UNIVERSIDADE FEDERAL DE PERNAMBUCO
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA

NICOLLE CHAVES CYSNEIROS

ANÁLISE SEMÂNTICA DE UM *MATCHER* DE ONTOLOGIAS

TRABALHO DE GRADUAÇÃO

RECIFE
2015

NICOLLE CHAVES CYSNEIROS

ANÁLISE SEMÂNTICA DE UM *MATCHER* DE ONTOLOGIAS

TRABALHO DE GRADUAÇÃO

Trabalho de Graduação apresentado à graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco para obtenção do grau de Bacharel em Ciência da Computação.

Orientadora – Prof.ª Ana Carolina Salgado (acs@cin.ufpe.br)

RECIFE
2015

NICOLLE CHAVES CYSNEIROS

ANÁLISE SEMÂNTICA DE UM *MATCHER* DE ONTOLOGIAS

TRABALHO DE GRADUAÇÃO

Trabalho de Graduação apresentado à graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco para obtenção do grau de Bacharel em Ciência da Computação.

Recife, ____ de fevereiro de 2015.

BANCA EXAMINADORA

_____

Prof.ª Ana Carolina Salgado
(Orientadora)

_____

Prof.ª Bernadette Farias Lóscio
(Avaliadora)

AGRADECIMENTOS

Agradeço aos meus pais pelo incentivo e pelo apoio incondicional. À Prof.ª Ana Carolina pela orientação e pela ajuda sempre solícita e rápida durante a elaboração desse projeto. Aos meus amigos do Centro de Informática com os quais eu pude conversar sobre as dificuldades enfrentadas.

RESUMO

O volume de dados disponíveis em fontes de dados distintas e conectadas vem crescendo, juntamente com a necessidade de extrair informações consistentes dessas fontes. O problema de integração de dados descritos por esquemas diferentes envolve a geração de um mapeamento entre conceitos do esquema, conhecida como a operação de *matching* de esquemas. Ao longo dos últimos anos, diversas técnicas de *matching* de esquemas foram desenvolvidas, incluindo algoritmos que buscam estender as correspondências entre termos para além do nível sintático, chegando ao nível semântico do relacionamento entre conceitos. Esse Trabalho de Graduação propõe a análise das correspondências semânticas geradas pelo *matcher* semântico *SemMatcher*.

SUMÁRIO

1. INTRODUÇÃO

Este capítulo fornecerá uma contextualização dos assuntos abordados neste projeto e a motivação para o trabalho produzido. Também serão apresentados os principais objetivos deste Trabalho de Graduação, bem como a estrutura do restante do documento.

1.1. Contextualização e Motivação

Com o crescimento da disponibilidade de informação provida pelo aumento da quantidade de computadores conectados em rede, fez-se necessária a criação de ferramentas que auxiliem o usuário a ter acesso aos dados armazenados em máquinas diferentes e organizados de acordo com esquemas diferentes. Para as aplicações que envolvem a integração dessas fontes de dados heterogêneas, o *matching* de esquemas é uma importante operação de comparação de esquemas que recebe como entrada dois esquemas de dados e retorna um alinhamento identificando os elementos correspondentes (MADHAVAN , BERNSTEIN e RAHM , 2001).

Ao longo dos últimos anos, a operação de *matching* vem sendo estudada e novas técnicas foram desenvolvidas, incluindo algoritmos que buscam expandir a semântica das correspondências entre elementos dos esquemas para além da simples correspondência sintática dos termos (BERNSTEIN , MADHAVAN e RAHM , 2011). Nesse contexto, o trabalho de (PEREIRA, 2008) propõe o desenvolvimento de um *matcher* semântico de ontologias que recebe como parâmetros os esquemas respresentados por ontologias das duas fontes de dados que se deseja integrar e uma ontologia que descreve o conhecimento do domínio onde os dois primeiros esquemas estão inseridos.

A saída do *matcher* semântico é um conjunto de correspondências semânticas entre os elementos das ontologias de entrada. Tais relacionamentos são encontrados baseados em regras semânticas apresentadas no trabalho de (SOUZA, ARRUDA, *et al.*, 2009), como equivalência, especialização, generalização, agregação (parte e todo) e disjunção. Para cada uma dessas correspondências semânticas é atribuído um peso no intervalo de [0, 1] que será utilizado no cálculo da medida de similaridade semântica entre os esquemas de entrada.

O *matcher* semântico apresentado em (PEREIRA, 2008) foi desenvolvido e testado em cenários envolvendo integração de dados e recuperação de informação de diferentes pontos. Um exemplo de uso do SemMatcher é no processo de formação de clusters semânticos, onde a medida de similaridade semântica é utilizada para determinar em qual

cluster, o ponto entrante deve se conectar. Além desse cenário, o SemMatcher também é utilizado no processo de roteamento da consulta, especificamente os pesos das correspondências são usadas no cálculo da medida de enriquecimento semântico da consulta durante o processo de roteamento da mesma, como apresentado no trabalho de (FREIRE, 2014).

1.2. Objetivos

Visto que o SemMatcher pode ser utilizado em diferentes cenários, este Trabalho de Graduação tem como objetivo geral tornar a ferramenta independente do ambiente onde ela é utilizada, de forma que ela possa ser incorporada por outros sistemas. Para atingir o objetivo geral, alguns objetivos específicos foram traçados:

i. Tornar os pesos associados às correspondências semânticas valores parametrizados, passados como argumentos dos principais métodos fornecidos pela ferramenta SemMatcher.
ii. A criação de uma API (*Application Programming Interface*) da ferramenta para que a mesma possa ser reutilizada em outras aplicações;
iii. A avaliação da ferramenta dentro do contexto de roteamento de consultas.

O projeto ainda comportará uma fase de teste onde será necessária a criação de ontologias escritas em Linguagem OWL e cenários específicos de uso para avaliar o impacto dos pesos das correspondências semânticas.

1.3. Estrutura do Documento

O restante do documento será organizado da seguinte forma:

- Capítulo 2: Apresenta definições e conceitos relacionados ao processo de *matching* de ontologias e faz um breve estudo sobre as principais ferramentas de *matcher* já desenvolvidas.
- Capítulo 3: Descreve uma visão geral sobre a ferramenta *SemMatcher* e o ambiente em que ela é utilizada (sistema SPEED), bem como os cenários em que os pesos atribuídos às correspondências semânticas têm grande importância.
- Capítulo 4: Descreve experimentos realizados com a ferramenta *SemMatcher* dentro do contexto de roteamento de consultas no ambiente SPEED e analisa os resultados obtidos.
- Capítulo 5: Faz um apanhado geral sobre as principais conclusões feitas a partir do trabalho desenvolvido, bem como sugestões sobre possíveis trabalhos futuros.

2. *MATCHING* DE ONTOLOGIAS

Neste capítulo serão apresentados definições e conceitos sobre a operação de *matching* de ontologias que são importantes para o entendimento desse trabalho, bem como algumas aplicações onde este tipo de processamento é de fundamental importância. Nessa seção também será feito um estudo sobre algumas ferramentas de *matching* já existentes.

2.1. Ontologias

Dentro do contexto da Ciência da Informação, ontologia é um conjunto de elementos (classes, atributos e relacionamentos) projetado para modelar um domínio de conhecimento (GRUBER, 2009). Ontologias são usadas para descrever, compartilhar e facilitar o reuso de conhecimento. A linguagem padrão adotada pelo W3C para a representação de ontologias é a OWL (*Web Ontology Language*), uma extensão da linguagem RDF (*Resource Description Framework*) e tem sua estrutura similar a um documento XML (BECHHOFER, HARMELEN, *et al.*, 2004). Os axiomas lógicos expressos em OWL são triplas formadas por um sujeito, um predicado e um objeto, onde o sujeito e o objeto são conceitos e o predicado é o tipo de relacionamento existente entre esses conceitos. Através de ontologias é possível representar o conteúdo de diferentes fontes de dados, agregando significado semântico aos dados armazenados.

Dadas duas ou mais ontologias sobre o mesmo domínio de conhecimento, é possível obter uma única ontologia que engloba as informações provenientes das diferentes fontes (operação de *merging*) ou obter um conjunto de correspondências entre os conceitos descritos nas diferentes ontologias, mantendo-as separadas em diferentes fontes (operação de *matching*) (NOY e MUSEN, 2000).

2.2. A Operação de *Matching* de Ontologias

O *matching* de ontologias é definido como uma operação que recebe como entrada duas ontologias O_1 e O_2 e fornece como resultado um mapeamento entre os elementos das ontologias (que podem representar esquemas de dados) passados como parâmetro (MADHAVAN , BERNSTEIN e RAHM , 2001). Opcionalmente, o *matcher* também pode receber artefatos que serão utilizados durante o processo de *matching*, como dicionários de palavras, enciclopédias ou um esquema do domínio.

As ferramentas que realizam a operação de *matching* podem aplicar diferentes técnicas para comparar os conceitos presentes nas ontologias que estão sendo manipuladas. Tais

ferramentas podem ser classificadas de acordo com os seguintes critérios (RAHM e BERNSTEIN, 2001):

a. Tipo de Informações

Com relação ao tipo de informações das ontologias que serão consideradas durante a realização do *matching*, a ferramenta pode utilizar a descrição da ontologia (esquemas) ou as instâncias armazenadas. Nos *matchers* baseados em esquemas, o nome dos conceitos e de suas propriedades serão avaliados e comparados durante a operação de *matching*. É a técnica comumente utilizada pelas ferramentas disponíveis.

Nos *matchers* baseados em instâncias, os indivíduos que estão armazenados nas ontologias são utilizados para realizar o *matching* de esquemas. Esse processo é feito utilizando técnicas de aprendizagem de máquina para extrair conhecimento sobre a organização das instâncias armazenadas nas ontologias a partir das mesmas.

b. Granularidade

A granularidade do *matcher* define qual o nível da informação que será utilizada para realizar a comparação entre os esquemas. No caso de *matching* em nível de elementos, cada conceito que compõe o esquema é considerado.

Em *matching* em nível de estrutura, os elementos que formam a estrutura são avaliados em conjunto. Nessa técnica, o *matching* pode ser feito parcialmente (quando foram achadas correspondências para apenas alguns elementos da estrutura) ou completamente (quando todos os elementos foram correspondidos).

c. Cardinalidade

A cardinalidade do *matcher* define como os mapeamentos entre os elementos das ontologias passadas como entrada será feito. A cardinalidade pode ser de 1:1 (quando um elemento da primeira ontologia é mapeado para apenas um elemento da segunda ontologia), 1:n (quando um elemento da primeira ontologia é mapeado para vários elementos da segunda ontologia), n:1 (quando vários elementos da primeira ontologia são mapeados para um elemento da segunda ontologia) ou n:m (quando vários elementos da primeira ontologia são mapeados para vários elementos da segunda ontologia).

d. Combinação de *Matchers*

Visto que existe uma grande variedade de possíveis técnicas que podem ser utilizadas para a implementação de um *matcher* de esquemas, é possível combinar essas técnicas para formar uma combinação de *matchers*, que pode dar origem a um *matcher* híbrido ou um

matcher composto. No caso de um *matcher* híbrido, um único *matcher* é desenvolvido utilizando diferentes técnicas para a implementação. Já no *matcher* composto, vários *matchers*, cada um implementando uma técnica distinta, são usados para realizar a operação e seus resultados são combinados posteriormente.

2.3. Aplicações

O *matching* de ontologias é uma importante operação em aplicações que manipulam dados provenientes de diferentes fontes. Seja para realizar a integração de dados/ontologias ou o carregamento de um *data warehouse*, o *matching* dos esquemas das fontes de dados é um pré-requisito para que a aplicação possa funcionar corretamente. Dentro desse cenário, a operação de *matching* é feita em tempo de análise do projeto, podendo ser executada de forma manual ou semiautomática (SHVAIKO e EUZENAT, 2008).

Um cenário mais desafiador para o uso de um *matcher* de ontologias pode ser encontrado em aplicações que tratam de fontes de dados mais dinâmicas, como PDMS (*Peer Data Management System*) e serviços na Web. A operação de *matching* nesse contexto é empregada no processamento de consultas e requisições, e é um problema que precisa ser tratado em tempo de execução e deve prover resultados com um bom nível de confiabilidade.

2.4. Ferramentas para *Matching* de Ontologias

a. COMA

Uma das ferramentas que mais se destaca com relação às técnicas utilizadas é o COMA (MASSMANN, RAUNICH, *et al.*, 2011), pois ela é um *matcher* composto que permite a utilização de *matchers* que implementam diferentes técnicas dentro da mesma ferramenta.

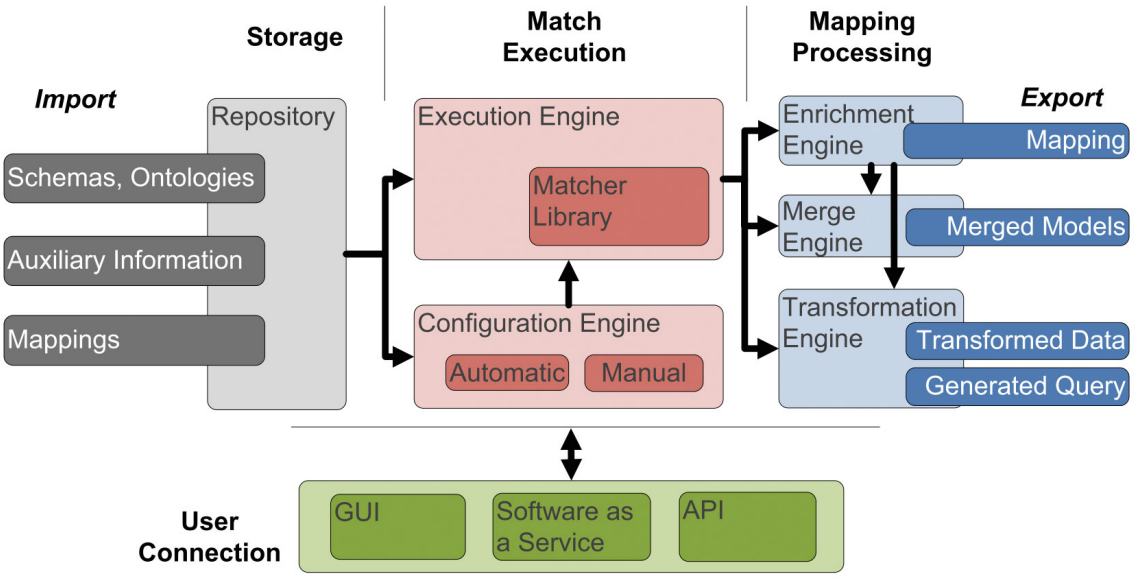

Figura 1 Arquitetura geral da ferramenta COMA

A Figura 1 ilustra a arquitetura da ferramenta, que pode ser genericamente dividida em três partes: *storage*, *match execution* e *mapping processing*. A parte de armazenamento (*storage*) da ferramenta é responsável por armazenar os dados relacionados ao processo de *matching*, como os esquemas e ontologias processados, informações auxiliares ao processo, além de mapeamentos entre conceitos já existentes. Esses dados são transformados em um modelo de dados genérico e armazenados em um banco de dados relacional.

Já na parte de *match execution,* o processamento automático do *match* é feito na *execution engine*, que realiza a combinação de diferentes blocos de *matching* que implementam estratégias diferentes. Esse processamento é realizado em três etapas: inicialmente é necessário identificar quais os componentes dos esquemas de entradas que são relevantes para a operação de *matching*, em seguida, diferentes *matchers* são utilizados para computar a similaridade entre os elementos dos esquemas e, finalmente, essas similaridades são combinadas para formar o mapeamento final entre os esquemas. Para configurar quais tipos de *matchers* serão utilizados e como as similaridades devem ser combinadas, o *match customizer* é utilizado.

Na etapa de *mapping processing*, o *enrichiment engine* é o componente responsável por enriquecer as correspondências 1:1 encontradas, através da associação de funções para transformação de dados. Por exemplo, dadas as correspondências *Name - FirstName* e *Name - LastName*, a ferramenta irá, de forma semiautomática, especificar que o elemento Name do primeiro esquema deverá sofrer a operação de separação (*split*) para que os elementos *FirstName* e *LastName* do segundo esquema possam ser obtidos. Com relação às

correspondências semânticas entre elementos, o COMA é capaz de identificar relacionamentos de especialização e generalização através do processamento de relações do tipo *is-a* e *inverse is-a* entre conceitos. Os demais componentes do *mapping processing* realizam operações sobre os mapeamentos encontrados, como a operação de *merging*.

A partir dessa visão global sobre a ferramenta, é possível perceber que o COMA se destaca por fazer uso de vários *matchers* que aplicam diferentes técnicas tradicionais de *matching*. Dessa forma, tal ferramenta atende as necessidades de um grande número de aplicações que necessitam realizar a correspondência de esquemas.

b. HMatch

O *HMatch* 2.0 (CASTANO, FERRARA e MONTANELLI, 2007) também é um *matcher* composto, mas ele busca fornecer uma arquitetura mais flexível, provendo um *framework* onde cada componente da ferramenta pode ser utilizado de forma independente ou em combinação com outros componentes. A Figura 2 apresenta um esquema da interação entre os componentes que formam o *HMatch*.

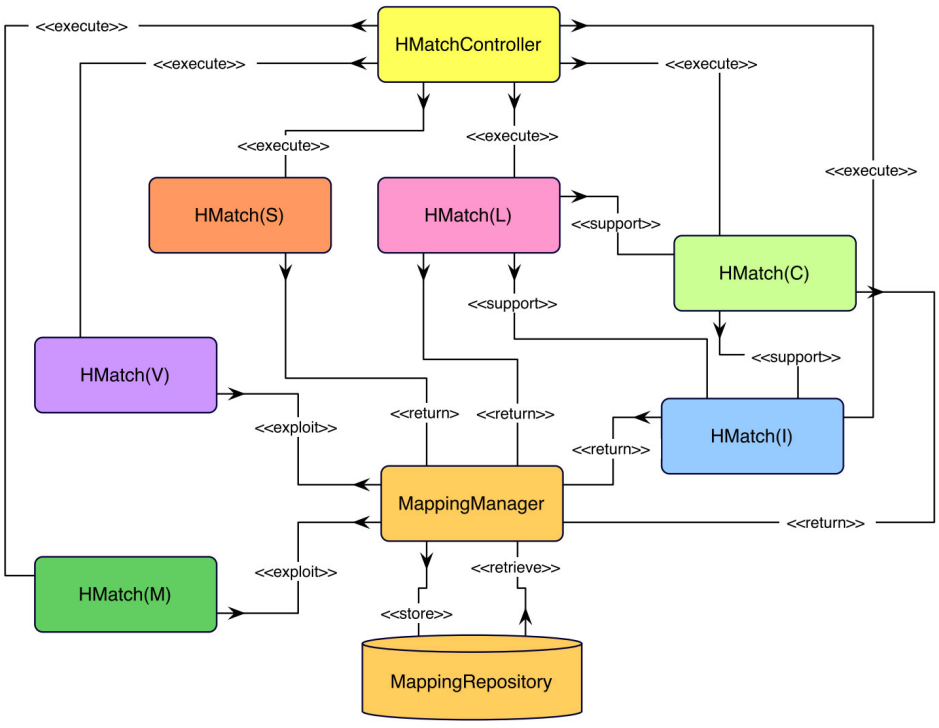

Figura 2 Visão alto nível dos componentes do HMatch 2.0 e suas interações

O *HMatchController* é o componente responsável por administrar as configurações da ferramenta e o fluxo de execução da aplicação, além de prover uma interface para os demais componentes. O *HMatch(L)* é o componente dedicado à execução do *matching* linguístico dos

nomes dos elementos das ontologias, se baseando na comparação sintática e semântica (através de um sistema léxico) dos mesmos. O componente *HMatch(S)* avalia a similaridade de dois elementos de acordo com a estrutura das ontologias que estão sendo comparadas. O *HMatch(I)* realiza a operação de *matching* se baseando nas instâncias armazenadas nas ontologias. O componente *HMatch(C)* determina o nível de afinidade semântica entre os conceitos baseando-se no contexto. Já o componente *HMatch(M)* realiza a análise e combinação dos mapeamentos gerados pelos demais componentes. Finalmente, o *MappingManager* é responsável por armazenar, recuperar e processar os mapeamentos resultantes.

Visto que o objetivo principal desse trabalho é a análise semântica de uma ferramenta de *matching*, o entendimento mais detalhado do funcionamento do componente *HMatch(C)* se faz importante para estabelecer parâmetros de comparação com o funcionamento da ferramenta *SemMatcher*. Para determinar o mapeamento entre duas ontologias, a ferramenta calcula a medida de similaridade semântica no range de [0, 1] de acordo com a complexidade semântica escolhida (CASTANO, FERRARA e MONTANELLI, 2006). A complexidade semântica determina o nível de complexidade com que a ontologia será analisada, cada nível irá focar em diferentes tipos de elementos da ontologia. Existem 4 modelos de *matching* definidos no *HMatch*, onde cada um aplica diferentes níveis de complexidade semântica:

- ***Surface***: nesse nível de complexidade semântica, apenas o nome do conceito é considerado no processo de *matching*;
- ***Shallow***: o nome e as propriedades dos conceitos são levados em consideração nesse nível;
- ***Deep***: além dos elementos considerados no nível ***Shallow***, esse nível de complexidade também considera as relações semânticas do conceito com outros conceitos da ontologia.
- ***Intensive***: esse nível compreende os elementos considerados no nível ***Deep*** e também o valor das propriedades do conceito sendo analisado.

Para calcular a medida de similaridade semântica, o *HMatch* combina as medidas de similaridade linguística e de similaridade contextual. A medida de similaridade linguística é obtida de acordo com a comparação dos nomes dos elementos da ontologia e seus significados. Para tanto, a ferramenta utiliza uma enciclopédia de termos derivada do sistema léxico *WordNet* para obter uma medida da relação sintática entre os conceitos.

Já a medida de similaridade contextual é calculada de acordo com as propriedades do conceito e com as relações semânticas deste com os conceitos adjacentes. Sobre as propriedades do conceito, estas podem ser classificadas como **forte** (caso a propriedade seja obrigatória) ou **fraca** (caso a propriedade seja opcional). O *HMatch* atribui um peso P a estas classificações, onde $P_{forte} \geq P_{fraca}$. No que diz respeito às relações semânticas entre conceitos, a ferramenta analisa 4 classificações:

- ***Same-as:*** caso os conceitos comparados sejam equivalentes
- ***Kind-of:*** caso os conceitos apresentem uma relação de especialização
- ***Part-of:*** caso os conceitos apresentem uma relação de composição
- ***Associates:*** caso os conceitos apresentem uma associação semântica positiva, ou seja, os conceitos possuem algum tipo de relacionamento arbitrário.

Para cada classificação de relação semântica, a ferramenta atribui um peso P, onde $P_{same-as} \geq P_{kind-of} \geq P_{part-of} \geq P_{associates}$

A Figura 3 ilustra o processo de *matching* realizado pelo componente *HMatch(C)*, mostrando os pontos onde os conceitos explicados acima são aplicados.

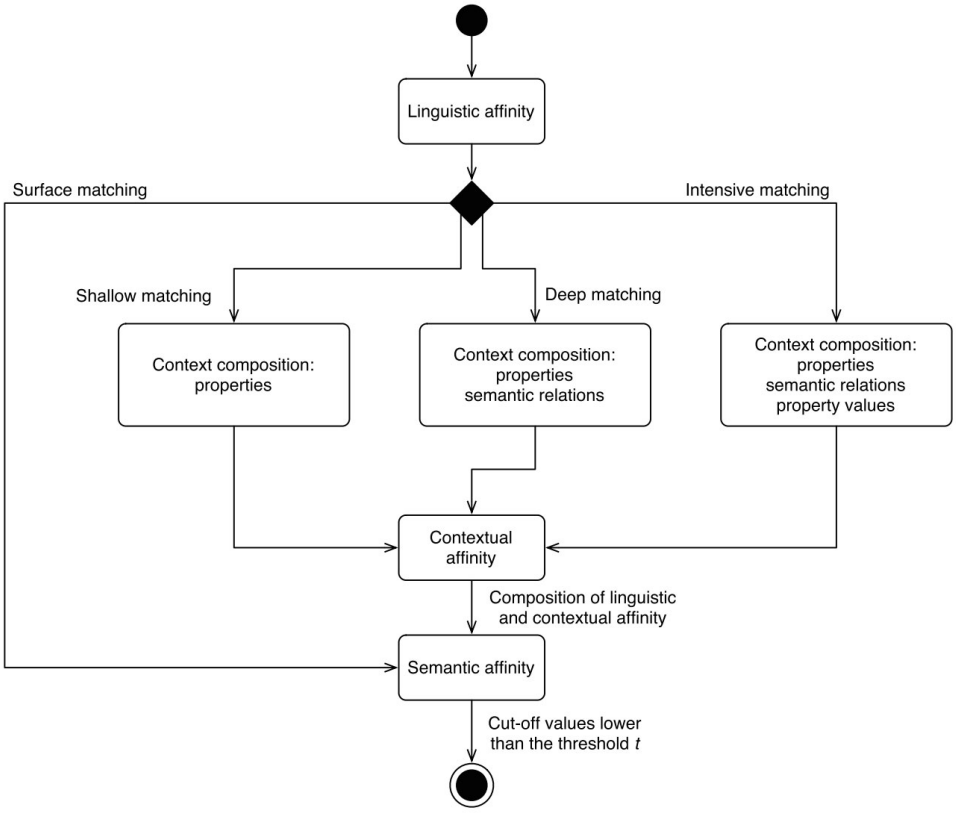

Figura 3 Processo de *matching* realizado pelo componente *HMatch(C)*

c. Comparando as ferramentas apresentadas

A Tabela 1 mostra a comparação entre as ferramentas detalhadas anteriormente com relação às técnicas de *matching* adotadas, bem como a especificação do uso de elementos do contexto para a geração de relações semânticas entre os conceitos das ontologias passadas como parâmetro para tais ferramentas.

	COMA ++	***HMatch***
Tipo de Informação	Esquema Instância	Esquema Instância
Granularidade	Elementos Estrutura	Elementos Estrutura
Combinação	Composto	Composto
Tipo de relacionamento entre elementos	*is-a* *inverse-is-a*	*Same-as* *Kind-of* *Part-of* *Associates*

Tabela 1 Comparação entre as ferramentas de *matching*

2.5. Considerações Finais

Este capítulo introduziu conceitos importantes para o entendimento do trabalho desenvolvido, como Ontologias e a operação de *Matching* de Ontologias. Também foi realizado um estudo acerca das principais ferramentas de *matching* disponíveis atualmente. O próximo capítulo desse documento irá descrever o estado atual do *SemMatcher* e seu uso dentro do projeto SPEED, bem como quais alterações foram implementadas para que a análise da ferramenta fosse feita.

3. SPEED E *SEMMATCHER*

Nesse capítulo será apresentada uma visão geral da ferramenta *SemMatcher* e uma descrição acerca da arquitetura do sistema SPEED. Essa sessão também especifica o processo de roteamento e reformulação de consulta no sistema e a importância do *SemMatcher* nesse contexto.

3.1. Visão Geral da Ferramenta

O *SemMatcher* é um *matcher* semântico de ontologias que recebe como entrada dois esquemas, representados por ontologias, de fontes de dados e uma ontologia de domínio usada na normalização dos termos dos outros dois esquemas (PEREIRA, 2008). Essa é uma ferramenta que se destaca por encontrar correspondências semânticas entre os elementos dos esquemas. A Figura 4 mostra um esquema do processo de funcionamento do *SemMatcher*.

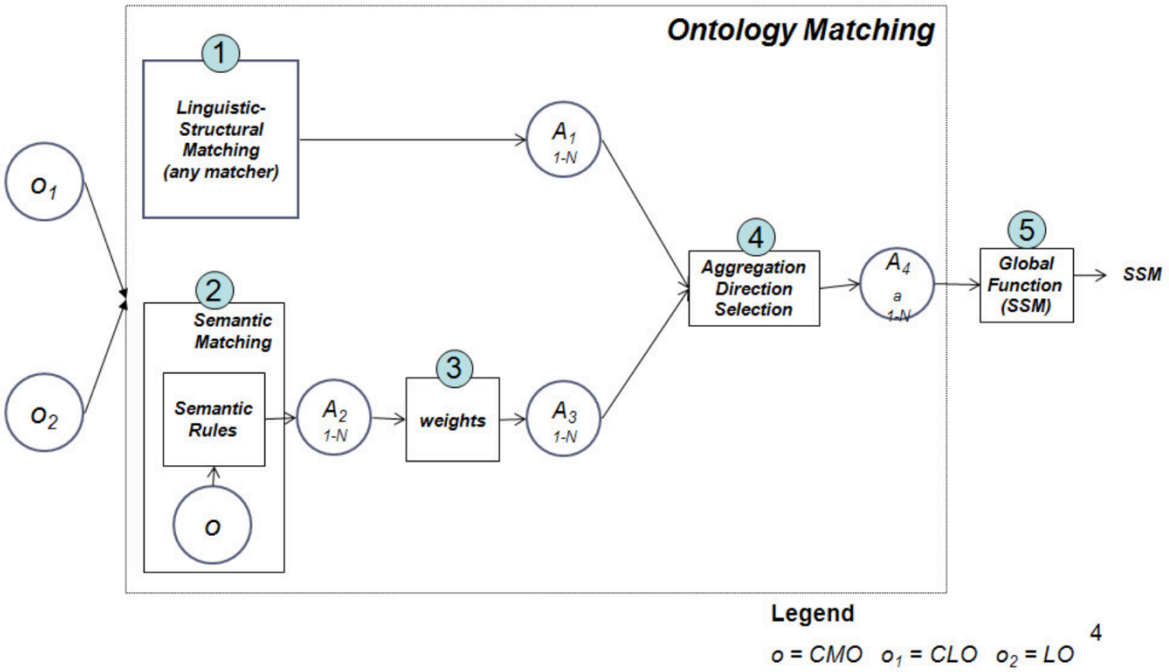

Figura 4 Esquema do processo do *SemMatcher*

Na Figura 4, as duas ontologias passadas como entrada são O1 e O2, enquanto a ontologia de domínio é representada por O. De forma paralela, o *SemMatcher* realiza o *matching* linguístico-estrutural, utilizando a ferramenta HMatch, e o *matching* semântico. Ao final dessa etapa, tem-se dois pares de alinhamentos 1:N gerados pelas duas técnicas de *matching* utilizadas. Os alinhamentos são combinados e a medida de similaridade da correspondência encontrada é dada através do cálculo da média ponderada entre a medida de similaridade da correspondência que foi gerada pelo *matcher* linguístico-estrutural e a medida

de similaridade da mesma correspondência que foi gerada pelo *matcher* semântico. Dessa forma, o *SemMatcher* obtém um único alinhamento 1:N.

a. *Matching* Semântico

A operação de *matching* semântico implementado pelo *SemMatcher* (componente 2 da Figura 4) gera correspondências semânticas, definidas no trabalho de (SOUZA, ARRUDA, *et al.*, 2009), entre os elementos das duas ontologias passadas como parâmetro para a ferramenta. Essas correspondências são mostradas na Figura 5.

1. i:x $\equiv\!\!\Rightarrow$ j:y, uma correspondência *isEquivalentTo*. (Equivalência)
2. i:x $\sqsubseteq\!\!\Rightarrow$ j:y, uma correspondência *isSubConceptOf*. (Especialização)
3. i:x $\sqsupseteq\!\!\Rightarrow$ j:y, uma correspondência *isSuperConceptOf*. (Generalização)
4. i:x $\rhd\!\!\Rightarrow$ j:y, uma correspondência *isPartOf*. (Agregação – parte)
5. i:x $\lhd\!\!\Rightarrow$ j:y, uma correspondência *isWholeOf*. (Agregação – todo)
6. i:x $\approx\!\!\Rightarrow$ j:y, uma correspondência *isCloseTo*. (Proximidade)
7. i:x $\not\equiv\!\!\Rightarrow$ j:y, uma correspondência *isDisjointWith*. (Disjunto de)

Onde x e y são elementos (conceitos ou propriedades) pertencentes a uma ontologia Oi e outra ontologia Oj respectivamente representando duas ontologias do sistema.

Figura 5 Regras semânticas utilizadas pelo *SemMatcher*

Para encontrar a correspondência semântica entre elementos, a ferramenta faz uso da ontologia de domínio: o relacionamento entre os elementos das ontologias de entrada é verificado na ontologia de domínio e, se tal relacionamento tiver uma das sete correspondências semânticas associada, essa correspondência será adicionada ao mapeamento entre as ontologias. Uma vez que as correspondências semânticas foram estabelecidas, é atribuído um peso a tal correspondência. Esse peso varia de acordo com o tipo de correspondência e é utilizado no cálculo da medida de similaridade semântica entre as duas ontologias de entrada. Originalmente, o valor desses pesos foi definido arbitrariamente e incluído no código da implementação com os valores descritos na Tabela 2.

Correspondência Semântica	Peso
isEquivalentTo	1.0
isSubConceptOf	0.8
isSuperConceptOf	0.8
isCloseTo	0.7
isPartOf	0.3
isWholeOf	0.3
isDisjointWith	0.0

Tabela 2 Pesos associados às regras semânticas

b. Medida de Similaridade Semântica

Para calcular a medida de similaridade entre as duas ontologias passadas como parâmetro, o *SemMatcher* implementa duas funções - DICE (Equação 4) e *Average* (Equação 2) - e possibilita que o usuário escolha qual delas deve ser usada para o cálculo.

$$DICE(O_1, O_2) = \frac{|A_{12}| + |A_{21}|}{|O_1| + |O_2|}$$

onde $|A_{XY}|$ é a quantidade de alinhamentos de O_X para O_Y e $|O_X|$ é a quantidade de conceitos em O_X

Equação 1 Função DICE para o cálculo da medida de similaridade

$$Average(O_1, O_2) = \frac{\sum_{i=1}^{|A_{12}|} n_i + \sum_{i=1}^{|A_{21}|} n_i}{|O_1| + |O_2|}$$

onde n_i é o peso da correspondência i no alinhamento $|A_{XY}|$ e $|O_X|$ é a quantidade de conceitos em O_X

Equação 2 Função *Average* para o cálculo de medida de similaridade

Dessa forma, o *SemMatcher* fornece como saída um alinhamento 1:N entre os elementos das ontologias e um valor para a medida de similaridade semântica global entre as ontologias.

3.2. O Sistema SPEED

Originalmente, o *SemMatcher* foi desenvolvido como parte do sistema SPEED (*Semantic PEEr-to-Peer Data Management System*), um PDMS baseado em semântica utilizando a arquitetura de *super-peers*. No SPEED, os pontos da rede são organizados em comunidades semânticas, onde cada uma dessas comunidades é representada por um ponto semântico e estes, por sua vez, estão organizados em uma topologia DHT. Dentro de cada comunidade semântica, existem os pontos de dados e os pontos de integração. Os pontos de

dados são fontes de dados, cujo esquema é representado através de ontologias e que compartilham seus dados armazenados com os outros pontos da rede. Já os pontos de integração são pontos de dados com maior capacidade computacional e são responsáveis por integrar os demais pontos de dados, formando um *cluster* semântico. A Figura 6 ilustra a organização da arquitetura do sistema SPEED.

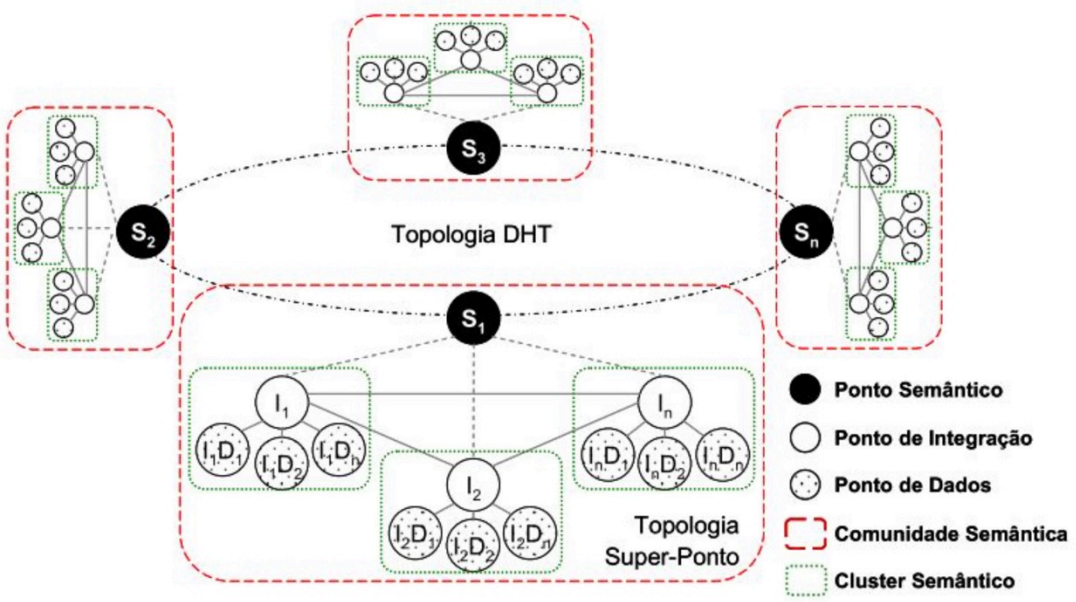

Figura 6 Arquitetura do SPEED

Dada a definição da arquitetura, pode-se notar que o SPEED precisa realizar a operação de *matching* de ontologias em pelo menos dois momentos:

i. Quando um ponto entra em uma comunidade semântica, é necessário calcular a medida de similaridade semântica entre este e os pontos de integração da comunidade para saber em qual *cluster* semântico o novo ponto deve ser inserido.
ii. Quando uma consulta é submetida em um ponto de dados, é necessário identificar as correspondências entre os termos presentes na consulta e os elementos presentes no esquema de dados do ponto para onde a consulta será propagada para reescrever a consulta de forma que o próximo ponto possa respondê-la.

3.3. Roteamento da Consulta no SPEED

Além das duas situações descritas na seção anterior em que o SPEED faz uso do *SemMatcher*, o sistema também precisa realizar a operação de *matching* de esquemas no momento do roteamento da consulta, ou seja, na hora de determinar para onde a consulta deverá seguir a partir daquele ponto. No trabalho de (FREIRE, 2014) é apresentada uma abordagem semântica para o roteamento da consulta em um PDMS que utiliza informações

semânticas, como as correspondências semânticas geradas pelo *SemMatcher*, para determinar qual conjunto de pontos da rede estão mais aptos a responder à consulta submetida.

a. *SemRef* - Reformulação Semântica

Quando uma consulta é passada de um ponto a outro na rede, a consulta é reformulada de acordo com o alinhamento entre a ontologia do ponto de origem e a ontologia do ponto de destino. No SPEED, esse processo de reformulação de consulta é feito utilizando uma estratégia baseada em semântica, como apresentado no trabalho de (SOUZA, ARRUDA, *et al.*, 2009).

Para evitar possíveis reformulações vazias - quando o ponto de destino não armazena um conceito que tenha correspondência exata com os termos da consulta - a reformulação entre dois pontos da rede é feita de forma enriquecida, utilizando técnicas de expansão e personalização da consulta. O processo de expansão é feito a partir da adição de termos semanticamente relacionados aos termos presentes na consulta, tomando como base as correspondências semânticas geradas pelo *SemMatcher* e a preferência do usuário para o tipo de enriquecimento que deve ser executado, como mostra a Tabela 3:

Tipo de Enriquecimento	Conceitos Adicionados na Reformulação
Approximate	Conceitos com correspondência *isCloseTo*
Specialize	Conceitos com correspondência *isSubConceptOf*
Generalize	Conceitos com correspondência *isSuperConceptOf*
Compose	Conceitos com correspondência *isPartOf* ou *isWholeOf*

Tabela 3 Tipos de enriquecimento da abordagem *SemRef*

Apesar do benefício de trazer uma resposta enriquecida para o usuário, com mais conceitos além daqueles solicitados, a abordagem de reformulação de consulta do *SemRef* pode gerar o problema da perda semântica da consulta original, caso a consulta seja roteada e reformulada para uma grande quantidade de pontos na rede.

b. SemRouting

A abordagem *SemRouting* proposta em (FREIRE, 2014) tem o objetivo de determinar o melhor conjunto de pontos capazes de responder à consulta submetida, preservando a semântica da consulta da melhor forma possível. A semântica da consulta é entendida como o conjunto de termos que faz parte da consulta original submetida. Para atingir tal objetivo, o

SemRouting realiza a análise e preservação semântica da consulta, gerando a referência semântica da consulta original e efetuando uma poda semântica de termos cada vez que a consulta é reformulada.

A referência semântica é uma estrutura que armazena todos os termos que têm relação semântica direta com os termos originais da consulta, i.e., todos os termos que possuem correspondência semântica (*isEquivalentTo, isSubConceptOf, isSuperConceptOf, isPartOf, isWholeOf, isCloseTo*) com os termos da consulta original. Já a poda semântica é definida como o processo de retirar da consulta reformulada os conceitos que não possuem relação semântica direta com os termos da consulta original.

Além de realizar a preservação semântica da consulta através da poda, o *SemRouting* também define critérios de qualidade usados para classificar os melhores pontos para responder à consulta. Dentre esses critérios, está o critério de relevância que é utilizado para eliminar os pontos que não estão aptos a trazer respostas relevantes à consulta. Para calcular a relevância, o *SemRouting* faz uso de duas funções descritas pela Equação 3 e pela Equação 4. O cálculo final da relevância é feito a partir da soma do resultado das duas funções.

$$PC(Q_{ref}) = \frac{|t_{eq}|}{|t_{orig}|}$$

onde Q_{ref} é a consulta reformulada,
$|t_{eq}|$ é a quantidade de termos da consulta original presente em Q_{ref} e
$|t_{orig}|$ é a quantidade de termos da consulta original

Equação 3 Função que calcula a Preservação da Consulta

$$EC(Q_{ref}) = \frac{\sum_{i=1}^{|t_{orig}|}(\sum_{j=1}^{|t_{exp_i}|} RS_{sim}(t_{orig}^{i}, t_{exp_i}^{j}))/|t_{exp_i}|}{|t_{orig}|}$$

onde Q_{ref} é a consulta reformulada,
$|t_{orig}|$ é a quantidade de termos da consulta original,
$|t_{exp_i}|$ é a quantidade de termos expandidos a partir do termo i da consulta original,
RS_{sim} é a função que retorna o peso associado à correspondência existente entre o termo i da consulta original e o termo j expandido a partir do mesmo de acordo com a referência semântica.

Equação 4 Função que calcula o Enriquecimento da Consulta reformulada

Analisando a função que calcula o enriquecimento da consulta, nota-se que os valores dos pesos atribuídos às correspondências semânticas geradas pelo *SemMatcher* causam

importante impacto no resultado final do cálculo. Visto que a relevância de um ponto é calculada levando em consideração o valor de enriquecimento da consulta, temos que para realizar experimentos envolvendo a medida de relevância dos pontos da rede utilizando a abordagem SemRouting, é necessário tornar os pesos utilizados pelo SemMatcher parametrizáveis

3.4. Modificações implementadas no *SemMatcher* e no SPEED

Dada a importância dos valores individuais dos pesos associados às correspondências semânticas para o processo de roteamento da consulta dentro do SPEED, foram feitas alterações na ferramenta *SemMatcher* para que a mesma possa fornecer ao usuário um maior controle sobre os valores do peso e que o mesmo possa utilizar a ferramenta numa maior gama de aplicações.

Na versão original do *SemMatcher*, os pesos de cada correspondência semântica tinham seus valores definidos dentro de uma classe de Java, ou seja, caso o usuário desejasse alterar tais valores, este teria que modificar o código fonte da ferramenta. Para resolver essa limitação, a ferramenta foi modificada para que o usuário possa passar um conjunto de valores para os pesos como parâmetro das principais funções fornecidas pelo *SemMatcher*. Além disso, a interface com o usuário da ferramenta também foi alterada para prover um espaço (painel *Weight Properties*) onde o usuário possa informar os valores preferíveis para os pesos das correspondências, como mostrado na Figura 7.

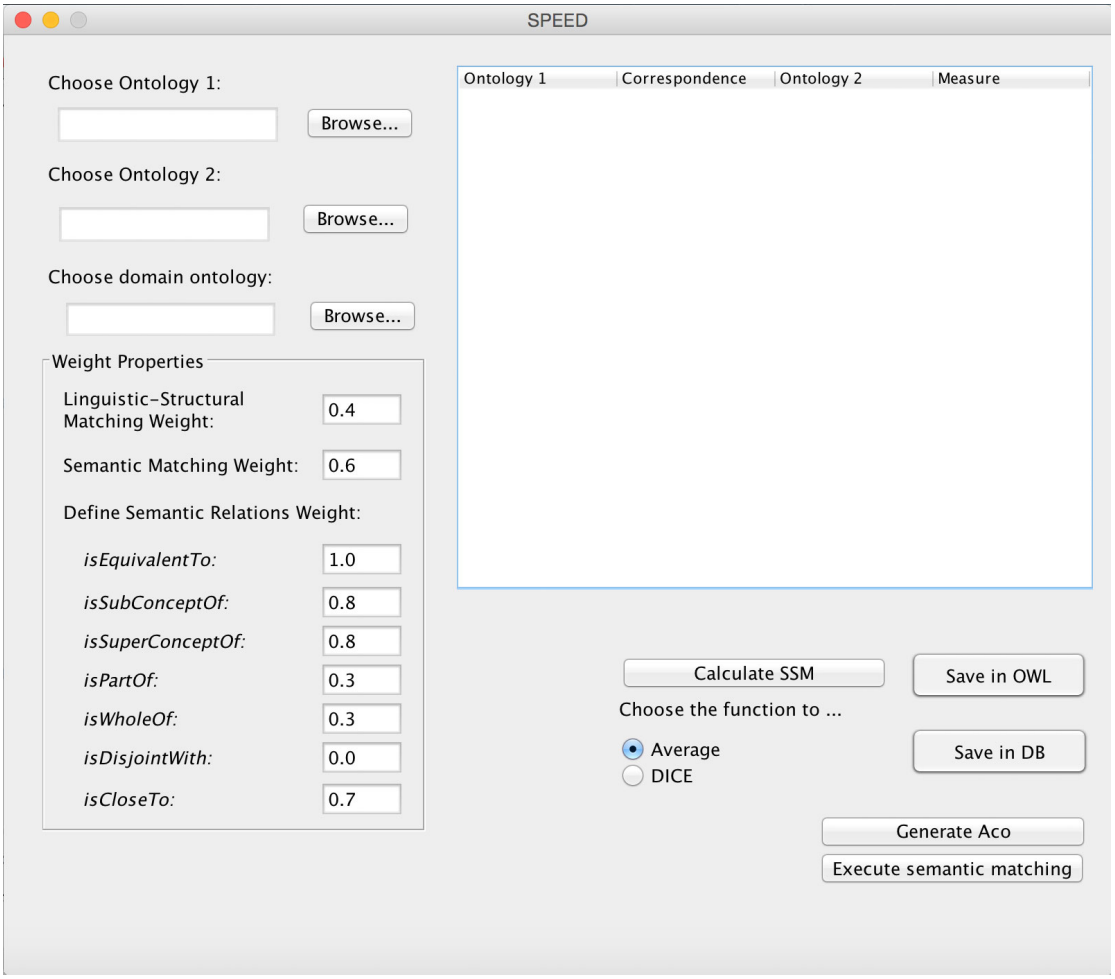

Figura 7 Nova interface com o usuário da ferramenta *SemMatcher*

A partir da inclusão dos valores dos pesos como parâmetro para os principais métodos do *SemMatcher*, foi feita a modificação do sistema SPEED de forma que o administrador do sistema possa escolher dois conjuntos de pesos para serem utilizados durante o processo de *matching*: um conjunto para o processo de formação dos clusters na rede e outro conjunto para o processo de reformulação e roteamento da consulta.

Outra modificação feita no *SemMatcher* foi o modo como a ferramenta trata o caminho do arquivo OWL fornecido pelo usuário. Originalmente, a ferramenta apenas considerava o padrão de caminho de arquivos utilizado pelo sistema operacional Windows. Essa limitação impede que o programa seja executado em computadores que não possuem o Windows como sistema operacional. Uma alteração foi feita de modo que o *SemMatcher* possa trabalhar com os arquivos selecionados pelo usuário independentemente do sistema operacional utilizado.

A partir das modificações implementadas, foi criada uma API para a ferramenta, onde os principais métodos disponíveis para o usuário estão presentes em uma Fachada, cuja

documentação está disponível no Apêndice A. Tanto a API quando a documentação completa da mesma podem ser encontradas em (CYSNEIROS, 2015)

3.5. Considerações Finais

Este capítulo proveu uma visão geral sobre a ferramenta *SemMatcher*, explicando sobre como as correspondências semânticas são encontradas e sobre como a medida de similaridade semântica entre ontologias é calculada. Também foi explicado o papel do *SemMatcher* dentro do sistema SPEED e a importância dos valores individuais dos pesos associados às correspondências semânticas para o processo de roteamento da consulta. Finalmente, as principais alterações feitas na ferramenta durante a elaboração desse projeto foram listadas. O próximo capítulo irá descrever e discutir os resultados de experimentos realizados com o *SemMatcher* dentro do cenário de roteamento de consultas.

4. EXPERIMENTO E ANÁLISE DOS RESULTADOS

Nesse capítulo será apresentada a descrição e análise do experimento realizado no sistema SPEED. O principal objetivo desse experimento é verificar quais as consequências geradas pela alteração dos valores dos pesos das correspondências geradas pelo *SemMatcher* no cenário de roteamento da consulta utilizando a abordagem *SemRouting*. A seguir, o cenário configurado no SPEED será descrito, bem como o processo realizado pelo programa. Finalmente, os resultados obtidos serão analisados.

4.1. Cenário do Experimento

O cenário projetado para esse experimento consiste de uma rede formada por cinco pontos de integração (PI 2178, P1 2278, P1 2378, PI 2478 e PI 2578) conectados de forma sequencial, como mostra a Figura 8. Visto que no sistema SPEED, os pontos de integração da rede são responsáveis por realizar o roteamento da consulta, os demais tipos de pontos da arquitetura SPEED não foram incluídos nesse cenário.

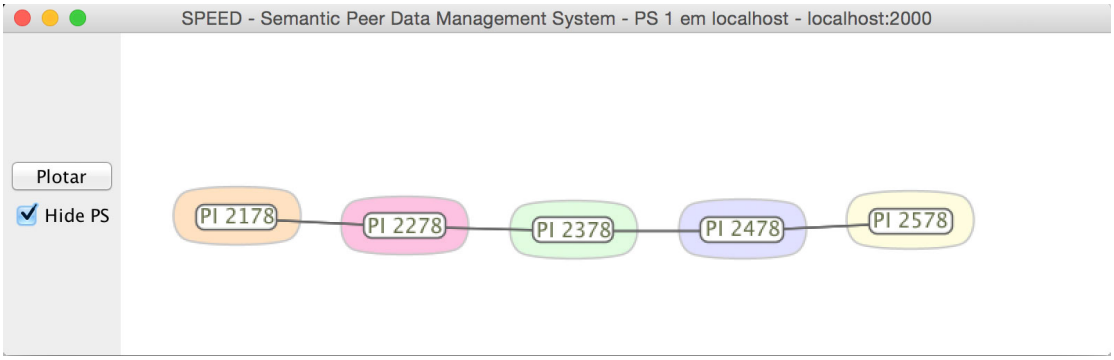

Figura 8 Configuração da Rede formada para o experimento

Os cinco pontos de integração estão inseridos em uma comunidade semântica que trata do domínio de conhecimento Educação, cuja ontologia de domínio é ilustrada no Anexo A desse documento. As figuras a seguir mostram um subconjunto dos conceitos armazenados por cada ponto da rede.

Figura 9 Subconjunto de conceitos armazenados pelo ponto PI 2178

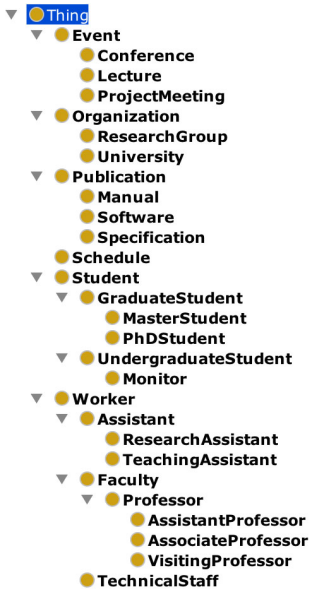

Figura 10 Subconjunto de conceitos armazenados pelo ponto PI 2278

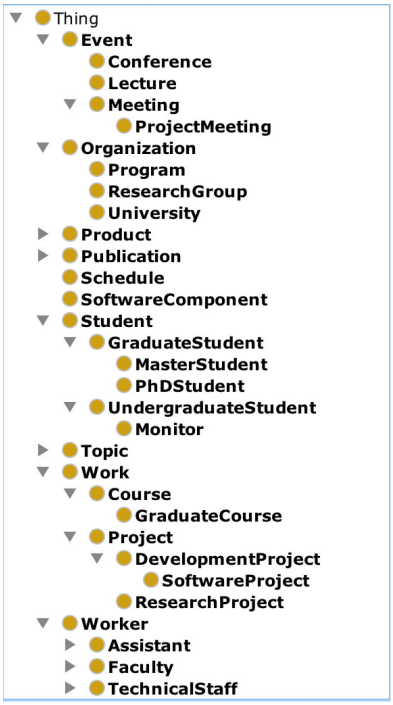

Figura 11 Subconjunto de conceitos armazenados pelo ponto PI 2378

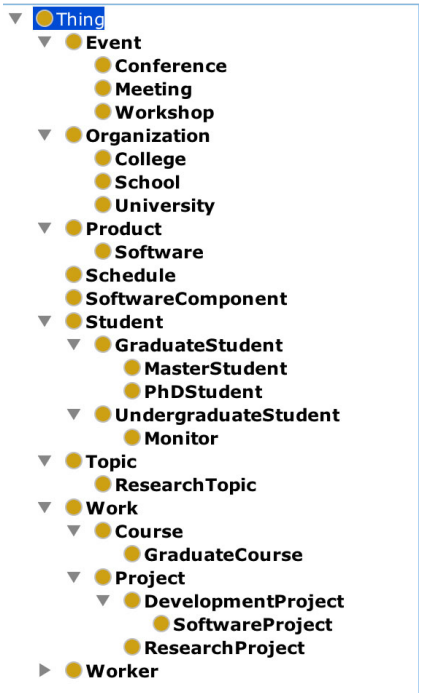

Figura 12 Subconjunto de conceitos armazenados pelo ponto PI 2478

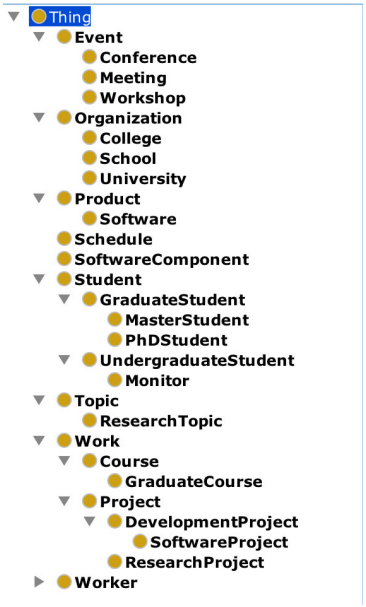

Figura 13 Subconjunto de conceitos armazenados pelo ponto PI 2578

4.2. Descrição do Experimento

Para a execução do experimento, a consulta $Q = \{GraduateStudent \cup Workshop\}$ foi submetida no ponto PI 2178 e todas as variáveis de enriquecimento $\{Approximate, Generalization, Specialization, Compose\}$ foram selecionadas para enriquecer a consulta. A consulta Q foi especificamente escolhida por proporcionar, dada a configuração da rede, um cenário onde a maior parte das correspondências semânticas geradas pelo *SemMatcher* foi utilizada no processo de expansão da consulta feita pela abordagem *SemRef*, permitindo uma análise mais completa dos pesos. A consulta submetida foi roteada na rede de pontos seguindo a abordagem *SemRouting*. Inicialmente, foi gerada a referência semântica (RS) para a consulta, como mostra a Figura 14.

RS = {(GraduateStudent, **GraduateStudent,** EQUIVALENT),
(GraduateStudent, **Student,** SUPER_CONCEPT),
(GraduateStudent, **PhdStudent,** SUB_CONCEPT),
(GraduateStudent, **MasterStudent,** SUB_CONCEPT),
(GraduateStudent, **ResearchProject,** PART),
(GraduateStudent, **Course,** PART),
(Workshop, **Workshop,** EQUIVALENT),
(Workshop, **Event,** SUPER_CONCEPT),
(Workshop, **Exhibition,** CLOSE),
(Workshop, **Conference,** CLOSE), (Workshop, **Meeting,** CLOSE),
(Workshop, **Lecture,** CLOSE)}

Figura 14 Referência Semântica gerada a partir da consulta submetida durante o experimento

Em seguida, a consulta foi roteada para os demais pontos da rede, sendo reformulada e sofrendo a poda semântica cada vez que era passada de um ponto de integração para outro.

Nesse momento também foi calculada a relevância do ponto para onde a consulta seguiria e seu valor foi registrado a cada execução. Para esse experimento não foi estabelecido um *threshold* para a relevância de um ponto com o objetivo de que a consulta fosse roteada para todos os pontos da rede.

Para avaliar as consequências de alterações dos valores associados aos pesos das correspondências semânticas geradas pelo *SemMatcher*, o processo descrito acima foi executado diversas vezes utilizando diferentes valores para os pesos. O *SemMatcher* atribui aos pesos das correspondências um valor no intervalo [0,1], sendo o valor 0.0 reservado para a correspondência entre termos disjuntos (regra semântica *DisjointWith*) e o valor 1.0 reservado para a correspondência entre termos equivalentes (regra semântica *EquivalentTo*). Dessa forma, os pesos das demais correspondências geradas pela ferramenta tiveram seus valores alterados dentro do intervalo [0.1, 0.9].

Além da definição do range de valores permitidos para os pesos, também foi determinada como deve ser a relação entre os mesmos. Tomando como base a relação entre os pesos definida pela ferramenta HMatch descrita no Capítulo 2 desse documento, temos que a relação de especialização (regra semântica *SubConceptOf*) tem um peso maior que a relação de composição (regras semânticas *PartOf* e *WholeOf*). Adicionando a relação de generalização (regra semântica *SuperConceptOf*) e aproximação (regra semântica *CloseTo*) geradas pelo *SemMatcher*, mas que não estão presentes na ferramenta HMatch, definimos o relacionamento entre os pesos (W) atribuídos a cada tipo de regra semântica como mostrado na Equação 5.

$$W_{SubConceptOf} \geq W_{SuperConceptOf} > W_{CloseOf} > W_{PartOf} \geq W_{WholeOf}$$

Equação 5 Relação entre pesos atribuídos a cada tipo de regra semântica utilizada pelo *SemMatcher*

Obedecendo ao intervalo e ao relacionamento entre os valores dos pesos definidos na Equação 5, diferentes conjuntos de pesos foram gerados de forma a explorar o intervalo possível de valores. Assim, foram utilizados conjuntos de pesos que possuíam valores máximos (próximo a 0.9), mínimos (próximos a 0.1) e medianos (próximos a 0.5), além de explorar a distribuição dos valores, como atribuindo ao maior peso o maior valor possível, ao menor peso o menor valor possível e distribuindo o valor do restante dos pesos ao longo do intervalo definido. Todos os conjuntos de pesos utilizados nesse experimento estão descritos na Tabela 5.

Dado que a configuração da rede e a consulta submetida permaneceram inalteradas durante a execução do experimento, temos que o resultado da reformulação da consulta com a

poda semântica no processo de roteamento da mesma será o mesmo, independentemente dos valores atribuídos aos pesos. A Tabela 4 mostra o resultado da reformulação da consulta para cada ponto na rede. Para facilitar o entendimento de como a expansão da consulta ocorreu, a consulta reformulada é apresentada como um conjunto de tuplas de alinhamentos gerados pelo *SemMatcher*.

Ponto de Origem	Ponto de Destino	Consulta Reformulada com Poda Semântica
PI 2178	PI 2278	$Q_{2178-2278}$ = {(Workshop, **Lecture,** CLOSE), (Workshop, **Conference,** CLOSE), (Workshop, **Event,** SUPER_CONCEPT), (GraduateStudent, **Student,** SUPER_CONCEPT), (GraduateStudent, **MasterStudent,** SUB_CONCEPT), (GraduateStudent, **PhDStudent,** SUB_CONCEPT), (GraduateStudent, **GraduateStudent,** EQUIVALENT)}
PI 2278	PI 2378	$Q_{2278-2378}$ = {(GraduateStudent, **Student,** SUPER_CONCEPT), (GraduateStudent, **Course,** PART), (GraduateStudent, **ResearchProject,** PART), (GraduateStudent, **MasterStudent,** SUB_CONCEPT), (GraduateStudent, **PhDStudent,** SUB_CONCEPT), (GraduateStudent, **GraduateStudent,** EQUIVALENT), (Workshop, **Meeting,** CLOSE), (Workshop, **Conference,** CLOSE), (Workshop, **Event,** SUPER_CONCEPT), (Workshop, **Lecture,** CLOSE)}
PI 2378	PI 2478	$Q_{2378-2478}$ = {(GraduateStudent, **Student,** SUPER_CONCEPT), (GraduateStudent, **Course,** PART), (GraduateStudent, **ResearchProject,** PART), (GraduateStudent, **MasterStudent,** SUB_CONCEPT), (GraduateStudent, **PhDStudent,** SUB_CONCEPT), (GraduateStudent, **GraduateStudent,** EQUIVALENT), (Workshop, **Workshop,** EQUIVALENT), (Workshop, **Conference,** CLOSE), (Workshop, **Event,** SUPER_CONCEPT), (Workshop, **Meeting,** CLOSE)}
PI 2478	PI 2578	(GraduateStudent, **Student,** SUPER_CONCEPT), (GraduateStudent, **Course,** PART), (GraduateStudent, **ResearchProject,** PART), (GraduateStudent, **MasterStudent,** SUB_CONCEPT), (GraduateStudent, **PhDStudent,** SUB_CONCEPT), (GraduateStudent, **GraduateStudent,** EQUIVALENT)

Tabela 4 Resultado da reformulação com poda semântica da consulta submetida a cada ponto da rede

No momento em que a consulta é reformulada, a relevância do ponto de destino é calculada a partir da soma da medida de preservação semântica (PC) e da medida de

enriquecimento semântico (EC) da consulta. Essas medidas foram registradas a cada execução do experimento e a Tabela 5 mostra os resultados obtidos para a medida de relevância para cada ponto na rede utilizando diferentes conjuntos de valores para os pesos das correspondências semânticas.

Pesos {SubConceptOf, SuperConceptOf, CloseTo, PartOf, WholeOf}	**Ponto**	**PC**	**EC**	**Relevância**
{0.8, 0.8, 0.7, 0.3, 0.3}	PI 2278	0.5	0.77	1.27
	PI 2378	0.5	0.66	1.16
	PI 2478	1.0	0.67	1.67
	PI 2578	0.5	0.3	0.8
{0.9, 0.9, 0.5, 0.1, 0.1}	PI 2278	0.5	0.77	1.27
	PI 2378	0.5	0.59	1.09
	PI 2478	1.0	0.61	1.61
	PI 2578	0.5	0.29	0.79
{0.9, 0.9, 0.8, 0.7, 0.7}	PI 2278	0.5	0.87	1.37
	PI 2378	0.5	0.82	1.32
	PI 2478	1.0	0.83	1.83
	PI 2578	0.5	0.41	0.91
{0.3, 0.3, 0.2, 0.1, 0.1}	PI 2278	0.5	0.27	0.77
	PI 2378	0.5	0.22	0.72
	PI 2478	1.0	0.23	1.23
	PI 2578	0.5	0.11	0.61
{0.5, 0.5, 0.4, 0.3, 0.3}	PI 2278	0.5	0.47	0.97
	PI 2378	0.5	0.42	0.92
	PI 2478	1.0	0.43	1.43
	PI 2578	0.5	0.21	0.71
{0.9, 0.7, 0.5, 0.3, 0.1}	PI 2278	0.5	0.70	1.20
	PI 2378	0.5	0.59	1.09
	PI 2478	1.0	0.59	1.59
	PI 2578	0.5	0.31	0.81

Tabela 5 Resultado do cálculo da medida de relevância dos pontos utilizando diferentes conjuntos de pesos

4.3. Análise dos Resultados

A partir dos resultados obtidos pelo experimento mostrados na Tabela 5, foi observado que os pontos com maior enriquecimento semântico (PI 2278), com maior relevância (PI 2478) e com menor relevância (PI 2578) permanecem os mesmos para diferentes conjuntos de pesos.

Esse resultado constante durante o experimento pode ser atribuído ao uso da medida de preservação semântica (PC) da consulta no cálculo da medida de relevância do ponto, já que tal medida visa destacar os pontos que trazem a maior quantidade de conceitos equivalentes à consulta original, como é o caso do ponto PI 2478. Já a constância do ponto

com a menor relevância está relacionada ao modo como o cálculo do enriquecimento semântico é feito: o denominador da equação utilizada é a quantidade de termos originais da consulta, ou seja, o valor do enriquecimento semântico será menor quando o ponto contribui apenas com termos expandidos a partir de poucos conceitos presentes da consulta original. Esse é o caso do ponto PI 2578, que contribui para o enriquecimento da consulta, apenas os termos expandidos a partir de um conceito (GraduateStudent) da consulta original, enquanto os outros pontos contribuem com termos expandidos a partir de todos os termos da consulta original.

De forma geral, os diferentes conjuntos de pesos para as correspondências semânticas não provocaram diferentes impactos no cálculo da medida de relevância de um ponto para o cenário de pontos proposto. Contudo, vale salientar que quando os pesos assumem valores baixos dentro do range permitido, os valores da relevância dos pontos ficam mais próximos, como é o caso dos pontos PI 2578 e PI 2378 para o conjunto de pesos {0.3, 0.3, 0.2, 0.1}. Esse fato pode dificultar a determinação de um valor para o *threshold* da relevância de um ponto na abordagem *SemRouting*.

4.4. Considerações Finais

Nesse capítulo, os experimentos realizados para a análise dos efeitos dos valores dos pesos associados às correspondências semânticas geradas pelo *SemMatcher* foram descritos e seus resultados analisados. O próximo capítulo trará uma discussão geral sobre o os assuntos que foram abordados durante esse projeto.

5. CONCLUSÃO

Este trabalho apresentou uma análise sobre a ferramenta de *matching* semântico de ontologias, o *SemMatcher*. Inicialmente, foram introduzidos conceitos e definições sobre a operação de *matching* de esquemas, bem como um breve estudo acerca de ferramentas *matcher*s comumente citadas na literatura sobre o assunto. Em seguida, uma descrição do ambiente em que a ferramenta *SemMatcher* é utilizada foi provida, com a descrição do sistema SPEED e do processo de roteamento e reformulação da consulta submetida pelo usuário do sistema em um dos pontos da rede. Finalmente, foi apresentado o experimento realizado com o *SemMatcher* e uma análise dos resultados obtidos.

5.1. Contribuições

A principal contribuição desse trabalho é o estudo feito acerca dos efeitos de se utilizar diferentes conjuntos de valores para os pesos associados às correspondências semânticas entre termos geradas pelo *SemMatcher*, dentro do contexto de roteamento da consulta. Ao final da realização do experimento, foi possível perceber que os pontos com maior e menor valor de relevância para a consulta submetida permaneceram os mesmos para todos os conjuntos de pesos testados, dado que esses conjuntos de valores obedeciam o intervalo [0.1, 0.9] e a relação especificada pela Equação 5.

Além disso, o trabalho também contribuiu com a criação de uma API para a ferramenta *SemMatcher*, de modo que a mesma possa ser utilizada mais facilmente em outros projetos.

5.2. Dificuldades Encontradas

A maior dificuldade encontrada durante o desenvolvimento desse trabalho foi o de encontrar experimentos similares com aquele descrito pelo projeto, visto que as referências literárias que descrevem ferramentas de *matching* semântico de esquemas não descrevem como os valores para os pesos utilizados foram encontrados.

5.3. Trabalhos Futuros

Considerando que esse projeto foi focado apenas da análise do *SemMatcher* dentro do contexto de roteamento e reformulação da consulta. A análise da ferramenta no momento de formação de *clusters* da rede poderá ser realizada no futuro com o objetivo de prover um estudo mais completo sobre a atuação do *SemMatcher* dentro do sistema SPEED.

REFERÊNCIA BIBLIOGRÁFICA

BECHHOFER, S. et al. OWL Web Ontology Language. **W3C Recommendation**, 10 February 2004. Disponivel em: <http://www.w3.org/TR/owl-ref/>. Acesso em: 04 January 2015.

BERNSTEIN , P.; MADHAVAN , J.; RAHM , E. **Generic Schema Matching, Ten Years Later**. VLDB Endowment. [S.l.]: [s.n.]. 2011. p. 695-701.

CASTANO, S.; FERRARA, A.; MONTANELLI, S. Matching ontologies in open networked systems: Techniques and applications. **Journal on Data Semantics**, Berlin, v. 5, p. 25-63, 2006.

CASTANO, S.; FERRARA, A.; MONTANELLI, S. **The HMatch 2.0 Suite for Ontology Matchmaking**. Università degli Studi di Milano. Milano. 2007.

CYSNEIROS, N. C. SemMatcher. **GitHub**, 23 February 2015. Disponivel em: <https://github.com/nicysneiros/SemMatcher>. Acesso em: 23 February 2015.

FREIRE, C. A. **Uma Abordagem para Roteamento de Consultas em PDMS baseada em Aspectos Semânticos e de Qualidade**. Tese (Doutorado) - Universidade Federal de Pernambuco. Recife. 2014.

GRUBER, T. Ontology. In: LIU, L.; ÖZSU, T. **Encyclopedia of Database Systems**. 1st Edition. ed. [S.l.]: Springer Publishing Company, v. 1, 2009. Acesso em: 04 January 2015.

MADHAVAN , J.; BERNSTEIN , P.; RAHM , E. **Generic Schema Matching with Cupid**. Microsoft Corporation. Redmond. 2001.

MASSMANN, S. et al. **Evolution of the COMA Match System**. The ISWC Workshop. 2011. p. 49-59.

NOY, N. F.; MUSEN, M. **Algorithm and Tool for Automated Ontology Merging and Alignment**. 17th National Conference on Artificial Intelligence. [S.l.]: AAAI. 2000.

PEREIRA, T. P. A. **Mapeamento Semântico de Ontologias no SPEED**. Dissertação (Graduação) - Universidade Federal de Pernambuco. Recife. 2008.

PIRES, C. E. **Um sistema de Gerenciamento Dados com Conectividade Baseada em Semântica**. Qualificação (Doutorado) - Universidade Federal de Pernambuco. Recife. 2007.

RAHM, E.; BERNSTEIN, P. A survey of approaches to automatic schema matching Erhard. **The VLDB Journal**, v. 10, n. 4, p. 334-350, November 2001.

SHVAIKO, P.; EUZENAT, J. **Ten challenges for ontology matching**. 7th International Conference on Ontologies, Data Bases, and Applications of Semantics. Monterey: Springer. 2008. p. 1163-1181.

SOUZA, D. et al. **Using Semantics to Enhance Query Reformulation in Dynamic Environments**. the 13th East European Conference on Advances in Databases and Information Systems. Latvia: Riga. 2009.

APÊNDICE A - DOCUMENTAÇÃO DA CLASSE FACHADA (FACADE)

speed.ontologymatcher.facade

Class Facade

java.lang.Object
    speed.ontologymatcher.facade.Facade

```
public class Facade
extends java.lang.Object
```

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method and Description
double	**calculateSSM**(java.lang.String fileCLOi, java.lang.String fileCLOj, java.lang.String fileCMO, **ESSMFunction** function, java.lang.String bestAlignmentsFile) Método que calcula a medida de similaridade semântica entre duas ontologias.
java.util.ArrayList<**Alignment**>	**executeSemanticMatching**(java.lang.String fileCLOi, java.lang.String fileCLOj, java.lang.String fileCMO) Método que executa o matching semântico entre a ontologia em CLOi e CLOj
java.util.ArrayList<**Alignment**>	**executeSemanticMatchingNormalized**(java.lang.String fileCLOi, java.lang.String fileCLOj, java.lang.String fileCMO) Método que executa o matching semântico considerando ontologias normalizadas entre a ontologia CLOi e CLOj
void	**generateAcoAlignments**(java.lang.String fileCLOi, java.lang.String fileCLOj, java.lang.String fileCMO)
java.util.Map<**WeightParam**,java.lang.Double>	**getDefaultWeight**() Métode que retorna os valores default para os pesos Equivalent: 1.0 SubConcept: 0.8 SuperConcept: 0.8 Close: 0.7 Part: 0.3 Whole: 0.3 Disjoint: 0.0
static **Facade**	**getInstance**()
void	**setWeights**(java.util.Map<**WeightParam**,java.lang.Double> weights) Método para configurar o conjunto de pesos que deve ser usado pela ferramenta

Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

ANEXO A - ONTOLOGIA DE DOMÍNIO