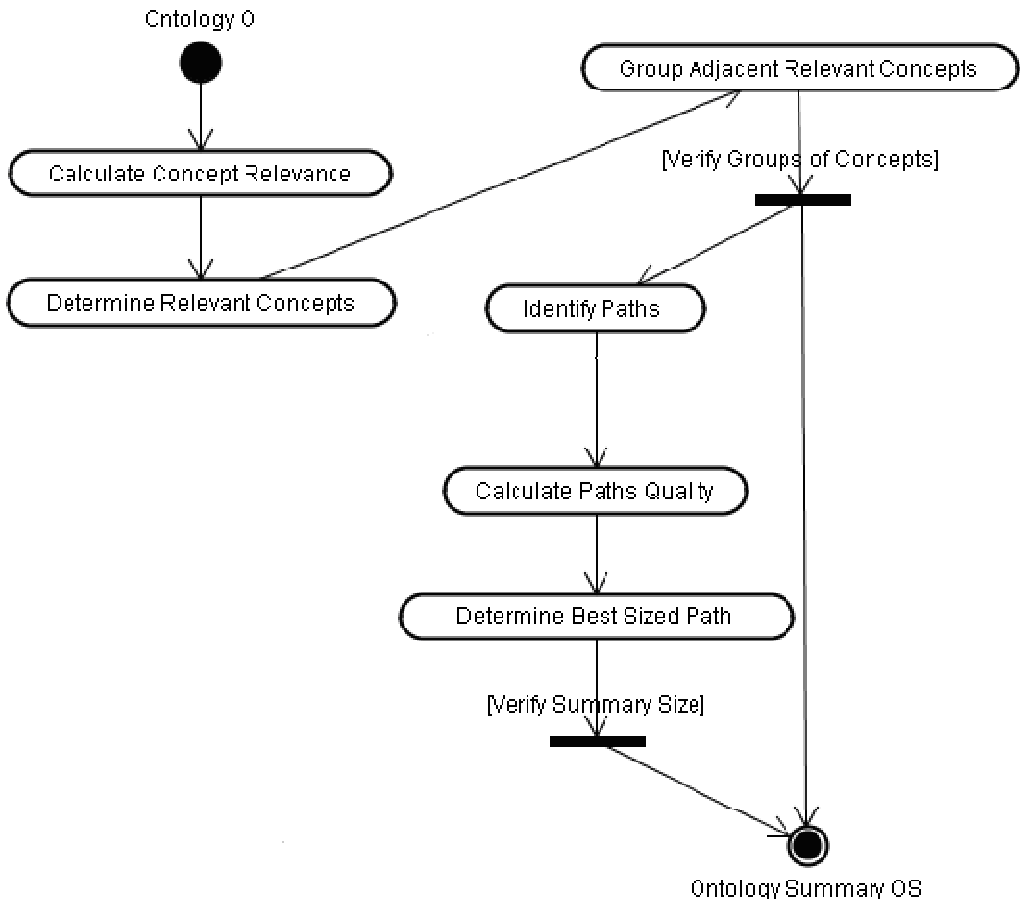

Figura 3. Passos envolvidos no processo de sumarização.

3.3.1 Passo 1: Calcular a relevância dos conceitos

Centralidade e freqüência são dois critérios que estão de algum modo correlacionados, mas que são inerentemente diferentes de modo que é possível encontrar resumos diferentes onde um possui mais centralidade e o outro mais freqüência. Um sumário ideal deve balancear as duas medidas. Não se pode afirmar que centralidade é mais importante que freqüência e vice-versa. Na verdade, a medida mais relevante depende do que o usuário necessita, entretanto, em alguns casos, ambas as medidas precisam ser consideradas juntas. Desta forma, como propõe [Pires, 2008], combinar centralidade e freqüência numa mesma fórmula envolvendo pesos parece ser a solução ideal para o cálculo da relevância. Dois parâmetros de peso devem então ser considerados, um para a centralidade e outro para a freqüência. A implementação considerou a seguinte fórmula para cálculo da relevância dos conceitos: *relevancia* (c_n) = μ *x cetralidade*(c_n) + β *x freqüência*(c_n) [Pires, 2008] onde: i) relevancia(c_n) $\in$ [0,1]; ii) $\mu + \beta = 1$.

3.3.2 Passo 2: Determinar os conceitos relevantes

Este passo consiste em identificar o *conjunto dos conceitos mais relevantes* (doravante denominado RC, onde RC $\subseteq$ C) de uma ontologia O. Vários critérios podem ser utilizados para a determinação de RC. O primeiro critério considera que RC têm um tamanho fixo, definido pelo usuário via parâmetro. Os conceitos são classificados em ordem decrescente de acordo com sua respectiva relevância. Depois, k conceitos mais relevantes são escolhidos, onde k é o parâmetro passado pelo usuário para limitar a quantidade de conceitos relevantes. O usuário pode usar k como o resultado de uma porcentagem da ontologia. Por exemplo, se uma ontologia possui 100 conceitos, o usuário pode limitar em 20 a quantidade de conceitos relevantes, sendo então 20% de conceitos relevantes e portanto k = 20. Então, Os 20 conceitos com com maior relevância farão parte de RC.

O segundo critério considera que RC pode assumir um tamanho varável. De modo a determinar os conceitos que serão incluídos em RC, primeiro a relevância média (RA) de todos os conceitos em C deve ser calculada:

$$RA(C) = \sum_{i=1}^{n} \frac{relevance(c_i)}{|C|}$$ [Pires, 2008]

Os conceitos relevantes serão aqueles cuja relevância é maior ou igual a relevância média dos conceitos. Formalmente,

$$\forall c_n \in C, \; if \; relevance(c_n) \geq RA(C) \Rightarrow c_n \in RC$$ [Pires, 2008]

Finalmente, o terceiro critério também assume que RC têm tamanho variável. Os conceitos a serem incluídos em RC serão aqueles cujas relevâncias for maior ou igual a uma relevância limiar (rt) informada como parâmetro. Formalmente,

$$\forall c_n \in C, \; if \; relevance(c_n) \geq rt \Rightarrow c_n \in RC$$ [Pires, 2008]

O presente trabalho implementa um modelo híbrido dos dois primeiros critérios, todos propostos por [Pires, 2008]. Desta forma, todos os conceitos cuja relevância for maior ou igual a RA serão candidatos a conceitos relevantes, e apenas aqueles k conceitos (parâmetro informado pelo usuário) dentro de RA com maior relevância serão considerados conceitos relevantes.

3.3.3 Passo 3: Agrupar conceitos relavantes adjacentes

Este passo consiste em formar grupos de conceitos, superconceitos, que são relevantes e adjacentes na ontologia a ser resumida. Um superconceito deve conter pelo menos 2 conceitos relevantes adjacentes. Tais agrupamentos são feitos para facilitar o cálculo dos caminhos entre nós relevantes (passo 4).

Considera-se somente conceitos adjacentes porque um dos principais objetivos é tornar o processo de sumarização automático e conceitos relevantes adjacentes são identificados de forma fácil e rápida. Para que não seja necessária intervenção humana, cada superconceito deve guardar também seus relacionamentos (isto é, suas arestas), para que o processo de desagrupamento de superconceitos possa ser feito automaticamente. Portanto, conclui-se que um superconceito é um subgrafo ou subontologia de O.

Ao se agrupar conceitos relevantes podem ocorrer uma de quatro situações:

i) nenhum par de conceitos relevantes adjacentes foi encontrado, portanto nenhum superconceito foi criado;

ii) vários grupos de conceitos relevantes adjacentes foram encontrados e alguns superconceitos criados, porém nenhum desses superconceitos está dentro do tamanho esperado do resumo (informado como parâmetro pelo usuário);

iii) apenas um superconceito (de tamanho menor ao esperado como resumo) foi criado, mais ainda restam conceitos relevantes não adjacentes. Para esses 3 casos, o processo de sumarização continua com o passo 4 (identificação dos caminhos).

Alguns superconceitos com tamanho igual ao tamanho esperado do resumo foram identificados. Nesse caso, aquele superconceito com tamanho igual ao esperado do resumo que possuir a maior relevância média será escolhido como resumo e o processo termina.

Apenas superconceitos com tamanho superior ao tamanho esperado do resumo foram identificados. Nesse caso, os superconceitos têm suas folhas de menor relevância eliminadas do superconceito até que estes tenham tamanho dentro do esperado como resumo e então aquele superconceito com maior relevância média é escolhido.

3.3.4 Passo 4: Identificar caminhos entre conceitos relevantes

Este passo consiste em detectar todos os caminhos entre conceitos (ou superconceitos) relevantes da ontologia. Cada caminho corresponde a uma subontologia de O e pode também conter conceitos não-relevantes que foram introduzidos para manter a coerência entre os conceitos do caminho. Processos para eliminar estes conceitos não-relevantes de um caminho entre dois conceitos relevantes podem ser aplicados, porém geralmente necessitam de intervenção do usuário ou da aplicação de regras sêmanticas para derivar um novo conceito que englobe um conceito relevante e um não relevante. A implementação deste trabalho não se vale do auxílio destas regras semânticas e, em se tratando de um modelo automático, não permite que o usuário faça parte do processo de identificação de caminhos. Por isso, ao final, o resumo pode ter conceitos não-relevantes.

3.3.5 Passo 5: Calcular a qualidade dos caminhos

Uma vez que múltiplos caminhos podem ser identificados, é necessário calcular a qualidade de cada caminho individualmente. As medidas de cobertura (*recall*) e precisão (*precision*) são comumentes utilizadas em recuperação de informação [Baeza and Ribeiro 1999]. Tais medidas são utilizadas para determinar o nível de cobertura e concisão de cada caminho OS_n, candidato a resumo. O *recall* indica que um caminho deve ser um pedaço de O que reflete a maior quantidade de conceitos relevantes possível, este é definido como a razão entre o número de conceitos relevantes da ontologia O e o número de conceitos relevantes existentes no caminho OS_n. O *precision* determina se um caminho é sucinto o bastante para facilitar a análise da ontologia inteira, este é definido como a razão entre a quantidade de conceitos

relevantes da ontologia O e a quantidade de conceitos do caminho OS_n. Formalmente,

$$\mathrm{Re}call = \frac{|OS_n \cup RC|}{|RC|} \qquad \mathrm{Pr}ecision = \frac{|OS_n \cup RC|}{|OS_n|} \quad \text{[Pires, 2008]}$$

Caminhos não podem ser comparados baseando-se tão somente em cobertura e precisão. Um caminho pode possuir grande cobertura e, em contrapartida, ter baixa precisão e vice-versa. Desta maneira, não é uma boa idéia como mostra [Pires, 2008], considerar estas medidas separadamente. Por essa razão, [Baeza and Ribeiro 1999] propõem utilizar uma média harmônica paramétrica dessas duas medidas, denominada *F-Measure*:

$$F - measure = \frac{\mathrm{Pr}ecision \times \mathrm{Re}call}{(1-\alpha) \times \mathrm{Pr}ecision + \alpha \times \mathrm{Re}call} \quad \text{[Pires, 2008]}$$

O parâmetro alfa tem o poder de ajustar a *F-measure* para considerar apenas a precisão ou apenas a cobertura, respectivamente, $\alpha = 0$ e $\alpha = 1$. Qualquer valor entre zero e um considerará quanto mais importante será uma medida sobre a outra no cálculo da qualidade dos caminhos. Muito comumente o valor $\alpha = 0.5$ é utilizado, sem privilégio de qualquer das medidas.

3.3.6 Passo 6: Determinar melhor caminho com tamanho esperado

Neste passo são filtrados os caminhos que possuem o tamanho esperado como sumário. O tamanho do esperado do sumário é calculado da seguinte forma: $T_R - \Delta \leq T \leq T_R + \Delta$, onde T_R é o tamanho de referência do sumário, Δ é uma variação em relação ao tamanho de referência e T é o tamanho esperado de um sumário, ou seja, um intervalo.

Uma vez filtrados os caminhos candidatos a sumário de acordo com o tamanho esperado, aquele com maior *F-measure* é escolhido. Caso haja algum empate de *F-measure* aquele caminho com maior relevância média deve ser escolhido.

3.4 Resumo do Processo

Resumidamente, o processo de sumarização de ontologias proposto pode ser descrito como a seguir. Dada uma ontologia O, o primeiro passo no processo de sumarização consiste em calcular a relevância de cada conceito considerando as medidas de centralidade e freqüência. Feito isso, os conceitos mais relevantes são determinados. Cada grupo de dois ou mais conceitos relevantes que são adjacentes na ontologia em questão é então agrupado de modo a formar um grupo de conceitos. Um grupo de conceitos é tratado como um único conceito (relevante). Desta forma, se apenas um grupo de conceitos é encontrado e este grupo de conceitos engloba todos os conceitos relevantes, então o processo de sumarização termina e o sumário corresponde à subontologia contendo todos os conceitos relevantes. Caso contrário, se existe pelo menos um conceito relevante não-adjacente, o processo de sumarização identifica todos os caminhos (até um determinado tamanho) entre conceitos relevantes (grupos de conceitos). Cada caminho corresponde a uma subontologia em O e pode conter conceitos não-relevantes.

Métricas usadas comumente na área de recuperação de informação são aplicadas para computar a qualidade de cada caminho. Finalmente, o melhor caminho é selecionado como o sumário da ontologia.

4 Aspectos da implementação

Neste capítulo serão apresentados alguns detalhes importantes da implementação, bem como instruções de como utilizar a ferramenta. Alguns experimentos serão descritos e validados a fim de demonstrar a coerência entre o que foi projetado (capítulo 3) e o que foi implementado.

4.1 Ferramenta Implementada

A ferramenta implementada foi denominada OWLSummarizer, uma vez que se trata de um sumarizador de arquivos OWL, isto é, ontologias codificadas no padrão OWL. A ferramenta foi implementada em Java. A razão para tal é que hoje em dia uma variedade de aplicações em formas de API (*Application Program Interface*) podem ser utilizadas em conjunto para solucionar diversos problemas. Desta maneira, a ferramenta aqui desenvolvida pode ser baixada e utilizada como uma API simples para resumir ontologias OWL, sendo facilmente anexável a qualquer outra aplicação java já existente.

A implementação está organizada de acordo como o digrama de pacotes da Figura 4. São 5 pacotes:

i) **summarizer:** Este é o pacote raiz que engloba todos os outros e contém a classe Summarizer, a partir da qual os comandos para sumarização são enviados;

ii) **summarizer.base:** Neste pacote estão contidas as classes básicas necessárias às diversas etapas do processo de sumarização. São elas: Graph (representa um grafo), Node (representa um nó em um grafo), Edge (representa uma aresta em um grafo) e Ontology (representa uma ontologia OWL);

iii) **summarizer.algorithm:** Neste pacote está contida a interface GraphSummaryCalculator que representa uma calculadora de resumos. Ainda nesse pacote existe a implementação de uma calculadora de resumos com base em *F-Measure*, denominada FmeasureGraphCalculator;

iv) **summarizer.parsing:** Aqui estão contidas todas as classes necessárias para a transformação de um arquivo OWL em um grafo e vice-versa. A classe responsável por fazer a conversão OWL ←→ Grafo é denominada O2GParser. Neste pacote existe ainda um conversor de arquivo XML contendo os mapeamentos ontológicos dos conceitos. Este conversor varre os mapeamentos do arquivo XML e partir dele gera uma instância da classe OntologyData que guarda informações que serão úteis para o cálculo da freqüência dos conceitos;

v) **summarizer.logging:** Aqui está contida a classe responsável por fazer o log da ferramenta.

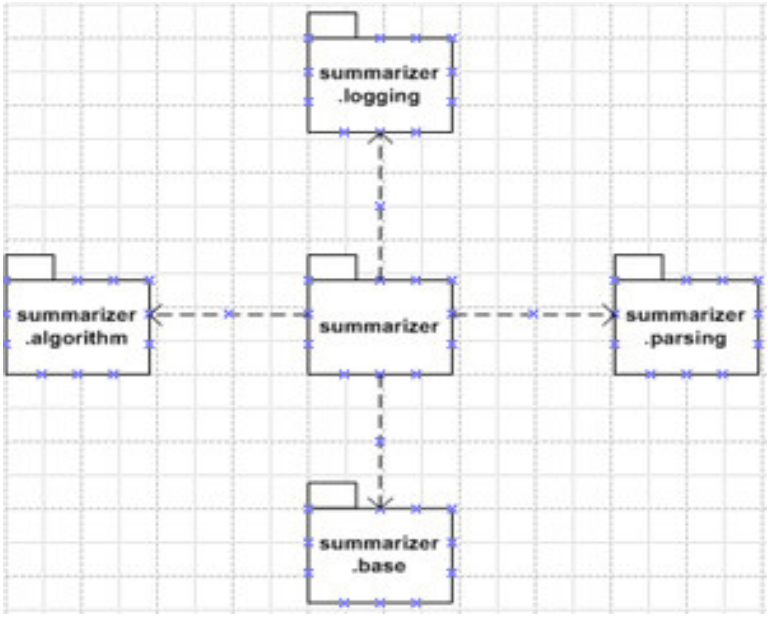

Figura 4. Diagrama de pacotes do OWLSummarizer

Embora o núcleo do processo de sumarização tenha sido especificado no capítulo três, algumas etapas anteriores e posteriores ao processo descrito são necessárias. As etapas anteriores basicamente consistem em tratar os parâmetros recebidos do usuário. Tais parâmetros são:

i) Nome do arquivo OWL de entrada a ser resumido;

ii) Nome do arquivo OWL de saída que conterá o resumo;

iii) Nome do arquivo XML contendo os mapeamentos ontológicos;

iv) Valor entre 0 e 1 informando o peso da centralidade no cálculo da relevância;

v) Valor entre 0 e 1 informando o peso da freqüência no cálculo da relevância;

vi) Valor entre 0 e 1 informando o alfa da média harmônica para cálculo da qualidade dos caminhos;

vii) Limite de nós relevantes na ontologia original;

viii) Tamanho de referência do resumo;

ix) Variação do tamanho de referência do resumo.

Apenas os parâmetros (i) e (ii) são obrigatórios na chamada do sumarizador, os parâmetros de (iv) à (ix) podem ser informados pelo arquivo summary.properties, como mostrado na Figura 5.

```
centrality_weight=0.5
frequency_weight=0.5
alpha_measure=0.5
relevant_concepts=6
summary_size_reference=6
summary_size_delta=2
```

Figura 5. Exemplo de summary.properties

Na Figura 5, os significados das variáveis lá informadas estão descritos abaixo:

i) centrality_weight: peso da centralidade. Exemplo: 0.5;

ii) frequency_weight: peso da frequência. Exemplo: 0.5;

iii) alpha_measure: alfa da média harmônica. Exemplo: 0.5;

iv) relevant_concepts: limite de conceitos relevantes da ontologia a ser resumida. Exemplo: 6;

v) summary_size_reference: tamanho de referência em conceitos da ontologia resumida. Exemplo: 6;

vi) summary_size_delta: variação do tamanho da ontologia resumida com base no tamanho de referência. Exemplo: 2.

Uma vez recebidos os parâmetros, a ferramenta converte a ontologia informada em um grafo. Ao converter a ontologia em um grafo, arestas relativas a auto-relacionamentos (quando origem e destino da aresta correspondem ao mesmo nó) normalizadas, escondidas do grafo para que não tenham efeito no cálculo do resumo, o mesmo ocorre com arestas múltiplas partindo de um nó para outro. Todas essas arestas múltiplas são normalizadas em uma única aresta que representa todas juntas. Ao final, depois que o resumo é encontrado, arestas múltiplas e auto-relacionamentos são desnormalizados, isto é, voltam a aparecer no grafo encontrado e só então o grafo é convertido na ontologia OWL resumida.

Caso um arquivo XML de mapeamentos tenha sido informado, a ferramenta, antes de iniciar o processo de sumarização, converte as informações contidas no arquivo XML para uma instância da classe OntologyData que será usada para cálculo das freqüências posteriormente. Apenas quando terminada essa conversão, o processo de sumarização inicia.

4.2 Usando a Ferramenta

Ao descompactar a ferramenta, a mesma pode ser usada a partir da chamada do arquivo summarize.bat, passando todos os parâmetros explicitamente na mesma ordem informada anteriormente ou apenas os dois parâmetros obrigatórios, isto é, o nome do arquivo OWL de entrada e o nome do arquivo OWL de saída, estando os outros parâmetros informados no summary.properties, com exceção do arquivo de mapeamentos que não é obrigatório. Exemplo de uso:

summarize Music.owl MusicSummary.owl

Como resultado será gerado o resumo da ontologia Music.owl no arquivo MusicSummary.owl, também será criado um log no formato da Figura 6 com o nome Music.log. A sintaxe completa é:

summarize <owl_filename> <owl_summarized_filename>
<frenquency_filename> <centrality_weight> <frequency_weight>
<alpha_measure> <relevant_concepts> <summary_size_reference> <summary_size_delta>

```
====================== RELEVANT NODES ======================

Node (Musician) relevance: 0.3235294117647059
Node (Group) relevance: 0.2647058823529412
Node (Music_piece) relevance: 0.20588235294117646
Node (Musical_Instrument) relevance: 0.17647058823529413
Node (String_instruments) relevance: 0.14705882352941177
Node (Class) relevance: 0.14705882352941177
Node (Quintet) relevance: 0.08823529411764706
Node (Trio) relevance: 0.08823529411764706
Node (Tempo) relevance: 0.08823529411764706
Node (Violinist) relevance: 0.058823529411764705
Node (Person) relevance: 0.058823529411764705
Node (Double_bass) relevance: 0.058823529411764705
Node (Performer) relevance: 0.058823529411764705
Node (Composer) relevance: 0.058823529411764705
Node (Violist) relevance: 0.058823529411764705
Node (Cello) relevance: 0.058823529411764705
Node (Violin) relevance: 0.058823529411764705
Node (Viola) relevance: 0.058823529411764705
Node (Double_Bassist) relevance: 0.058823529411764705
Node (Cellist) relevance: 0.058823529411764705
Node (Piano) relevance: 0.058823529411764705
Node (Movement) relevance: 0.058823529411764705
Node (Solo) relevance: 0.058823529411764705
Node (Pianist) relevance: 0.058823529411764705
Node (String_quintet) relevance: 0.029411764705882353
Node (Sonata) relevance: 0.029411764705882353
...
Node (Orchestra) relevance: 0.029411764705882353
Node (Piano_trio) relevance: 0.029411764705882353
Node (Quartet) relevance: 0.029411764705882353

Average Node Relevance: 0.03865546218487394

==================== END OF RELEVANT NODES ====================

All relevant nodes are adjacents! No need to find summary
candidates!


====================== SUMMARY CHOSEN ======================

Nodes: [Musical_Instrument, Music_piece, Group, Musician, Class,
String_instruments]
Edges: [(Musician, main_instrument, plays_instrument, Class),
(Group, consists_of_instruments - consist_of_members, Class),
(Class, played_repertory, Music_piece), (String_instruments, is-a,
Musical_Instrument), (Musical_Instrument, main_player, Class)]
Size: 6
Recall: 1.0
Precision: 1.0
F-Measure: 1.0
Relevance: 0.21078431372549014

==================== END OF SUMMARY CHOSEN ====================

Original number of classes: 35
Summarized number of classes: 6

Total time: 1.0 seconds.
```

Figura 6. Exemplo de arquivo de log. (Music.log)

4.3 Descrição de Experimentos

Com o objetivo de testar a ferramenta desenvolvida, foi selecionada uma ontologia (univ-cs.owl, vide Figura 7) e aplicadas duas conFigurações de resumos, descritas na tabela 1. A ontologia univ-cs.owl possui 53 conceitos.

Nome da Ontologia	Centralidade	Freqüência	Tamanho do Resumo	Delta	F-measure
univ-cs.owl	1.0	0.0	7 conceitos	2	0.5
univ-cs.owl	0.5	0.5	7 conceitos	2	0.5

Tabela 1. ConFigurações para geração do resumo das ontologias.

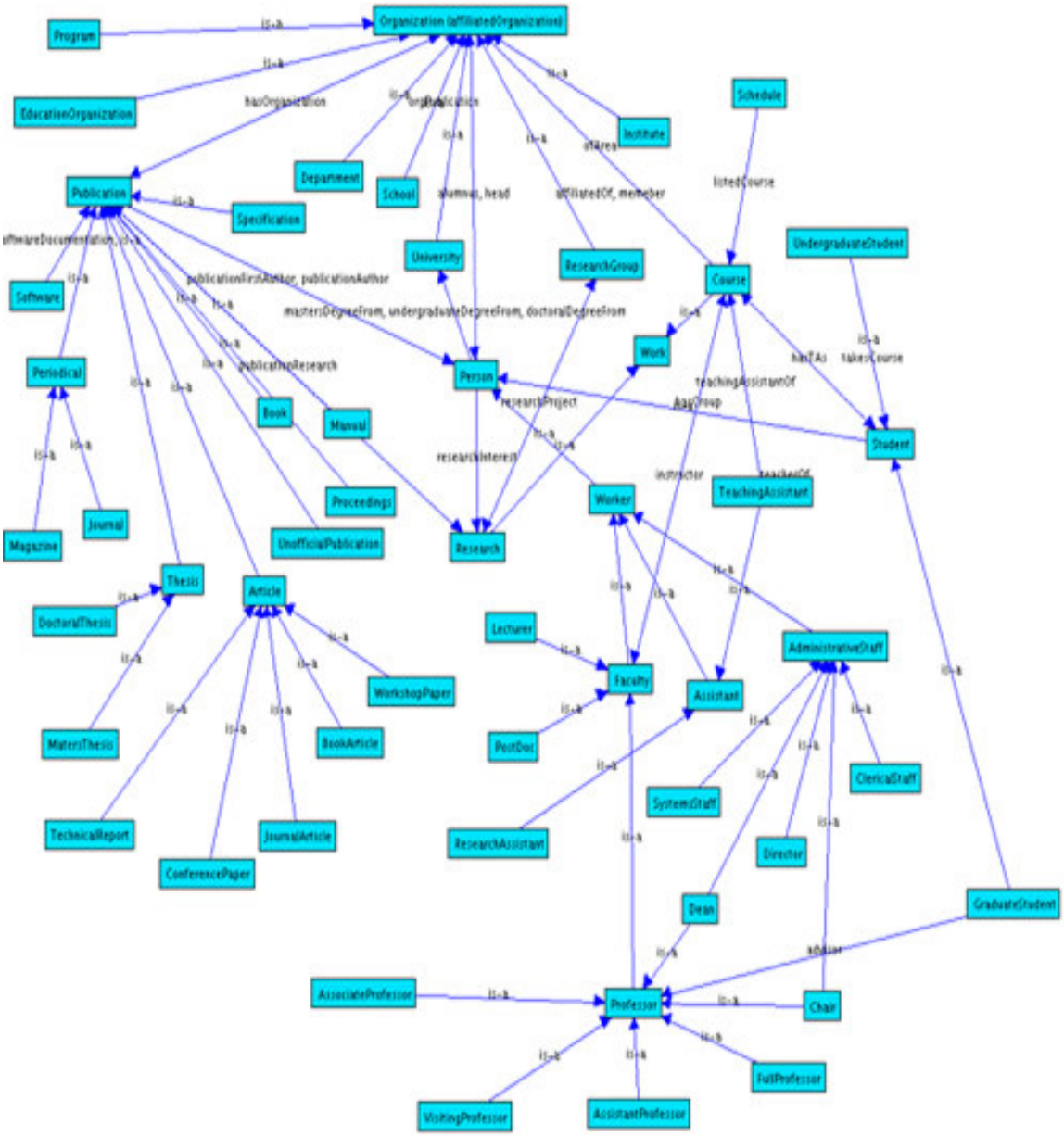

Figura 7. Grafo que representa a ontologia univ-cs.owl

4.4 Resultados Obtidos

Ao se calcular os conceitos relevantes da ontologia univ-cs.owl para o primeiro experimento, isto é, como conFigurado na primeira linha da tabela 1, seis conceitos relevantes foram determinados, como indicado na Figura 8. A razão pela qual apenas 7 conceitos foram determinados relevantes é porque se considerou o parâmetro que indica o limite de conceitos relevantes igual ao parâmetro que indica o tamanho do sumário pretendido, tal consideração é válida para ambos os experimentos da tabela 1, ficando então implícito, o parâmetro que indica o limite de conceitos relevantes.

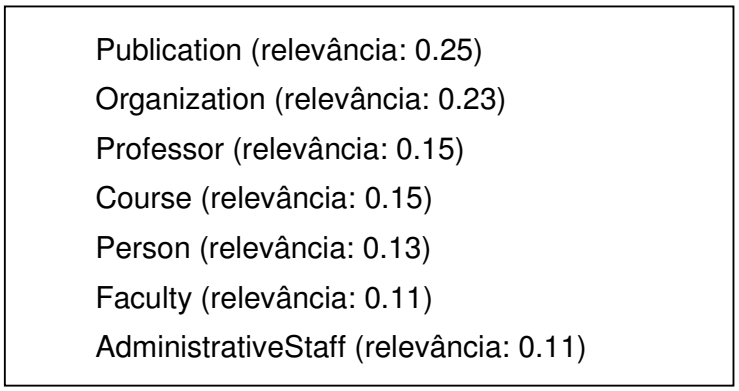

Figura 8. Conceitos relevantes da ontologia univ-cs.owl, durante experimento 1.

O algoritmo verificou que existia um superconceito contendo 6 dos 7 conceitos relevantes. Tal superconceito continha os seguintes subconceitos: “Person”, “Organization”, “Publication”, “Professor”, “Course” e “Faculty”. Após isso o algoritmo calculo todos os caminhos do superconceito até o conceito relevante restante “AdministrativeStaff”. O caminho com menor *F-Measure* encontrado incluiu o conceito não relevante “Worker” para alcançar “AdministrativeStaff”, como pode ser visto no sumário obtido da Figura 10.

Pôde-se observar que o resumo encontrado para o primeiro experimento de fato corresponde a um resumo que privilegia a centralidade, dado que os conceitos presentes no resumo são os mais centrais da ontologia em questão.

Para o segundo experimento, utilizou-se a mesma ontologia do primeiro experimento, porém dessa vez um arquivo de mapeamentos ontológicos foi passado como parâmetro para que a freqüência fosse levada em consideração no cálculo da relevância. O arquivo de mapeamentos passado como parâmetro apenas atribuía ocorrência de mapeamento para os conceitos “AdministrativeStaff” e “Article”, respectivamente 1 (uma) e 2 (duas) ocorrências de mapeamento. O restante dos conceitos terão então freqüência 0, devido a inexistência de mapeamentos para esses conceitos.

Ao se calcular os conceitos relevantes do segundo experimento, foram obtidos os conceitos relevantes da Figura 9. Observa-se que devido à freqüência, os conceitos “Article” e “AdministrativeStaff” passaram a ser os mais relevantes. Com essa nova conFiguração de conceitos relevantes, um superconceito englobando “Course”, “Person”, “Organization”, “Publication” e “Article” foi criado, restando ainda dois conceitos relevantes não adjacentes: “Professor” e “AdministrativeStaff”. Dessa forma, o algoritmo comparou os melhores caminhos entre o superconceito, “Professor” e “AdministrativeStaff”,

verificando que houve empate de *F-Measure* entre todos os caminhos, porém aquele com maior relevância média possuia como origem ou destino os conceitos “Professor” e “AdministrativeStaff”, gerando então o sumário da Figura 11.

Article (relevância: 0.56)
AdministrativeStaff (relevância: 0.31)
Publication (relevância: 0.13)
Organization (relevância: 0.12)
Professor (relevância: 0.08)
Course (relevância: 0.08)
Person (relevância: 0.07)

Figura 9. Conceitos relevantes da ontologia univ-cs.owl durante experimento 2.

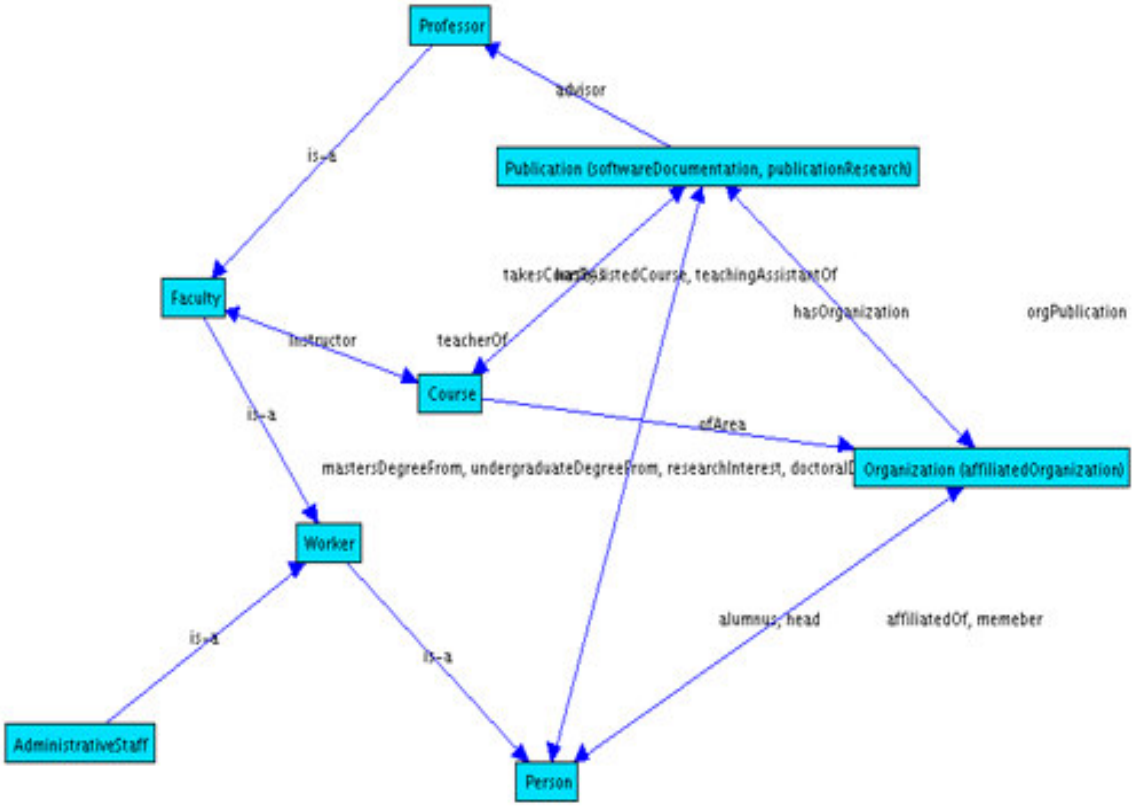

Figura 10. Resumo encontrado para a ontologia univ-cs.owl no experimento 1.

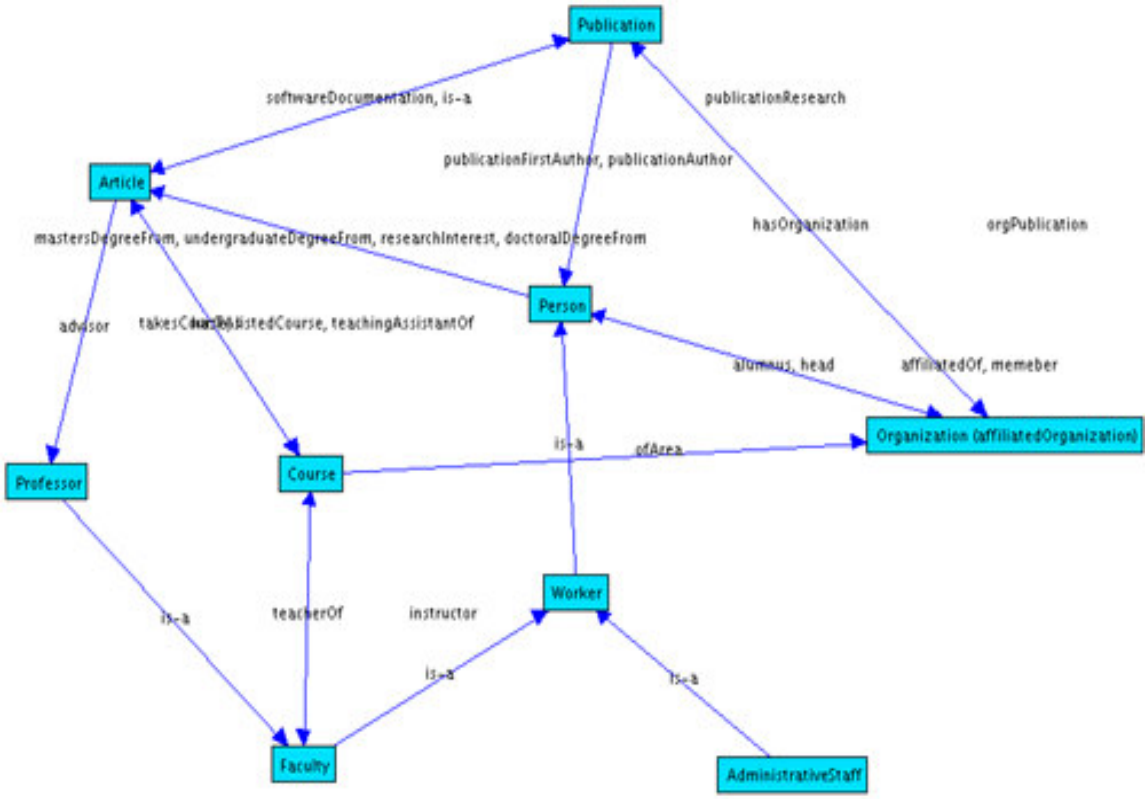

Figura 11. Resumo encontrado para a ontologia univ-cs.owl no experimento 2.

4.5 Considerações Finais

Atualmente, existem três situações específicas na qual a ferramenta não encontrará resumo, são elas:

i) Quando todos os caminhos candidatos a resumo encontrados estão abaixo do limite inferior do tamanho esperado como resumo;

ii) Quando todos os caminhos candidatos a resumo encontrados estão acima do limite superior do tamanho esperado como resumo;

iii) Quando existem caminhos candidatos a resumo tanto abaixo como acima da faixa de tamanho esperado como resumo.

Devido às situações (i), (ii) e (iii), percebe-se que o uso dos parâmetros passados para a ferramenta pode ser crítico quando se deseja encontrar um sumário de forma automática. Desta maneira, visando descobrir quais seriam os melhores parâmetros para cada situação, uma rede neural ou uma máquina de aprendizagem poderia ser cogitada para se aprender acerca da melhor forma de usar os parâmetros. Para tanto, seria necessário treinar tal máquina de aprendizagem para resumos oriundos de diferentes domínios a fim de descobrir como os parâmetros devem se comportar de modo a obter um melhor resumo. Assim, para situações nas quais o usuário não soubesse *a priori* qual a melhor maneira de fornecer os parâmetros, ele poderia utilizar um conjunto de ontologias de treinamento, bem como seus resumos de forma a auxiliá-lo na descoberta dos melhores parâmetros para encontrar resumos em um determinado domínio ou ontologia.

5 Conclusão

Este trabalho mostrou como é possível desenvolver uma ferramenta automática para o resumo de ontologias e como a idéia de freqüência é importante e possivelmente inovadora para o cálculo da relevância de conceitos. Além disso, a ferramenta implementada é parametrizada o suficiente para permitir ao usuário encontrar diferentes tipos de resumo de acordo com requisitos como tamanho, importância da freqüência e centralidade dos conceitos.

Técnicas oriundas da área de recuperação de informação foram utilizadas para auxiliar o cálculo do melhor caminho entre dois conceitos relevantes, bem como técnicas comumente utilizadas para a determinação de nós importantes em um grafo, como a idéia de centralidade.

Este documento, em conjunto com os dois experimentos demonstrados, evidencia que a ferramenta é capaz de produzir resumos customizáveis de forma útil. Além disso, a ferramenta pode ser facilmente portada para aplicações *Java* já existentes que necessitem resumir ontologias.

5.1 Dificuldades Encontradas

Algumas dificuldades foram encontradas durante o desenvolvimento do trabalho:

- A OWL-API utilizada para o tratamento de ontologias em OWL estava parcamente documentada e os exemplos nem sempre eram fáceis de entender;
- Dificuldade para encontrar ferramentas simples que permitissem a visualização de arquivos OWL representados como grafos;
- A implementação de um algoritmo para calcular todos os caminhos entre dois nós em um grafo foi especialmente trabalhosa, pois uma recursão com vários critérios de parada precisou ser desenvolvida de modo a tornar o algoritmo mais eficiente;
- Para que a ferramenta fosse devidamente testada, simulações com grafos com diversas configurações espaciais precisaram ser feitas manualmente a fim de descobrir como o software reagiria a essas situações. Exemplo de configuração espacial: grafos com diferentes granularidade (isto é, a distribuição dos nós em relação a como os mesmos estão concentrados) e tamanhos.

5.2 Trabalhos Futuros

Embora o processo implementado para encontrar resumos em arquivos OWL funcione satisfatoriamente para uma grande diversidade de casos, ainda existem pontos que podem ser melhorados futuramente:

i) tempo de resposta para se encontrar um resumo;

ii) capacidade de encontrar resumos sob condições bem específicas;

iii) maior capacidade de flexibilização do cálculo da relevância e da qualidade dos caminhos.

Quanto ao tempo de resposta da ferramenta, poderiam ser otimizadas certas computações que são feitas apenas ao final, quando todos os caminhos candidatos à sumário estão definidos. Por exemplo, o cálculo de *F-Measure* poderia ser feito juntamente com a identificação dos caminhos, o que faria o caminho identificado entre dois nós relevantes já possuir seu *F-Measure* calculado, sem que fosse necessário calcular o *F-Measure* de cada caminho numa etapa posterior. Também é possível promover um ganho de tempo de resposta eliminando-se instruções destinadas meramente à geração do arquivo de registro para fins de verificação do processo ou simplesmente tornar a geração deste arquivo opcional a partir de um parâmetro.

Visando solucionar as três considerações feitas na seção 4.5, que descrevem situações na qual a ferramenta na encontra resumos, três sugestões poderiam ser estudadas como um trabalho futuro:

- Para o caso (i) a ferramenta poderia ser modificada para simplesmente adicionar, de forma incremental, o conceito mais relevante entre aqueles adjacentes a um caminho candidato a sumário até que, onde T representa a faixa de tamanho esperado do sumário e $|OS_n|$ o tamanho de um caminho candidato a sumário. Esse procedimento seria feito para todos os caminhos candidatos imediatamente mais próximos de T. Quando todos estivessem com $|OS_n| \varepsilon T$, então o *F-Measure* de cada caminho OS_n seria atualizado e aquele com o novo maior *F-Measure* selecionado como sumário;
- Para o caso (ii), de modo semelhante ao caso (i), o algoritmo poderia simplesmente remover, de forma incremental, ou o conceito relevante origem do caminho ou o conceito relevante destino do caminho, optando por remover aquele com menor relevância. Tal procedimento deveria ser feito até que $|OS_n| \varepsilon T$, onde T representa a faixa de tamanho esperado do sumário e $|OS_n|$ o tamanho de um caminho candidato a sumário. Similar ao caso (i), esse procedimento seria feito para todos os caminhos candidatos imediatamente mais próximos de T. Quando todos estivessem com $|OS_n| \varepsilon T$, então o *F-Measure* de cada caminho OS_n seria atualizado e aquele com o novo maior *F-Measure* selecionado como sumário;
- A solução para o caso (iii) poderia ser a combinação das soluções para o caso (i) e (ii). Tais medidas certamente empobreceriam o resumo para esses casos, mas seria, de certa forma, o preço a se pagar por um resumo de tamanho insuficiente para representar uma ontologia.

Futuramente, é possível tornar a ferramenta ainda mais *customizável*. A forma de se calcular relevância e a qualidade dos caminhos adotadas poderiam ser informadas pelo usuário, possibilitando o uso de outras fórmulas para esses cálculos. Trabalhos futuros podem vir a descobrir melhores fórmulas além de uma soma ponderada para o cálculo da relevância ou uma maneira mais adequada que uma média harmônica (*F-Measure*) para o cálculo da qualidade dos caminhos.

Referências

[Castano at al. 1998] Castano, S., De Antonellis, V., Fugini, M. G., and Pernici, B. 1998. Conceptual Schema Analysis: Techniques and Applications. In Proc. of the ACM Transactions on Database Systems (TODS), Volume 23, Issue 3. Pages: 286-333.

[DAML+OIL] http://www.daml.org/2001/03/daml+oil-index

[Gaertler at al. 2004] Gaertler, M. and Wagner, D. 2004. Algorithms for Representing Network Centrality, Groups and Density, and Clustered Graph Representation. http://i11www.iti.uni-karlsruhe.de/cosin/documents/section_D6.pdf, ultimo acesso em 12/08/2008.

[Gruber, 1993] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199-220, 1993.

[Horrocks & Schneider] Ian Horrocks & Peter F. Patel-Schneider. Reducing OWL Entailment to Description Logic Satisfiability.

[OIL] http://www.ontoknowledge.org/oil

[Ontology Languages, Wikipedia] http://en.wikipedia.org/wiki/Ontology_(information_science)#Ontology_languages

[Perez & Corcho, 2002] Ontology Languages for the Semantic Web.

[Pires, 2007] Pires, C. E. S. 2007. Um Sistema P2P de Gerenciamento de Dados com Conectividade Baseada em Semântica. Exame de Qualificação e Proposta de Tese. UFPE/CIn. Disponível em http://www.cin.ufpe.br/~cesp

[Pires, 2008] Artigo em desenvolvimento por Carlos Eduardo Pires, denominado Building Ontologies a ser publicado ainda em 2008.

[Razmerita & Maedche 2003] Razmerita, L., Angehrn, A., & Maedche, A. 2003. "Ontology-Based User Modeling for Knowledge Management Systems". In: Lecture Notes in Computer Science: 213-217.

[Smith, 2008] Smith, B. Ontology (Science), in C. Eschenbach and M. Gruninger (eds.), Formal Ontology in Information Systems. Proceedings of FOIS 2008, Amsterdam/New York: ISO Press, 21-35.

[W3C, 2002] Feature Synopsis for OWL Lite and OWL: W3C Working Draft 29 July 2002. W3C (2002-07-29).

[W3C, 2004] OWL Web Ontology Language Guide: W3C Recommendation 10 February 2004. W3C (2004-02-10).

[W3C UCs, 2004] OWL Web Ontology Language Use Cases and Requirements: W3C Recommendation 10 February 2004. W3C (2004-02-10).

[W3C XML] http://www.w3.org/XML

[W3C RDF] http://www.w3.org/RDF

[WebOnt, 2004] Web-Ontology (WebOnt) Working Group (Closed). W3C.

[Yu and Jagadish 2006] Yu, C. and Jagadish, H. V. 2006. Schema Summarization. In Proc. of the 32nd International Conference on Very Large Data Bases, Seoul, Korea. Pages: 319-330.

[Zhang et al. 2007] Zhang, X., Cheng, G., and Qu, Y. 2007. Ontology summarization based on rdf sentence graph. In Proc. of the 16th International Conference on World Wide Web, Banff, Alberta, Canada. Pages: 707-716.

Assinaturas

Este Trabalho de Graduação é resultado dos esforços do aluno Victor Bezerra Alencar, sob a orientação da professora Ana Carolina Salgado e co-orientação do doutorando Carlos Eduardo Santos Pires, sob o título de "Uma Ferramenta para Sumarização de Ontologias". Todos abaixo estão de acordo com o conteúdo deste documento e os resultados deste Trabalho de Graduação.

_____

Ana Carolina Salgado

_____

Victor Bezerra Alencar