UNIVERSIDADE ESTADUAL PAULISTA
"Júlio de Mesquita Filho"
Pós-Graduação em Ciência da Computação

Rodrigo Cesar Antonialli

Framework para Integração Semântica de Dados Geoespaciais: Integração de Dados Geológicos

UNESP

Rio Claro - 2015

Rodrigo Cesar Antonialli

Framework para Integração de Dados Geoespaciais: Integração de Dados Geológicos

Orientador: Prof. Dr. Ivan Rizzo Guilherme

Dissertação de Mestrado elaborada junto ao Programa de Pós-Graduação em Ciência da Computação – Área de Concentração em Sistemas Inteligentes, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

UNESP

Rio Claro - 2015

Rodrigo Cesar Antonialli

Framework para Integração de Dados Geoespaciais: Integração de Dados Geológicos

Dissertação de Mestrado elaborada junto ao Programa de Pós-Graduação em Ciência da Computação – Área de Concentração em Sistemas Inteligentes, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Prof. Dr. Ivan Rizzo Guilherme

UNESP – Universidade Estadual Paulista Júlio de Mesquita Filho - Rio Claro

Orientador

Profa. Dra. Mara Abel

UFRGS – Universidade Federal do Rio Grande do Sul

Prof. Dr. Daniel Carlos Guimarães Pedronette

UNESP – Universidade Estadual Paulista Júlio de Mesquita Filho - Rio Claro

UNESP

Rio Claro - 2015

“We can only see a short distance ahead, but we can see plenty there that needs to be done.”

– Alan Turing

AGRADECIMENTOS

Agradeço inicialmente a toda minha família, em espacial e principalmente à minha esposa Anne Caroline, que teve todo amor e compreensão, possíveis e impossíveis, nas mais variadas experiências vividas durante o desenvolvimento deste trabalho. Sem sua companhia, amor e apoio este trabalho jamais seria realizado.

Agradeço ao meu orientador, Ivan Rizzo Guilherme, pela oportunidade que me foi dada para realizar este trabalho, e principalmente pelas diversas discussões enriquecedoras que tivemos a respeito dos temas da pesquisa e outros muito além.

À todos os geólogos e geofísicos, em especial ao Iata A. de Souza por suportar pacientemente infinitas horas de questionamentos, à Norberto Morales, Yociteru Hasui, Adilson V. Soares Jr., Maria Gabriela C. Vincentelli e Sérgio Caceres, assim como todos os participantes do Projeto Neotectônica que também tiveram a grandeza de compartilhar seu imenso conhecimento a respeito destas ciências que tanto fascinam. Os anos de aprendizado foram inestimáveis e fundamentais para muito além da execução desse trabalho.

Aos amigos e companheiros de trabalho Victor D. Mattos, Vinícius A. P. de Souza e Bruno Vedoveto, por dividirem muitas horas e linhas de código de uma experiência ótima no desenvolvimento do Sistema NEOTEC.

À todos os amigos e colegas com os quais tive oportunidade de conviver no tempo em que estive na graduação e no mestrado.

Agradeço também ao UNESPetro e à Rede de Geotectônica da Petrobrás, representados nas figuras de Dimas D. Brito e Gilmar V. Bueno, que forneceram toda a infraestrutura necessária para o desenvolvimento do Projeto Neotectônica e consequentemente para elaboração deste trabalho.

Finalmente, apesar de parecer estranho a alguns, gostaria de deixar registrado meu agradecimento à Meg, minha fiel companheira canina, que apesar de sequer poder ler ou compreender uma única palavra, nunca me abandou em nenhuma madrugada de trabalho e me ensinou o significado da amizade e amor incondicional.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ALGORITMOS

LISTA DE SIGLAS

HTTP – Hypertext Transfer Protocol IDE

OGC – Open Geospatial Consortium

OWL – Web Ontology Language

PNG – Portable Network Graphics

RDF – Resource Description Framework

SIG – Sistemas de Informação Geográfica

SLD – Styled Layer Descriptor

SPARQL – SPARQL Protocol And RDF Query Language

SWRL – Semantic Web Rule Language

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

XML – eXtensible Markup Language

WFS – Web Feature Service

WKT – Well Known Text

WMS – Web Map Service

WPS – Web Processing Service

W3C – World Wide Web Consortium

RESUMO

A necessidade de integrar dados é constante nos estudos e trabalhos na área de Geologia, parte das Geociências responsável pelo estudo de diversas características da Terra. Esta necessidade surge do fato de que os dados são, geralmente, produzidos por diferentes organizações e/ou grupos de pesquisas, que podem estudar áreas extensas e/ou relacionarem, nestes estudos, diferentes disciplinas associadas à Geologia. Esta descentralização na produção de informação faz com que os dados sejam descritos com diferentes esquemas e diferentes interpretações de conceitos a eles relacionados.

O volume de dados que pode ser gerado nos estudos de Geologia traz a necessidade de automatizar processo de integração. Entretanto, abordagens tradicionais baseadas em comparações sintáticas dos esquemas de dados podem não ser suficientes. Os dados geológicos possuem características complexas que dependem de aspectos conceituais para serem analisados. Portanto, a integração de dados geológicos requer uma abordagem em um nível mais completo de representação da informação, que considere os conceitos relacionados aos dados.

A partir do contexto apresentado, o *Framework para Integração de Dados Geoespaciais* proposto neste trabalho visa integrar dados geológicos com a utilização de ontologias, considerando como escopo fundamental as características geoespacias destes dados. Porém, as ontologias que descrevem conceitos relacionados à Geologia podem ser definidas com diferentes interpretações da relação entre estes conceitos. Para superar esta heterogeneidade semântica, o *framework* proposto utiliza o processo de alinhamento de ontologias.

O *framework* também aborda aspectos relacionados à disponibilidade dos dados geológicos, que podem ser públicos, de livre acesso e compartilhamento ou restritos, com acesso controlado, principalmente quando dizem respeito a atividades estratégicas, como a exploração de recursos minerais e energéticos.

No processo de integração proposto no *framework*, uma análise semântica dos dados é executada e, com base em regras, identifica casos de duplicação, conflitos ou combinações de dados geológicos.

ABSTRACT

Data integration is a constant need in researches and work in Geology, part of Geosciences responsible for studying several characteristics of Earth. This need comes from the fact that data are usually produced by different organizations and/or research groups, which may study large areas and/or relate different subjects associated with Geology within these studies. The decentralized information production cause the data to be described with different *schemas* and different interpretations of the concepts related to them.

The volume of data that can be generated in Geological studies brings the need to automate the integration process. However, traditional approaches based on syntactic comparison of data schema may not be enough to perform the task. Geological data have complex characteristics which rely on conceptual aspects to be analyzed. Thus, geological data integration a more complete level of information representation approach, that considers the concepts related to the data.

Within the presented context, the *Framework* for Geospatial Semantic Integration proposed in this work aims to integrate geological data based on ontologies, and to consider the fundamental geospatial characteristics of these data. However, the ontologies that describe concepts related to Geology may be defined with different interpretations of these concepts relations. To overcome this semantic heterogeneity, the proposed *framework* uses the ontology alignment process.

The *framework* also considers geological data availability aspects, which may vary as public, with free access and sharing possibilities or as restricted, with controlled access, mainly when the data is about strategic activities, as mineral and energetic resources exploration.

In the integration process proposed in the *framework*, a semantic analysis of the data is executed; based on rules, it identifies cases of duplications, conflicts or combinations of geological data.

1 Introdução

Nos estudos e trabalhos desenvolvidos na área de Geologia, parte das Geociências responsável pelo estudo de diversas características da Terra, é comum a necessidade de integrar informações de diferentes fontes. Essa necessidade surge do fato de que a produção de informações é geralmente dividida entre várias organizações e/ou grupos de pesquisa. Na maioria dos casos, essa divisão é motivada pela extensão territorial da área estudada, e/ou das diferentes disciplinas envolvidas nesses estudos, o que requer a participação de diversos profissionais de áreas relacionadas à Geologia.

Em razão da descentralização na produção das informações, diferentes fontes de dados geológicos podem apresentar descrições duplicadas ou complementares de uma determinada região ou área de estudo. Além disso, cada fonte de dados pode utilizar um esquema diferente, isto é, organizar as nomenclaturas e relações dos dados de maneira distinta. Mesmo com a adoção de padrões pré-estabelecidos pela indústria para interoperabilidade de dados, algumas diferenças podem ainda permanecer, pois cada padrão é construído sob uma determinada visão conceitual, geralmente associada ao domínio da informação que os padrões descrevem (WERLANG, 2015). Desta forma, para integrar as informações providas por estas diferentes fontes de dados, uma correspondência entre os esquemas utilizados deve ser estabelecida. O objetivo dessa correspondência é explicitar equivalências entre a organização dos termos e seus relacionamentos utilizados em cada esquema. Por meio dessas equivalências é possível uniformizar consultas, processos e visualização dos conjuntos de dados provenientes dessas diferentes fontes.

A tarefa de integração, isto é, a geração de correspondências entre os esquemas, geralmente é realizada por sistemas computacionais em razão da complexidade e do volume de dados a ser integrado. Tradicionalmente, muitos sistemas que realizam a integração implementam técnicas que consideram apenas o nível sintático da informação para automatizar a comparação de esquemas (MELNIK; GARCIA-MOLINA; RAHM, 2002) (RAHM; BERNSTEIN, 2001). Neste caso, as equivalências e divergências são determinadas apenas com a comparação textual dos termos que compõem os esquemas e/ou metadados. As diferenças de formato de representação e o

armazenamento dos dados são tratados por processos de transformação depois de ser estabelecida a relação entre esquemas.

Entretanto, em vários cenários, apenas a comparação sintática pode não ser suficiente para determinar todas as equivalências. Por exemplo, ao integrar duas bases de dados geológicos, uma tabela denominada "Estruturas" em uma das bases pode conter informações equivalentes àquelas armazenadas em uma tabela denominada "Falhas" em outra base, já que o termo Estrutura é mais abrangente que o termo Falha, isto é, Falha é um tipo de Estrutura. Assim, as duas tabelas com nomes diferentes poderiam armazenar informações a respeito de falhas geológicas dependendo dos contextos em que estão inseridas. O trabalho de Werlang (2015) também mostra que as diferenças existentes entre padrões de interoperabilidade de dados também precisa de um tratamento mais elaborado do que a simples comparação sintática de termos.

Para solucionar esse e outros problemas decorrentes da comparação sintática, a comparação de modelos deve ser realizada em um nível mais completo, o nível semântico da informação. Neste nível, além da comparação textual é possível utilizar a definição conceitual e os relacionamentos dos conceitos que compõem os esquemas ou metadados para definir equivalências.

Diversos trabalhos, discutidos nos Capítulos 2 e 3, mostram a comparação de esquemas utilizando aspectos semânticos descritos em ontologias. As ontologias são uma especificação explícita de uma conceitualização, isto é, uma descrição formal de conceitos e seus relacionamentos (GRUBER, 1993). Esta abordagem de comparação que utiliza ontologias requer um processo de anotação semântica, ou seja, um processo que realiza a associação dos termos que compõem os esquemas e metadados, de uma ou mais fontes de dados, a conceitos de uma ontologia. Esse processo pode resolver grande parte do problema de integração, pois, uma vez anotados, os dados poderão ser manipulados com um mesmo vocabulário e também ser interpretados de modo que seu significado seja levado em consideração.

Porém, devido às características dos estudos e trabalhos na área de geologia, assim como ocorre em vários outros domínios, diferentes conjuntos de dados podem ser representados por diferentes ontologias. Essas diferentes ontologias, que podem ser referentes a um mesmo domínio ou a domínios complementares, são construídas com

diferentes interpretações dos conceitos e seus relacionamentos. Além disso, ainda há muitos casos em que os dados não possuem associação com ontologias, e nem sequer metadados, mas também precisam ser integrados.

Neste cenário, torna-se necessário definir formas de solucionar dois problemas: integrar diferentes ontologias; e integrar os dados não associados a ontologias e/ou metadados. Para solucionar o segundo problema, podem ser estabelecidas formas para transformar os esquemas dos dados em ontologias simplificadas. Deste modo, a integração resume-se ao primeiro problema da integração de diferentes ontologias.

A partir dessa transformação de esquemas em ontologias, a solução final para integração de diferentes ontologias requer o processo de alinhamento de ontologias. O processo de alinhamento busca superar a heterogeneidade semântica entre diferentes fontes de informação associadas a diferentes ontologias, estabelecendo correspondências entre os conceitos das ontologias (EUZENAT; SHVAIKO, 2007).

Um outro importante aspecto a ser considerado na integração semântica é que o resultado do processo de integração dos dados pode ser disponibilizado em repositórios semânticos. Neste contexto, é importante identificar a disponibilidade dos dados integrados, que podem ser públicos, de livre acesso e compartilhamento, ou restritos, com acesso controlado. Como conseqüência, as consultas aos resultados podem ocorrer de forma local ou distribuída.

Com base nessas análises, o objetivo deste trabalho é propor um *framework* para integração semântica de dados geoespaciais, com ênfase nos dados geológicos, que utilize o alinhamento de ontologias como principal ferramenta de integração.

O *framework* proposto neste trabalho apresenta as funcionalidades necessárias para reconhecer diferentes tipos de fontes de dados. A partir do acesso a estas fontes de dados, é realizada a transformação do esquema de dados utilizado em cada fonte para um documento RDF. Com algumas operações de enriquecimento deste documento RDF, este passa a ser interpretado como uma ontologia. Uma das particularidades deste *framework* é o uso do alinhamento de ontologias, para estabelecer uma correspondência entre os termos utilizados no esquema de dados e os conceitos de ontologias de referência que são mantidas em um repositório semântico. Após a determinação do

alinhamento são descritas as instâncias de dados com o vocabulário das ontologias de referência.

O *framework* também aborda aspectos relacionados à disponibilidade dos dados geológicos, que podem ser públicos, de livre acesso e compartilhamento ou restritos, com acesso controlado, principalmente quando dizem respeito a atividades estratégicas, como a exploração de recursos minerais e energéticos. Neste contexto, para o acesso a dados públicos, foi definida uma estratégia de consulta distribuída e para dados restritos, foi definida uma estratégia de armazenamento dos dados em um repositório semântico.

Quando a estratégia de armazenamento é adotada, é realizado também um processo de análise semântica para verificar inconsistências semânticas nas descrições e no relacionamento dos dados. No caso dos relacionamentos, o processo procura identificar casos de duplicações e/ou conflitos, e possíveis combinações que podem ser estabelecidas. Esta identificação é feita por meio de regras codificadas junto às ontologias de referência, o que permite que a análise semântica seja facilmente modificada de acordo com as necessidades do usuário.

Os aspectos conceituais relacionados a ontologias e alinhamento de ontologias são apresentados no Capítulo 2. Estes aspectos auxiliam no entendimento destes conceitos e também da integração de dados baseada em ontologias. No Capítulo 3 são apresentados detalhes do contexto do problema a ser abordado por este trabalho, assim como alguns aspectos conceituais relacionados a este contexto.

A solução de integração de dados geológicos baseada em ontologias é apresentada no Capítulo 4 na forma de um *Framework* para Integração de Dados Geoespaciais. Nesta apresentação, os principais componentes da arquitetura do *framework* são caracterizados junto aos aspectos pertinentes às suas concepções.

A partir da definição teórica apresentada no Capítulo 4, um protótipo do *framework* é apresentado no Capítulo 5, com a finalidade de validar sua implementação prática e discutir os aspectos mais pertinentes ao nível de implementação do processo de integração semântica. A implementação do protótipo do *framework* permitiu ainda a realização de alguns estudos de caso de integração de dados geológicos, apresentados no Capítulo 6.

Finalmente, no Capítulo 7, são apresentadas as considerações a respeito do trabalho desenvolvido, que envolvem a identificação das principais contribuições do trabalho e as questões a serem investigadas em trabalhos futuros.

2 Aspectos Conceituais

Neste capitulo são apresentados e discutidos os principais conceitos necessários para o entendimento e a posterior contextualização do problema de integração semântica de dados geológicos. Os conceitos de ontologia e alinhamento de ontologias são definidos em detalhes por serem os principais conceitos utilizados no processo de integração de dados.

2.1 Ontologias

Uma ontologia, conforme definido por Gruber (1993), é uma especificação explícita de uma conceitualização, isto é, uma descrição formal de conceitos e seus relacionamentos.

As ontologias são classificadas de acordo com o os tipos de conceitos que especificam. Os tipos de ontologias podem ser:

- **Ontologias de Representação ou Meta-Ontologias**, que capturam as primitivas de representação utilizadas para formalizar o conhecimento em um dado sistema (BECK; PINTO, 2002).
- **Ontologias Gerais**, que classificam as diferentes categorias de entidades existentes no mundo. Noções muito gerais, que são independentes de um problema ou domínio, são representadas nestas ontologias (BECK; PINTO, 2002).
- **Ontologias de Domínio**, que são ontologias com conhecimento mais específico. O conhecimento representado neste tipo de ontologias é específico de um determinado domínio. Estas ontologias contêm vocabulários sobre conceitos de um domínio e suas relações ou sobre teorias que governam o domínio (BECK; PINTO, 2002).
- **Ontologias de Aplicação**, que descrevem o conhecimento que depende de um domínio particular e uma tarefa a ser realizada. Relacionam conceitos que

descrevem um domínio com conceitos que são parte da descrição de métodos de resolução de um problema. Estas ontologias explicitam o papel realizado por conceitos de um domínio em um determinado método de solução de problema (BECK; PINTO, 2002).

O tipo mais importante de ontologia a ser considerado neste trabalho são as ontologias de domínio. Um exemplo de ontologia de domínio relacionada à geologia é a *Ontology of Fractures* (ZHONG; AYDINA; MCGUINNESS, 2009), que descreve conceitos relacionados a falhas geológicas e mecanismos de deformação, que atuam na formação de falhas. Na Figura 1, é apresentado um trecho desta ontologia, exibindo alguns conceitos pertinentes ao domínio.

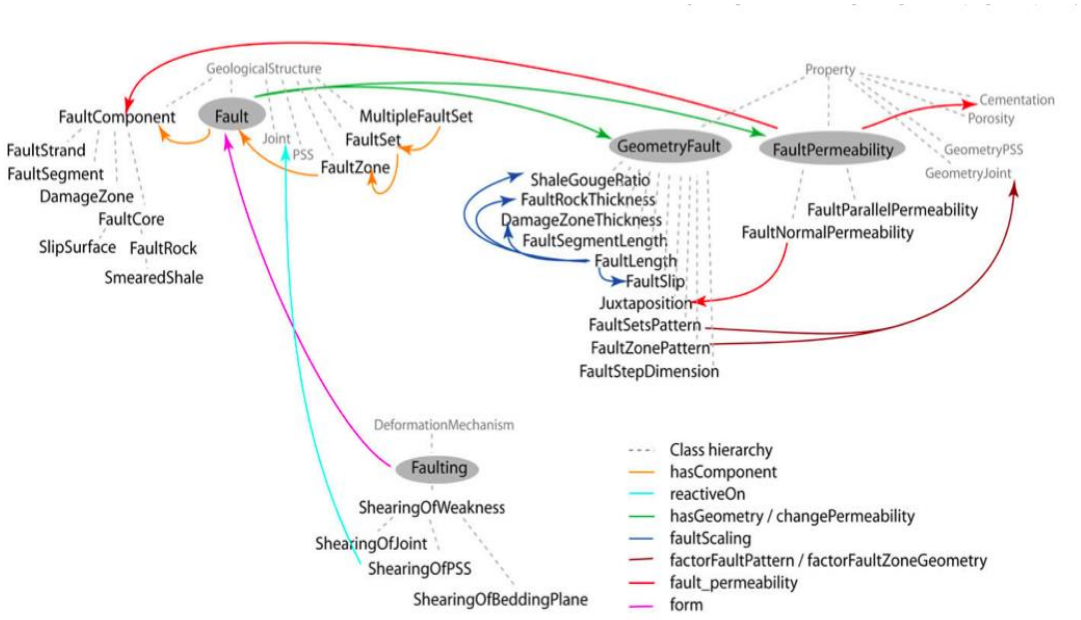

Figura 1: Fragmento de uma ontologia de domínio geológico.

Fonte: Zhong, Aydina e McGuinness (2009).

Além da definição inicialmente apresentada, Berners-Lee, Hendler e Lassila (2001) afirmam que uma ontologia é um documento que define formalmente o relacionamento entre termos. Além disso, apontam que ontologias são um dos pilares da Web Semântica, pois neste contexto as ontologias têm como finalidade descrever

semanticamente os recursos da Web, de forma que as máquinas possam processar esses recursos de uma forma mais significativa.

A Web Semântica é uma extensão da Web atual, em que existe uma estrutura para agregar significado ao conteúdo de páginas, fazendo com que a Web seja dirigida a dados, e não apenas às páginas. Esta estrutura permite com que computadores e pessoas possam trabalhar melhor em cooperação (BERNERS-LEE et al., 2001). Assim, para entender a representação computacional de ontologias e as tecnologias utilizadas para representá-las é preciso primeiro conhecer a estrutura e os conceitos da Web Semântica. Esta estrutura apresenta-se em camadas, e cada camada está associada a uma tecnologia e ao nível de representação de dados e informações que esta tecnologia permite. A estrutura segue a evolução do nível sintático para o nível semântico de representação da informação na Web. Esta estruturação facilita a adoção das tecnologias de maneira gradativa por parte dos desenvolvedores de sistemas Web. Na Figura 2 é ilustrado um esquema com a estrutura da Web Semântica[1].

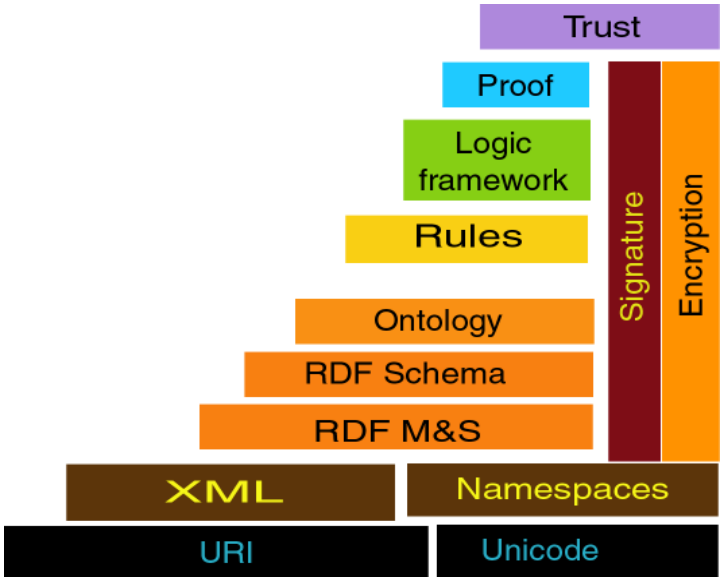

Figura 2: Estrutura das camadas que compõem a Web Semântica.

Fonte: Berners-Lee (2003).

[1] http://www.w3.org/2003/Talks/0922-rsoc-tbl/slide30-0.html

A estrutura ilustrada na Figura 2 apresenta as tecnologias fundamentais para compartilhamento e interoperabilidade de dados na Web. Segundo esta estrutura, qualquer recurso deve ser sempre identificado por um URI (*Uniform Resource Identifier*), e a codificação padrão para conteúdos textuais deve ser o *Unicode*[2], o que permite a interoperabilidade entre textos escritos em diferentes idiomas.

Para superar problemas provenientes da organização sintática das informações trocadas entre sistemas por meio da Web, utiliza-se o XML (*eXtensible Markup Language*). O XML é uma linguagem de marcação baseada em *tags* para escrever documentos estruturados com um vocabulário que pode ser definido pelo usuário e referenciado com o uso de *Namespaces* (ANTONIOU; HARMELEN, 2008).

A interoperabilidade alcançada pelo uso de XML e *Namespaces* atinge apenas o nível sintático da informação, e permite a padronização do formato de mensagens trocadas entre sistemas. Para elevar esta interoperabilidade ao nível da informação propriamente dita, apresenta-se o RDF (*Resource Description Framework*), um *framework* utilizado para representar relacionamentos entre recursos da Web. O RDF fornece uma estrutura comum para expressar descrições e relacionamentos e para que informações possam ser trocadas entre aplicações sem perda de sentido (MANOLA, 2013).

A ideia básica do *framework* RDF é identificar recursos por um URI, e descrever estes recursos de forma simples em termos de suas propriedades e dos valores dessas propriedades. Desta forma, um modelo RDF permite representar o relacionamento entre recursos por meio de estruturas compostas por um sujeito (um recurso), um predicado (uma propriedade) e um objeto (valor de uma propriedade ou recurso), formando uma sentença (MANOLA, 2013). Apesar de haver variações de sintaxe e serialização, a organização padrão de um documento RDF segue os padrões definidos pelo XML e o uso de *Namespaces* para referenciar documentos RDF externos, publicados na Web.

[2] http://www.unicode.org/standard/WhatIsUnicode.html

O vocabulário fundamental utilizado pelo RDF é definido pelo *RDF Schema*[3], uma extensão semântica do RDF que fornece mecanismos para descrever grupos de recursos e suas relações (WORLD WIDE WEB CONSORTIUM, 2004). Tais mecanismos envolvem hierarquia de classes, a definição de domínios, e os intervalos dos valores de propriedades.

Apesar destes mecanismos e do vocabulário associado, de acordo com Antoniou e Harmelen (2008), o *RDF Schema* é uma linguagem primitiva e, portanto, é necessária uma linguagem que permita a representação de relacionamentos mais complexos entre entidades. Assim, no próximo nível da hierarquia está o padrão estabelecido pela W3C atualmente como sendo esta linguagem mais completa para representação do conhecimento: a OWL (*Web Ontology Language*) (W3C OWL WORKING GROUP, 2013). A linguagem OWL é considerada como o padrão para construção de ontologias.

A OWL[4] estende o vocabulário fornecido pelo RDF *Schema*, e permite maior expressão nas declarações sobre recursos. Permite também que mecanismos de inferência sejam capazes de gerar conclusões mais elaboradas a respeito das declarações (ANTONIOU; HARMELEN, 2008). Esta capacidade está associada à relação da OWL com a Lógica Descritiva, uma família das linguagens formais de representação do conhecimento.

A proposição de sentenças possíveis de serem construídas com a OWL ainda possuem certas limitações na descrição do conhecimento de um modo geral. Portanto, para complementar esta capacidade são definidas linguagens de regras. Estas regras são definidas para permitir gerar novas relações conceituais aplicadas ao conhecimento explícito em OWL. Apesar de haver outras iniciativas, a principal linguagem de regra, considerada como uma extensão da OWL é o SWRL[5] (*Semantic Web Rule Language*).

Os conceitos representados na Figura 2 a partir da camada de regras (*RULES*), bem como os conceitos de *Signature* e *Encription*, não são pertinentes ao entendimento da representação computacional de ontologias e, portanto, não pertencem ao escopo deste trabalho.

---

[3] http://www.w3.org/2000/01/rdf-schema
[4] http://www.w3.org/2002/07/owl
[5] http://www.w3.org/Submission/SWRL/

Finalmente, para armazenar as ontologias descritas com as tecnologias citadas, utilizam-se Repositórios Semânticos. Estes repositórios são estruturas de dados desenvolvidas especialmente para o armazenamento de informações codificadas no padrão RDF, conhecidas como *triplestore*.

As ontologias geralmente são dividas entre um componente terminológico (*TBox*) e um conjunto de fatos (*ABox*) (GRUBER, 1993). O *TBox* é composto por conceitualizações que são utilizadas para descrever o conjunto de fatos que compõem o *ABox*. As conceitualizações do *TBox* expressam o conhecimento de um modo geral, e os fatos do *ABox* são instâncias, ou ocorrências deste conhecimento. Por exemplo, os conceitos de Falha, Estrutura e Geometria são conceitualizações que podem ser descritas no *TBox*. A Falha de San Andreas, uma estrutura descrita em um afloramento e uma linha que representa uma feição geológica em um mapa podem, respectivamente, ser instâncias desses conceitos presentes no *ABox*.

Apesar de haver variações, a organização mais comum dos repositórios semânticos acompanha a divisão das ontologias descrita acima, na qual o *TBox* é armazenado separado do *ABox*. Esta organização é mais utilizada, pois assim estes conteúdos podem evoluir de maneira distinta, já que as instâncias, que representam fatos, tendem a aumentar muito mais que o conhecimento descrito. Além disso, com esta organização, as conceitualizações podem ser acessadas, ou compartilhadas, de modo independente dos fatos.

Os repositórios semânticos podem variar também quanto ao armazenar apenas o conhecimento e fatos explícitos, ou armazenar também as inferências geradas pelos raciocinadores. No segundo caso, as inferências são atualizadas a cada operação de atualização do repositório.

2.2 Alinhamento de Ontologias

O processo de alinhamento de ontologias permite encontrar as correspondências entre entidades semanticamente relacionadas de duas ou mais ontologias (SHVAIKO; EUZENAT, 2013). Estas correspondências podem ser utilizadas para várias tarefas,

como a união de ontologias, expandindo o conhecimento representado; para obter resposta de consultas a diferentes fontes de informação ou para atingir outros objetivos. Assim, o alinhamento de ontologia pode ser utilizado para superar a heterogeneidade semântica correspondentes às diferenças semânticas entre duas ontologias.

De acordo com Shvaiko e Euzenat (2013), para realizar o alinhamento de ontologias geralmente é preciso executar dois passos: 1) Definir a equivalência entre as entidades que representam uma definição e, 2) interpretar o alinhamento de acordo com a necessidade da aplicação que fará uso deste alinhamento. Estas entidades alinhadas podem ser classes, propriedades ou indivíduos de uma ontologia.

Na Figura 3 é apresentada uma visão geral do processo de alinhamento, que utiliza a nomenclatura O_1 e O_2 para as diferentes ontologias que devem ser alinhadas. Além das ontologias, um alinhamento de referência (A) previamente estabelecido, também pode ser utilizado. O processo pode utilizar recursos externos (*resources*), como dicionários ou ontologias gerais (MASCARDI; LOCORO; ROSSO, 2010), além de permitir o ajuste de parâmetros de configuração (*parameters*). O resultado (A') do algoritmo de alinhamento é o conjunto de correspondências entre as entidades de O_1 e O_2.

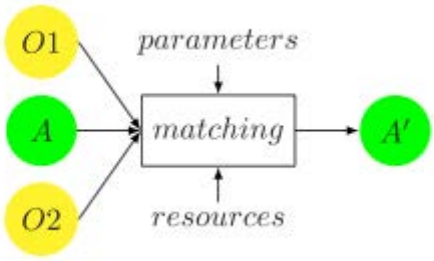

Figura 3: Visão geral de um processo de alinhamento de ontologias.

Fonte: Euzenat e Shvaiko, 2013.

Com relação aos tipos de alinhamento, estes podem variar de acordo com a cardinalidade das equivalências. Um alinhamento pode ter cardinalidade 1:1 (um-para-um), relacionando uma entidade de O_1 a uma entidade de O_2; 1:N (um-para-muitos), relacionando uma entidade de O_1 a uma ou mais entidades de O_2; N:1 (muitos-para-um),

relacionando uma ou mais entidades de O_1 a uma entidade de O_2 ou N:M (muitos-para-muitos), relacionando uma ou mais entidades de O_1 com uma ou mais entidades de O_2. Pode ser ainda que uma entidade de uma ontologia corresponda a um conjunto de relações entre entidades de outra ontologia, um tipo de relacionamento mais complexo, em que um determinado conceito pode ser representado de maneira concreta, ou diluído em descrições mais refinadas.

Ainda segundo Shvaiko e Euzenat (2013), geralmente um alinhamento é representado pela seguinte relação:

$$(id, e_1, e_2, r)$$

Onde:

id é um identificador para a correspondência;

$\boldsymbol{e_1}$ e $\boldsymbol{e_2}$ são entidades das ontologias O_1 e O_2 respectivamente;

r é uma relação entre $\mathbf{e_1}$ e $\mathbf{e_2}$, como por exemplo, equivalência, relação hierárquica de classes (superclasse ou subclasse), disjunção, etc.

A representação do alinhamento também é acompanhada de alguns metadados, como o método utilizado para gerar o alinhamento e o grau de confiabilidade da relação estabelecida entre duas entidades.

Para propor uma abordagem de integração de dados que utilize o alinhamento de ontologias, é preciso entender o funcionamento do processo de um modo geral. As abordagens atuais de sistemas de alinhamento de ontologias seguem um mesmo modelo de estruturação e implementação. Este modelo é um *framework* genérico para o processo e a avaliação de alinhamento de ontologias, definido por Ngo, Bellahsene, Todorov (2013), e apresentado na Figura 4.

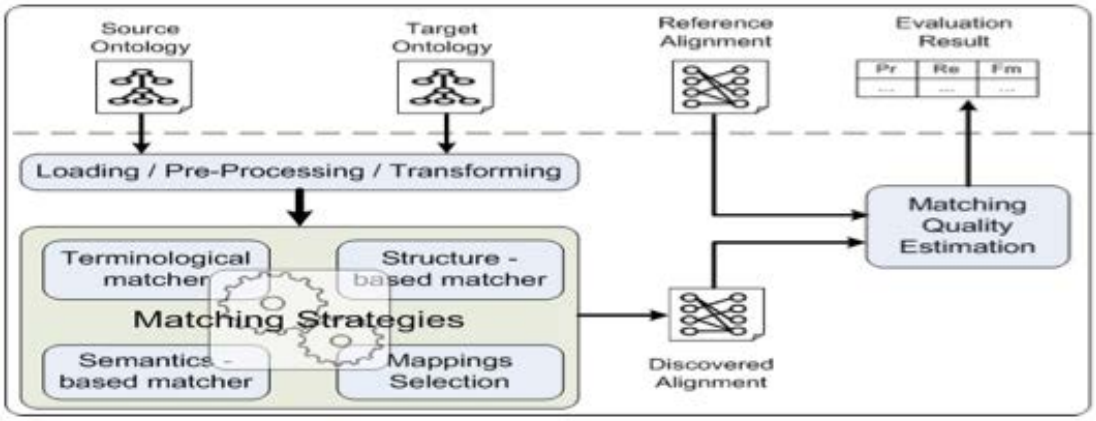

Figura 4: *Framework* genérico de sistemas de alinhamento de ontologias.

Fonte: Ngo; Bellahsene; Todorov, 2013.

Na Figura 4, o processo descrito no *framework* tem início com a definição de ao menos duas ontologias a serem alinhadas, denominadas de *Source Ontology* e *Target Ontology*, respectivamente.

A primeira etapa envolve o pré-processamento das entradas, geralmente para carregar as ontologias em memória e/ou realizar transformações para os formatos necessários à fase seguinte.

O processo de alinhamento é então realizado através da execução das seguintes estratégias: o Casamento Terminológico, que descobre correspondências a partir da comparação de termos; o Casamento Baseado em Estrutura, que compara as estruturas das ontologias; e o Casamento Baseado em Semântica, que analisa as relações semânticas entre os conceitos. Por fim, a etapa de filtragem seleciona os melhores candidatos dentre as correspondências encontradas. Essa última etapa é executada quando há mais de uma correspondência possível para uma entidade.

Para realizar a validação do processo de alinhamento, o alinhamento resultante do processo é comparado com um alinhamento de referência na etapa de estimativa de qualidade. O alinhamento de referência deve ter sido previamente estabelecido, geralmente por especialistas do domínio das ontologias alinhadas. O resultado da comparação é medido por métricas provenientes das áreas de reconhecimento de padrões e recuperação de informação.

2.3 Considerações sobre o Alinhamento de Ontologias

As pesquisas de alinhamento de ontologias vêm sendo largamente desenvolvidas e têm melhorado a qualidade dos resultados do processo nos últimos anos (SHVAIKO; EUZENAT, 2013). Os trabalhos publicados recentemente sobre abordagens de alinhamento de ontologias podem ser divididos de acordo com o método que utilizam: 1) métodos com abordagem determinística; e 2) métodos com abordagem não determinística (NGOMO; LYKO, 2013).

As abordagens determinísticas aplicam algoritmos de busca linear, de forma que todos os casos possíveis são comparados e os resultados ao final de um processo de alinhamento são determinísticos. Neste caso, após a busca por todas as possibilidades, todos os casos de alinhamento de entidades devem fazer parte dos resultados. As abordagens não determinísticas, por sua vez, aplicam algoritmos de busca não lineares, geralmente baseados em algoritmos evolutivos, redes neurais e outras técnicas de Inteligência Artificial.

Ainda não há uma definição absoluta em relação a qual tipo de método deve ser utilizado na implementação de um algoritmo de alinhamento de ontologias. Há trabalhos que apontam vantagens aos métodos determinísticos como Ngomo e Lyko (2013) e trabalhos que apontam vantagens aos métodos não determinísticos, como Ngo, Bellahsene, Todorov (2013). Esta discussão não faz parte do escopo deste trabalho, já que neste caso o foco é a capacidade destes algoritmos de realizar a integração da semântica associada aos dados geológicos geoespaciais. Entretanto, os trabalhos apresentados e discutidos nesta seção seguem abordagens não determinísticas, pois estes têm sido os trabalhos desenvolvidos mais recentemente.

Isele e Bizer (2012) desenvolveram um sistema que utiliza programação genética para o aprendizado das regras que determinem a similaridade de duas entidades representadas em RDF. Este algoritmo de alinhamento foca diretamente nas instâncias de dados para determinar equivalências. As regras geradas definem quais operações

devem ser realizadas com as propriedades de cada par de instâncias para determinar a similaridade entre as mesmas.

A técnica apresentada por Isele e Bizer (2012) mostra como um algoritmo evolutivo pode ser utilizado de forma eficiente para gerar regras de comparação de entidades declaradas em ontologias distintas. Entretanto, a técnica se restringe apenas a etapa do Casamento Terminológico, ou seja, as regras encontradas pelo processo utilizam apenas operações de comparação e transformação de texto. Além disso, esse trabalho aborda apenas o alinhamento de instâncias, não considerando o alinhamento de esquemas, ou seja, dos conceitos que definem as instâncias de dados.

O trabalho desenvolvido por Acampora, Loia e Salerno (2012) utiliza uma abordagem evolutiva híbrida para solucionar o alinhamento de ontologias, e desta vez considera apenas o alinhamento de esquemas, isto é, os conceitos definidos nas ontologias, e não as instâncias de cada conceito. O algoritmo apresentado, denominado de algoritmo memético, é uma combinação de um algoritmo genético com o algoritmo *Hill Climbing* (RUSSEL; NORVIG, 2003). Os resultados do trabalho de Acampora, Loia e Salerno (2012) mostram que a combinação das duas técnicas tem um desempenho muito superior a apenas utilizar o algoritmo genético. Além disso, o trabalho também evolui no sentido de considerar, além do Casamento Terminológico, o Casamento Baseado em Estrutura para determinar similaridade entre as entidades das ontologias.

Outros trabalhos utilizam abordagens não determinísticas, como Bock e Hettenhausen (2012) que apresentam uma solução de alinhamento com a utilização da Otimização por Enxame de Partículas; Djeddi e Khadir (2013) que utilizam uma rede neural artificial para alinhar ontologias; e Liu, Qin e Wang (2013) que optaram pelo uso de SVM (Máquina de Vetores de Suporte) para realizar o alinhamento. Esses trabalhos vêm mostrando que técnicas de Inteligência Artificial podem ser utilizadas para solucionar o problema de alinhamento de ontologias em larga escala e obter bons resultados.

Entretanto, mesmo utilizando técnicas cada vez mais avançadas, tem-se atingido um estágio de aparente estagnação no desenvolvimento de sistemas de alinhamento de ontologias, já que a velocidade de evolução das pesquisas na área tem diminuído

(SHVAIKO; EUZENAT, 2013). Tal diminuição pode ser justificada pelo aumento da complexidade dos problemas a serem resolvidos. Isto ocorre porque, até o momento, as técnicas de alinhamento abrangem com maior frequência as estratégias de Casamento Terminológico e Casamento Baseado em Estrutura. Poucos trabalhos abordam o Casamento Baseado em Semântica (NAGY; VARGAS-VERA; MOTTA, 2006) (JEAN-MARY; SHIRONOSHITA; KABUKA, 2009) (KHETARPAUL; GUPTA; CHAUHAN, 2011), que é mais complexo de ser implementado.

Além da exploração superficial do Casamento Baseado em Semântica, há outros aspectos que exigem uma avaliação mais detalhada para serem abordados. Estes aspectos são apresentados pelo trabalho de Shvaiko e Euzenat (2013) que mostra um quadro do estado da arte de soluções de alinhamento e desafios a serem enfrentados no futuro, apontando alguns aspectos que não foram abordados até o momento, ou que foram explorados apenas por alguns trabalhos. Os desafios destacados são apresentados a seguir.

- **Avaliação do processo de alinhamento em larga escala -** Os testes realizados com ontologias de grande escala, que podem conter milhares ou até milhões de entidades, precisam de um grande esforço para definir um alinhamento de referência. Automatizar o processo, ou parte do processo de obtenção destes alinhamentos seria um grande avanço nesta questão. Além disso, métricas mais precisas e adaptadas para cada tipo de aplicação podem tornar o processo de avaliação melhor.
- **Eficiência das técnicas de alinhamento -** Basicamente envolve o desempenho dos sistemas de alinhamento que, segundo Shvaiko e Euzenat (2013), não têm preocupação com o consumo de recursos. Pensando no tempo de resposta e até mesmo no uso de sistemas de alinhamento de ontologias em dispositivos móveis, melhorias de desempenho poderiam ser mais bem exploradas utilizando técnicas como, por exemplo, a paralelização do processo.
- **Reaproveitamento de conhecimento -** Técnicas de alinhamento poderiam se beneficiar de conhecimento pré-existente, tais como documentos e outros recursos anotados com os conceitos das ontologias sendo alinhadas, e também

dicionários e glossários para determinar a relação entre duas entidades. A utilização de conhecimento existente poderia auxiliar por um lado, mas também causar problemas com correspondências incorretas sendo geradas, devido a erros ou incoerências nas fontes de conhecimento utilizadas. Neste caso, seria necessário estudar maneiras de utilizar este conhecimento da melhor forma possível.

- **Melhorar a seleção, combinação e configuração de estratégias -** Nos sistemas atuais, diferentes estratégias terminológicas, estruturais ou semânticas são utilizadas em conjunto. Os resultados de cada estratégia são agregados para determinar a similaridade de entidades. Para aprimorar a combinação destas estratégias, os sistemas poderiam se adaptar, selecionando os tipos de agregação para cada caso, além de definir as configurações de cada combinador em tempo de execução, também de acordo com um determinado caso de uso.
- **Envolvimento do usuário -** Os usuários podem auxiliar o processo de alinhamento, contribuindo para melhorar a qualidade de seus resultados. Entretanto, as formas de interação do usuário com os sistemas de alinhamento precisam ser estudadas para que a tarefa não seja um ônus para o usuário, principalmente quando ontologias de larga escala estão sendo alinhadas.
- **Explicação dos resultados do processo de alinhamento -** Os usuários precisam entender como um alinhamento foi determinado para que possam editar este alinhamento e fornecer um *feedback* para o sistema. É preciso determinar formas simples, objetivas e precisas para apresentar a explicação do alinhamento para o usuário. O desafio é ainda maior em sistemas que utilizam técnicas de aprendizado de máquina e otimização discreta, ou seja, abordagens não determinísticas, que não produzem uma explicação simples ou simbólica.
- **Alinhamento social e colaborativo -** A interação colaborativa pode ser utilizada por sistemas de alinhamento para obter ou aprimorar resultados, com usuários discutindo e avaliando alinhamentos ou determinando alinhamentos por meio de tarefas corriqueiras. O desafio é lidar com um grande número de alinhamentos, entender o processo utilizado pelos usuários para determinar um alinhamento e tratar casos de usuários maliciosos que poderiam gerar resultados

incorretos. Além disso, faz parte do problema fazer da avaliação ou geração de alinhamentos uma tarefa simples para ser realizada por não especialistas.

- **Gerenciamento dos alinhamentos - infraestrutura e suporte -** Envolve o estudo e desenvolvimento de ferramentas, *frameworks* e/ou plataformas para o armazenamento e compartilhamento de alinhamentos, visando à interoperabilidade entre diversos sistemas.

Apesar de não estarem listados como um tópico para desafio futuro há também problemas a serem resolvidos com tipos específicos de informação, como por exemplo, considerar dados espaciais e temporais para serem alinhados. Este tipo de informação possui características específicas que podem influenciar na determinação de similaridade de duas entidades. Este tema é discutido com mais detalhes nas seções 3.1 a 3.3.

Quanto ao tratamento de dados específicos, o sistema de alinhamento pode ser genérico, e permitir o alinhamento de quaisquer tipos de ontologias, de qualquer domínio, ou ser especializado, e permitir o alinhamento de ontologias de um domínio determinado. Este último geralmente utiliza técnicas e recursos específicos para os dados característicos do domínio considerado.

Por utilizar técnicas e recursos voltados ao domínio em que trabalham, os sistemas especializados acabam por ter um melhor desempenho em suas áreas, porém podem ter problemas se utilizados fora do domínio para os quais foram propostos.

2.4 Integração de Dados baseada em Ontologias

O uso de ontologias na integração de dados é uma abordagem relativamente recente. No contexto da integração de dados geológicos, duas abordagens têm sido desenvolvidas: a disponibilidade de dados não anotados semanticamente, que podem ou não estar associados à metadados, mas que não possuem semântica explicitamente associada e que seja passível de processamento pelas máquinas; e a disponibilidade de

dados anotados semanticamente, ou seja, dados que estão associados a conceitos de ontologias.

No trabalho de Hwang, Nam, Ryu (2012), é apresentado um sistema de informação geológica que utiliza uma ontologia do domínio espaço-temporal para integrar mapas geológicos. Essa ontologia envolve os conceitos de classificação de rochas e idade geológica e foi utilizada para construir um mapa geológico da Coréia, além de definir os conceitos e simbologias necessários para a composição do mapa. Neste trabalho, apenas uma ontologia foi desenvolvida e utilizada, mas no trabalho de LIN; LUDÄSCHER (2003) é apresentado um cenário em que há dados pré-existentes, de diferentes domínios, associados a diferentes ontologias. Essas diferentes ontologias podem ser alinhadas, tornando o processo de integração automatizado mais eficiente.

Existem casos em que os dados a serem integrados não estão associados a uma ontologia. Geralmente, este é o caso da produção de dados por diferentes equipes, em diferentes regiões. Mesmo que estes dados ainda estejam associados à metadados, algumas diferenças ainda podem permanecer, já que os padrões e definições dos metadados também podem variar. Definir uma ontologia comum entre os envolvidos neste tipo de projeto pode levar muito tempo, aumentar o prazo de conclusão do trabalho, e consequentemente aumentar seu custo.

Para lidar com dados não associados a ontologias é comum utilizar a abordagem da anotação semântica, que visa associar os conceitos de uma ontologia com os termos que descrevem os dados (KLIEN, 2008) (MACARIO, 2009).

Neste sentido, para endereçar dados previamente associados a ontologias e dados não anotados, a proposta desenvolvida neste trabalho utiliza uma forma de associar conceitos aos metadados e esquemas de dados, que torne possível a integração de dados sob um único tipo de processo, ou seja, o alinhamento de ontologias, com a finalidade de minimizar o impacto para processamento prévio por parte do usuário. Essa forma de anotação semântica é baseada na transformação do esquema de dados para RDF, e é descrita com mais detalhes no Capítulo 5.

2.5 Conclusão do Capítulo

Neste Capítulo, foram apresentados os diversos aspectos conceituais envolvidos no desenvolvimento deste estudo, tais como: Ontologias e sua representação computacional por meio de RDF e OWL dentro do contexto da Web Semântica; O conceito de Alinhamento de Ontologias e o *framework* geral seguido pelos sistemas que implementam esta técnica, além de considerações pertinentes a quaisquer trabalhos que utilizem um processo de alinhamento de ontologias como solução para determinados problemas.

O estudo destes aspectos conceituais foi necessário para elaborar a proposta de integração de dados geológicos baseada em ontologias, no sentido de que com o conhecimento a respeito destes aspectos, as principais características da solução proposta puderam ser determinadas de maneira concisa, levando em consideração as possibilidades e limitações de cada aspecto conceitual.

3 Contextualização do Problema

Inicialmente, neste capítulo, são apresentados os aspectos gerais do tipo de informação que está sendo considerada como alvo de um processo de integração neste trabalho. Neste contexto, é importante entender alguns conceitos de Geologia e qual o foco de estudo desta ciência.

Posteriormente, são apresentadas as ontologias construídas para representar o conhecimento da Geologia que são pertinentes ao escopo deste trabalho.

3.1 Contextualização dos Dados Geológicos

A Geologia é a ciência que estuda a Terra em sua constituição, estruturação e processos que nela atuaram e atuam. A constituição diz respeito aos minerais e rochas, que são os materiais componentes da Terra. A estruturação diz respeito ao arranjo espacial desses componentes. Os componentes são vistos da escala atômica até a escala terrestre. Os processos que atuaram e atuam são aqueles que originaram a constituição e estruturação. Relacionam-se também com movimentos de massas das profundezas da Terra e da sua casca rígida (placas litosféricas) que ocorreram ao longo do tempo geológico e continuam em operação.

Todas as informações a respeito da constituição, da estruturação e dos processos envolvidos são coletadas e armazenadas para que possam ser processadas por sistemas computacionais no auxílio à interpretação de cenários complexos por parte dos geólogos. Esta interpretação é geralmente realizada a partir de mapas ou modelos geológicos.

Os estudos de geologia que utilizam as informações citadas têm amplo impacto em diversas áreas e atividades, como análises de risco ambiental, exploração de recursos energéticos (óleo e gás) e minerais, chegando até à geologia médica, parte desta ciência que estuda a influência de fatores geológicos ambientais na saúde do homem. Para realizar estes estudos e produzir de resultados nestas áreas, é preciso coletar e analisar

todo este conjunto de dados geológicos. Esta análise geralmente requer a integração dos dados de diferentes fontes e aspectos.

A integração de dados se faz necessária pois a produção de dados geológicos, é descentralizada. Esta característica é influenciada por diversos fatores, tais como a extensão territorial estudada e as diferentes disciplinas envolvidas nestes estudos.

Há ainda fatores históricos e tecnológicos relacionados à integração de dados geológicos, pois grande parte dos dados coletados encontram-se impressos. Quando se realiza levantamentos bibliográficos, é possível encontrar muitos mapas que podem ter sido produzidos em tempos em que a tecnologia para a produção em formato digital não estava disponível ou não era facilmente acessível. Reutilizar estas informações exige a digitalização destes mapas e sua integração para compor uma caracterização regional, por exemplo.

Outra questão relacionada à integração de dados geológicos é o caráter de confidencialidade que pode ser atribuído a estes dados. Devido à natureza dos estudos de Geologia, os dados geológicos estão diretamente associados às atividades de alto impacto econômico e estratégico para o país, tais como a produção de óleo e gás e extração de recursos minerais, fonte de matéria prima para indústrias dos mais variados setores. Esta característica faz com que muitas vezes estudos e projetos de pesquisa sejam realizados em caráter de confidencialidade, ou seja, o acesso aos dados geológicos é restrito apenas aos envolvidos diretamente na sua produção ou integração. Há casos em que este caráter é estabelecido apenas durante a fase de execução de um projeto, mas mesmo assim esta restrição de acesso impacta na manipulação dos dados.

A partir da questão da confidencialidade, a implementação de um sistema de integração deve ser definida com alguns cuidados. Neste caso, a utilização de um repositório local, interno ao sistema de integração, é necessária para garantir o controle de acesso aos dados, mas esta abordagem contrasta com as iniciativas relacionadas à Web Semântica, e ao movimento mais recente conhecido como Linked Data, que geralmente focam na integração de dados de domínio público, disponíveis livremente na Web (BERNERS-LEE, 2006). Considerando essas questões, a proposta apresentada neste trabalho aborda dois tipos de dados ao que se refere à confidencialidade: dados de acesso público e dados de acesso restrito.

Esta questão da confidencialidade que foi discutida impacta em nível de implementação, e faz com que seja preciso definir duas estratégias: uma para consultas distribuídas, geralmente possíveis em repositórios públicos; e outra para consultas locais, que considera um repositório interno. Entretanto, se ambos os tipos de dados, públicos e privados, forem manipulados no sentido de integração, as duas estratégias podem ser mescladas e implementadas em um único ambiente.

3.2 Ontologias no Domínio de Geociências

Uma vez que a abordagem proposta neste trabalho tem como objetivo a integração semântica de dados geológicos, é importante entender a descrição e representação dos conceitos envolvidos no domínio geológico em ontologias. O domínio em questão pode ser muito extenso, portanto, neste trabalho, o foco principal é abordar o aspecto mais geral dos dados geológicos, que é a localização espacial.

A relação de informações geológicas com a localização espacial é uma característica intrínseca a todas as Geociências (JANOWICZ, 2012). Portanto, é importante entender a representação de informações espaciais em ontologias.

De modo geral, um dos principais recursos utilizados nos estudos no domínio de Geociências são os mapas. No âmbito da Computação, os Sistemas de Informações Geográficas (SIG) têm sido desenvolvidos com a finalidade da representação digital de mapas e informações relacionadas (CÂMARA; DAVIS; MONTEIRO, 2001). Os SIG permitem a interação de usuários com mapas digitais e com os dados que compõem ou acompanham estes mapas. No atual cenário de estudo das Geociências, os SIG desempenham um papel fundamental na geração, na análise e no compartilhamento de dados geográficos de diferentes áreas do conhecimento (CÂMARA; DAVIS; MONTEIRO, 2001).

Diante da necessidade do compartilhamento de dados geológicos apresentada na seção 3.1, é possível estabelecer também a necessidade de interoperabilidade entre os SIG. Entretanto, a abordagem tradicional utilizada por SIG para melhorar a interoperabilidade tem sido a associação de metadados aos dados produzidos. O uso de

metadados, apesar de melhorar significativamente a qualidade dos dados, não é suficiente para superar alguns problemas de integração relacionados à semântica e permitir a automatização do processo de integração. Portanto, a interoperabilidade vem sendo endereçada por meio do uso de ontologias para o domínio das Geociências. Estas ontologias vêm aumentando cada vez mais e são fundamentais para abranger a vasta heterogeneidade das informações oriundas das ciências naturais (JANOWICZ, 2012).

Porém, mesmo com o uso de ontologias para representação da informação em nível semântico, as características de distribuição das informações das Geociências também fazem com que, para um determinado domínio, seja possível existir diferentes formas de representação. Como consequência, podem existir diversas ontologias para descrever um mesmo domínio. Assim, é necessário realizar o alinhamento dessas ontologias para que os sistemas sejam capazes de processar as diferentes formas do conhecimento expresso e possam interagir de maneira eficiente e automatizada.

As ontologias que descrevem as representações de dados espaciais são conhecidas como ontologias geoespaciais, e fornecem vocabulário necessário para representar desde coordenadas geográficas até relações topológicas entre diferentes geometrias. Além destas representações, é comum que estas ontologias também representem conceitos relacionados ao tempo, já que estes dois domínios podem estar constantemente relacionados, ou seja, são comuns casos em que a localização geográfica varia com o tempo, caso de qualquer descrição de trajetória.

Com base em todos os aspectos discutidos, identifica-se a necessidade de analisar as características dos dados geoespaciais e temporais para realizar o alinhamento de ontologias geoespaciais, e a partir disso, realizar a integração de dados geológicos em nível semântico.

3.2.1 Alinhamento de Ontologias Geoespaciais

Apenas a etapa de Casamento Terminológico, apresentado na seção 2.2, não é suficiente para determinar a similaridade entre duas instâncias de dados geológicos quando estes são caracterizados como entidades de ontologias. Isto se deve ao fato de

que as informações espaciais precisam ser consideradas para distinguir entidades que sejam dependentes de sua localização para serem definidas (DU et al.; 2013).

Du et al. (2013) apresentam alguns exemplos de informações espaciais representadas em ontologias geoespaciais. Porém, além da distinção de localização, no referido trabalho é identificado outro tipo de relacionamento pertinente à identificação de equivalência de instâncias de dados, definido pela relação "parte de", e associado aos conceitos estudados na Mereologia[6]. Este relacionamento permite definir que uma entidade representada em uma ontologia pode ser parte de outra entidade representada em outra ontologia. Por exemplo, uma zona de falhas pode ser representada como um único indivíduo em uma ontologia e como um conjunto de falhas associadas que compõem a mesma zona em outra ontologia.

A importância do trabalho de Du et al. é mostrar que se apenas as informações textuais, como as *labels* das entidades, forem consideradas para o alinhamento de instâncias, os resultados podem não ser satisfatórios. Ao considerar informações espaciais, os resultados obtidos são muito melhores do que os de outros sistemas que realizam apenas o Casamento Terminológico (DU et al.; 2013).

Entretanto, para considerar informações espaciais no processo de alinhamento de ontologias, é preciso analisar diversas características destas informações, e principalmente as diferentes formas que estes tipos de dados podem ser representados.

Segundo levantamento bibliográfico realizado em Shvaiko e Euzenat (2013), há poucos trabalhos que abordam o tema espaço ou tempo durante um processo de alinhamento. Uma justificativa para isto é a dificuldade para representar informações espaciais e temporais em ontologias. Desta forma, neste trabalho, foram levantadas as principais características de dados geoespaciais e o que deve ser considerado para realizar o alinhamento de informações deste tipo. Este levantamento é apresentado nas próximas seções.

[6] http://plato.stanford.edu/entries/mereology/

3.2.2 Sistemas de Referência Espacial

O primeiro fator a ser considerado quando se trata de informações geoespaciais, ou coordenadas geográficas, é o Sistema de Referência Espacial (SRS - *Spatial Reference System*).

O Sistema de Referência Espacial é formado pela combinação do Sistema de Coordenadas e da Projeção Cartográfica. O Sistema de Coordenadas indica a forma com que um determinado ponto é encontrado na superfície da Terra. A Projeção Cartográfica indica o modo com que o globo terrestre é representado em uma superfície plana, ou seja, em um mapa. Desta forma, a localização espacial de qualquer ponto na superfície da Terra depende diretamente do Sistema de Coordenadas e da Projeção Cartográfica que são utilizados para representar a informação.

O sistema mais utilizado para representar informações de localização é o Sistema de Coordenadas Geográficas[7], que localiza um ponto na superfície da Terra a partir de duas ou três informações: a Latitude, que é o ângulo entre um ponto qualquer e o Equador; a Longitude, que é o ângulo de um ponto qualquer ao longo do Equador; e a Altitude, que representa a altitude do local a partir do nível do mar (GASPAR, 2005).

A combinação de Latitude e Longitude formam as coordenadas de um ponto em um espaço bidimensional. A informação de Altitude complementa a informação da localização e determina um ponto em um espaço tridimensional. Dependendo da projeção cartográfica utilizada, os valores das coordenadas podem variar. A projeção mais comum para representar Coordenadas Geográficas em escala global é a WGS 84[8].

Há ainda outros Sistemas de Coordenadas amplamente utilizados, como o UTM (*Universal Transversa de Mercator*), que utilizam três informações para localizar um ponto no espaço bidimensional, dividindo o globo terrestre em 60 fusos, referenciando um ponto com coordenadas X e Y relativas a cada fuso (desta forma é preciso utilizar as coordenadas X e Y e o fuso para localizar um ponto neste sistema). Além disso, este sistema utiliza uma projeção diferente, denominada *Transversa de Mercator* (GASPAR, 2005).

[7] http://goo.gl/Qy4bU1
[8] http://spatialreference.org/ref/epsg/4326/

A lista de sistemas de coordenadas com as respectivas projeções utilizadas é longa. Esta lista é organizada por algumas instituições, sendo a principal delas a *European Petroleum Survey Group*[9] (EPSG), que associa um código numérico (conhecido como código EPSG) a cada Sistema de Referência Espacial. Este código simplifica a identificação do Sistema de Coordenadas e Projeção Cartográfica que são utilizados para indicar uma localização.

Quando informações espaciais são representadas em ontologias, é preciso definir o Sistema de Referência Espacial utilizado, para que as coordenadas sejam utilizadas de maneira correta. Portanto, para alinhar estas ontologias, e fazer comparações entre as localizações de suas entidades, é preciso identificar o Sistema de Referência Espacial utilizado em cada ontologia e realizar uma transformação entre Sistema de Coordenadas e Projeções Cartográficas quando necessário.

Uma forma de identificar os Sistemas de Referência Espacial em ontologias é utilizar vocabulários comuns que definem estes conceitos. Dois exemplos de vocabulários utilizados para esta finalidade são a ontologia WGS84-POS[10] e a Ontologia de Sistemas de Coordenadas (IGNF) [11]. Ambos possuem definições que tornam possíveis a representação de coordenadas geográficas e a representação de geometrias espaciais associadas a um Sistema de Referência Espacial.

3.2.3 Escala de Representação

Outro fator que deve ser considerado em relação às coordenadas geográficas é a escala de representação, ou seja, a relação entre as medidas reais e do desenho de um mapa. A escala determina a precisão com que uma informação espacial é representada, e este fator precisa ser considerado quando há intenção de realizar integração de dados espaciais. Deste modo, se as entidades de duas ontologias sendo alinhadas forem representadas em escalas diferentes, as comparações entre duas geometrias podem gerar

[9] http://www.epsg.org/
[10] http://www.w3.org/2003/01/geo/wgs84_pos
[11] http://data.ign.fr/def/ignf.html

conclusões incorretas, pois pode haver certo grau de divergência entre as geometrias representadas em escalas diferentes.

Considere o seguinte exemplo: Em uma ontologia, os indivíduos representam dados definidos com base em um mapeamento realizado em uma escala pequena (1:50.000[12]); Em outra ontologia, os indivíduos representam dados definidos com base em um mapeamento realizado em uma escala média (1:250.000 por exemplo). Um mesmo objeto geográfico representado em ambas as ontologias terá sua representação geométrica com uma precisão diferente em cada escala. Esta diferença decorre do nível de detalhe que é possível identificar ao analisar um mapa de cada escala, pois quanto menor a escala, maior o nível de detalhes.

Entretanto, mesmo que esta diferença devido à escala possa ser contornada, esta variação pode ser confundida com o posicionamento incorreto de uma entidade mapeada. É comum ocorrer este tipo de problema quando a informação espacial é obtida a partir de uma fonte limitada, como por exemplo, um mapa impresso.

Possivelmente, a melhor solução para este tipo de problema é a intervenção do usuário no processo de alinhamento. Entretanto, segundo o trabalho de Shvaiko e Euzenat (2013), a interação do usuário com o processo de alinhamento é um dos desafios que ainda precisam ser explorados na área. Esta questão trata-se de um desafio, pois é importante que a necessidade de interação do usuário não impacte negativamente em sua experiência de uso de sistemas de alinhamento e/ou integração de dados.

3.2.4 Geometria

Outro ponto importante a ser considerado durante a integração de dados geoespaciais, é a geometria utilizada para representar uma entidade. Os três tipos básicos de geometria para representar uma entidade em um mapa são: ponto, linha e polígono.

Dependendo do contexto, uma mesma entidade pode ser representada por um ponto em uma ontologia, ou por um polígono em outra ontologia (ou qualquer

[12] Notação de escala cartográfica: Seja 1:50.000, então 1 unidade de medida do mapa representa 50.000 unidades da medida real.

combinação entre os tipos básicos). É preciso estabelecer uma forma de determinar a similaridade espacial destes dados, considerando sua geometria distinta em diferentes ontologias.

Neste caso, o sistema de alinhamento também deve ser capaz de realizar operações espaciais para determinar a relação topológica entre as geometrias que representam as entidades. Com estas operações, é possível determinar regras que indiquem possíveis similaridades. Por exemplo, seja uma entidade de uma ontologia A, representada por um ponto, com *label* igual à de outra entidade de uma ontologia B, representada por um polígono. Uma regra topológica poderia dizer que se o ponto da entidade da ontologia A for próximo ao centro ou centroide do polígono da entidade da ontologia B, então existe uma possibilidade das entidades serem similares.

Alguns vocabulários publicados fornecem conceitos para descrever a representação geométrica de entidades, como por exemplo, o GEOSPARQL[13] e NeoGeo[14], que permitem representar geometrias simples como linhas, pontos e polígonos, além de tipos derivados, como múltiplas linhas, múltiplos polígonos, etc.

3.2.5 Tempo

Uma característica muito comum associada a dados espaciais é o fator tempo. Muitas informações podem variar de acordo com o tempo, modificando, por exemplo, sua localização ou representação geométrica (JANOWICZ, 2012).

Os movimentos de trajetória são exemplos comuns da relação espaço-tempo. A movimentação de um indivíduo é caracterizada em ontologias pelo registro de sua posição espacial em cada período de tempo em que a sua trajetória foi medida. Além disso, pode haver casos em que exista a variação na representação espacial de uma determinada entidade conforme a variação do tempo. Um exemplo disso são os mapeamentos de falhas geológicas em diferentes períodos de tempo geológico.

[13] http://www.opengeospatial.org/standards/geosparql

[14] http://geovocab.org/geometry.html

Suponha que uma falha geológica seja representada na superfície por uma linha. Esta representação pode ser exibida no momento em que a falha se originou. Se em algum momento posterior da história houve alguma perturbação na estrutura e/ou terreno ao redor desta falha, ela pode sofrer deformações. Neste caso, a representação mais atual da falha pode ser diferente do que a representação original, dada em um período de tempo anterior. O exemplo fica mais claro na Figura 5, em que uma estrutura A é apresentada sem alteração. Em seguida, as estruturas B, C e D aparecem deformando a estrutura original, formando A_1 e A_2. Pode-se verificar por esta deformação, que as estruturas B, C e D são mais recentes que a primeira, ou seja, formaram-se posteriormente.

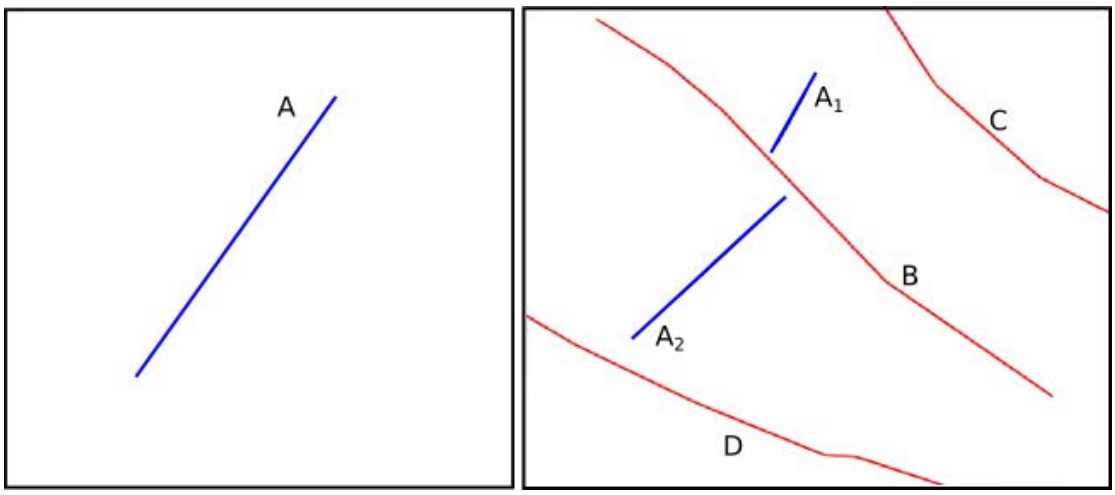

Figura 5: Exemplo de estruturas deformadas.
As estruturas mais novas (B, C e D) deformam uma estrutura mais antiga (A), gerando novas representações (A_1 e A_2).

Considerando diferentes ontologias, pode haver diferenças no formato e na precisão do registro de tempo. Em uma ontologia, o tempo pode ser representado em forma de data, como dia, mês e ano. Em outra, o tempo pode ser representado em forma de data, mas acrescido de informações de horas. Já em uma terceira ontologia, o tempo pode ser indicado de acordo com o *Unix Timestamp*, geralmente utilizado em sistemas

computacionais, e que conta os segundos desde uma data arbitrária[15] para representar um dado instante.

Quando se trata de um domínio específico, a contagem do tempo pode ser relativa, gerando diferentes sistemas para esta contagem. Este é o caso do tempo geológico, que possui uma nomenclatura específica para cada período de tempo e pode variar de acordo com algumas interpretações da comunidade científica da área (MA; FOX, 2013). Estas interpretações estão fora do escopo deste trabalho.

3.3 Ontologias no Domínio Geológico

Apesar de haver documentações e publicações a respeito da utilização e definição de ontologias de domínio geológico, poucas dentre aquelas publicações que foram encontradas estão completamente acessíveis e compartilhadas. Os maiores destaques de ontologias relacionadas ao domínio geológico são a SWEET[16] (*Semantic Web for Earth and Environmental Terminology*) e a *Ontology of Fractures* (ZHONG; AYDINA; MCGUINNESS, 2009).

A SWEET é um conjunto de ontologias desenvolvidas pela NASA (*National Aeronautics and Space Administration* – Administração Nacional da Aeronáutica e do Espaço) que abrangem os mais variados conceitos relacionados ao estudo do planeta Terra. Estes conceitos variam entre fenômenos físicos, atividades humanas, processos, materiais, e outros relacionados. Alguns dos conceitos mais comuns do domínio de Geologia estão definidos na SWEET, e por esta ser considerada uma ontologia geral dos domínios relacionados às Geociências, sua reutilização na construção de ontologias do domínio geológico torna-se fundamental para integração das mesmas.

A ontologia *Ontology of Fractures* (ZHONG; AYDINA; MCGUINNESS, 2009) contém diversos conceitos relacionados à Geologia Estrutural, ao definir estruturas geológicas e seus mecanismos de deformação. Esta ontologia define conceitos que podem ser utilizados em conjunto com diversos subdomínios da geologia para o estudo

[15] *Unix Timestamp* conta os segundos desde 1 de Janeiro de 1970.
[16] http://sweet.jpl.nasa.gov/ontology/

de diversas atividades, como a exploração de óleo e gás por exemplo. Entretanto, esta ontologia não foi encontrada disponível de maneira acessível para um estudo aprofundado.

Diversos outros exemplos de ontologias relacionados ao domínio geológico e a seus subdomínios podem ser encontrados, porém devido à complexidade dos conceitos a serem definidos existem alguns problemas na elaboração de ontologias geológicas. Estes problemas estão relacionados à complexidade da própria formação da Terra, que abrange muitos domínios, conceitos e relações complexas. Mesmo com estes desafios, é possível encontrar cada vez mais ontologias relacionadas a estes domínios, principalmente quando se trata do desenvolvimento de soluções na indústria de óleo e gás. Dado ao volume de ontologias, o processo de alinhamento é fundamental para permitir que os dados representados por estas ontologias possam ser integrados e utilizados em estudos e trabalhos na área.

Devido à dificuldade de acessibilidade das ontologias de domínio encontradas na literatura, novas ontologias de referência foram construídas para concepção do *framework* proposto neste trabalho.

3.3.1 Ontologias Neotectônica

As ontologias utilizadas neste trabalho foram construídas no contexto do projeto que serviu de base para implementação do protótipo do *framework* de integração. Nesta seção são apresentadas as ontologias que compõem as denominadas *Ontologias Neotectônicas*, um conjunto de ontologias que descrevem o conhecimento necessário para representar as informações pertinentes ao projeto Mapa Neotectônico do Brasil, que será apresentado com maiores detalhes na seção 5.2.

A primeira ontologia de domínio, denominada *Ontologia Neotectônica*, descreve conceitos relacionados principalmente à Geologia Estrutural, como falhas geológicas, as variações de tipos de falhas, lineamentos, afloramentos, entre outros. Também envolve a descrição das propriedades essenciais que definem cada entidade, e as relações fundamentais entre cada conceito. A *Ontologia Neotectônica* foi desenvolvida

principalmente por meio da reutilização de conceitos definidos pela ontologia SWEET. Devido ao seu tamanho e complexidade apenas uma parte da hierarquia de classes da *Ontologia Neotectônica* é apresentada na Figura 6.

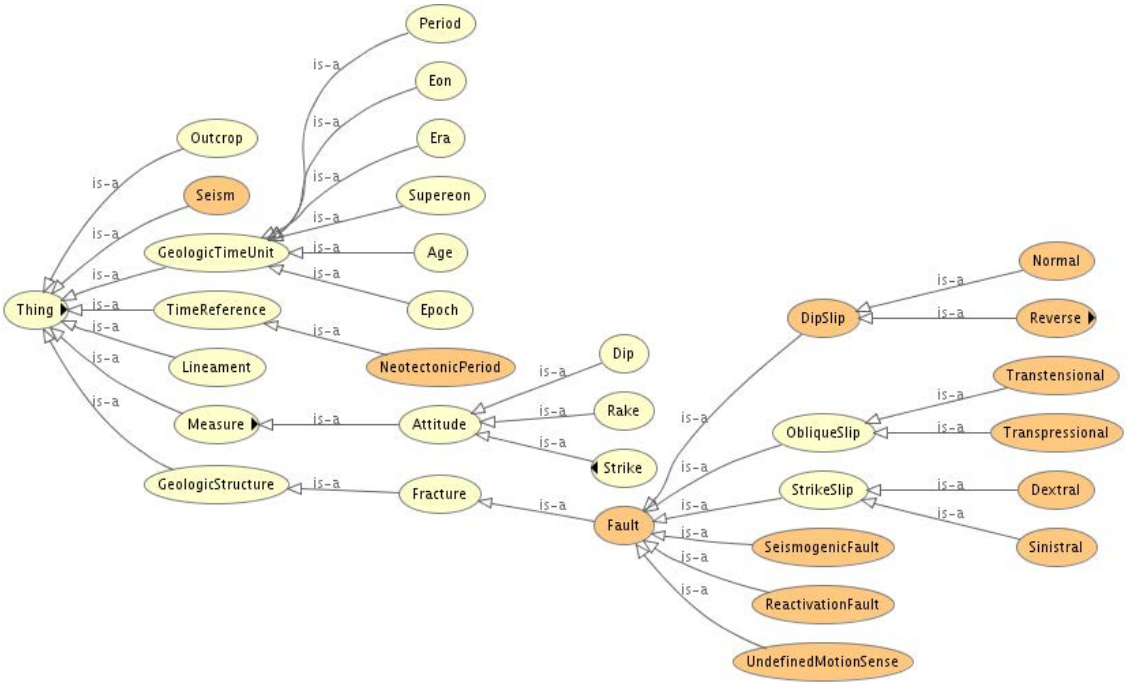

Figura 6: Parte dos conceitos da *Ontologia Neotectônica*.

A segunda ontologia de domínio foi denominada *Ontologia Neotectônica de Referências Bibliográficas*, e descreve conceitos relacionados à bibliografia. Esta ontologia é necessária pois grande parte dos dados compilados pelo projeto foi coletada em publicações científicas. Esta ontologia foi desenvolvida por meio da reutilização de diversas ontologias, dentre as quais as principais são a *The Bibliographic Ontology*[17], *Dublin Core Terms*[18] e FOAF[19] (*Friend of a Friend*). Parte da *Ontologia Neotectônica de Referências Bibliográficas* é apresentada na Figura 7.

[17] http://bibliontology.com/

[18] http://dublincore.org/documents/dcmi-terms/

[19] http://xmlns.com/foaf/spec/

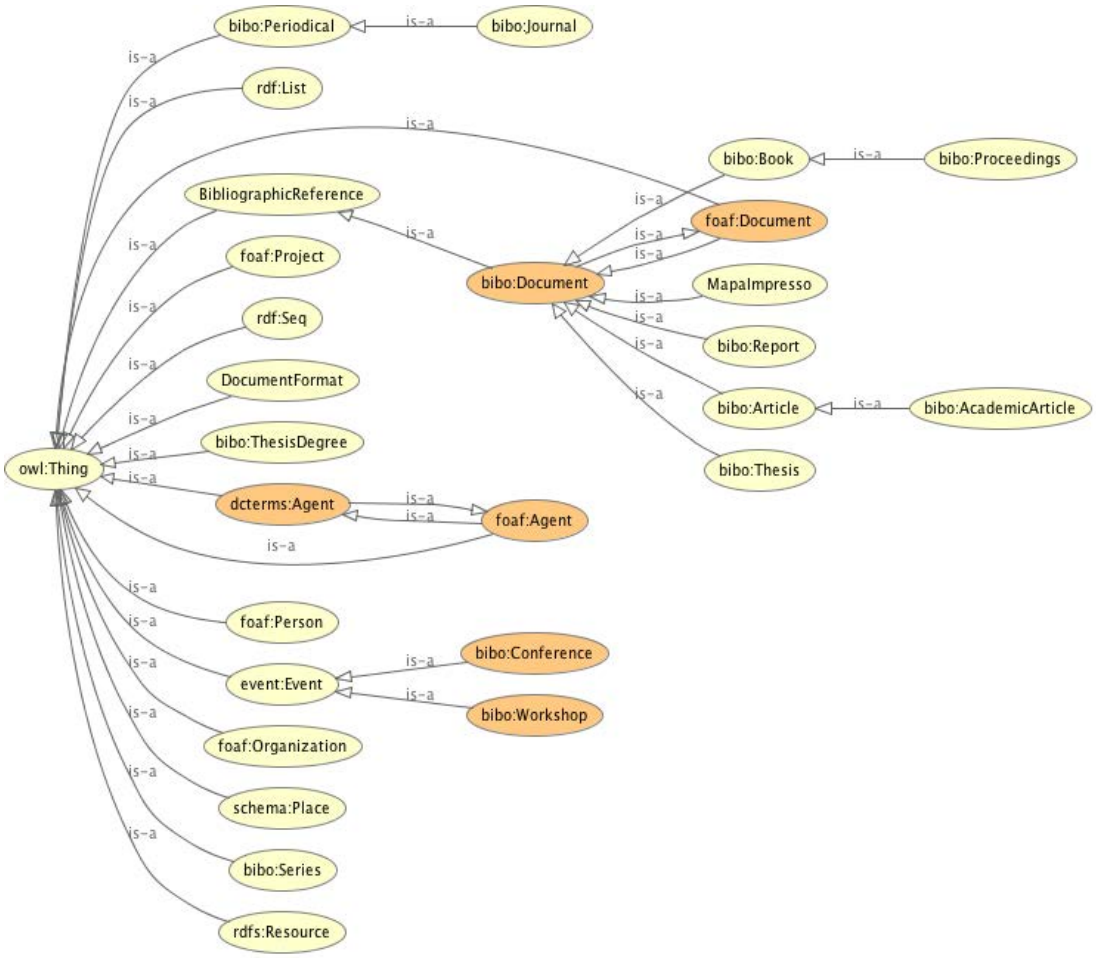

Figura 7: Parte da *Ontologia Neotectônica de Referências Bibliográficas*.

A terceira ontologia de domínio foi denominada *Ontologia Neotectônica Espacial*, e descreve conceitos relacionados com o domínio geoespacial. Esta ontologia foi desenvolvida com a reutilização e expansão do vocabulário GEOSPARQL, assim como a reutilização da IGNF citada na seção 3.2.2. Além dos conceitos descritos pelos vocabulários comuns, novos conceitos relacionados a formatos de representação de dados geográficos e sistemas de coordenadas também foram descritos. Parte da *Ontologia Neotectônica Espacial* é apresentada na Figura 8.

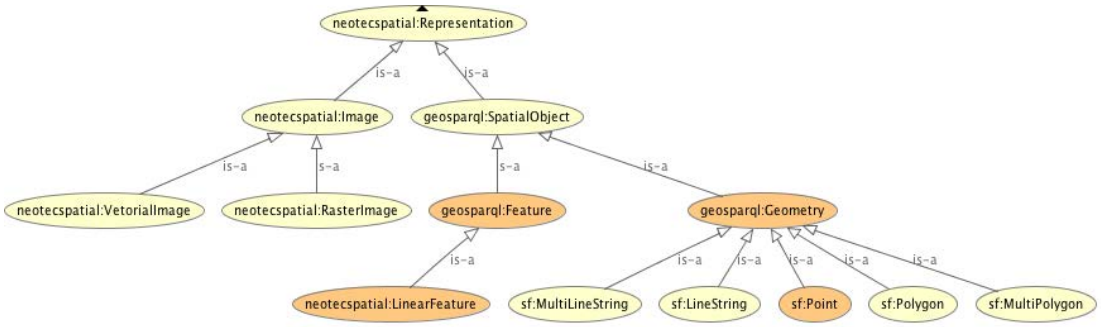

Figura 8: Parte da *Ontologia Neotectônica Espacial*.

Finalmente, a ontologia denominada *Ontologia Neotectônica de Aplicação* foi desenvolvida como ontologia de aplicação. Esta ontologia importa as demais ontologias e complementa a descrição dos conceitos destas com outros conceitos e propriedades que são pertinentes especificamente ao escopo do projeto, como os conceitos que associam uma sigla de controle para as feições geoespaciais e atributos geológicos mais restritos. O aspecto mais importante desta ontologia é que ela armazena as regras utilizadas para a análise semântica de instâncias de dados geológicos. Estas regras, definidas com a linguagem SWRL, são apresentadas com mais detalhes na seção 5.4.1. Em sua maioria utilizam os vocabulários das ontologias de domínio, mas são mantidas na ontologia de aplicação pois podem ser personalizadas de acordo com o contexto da aplicação. Parte da *Ontologia Neotectônica de Aplicação* é apresentada na Figura 9.

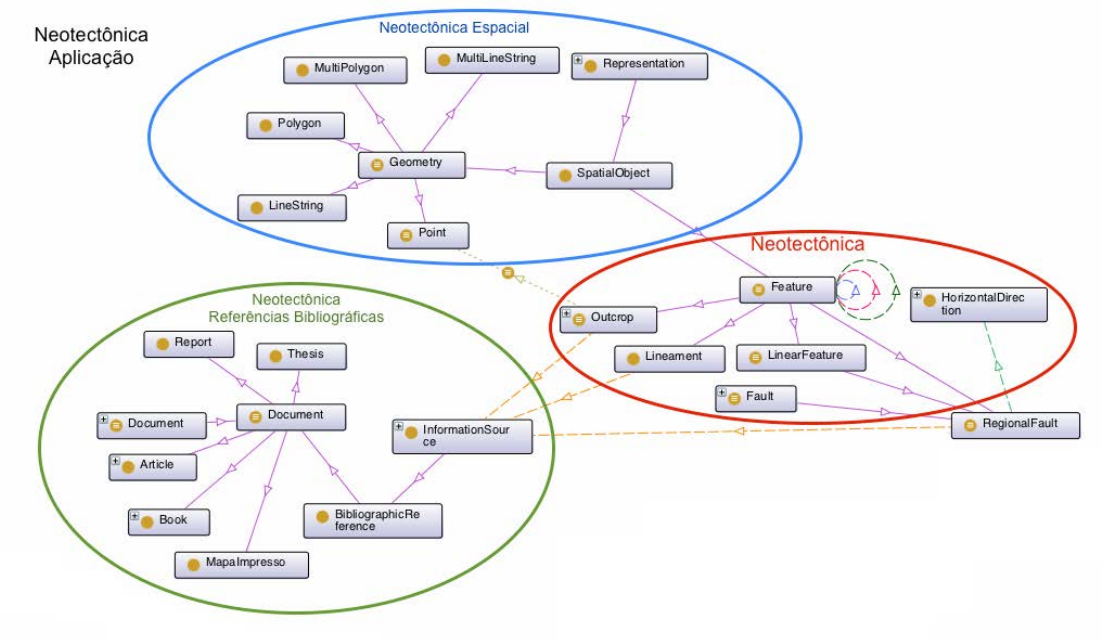

Figura 9: Parte da *Ontologia Neotectônica de Aplicação*.

O papel das Ontologias Neotectônica no escopo deste trabalho é serem utilizadas como referência para o processo de integração semântica de dados geológicos.

3.4 Integração de Dados e Alinhamento de Ontologias

Para finalizar a contextualização do problema, e como forma de corroborar a proposta apresentada neste trabalho, esta seção apresenta alguns trabalhos que relacionam os conceitos de ontologias, alinhamento de ontologias, e representação de dados geoespaciais para solucionar problemas de integração no domínio geral das Geociências.

Cruz et al. (2013) desenvolveram um *framework* para integrar, visualizar e analisar dados espaciais e temporais. O objetivo do *framework* é permitir o acesso simultâneo a conjuntos de dados heterogêneos e permitir a integração dos dados para uma visualização e análise uniforme. Este *framework* permite o acesso a dados padronizados e não padronizados, e realiza a extração de ontologias dos esquemas desses dados. A partir das ontologias extraídas, um processo de alinhamento é realizado

para integrar as informações. Inserido no processo de alinhamento, destaca-se o casamento de dados espaciais e temporais, em que são analisados casos de geocodificação, que é o processo de definir de uma posição geográfica a partir do nome de uma entidade, como por exemplo, o nome de uma cidade; e casos em que há heterogeneidade proveniente da forma com que os dados foram coletados e representados.

O trabalho de Cruz et al. (2013) é um dos principais exemplos de aplicação de alinhamento de ontologias para integração de dados geoespaciais, portanto contribui positivamente para a elaboração da proposta deste trabalho.

Giannopoulos et al. (2014) apresenta o uso do alinhamento de ontologias com foco na integração de dados geoespaciais, considerando as diferentes formas de representação de coordenadas geográficas com vocabulários RDF. O sistema desenvolvido no trabalho de Giannopoulos et al. (2014) é capaz de reconhecer diferentes representações de coordenadas geográficas com a aplicação de expressões regulares. Uma vez que estejam reconhecidas as coordenadas, o sistema transforma a representação para o formato definido pelo padrão GEOSPARQL[20]. A transformação neste caso permite com que os dados espaciais possam ser processados e comparados de maneira uniforme, uma importante contribuição para integração de dados de qualquer um dos subdomínios das Geociências.

Além dos trabalhos citados, existem ferramentas que permitem a manipulação de informações geográficas, em nível de implementação, como por exemplo, o *framework* Apache Jena[21], que em sua extensão espacial, reconhece formatos definidos pelo GEOSPARQL, WGS84-POS e ainda permite a definição de formatos customizados com os quais o usuário possa ter representado informações geoespaciais.

[20] https://portal.opengeospatial.org/files/?artifactid=47664

[21] https://jena.apache.org/

3.5 Conclusão do Capítulo

Neste capítulo foi apresentado um detalhamento do contexto do problema a ser tratado neste estudo. Neste detalhamento, foi caracterizada a natureza dos dados geológicos a serem integrados e a representação destas informações em ontologias, com destaque às informações geoespaciais, característica fundamental associada a qualquer dado geológico. Após a discussão sobre estes aspectos, uma breve relação entre os conceitos apresentados e a solução de integração proposta foi estabelecida com base em trabalhos relacionados.

Os conceitos apresentados neste capítulo permitiram identificar alguns componentes do *framework* e principalmente as duas estratégias de manipulação de dados: o armazenamento local e a consulta distribuída.

4 *Framework para Integração de Dados Geoespaciais*

Neste capítulo é apresentado o *Framework para Integração de Dados Geoespaciais*, a solução proposta neste trabalho para realizar a integração de dados geológicos baseada em ontologias.

Na próxima seção é apresentada uma visão geral da arquitetura do *framework*. Nas seções seguintes, são apresentados cada um dos componentes da arquitetura e as considerações práticas e teóricas de suas concepções e implementações.

4.1 Visão Geral do *Framework*

Com base nos aspectos conceituais, apresentados no Capítulo 2, necessários ao desenvolvimento deste trabalho e, no contexto do problema a ser resolvido, apresentados no Capítulo 3, foi definido um *framework* capaz de utilizar ontologias para a integração semântica de dados geológicos. Esta solução é apresentada como um *framework*, pois oferece uma estrutura básica de componentes e comportamentos, mas pode ser adaptada de acordo com as estratégias de acesso aos dados. Além disso, a configuração de um *framework* que utiliza ontologias pode ser adaptada para qualquer subdomínio das Geociências, pois uma das características desta solução é o processamento de informações geoespaciais durante o processo de integração de dados.

A visão geral arquitetura do *framework*, apresentada na Figura 10, consiste basicamente de uma camada de fonte de dados (*Data Layer*), que representa o acesso às fontes de dados a serem integradas; uma camada semântica (*Semantic Layer*) que realiza o processo de integração e uma camada de aplicação (*Application Layer*) que permite a interação de usuários no processo.

As etapas de integração ocorrem predominantemente na camada semântica. Esta camada é composta pelo *Semantic Middleware*, que organiza as interações do *Alignment Server* e do *Semantic Repository* de acordo com o tipo de estratégia de acesso a dados que foi adotada. Uma das estratégias é utilizada para realizar consultas semânticas a repositórios de dados distribuídos (*Distributed Query Strategy* - DQS) , e outra para

realizar consultas semânticas em um repositório local (*Local Storage Strategy - LSS*), no qual os dados integrados são armazenados e existe o controle sobre o acesso a estes dados. Há ainda a possibilidade de configurar as duas estratégias simultaneamente. A possibilidade de configurar o *framework* com estas estratégias permite que tanto dados públicos quanto dados com acesso restrito sejam manipulados para integração, um requisito discutido conceitualmente na seção 3.1.

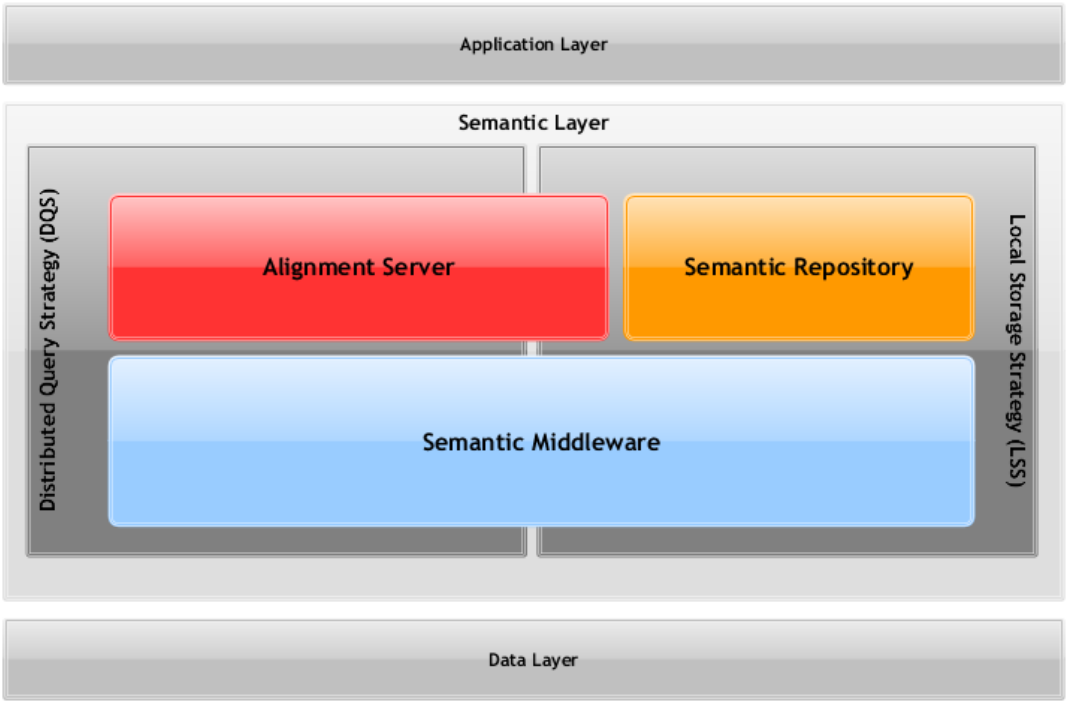

Figura 10: *Framework* de Integração Geoespacial Semântica: Visão Geral.

Os detalhes pertinentes a cada componente pertencente à organização desta arquitetura são apresentados nas seções seguintes. A ordem de apresentação segue a partir da fonte de dados, seguida pela camada de aplicação e finalmente pela camada semântica, deixada para o final devido à sua complexidade.

4.2 *Data Layer*

Nesta camada estão disponíveis os repositórios de dados a serem integrados. Conforme discutido na seção 3.1, as fontes de dados geológicos, ou datasets, podem disponibilizar os dados com diferentes esquemas e formatos. Neste trabalho foram considerados os formatos estruturados e semiestruturados, isto é, aqueles que possuem um rigor em relação à estruturação dos dados. Dentre as formas mais utilizadas para disponibilizar dados espaciais, foram considerados neste estudo: o shapefile, formato de arquivo utilizado pelos principais SIG; o banco de dados espacial open source PostgreSQL/PostGIS, também amplamente utilizado pela comunidade de desenvolvimento de SIG; o padrão WFS22 (Web Feature Service), definido pela OGC23 (Open Geospatial Consortium) como padrão de serviço para interoperabilidade de dados espaciais na Web; e as planilhas de dados, que podem organizar informações de maneira estruturada, em tabelas, ou semiestruturadas, em tabelas com células mescladas, acompanhadas de fórmulas e figuras.

Quando houver um conjunto de dados distribuídos, a fonte de dados deve ser registrada em uma estrutura na camada Data Layer. Este registro tem a finalidade de indicar aos demais componentes do *framework* quais as fontes de dados disponíveis para integração. A partir do registro de fontes de dados ou da apresentação direta de um conjunto de dados, como por exemplo, arquivos shapefile e planilhas, os dados e metadados ficam disponíveis para acesso pelos componentes das demais camadas.

4.3 *Application Layer*

Os componentes desta camada implementam as interfaces gráficas de usuário, que permitem a interação com as funcionalidades da integração de dados.

[22] http://www.opengeospatial.org/standards/wfs

[23] http://www.opengeospatial.org/

A aplicação deve consistir principalmente em uma interface Web SIG capaz de permitir consultas e apresentar mapas interativos. Estes mapas podem ser instâncias individuais, disponibilizadas pelo próprio sistema e/ou resultados de integração de dados geológicos.

Além de disponibilizar os mapas, a interface deve permitir acesso às funcionalidades de integração. Entre estas funcionalidades estão:

- Configuração e definição da ontologia de domínio do sistema.
- Cadastro e seleção das fontes de dados que devem ser reconhecidas pelo sistema quando houver acesso e consulta a *datasets* distribuídos.
- Acesso à gerência dos processos de alinhamentos disponibilizados no Servidor de Alinhamentos.
- Configuração dos parâmetros para execução do processo de integração. Os parâmetros podem ser: seleção das fontes de dados que serão integradas; seleção do processo de alinhamento; e configurações específicas do processo selecionado.

Este conjunto de funcionalidades deve garantir que o usuário possa manipular a configuração e execução do processo de integração de uma forma geral. Outras funcionalidades podem ser consideradas na implementação da aplicação, de acordo com as necessidades de contexto.

Ainda na camada de Aplicação, uma interface simplificada, baseada em busca e/ou navegação hierárquica por conceitos deve permitir ao usuário acessar as fontes de dados reconhecidas pelo sistema e as instâncias de dados de cada fonte.

As características citadas acima para a definição da aplicação não fazem parte da configuração básica do *framework*, pois podem variar de acordo com os requisitos e contextos da aplicação, portanto, estas características possuem caráter de sugestão.

4.4 *Semantic Layer*

A *Semantic Layer* consiste dos componentes que utilizam tecnologias da Web Semântica para realizar a integração de dados geológicos. A organização e a interação destes componentes dependem da estratégia de acesso a dados que é implementada. Na Figura 11, os componentes da *Semantic Layer* são apresentados com os detalhes das interações entre os componentes.

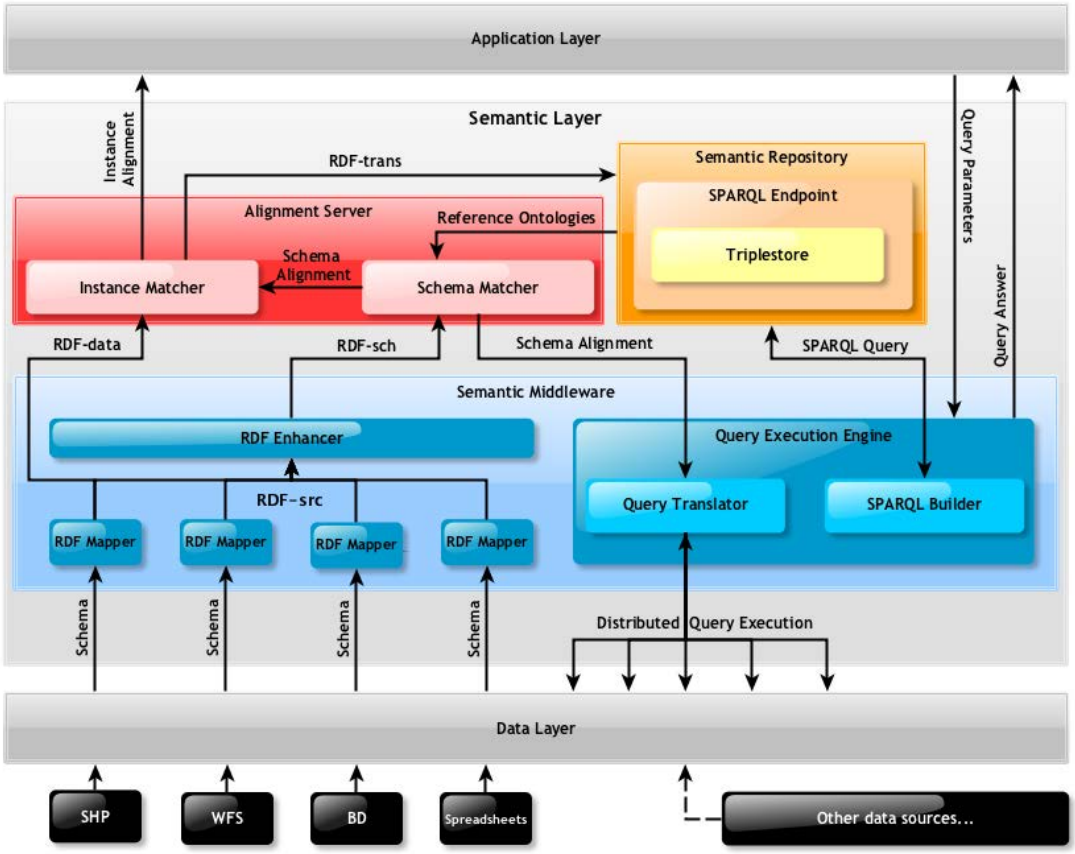

Figura 11: *Framework para Integração de Dados Geoespaciais – Semantic Layer.*

Na Figura 11, as interações das demais camadas com os componentes da *Semantic Layer* são também apresentadas. As *Reference Ontologies*, apesar de não estarem diretamente representadas na figura, estão armazenadas no *Semantic Repository*. Estas ontologias utilizadas como referência pelo *framework*.

O processo de integração é iniciado com a apresentação de um conjunto de dados à *Data Layer*. Nesta etapa inicial, existe a opção de apresentar os dados diretamente, apenas para consumo imediato, ou o registro das fontes de dados para que estas sejam acessadas posteriormente, conforme discutido na seção 4.2. Uma vez que os dados estejam disponibilizados na *Data Layer*, o processo de integração é iniciado pelo *Semantic Middleware*, que direcionará o fluxo de informação entre os componentes, seguindo os requisitos das estratégias de acesso a dados para a qual o *framework* estiver configurado.

As próximas seções apresentam uma descrição dos componentes da *Semantic Layer* e as tarefas que estes componentes são responsáveis por realizar.

4.4.1 *Semantic Middleware*

O *Semantic Middleware*, apresentado na Figura 11, é o componente responsável por coordenar a sequência de ações dos demais componentes da arquitetura do *framework* durante o processo de integração, ou seja, este componente controla a comunicação entre seus componentes internos e externos, como o *Alignment Server* e o *Semantic Repository*. Também é responsabilidade do *Semantic Middleware* intermediar as ações do usuário, que ocorrem na *Application Layer*, com a execução da integração ou com a consulta aos dados integrados.

O *Semantic Middleware* possui componentes independentes que realizam tarefas específicas em cada etapa da integração ou consulta. Estes componentes e suas funções são descritos a seguir.

4.4.2 *RDF Mapper*

Este componente, representado na Figura 12 com suas principais interações, é o responsável por transcrever o esquema dos dados disponíveis na *Data Layer* para o formato RDF.

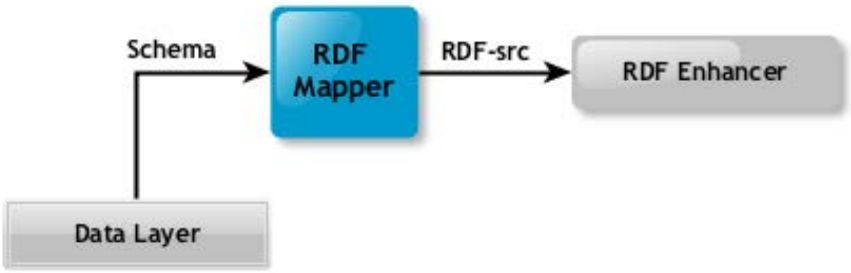

Figura 12: Interações do *RDF Mapper*.

O papel do *RDF Mapper* é transformar o esquema de uma fonte de dados (*Schema*) em um arquivo no formato RDF que descreva os tipos de dados e atributos deste esquema (*RDF-src*). A implementação desta etapa é independente da estratégia de integração adotada. Porém, na arquitetura do *framework* deve haver uma implemementação do *RDF Mapper* para cada tipo de fonte de dados. Isto se deve ao fato de que a forma de acessar o esquema de dados em cada tipo de fonte é diferente. Por exemplo, para acessar o esquema em *shapefiles* é preciso uma API específica; com o WFS, é preciso realizar requisições HTTP[24] e ler um arquivo XML; e no PostgreSQL/PostGIS é preciso realizar consultas SQL.

No caso dos formatos considerados como fonte de dados neste trabalho, tanto o *shapefile*, quanto o PostgreSQL/PostGIS e o WFS utilizam o modelo relacional para organizar dados. No caso do *shapefile*, os dados são armazenados em um arquivo DBF (*Database Format*), estruturados em formato de tabela. O WFS realiza o mapeamento de uma tabela para uma descrição XML. Já o PostgreSQL/PostGIS é um SGBD (Sistema de Gerenciamento de Banco de Dados) tradicional que implementa o modelo relacional. Entretanto, as planilhas de dados podem apresentar diversas configurações de organização de dados, isto é, mesclar células, calcular fórmulas e até mesmo anexar figuras ou arquivos como conteúdos. Desta forma, as planilhas de dados são consideradas como fonte de dados semiestruturados, pois, apesar de obedecerem a um padrão de organização do formato de planilha, podem apresentar diferenças substanciais

[24] Hypertext Transfer Protocol: http://www.w3.org/Protocols/

neste aspecto. Assim, o tratamento de planilhas deve ser determinado de acordo com regras pré-definidas para o envio de dados quando o *framework* for implementado.

A partir destas observações, a transformação em RDF do esquema de dados, considerando os principais formatos estruturados adotados, pode ser resumida como uma transformação do modelo relacional para um modelo baseado em grafos, utilizado pelo RDF, conforme é apresentado na Figura 13. Esta transformação do modelo relacional para RDF pode ser feita por meio do modelo de mapeamento direto[25], no qual: uma tabela é mapeada para uma classe OWL, e os atributos simples são mapeados para propriedades de dados. As chaves estrangeiras de um modelo são mapeadas para propriedade de objetos e as classes são relacionadas por estas propriedades.

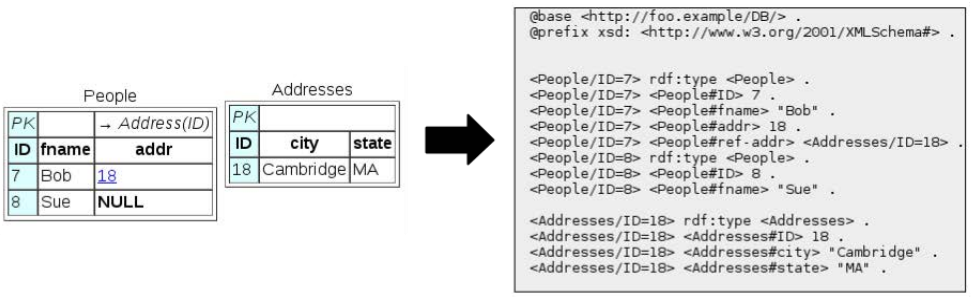

Figura 13: Exemplo de transformação de modelo relacional em RDF.
Fonte: WORLD WIDE WEB CONSORTIUM, 2012.

Na Figura 13, por exemplo, a tabela *People* é mapeada para uma classe *People*, e os registros desta tabela são mapeados para indivíduos do tipo *People*, ou seja, cada registro representa uma pessoa. Observe, ainda na Figura 13, que o atributo *addr* da tabela *People* é mapeado para uma propriedade de objeto, e o valor deste atributo para um indivíduo do tipo *People* é outro indivíduo do tipo *Addresses*, classe criada a partir da tabela *Addresses*.

As transformações realizadas nesta etapa não são armazenadas, pois pode haver mudanças na estrutura dos modelos de dados dos *datasets*. Além disso, não se trata de

[25] http://www.w3.org/TR/rdb-direct-mapping/

um processo complexo, podendo ser executado com frequência sem impacto em desempenho.

O arquivo denominado *RDF-src*, gerado nesta etapa a partir do esquema de dados, é encaminhado para o *RDF Enhancer*, cujo papel é apresentado a seguir.

4.4.3 *RDF Enhancer*

Este componente, representado na Figura 14 com suas principais interações, é responsável pela execução do processo de enriquecimento do RDF (*RDF-src*) gerado pelo *RDF Mapper*.

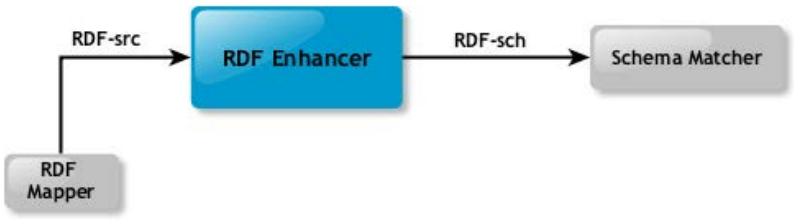

Figura 14: Interações do *RDF Enhancer*.

O enriquecimento é um processo que busca expandir a descrição inicialmente gerada pelo mapeamento de esquema em RDF (RDF-src). O enriquecimento do RDF tem como finalidade ampliar a eficiência no processo de alinhamento.

Para o domínio de geociências, foram implementadas três etapas de enriquecimento: a padronização de coordenadas; a expansão de vocabulário; e, o uso de metadados, quando disponíveis.

A etapa de padronização de coordenadas consiste em identificar a presença de coordenadas geográficas nos conceitos que compõe o *RDF-src* e descrever estas coordenadas com vocabulário padrão (GEOSPARQL e IGNF[26]). A partir da identificação de coordenadas, é possível filtrar os conceitos da ontologia de domínio

[26] http://data.ign.fr/def/ignf.html

que também possuem coordenadas geográficas associadas, de modo a contribuir para o alinhamento encontrar correspondências mais facilmente.

A etapa de expansão de vocabulário também pode contribuir para aprimorar o processo de alinhamento posterior. Com a utilização de dicionários, principalmente aqueles específicos do domínio, os nomes dos conceitos podem ser expandidos, utilizando-se sinônimos, por exemplo.

A etapa que utiliza metadados para enriquecer o *RDF-src* é condicionada à disponibilidade de metadados no conjunto de dados. Quando houver esta disponibilidade, os metadados podem ser utilizados para enriquecer a descrição *RDF-src*, principalmente se os metadados seguirem vocabulários padronizados, o que pode facilitar ainda mais o processo de alinhamento. Geralmente é no conjunto de metadados que há informações importantes, como a escala de representação e detalhes sobre o Sistema de Referência Espacial utilizado nos dados.

O resultado gerado pelo *RDF Enhancer* é um arquivo RDF, denominado *RDF-sch*, que possui conceitos e relações enriquecidos. Após esta etapa, o *RDF-sch* está em condições de ser alinhado com as *Reference Ontologies*. Assim, o *RDF-sch* é encaminhado para o *Schema Matcher*, parte do *Alignment Server* apresentado a seguir.

4.4.4 *Alignment Server*

Este componente deve ser utilizado para gerar, armazenar e compartilhar os alinhamentos, além de editá-los de diversas formas. O *Alignment Server* também deve possibilitar a personalização e a execução de diferentes métodos de alinhamento, dentre outras funcionalidades relacionadas. A implementação do *Alignment Server* utilizada para concepção deste *framework* segue a proposta de David et al. (2011), que especifica a implementação de um servidor de alinhamento utilizando a *Alignment API*[27], uma API Java para manipular alinhamentos de ontologias.

[27] http://alignapi.gforge.inria.fr/

A utilização do Alignment Server com as características descritas acima permitem dar a arquitetura do *framework* flexibilidade e a capacidade de adaptação. A flexibilidade decorre do fato do servidor ser capaz de gerenciar métodos de alinhamento, e permitir aos usuários e administradores do sistema definir quais métodos ficam disponíveis para execução. A interface de serviços fornecida pelo servidor permite ainda integrá-lo com outros sistemas. Além da capacidade de incorporar novos métodos de alinhamento, a capacidade de adaptação também é garantida com o gerenciamento dos resultados dos alinhamentos. Ao reutilizar alinhamentos, os processos podem obter resultados de maneira mais rápida e, dependendo da forma de reuso, de maneira mais precisa.

O *Alignment Server* deve possuir dois métodos para realizar seu papel na integração de dados geológicos, ou pelo menos um método que seja configurável para separar o alinhamento de esquemas e alinhamento de instâncias. Esta divisão é motivada pelos aspectos discutidos em todo o contexto do Capítulo 3. Na Figura 11, os métodos são representados pelo *Schema Matcher* e *Instances Matcher*, descritos a seguir.

4.4.5 *Schema Matcher*

Este componente representa um método, ou parte de um método de alinhamento que seja capaz de realizar o alinhamento de esquemas de duas ontologias. No que diz respeito a ontologias, o esquema é composto pelas descrições terminológicas de conceitos e propriedades. Na Figura 15, é apresentado o componente *Schema Matcher*, com suas principais interações.

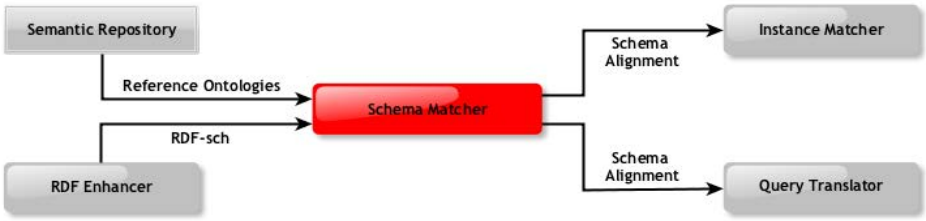

Figura 15: Interações do *Schema Matcher*.

No caso deste *framework*, o *RDF-sch*, gerado pelo *RDF Enhancer* é alinhado com as *Reference Ontologies*. O resultado gerado, *Schema Alignment*, é um conjunto de correspondências entre os conceitos e relações do RDF-sch com os conceitos e relações das *Reference Ontologies*, codificado no formato[28] definido pela *Alignment API*.

O processo de alinhamento para ser executado nesta etapa pode ser qualquer método que siga o modelo geral de organização do processo de alinhamento apresentado na Figura 4 da seção 2.2. Os detalhes pertinentes à implementação serão apresentados e discutidos no Capítulo 5, que descreve o protótipo implementado a partir do *framework* proposto.

O alinhamento realizado pelo *Schema Matcher* é executado em qualquer configuração de estratégia de acesso a dados, e o resultado gerado é sempre armazenado no *Alignment Server*. Entretanto, a utilização do *Schema Alignment* é realizada de maneira distinta em cada estratégia. Na *DQS* o *Schema Alignment* é utilizado pelo *Query Translator*, e na *LSS*, é utilizado pelo *Instance Matcher*. Os detalhes de utilização deste resultado são apresentados juntos às descrições destes componentes.

É importante destacar que o resultado gerado pelo *Schema Matcher* pode ser validado pelo usuário antes de ser definitivamente armazenado e utilizado nas demais etapas da integração.

[28] http://alignapi.gforge.inria.fr/format.html

4.4.6 *Instance Matcher*

Este componente, representado na Figura 16 com suas principais interações, é o responsável pelo processo de integração das instâncias de dados geológicos. A execução deste componente irá ocorrer após a execução do *Schema Matcher*, quando a *LSS* é implementada, pois os dados precisam ser padronizados e analisados antes do armazenamento. Entretanto, o *Instance Matcher* pode ser utilizado também na implementação da *DQS*, após a realização das consultas distribuídas, antes de apresentar os resultados para a *Application Layer*.

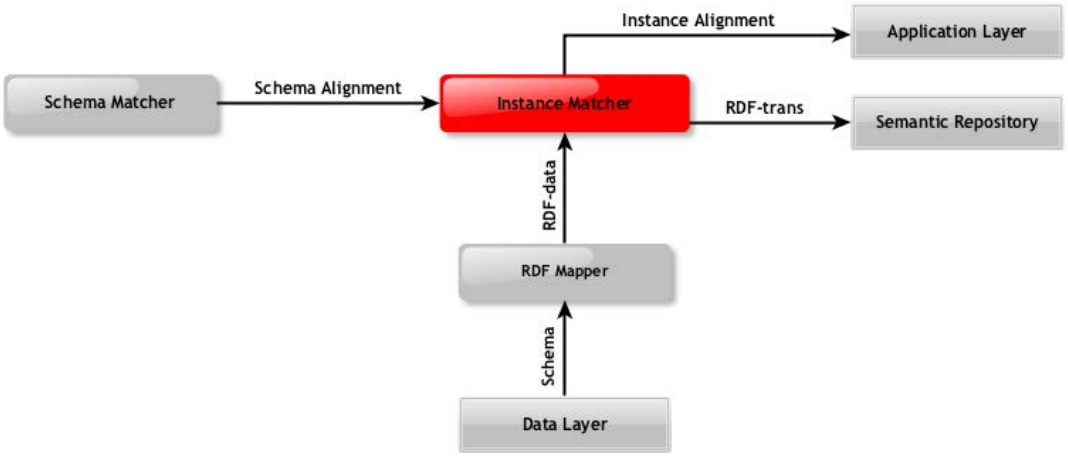

Figura 16: Interações do *Instance Matcher*.

Conforme apresentado na Figura 16, o Instance Matcher utiliza como entradas o *Schema Alignment* gerado pelo *Schema Matcher* e os dados provenientes da *Data Layer* que, na implementação da *LSS*, também são mapeados para RDF pelo *RDF Mapper*. O *Schema Alignment* é utilizado para traduzir a descrição das instâncias de dados (*RDF-data*), que posteriormente, passam por uma análise semântica, cujo objetivo é expandir as classificações e atributos dos dados e identificar possíveis inconsistências.

Parte do resultado gerado por este processo, o *Instance Alignment*, fica armazenado no *Alignment Server* e disponível para ser acessado na *Application Layer*. Este alinhamento contém os casos de possíveis combinações, duplicações e conflitos

entre as instâncias de dados integradas. As instâncias de dados, agora descritas com o vocabulário das *Reference Ontologies* (*RDF-trans*), são armazenadas no *Semantic Repository*, descrito com mais detalhes na seção 4.4.7.

Os detalhes pertinentes às etapas de tradução e análise semântica são discutidos no Capítulo 5.

4.4.7 *Semantic Repository*

Este componente consiste no repositório de armazenamento de ontologias, e é representado na Figura 17 com suas principais interações.

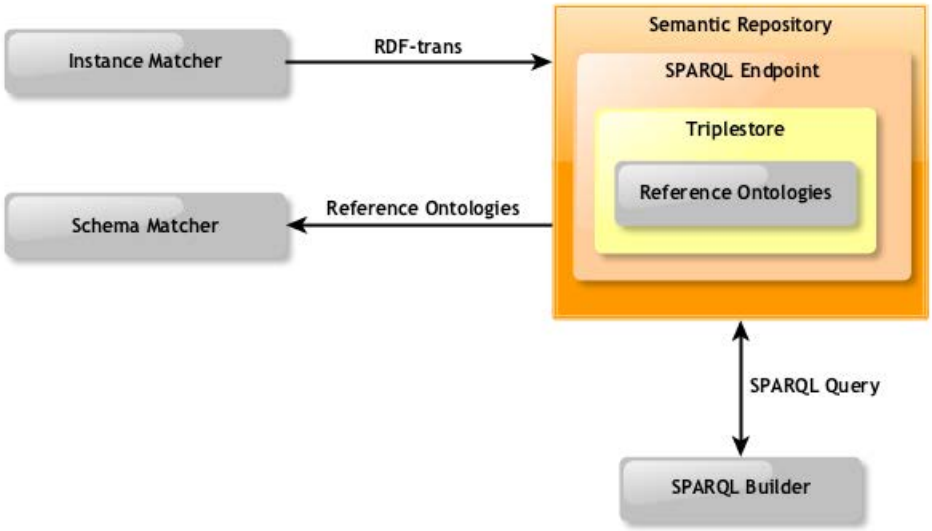

Figura 17: Interações do *Semantic Repository*.

O *Semantic Repository* consiste basicamente de um *triplestore*, um repositório de dados com propósito de armazenar triplas RDF, acessível através de um *Endpoint SPARQL*, que fornece, via interface de serviços, a possibilidade tanto de consultas quanto para atualizações.

O papel fundamental do *Semantic Repository* é armazenar as *Reference Ontologies*, ou seja, as ontologias de domínio utilizadas como referência durante o processo de integração e também para definir o vocabulário utilizado na interação do

usuário na *Application Layer*. Porém, quando a *LSS* é implementada, o *Semantic Repository* também mantém os dados geológicos semanticamente integrados, descritos com os vocabulários de referência. Neste contexto, existem duas abordagens possíveis: armazenar os dados como indivíduos da ontologia de domínio do sistema ou armazená-los em repositório diferente do *triplestore*, mas indexado pela ontologia. Neste caso, a escolha da abordagem depende de fatores exclusivos de implementação e não deve interferir no processo como um todo.

Uma colocação importante sobre o armazenamento das Reference Ontologies é a execução de inferências sobre as triplas armazenadas, isto é, armazenar a ontologia em um triplestore com capacidade de executar inferências. Esta estratégia diminui os tempos de consultas, pois estas já serão realizadas sobre o conhecimento inferido, sem a necessidade de processamento adicional, uma vez que as inferências serão geradas quando o conhecimento for adicionado no triplestore. Entretanto, as operações de atualização no *triplestore* podem ter seu desempenho prejudicado, uma vez que a cada modificação as inferências precisam ser atualizadas. Dependendo do tamanho e complexidade das *Reference Ontologies*, este processo pode levar algum tempo.

4.4.8 *Query Execution Engine*

Os componentes apresentados implementam as funcionalidades principais que são utilizadas na etapa de integração do *framework*. Nesta seção, assim como nas seções 4.4.9 e 4.4.10, são apresentados os componentes cujas funcionalidades são aplicadas no momento de consulta aos dados integrados pelo *framework*.

A realização de consultas é a etapa mais dependente do tipo de estratégia de acesso a dados implementada na arquitetura do *framework*. Quando a DQS é implementada, os dados permanecem em suas respectivas fontes, e consultas distribuídas são realizadas para recuperar os dados que correspondem aos critérios de busca definidos pelo usuário. Na implementação da *LSS*, como os dados foram armazenados no *Semantic Repository*, uma consulta interna neste componente é suficiente para recuperar os dados desejados.

O componente *Query Execution Engine*, representado na Figura 18 implementa uma interface comum para receber os parâmetros de busca (*Query Parameters*) e retornar os resultados (*Query Answer*) para a *Application Layer*.

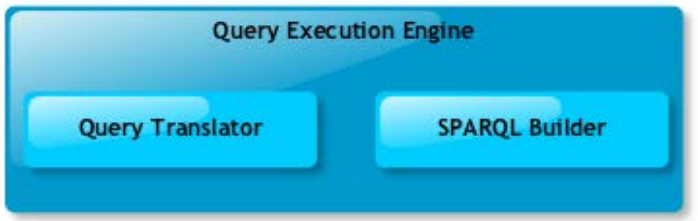

Figura 18: *Query Execution Engine*.

A execução das consultas é delegada para dois componentes específicos: o *Query Translator* para executar consultas distribuídas, e o *SPARQL Builder* para executar consultas locais, descritos nas seções 4.4.9 e 4.4.10 respectivamente. Porém, apesar desta divisão, estes componentes podem ser organizados e sincronizados para trabalhar simultaneamente, assim como os componentes da etapa de integração, de modo a garantir que o *framework* possa ser utilizado com as duas estratégias de acesso a dados ao mesmo tempo. Entretanto, para que isto seja possível, é preciso que a aplicação implementada na *Application Layer* forneça mecanismos de controle que permitam alternar entre as estratégias de acordo com o interesse do usuário.

A seguir, são apresentados detalhes dos componentes responsáveis pela execução das consultas.

4.4.9 *Query Translator*

Este componente, representado na Figura 19 com suas principais interações, é responsável por realizar as operações necessárias para a execução de consultas distribuídas.

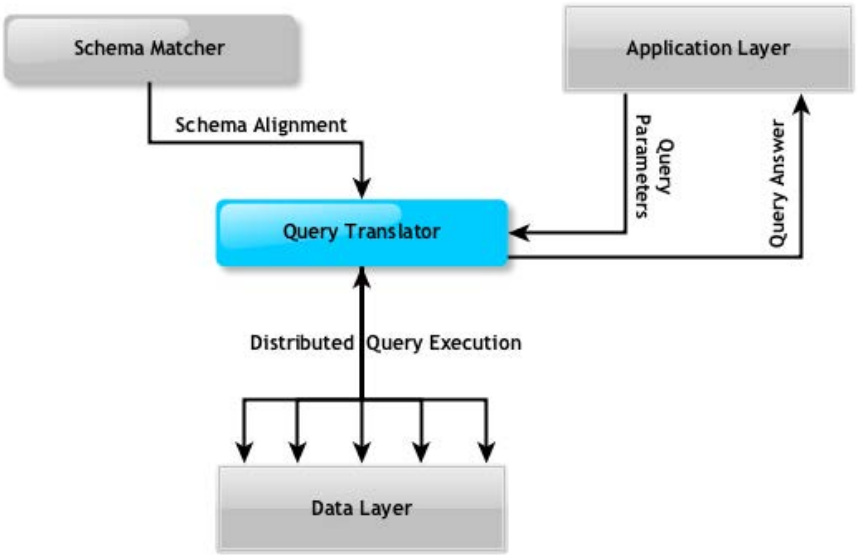

Figura 19: Interações do *Query Translator*.

O *Query Translator* recebe os parâmetros da consulta (*Query Parameters*) que foram enviados para o *Query Execution Engine*. Após receber os parâmetros, solicita para o *Schema Matcher* o *Schema Alignment* de cada fonte de dados registrada na *Data Layer* e inicia o processo de tradução de consulta. Neste processo, os parâmetros de consulta são traduzidos com o vocabulário utilizado pela fonte de dados e, de acordo com o formato da fonte, cada consulta é construída. No caso de arquivos no formato *shapefiles* e planilhas, as consultas são realizadas por APIs específicas, mas geralmente, este formato não é recomendado para utilizar nesta estratégia. No caso do padrão WFS uma consulta é construída nos formatos definidos pelo padrão. No caso do PostgreSQL/PostGIS, uma consulta SQL é construída. Com as consultas traduzidas, o *Query Translator* dispara cada uma das consultas para suas fontes de dados de destino, utilizando os parâmetros de acesso disponíveis no registro da *Data Layer*. Quando todos os resultados são retornados para o *Query Translator*, os dados são escritos em RDF e traduzidos para o vocabulário das *Reference Ontologies*. Finalmente, os dados descritos em RDF são enviados para a *Application Layer* por meio da interface do *Query Execution Engine*.

4.4.10 *SPARQL Builder*

Este componente, representado na Figura 20 com suas principais interações, é responsável por realizar consultas SPARQL diretamente no *Semantic Repository*.

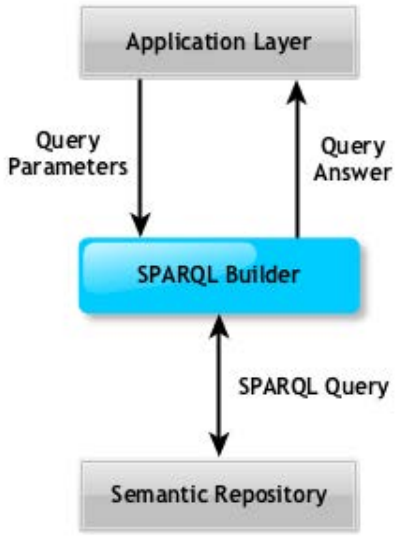

Figura 20: Interações do *SPARQL Builder*.

O *SPARQL Builder* recebe os parâmetros da consulta (*Query Parameters*) que foram enviados para o *Query Execution Engine* pela aplicação e constrói uma consulta SPARQL de acordo com os parâmetros e utilizando o vocabulário das *Reference Ontologies*. O SPARQL definido é enviado ao *SPARQL Endpoint* do *Semantic Repository*, que retorna o resultado da consulta. O resultado, neste caso já em RDF, é enviado para a *Application Layer* por meio da interface do *Query Execution Engine*.

Outra característica do *SPARQL Builder* é que este componente não lida apenas com consultas. Devido ao fato de ter acesso ao repositório local, também pode executar atualizações no *Semantic Repository*. Para realizar estas operações, a aplicação implementada na *Application Layer* deve fornecer a interface necessária para o usuário acrescentar, modificar ou excluir dados.

4.5 Conclusão do Capítulo

Neste capítulo foram apresentado os detalhes dos componentes do *Framework para Integração de Dados Geoespaciais*, que é concebido como uma solução para a integração semântica de dados geológicos.

Além do uso de ontologias e tecnologias associadas, como o RDF, a arquitetura do *framework* endereça questões conceituais pertinentes aos cenários de integração de dados geológicos, como o uso de dados públicos e distribuídos, endereçado pela *Distributed Query Strategy* e a manipulação de dados restritos, endereçada pela *Local Storage Strategy*. E apesar da concepção independente, a organização de componentes de serviços do *framework* permite que as estratégias sejam utilizadas simultaneamente.

A técnica de transformar os esquemas de dados em ontologias e utilizar o alinhamento de ontologias como base do processo de integração, garante a capacidade do *framework* de integrar dados geológicos sempre considerando a semântica associada a eles, independente dos dados estarem publicados com anotação semântica prévia ou não.

No próximo Capítulo, é apresentada a implementação de um protótipo do *Framework para Integração de Dados Geoespaciais*. Este protótipo serviu de base para analisar de modo mais detalhado os diversos aspectos relacionados à implementação dos componentes do *framework*.

5 Implementação do *Framework para Integração de Dados Geoespaciais*

Neste capítulo é apresentado um protótipo do *Framework para Integração de Dados Geoespaciais*. Este protótipo foi desenvolvido com o propósito de validar a organização conceitual do *framework* apresentado no Capítulo 4 e consequentemente, identificar aspectos pertinentes à implementação de cada componente.

5.1 Visão Geral da Implementação

Como forma de validar o *framework* apresentado no capítulo anterior, um protótipo com as principais funcionalidades foi implementado dentro do escopo de um projeto de pesquisa na área de Geologia. O protótipo implementado foi incorporado em um Sistema de Integração de Dados (SID), cujo objetivo principal é armazenar dados geológicos e disponibilizá-los em diversos formatos padronizados. Devido aos requisitos do SID, a implementação adota os conceitos da *LSS,* apresentada junto à descrição do *framework* no Capítulo 4. Na Figura 21, é apresentada a arquitetura do SID com a indicação dos componentes do *framework* que foram implementados no protótipo.

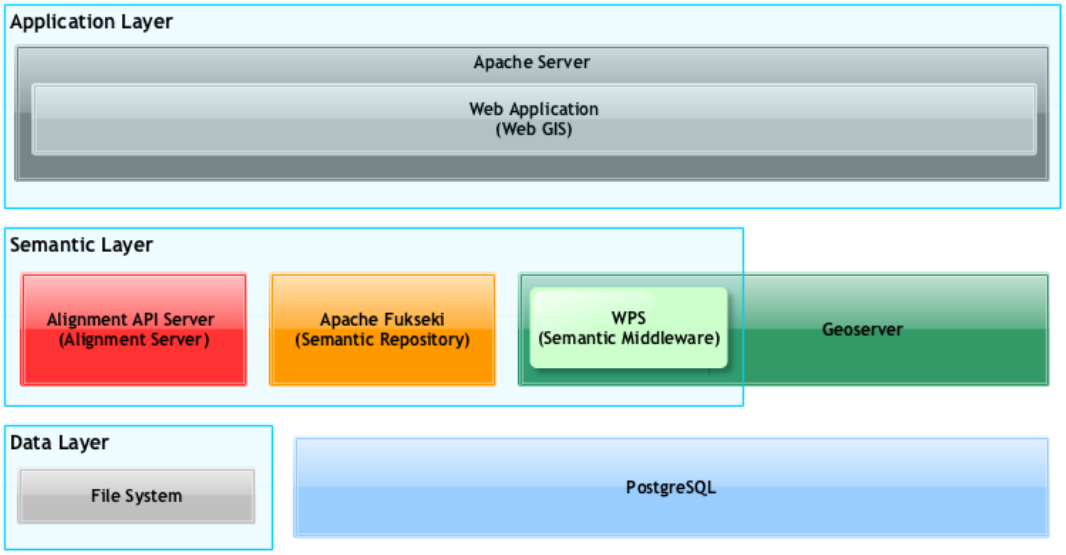

Figura 21: Arquitetura do Sistema de Integração de Dados.

A *Web Application* do Sistema de Integração de Dados equivale a *Application Layer* do *framework*, ou seja, trata-se de uma interface de um Sistema de Informação Geográfica Web (*Web GIS*) com as principais funcionalidades de manipulação de dados definidas na concepção do *framework*, conforme destacado na seção 4.3. Há também funcionalidades de interação com mapas e outros requisitos específicos do sistema. A *Web Application* foi implementada para ser executada no Apache Server e desenvolvida com as tecnologias *Javascript* / *OpenLayers* do lado cliente, e PHP no servidor.

A *Semantic Layer* proposta no *framework* foi implementada de modo a aproveitar a estrutura pré-existente na arquitetura do SID. Assim, conforme a Figura 21, o *Semantic Middleware* foi implementado como um processo WPS[29] (*Web Processing Service*) dentro do Geoserver. O Geoserver é um servidor de mapas que implementa a especificação do padrão de serviços da OGC. O WPS é o padrão OGC para a especificação de processos, e pode ser utilizado para manipular dados espaciais ou modificado para executar outras tarefas. Esta implementação é apresentada com maiores detalhes na seção 5.3. Os demais componentes da *Semantic Layer* foram adicionados à arquitetura original do SID. Desta forma, o *Alignment Server* implementado pela

[29] http://www.opengeospatial.org/standards/wps

Alignment API foi utilizado sem nenhuma personalização. Na implementação do *Semantic Repository* foi utilizado o Apache Fuseki[30], um servidor que fornece acesso a repositórios *triplestore* através de um *Endpoint SPARQL* e é implementado com a API Apache Jena[31].

A *Data Layer* foi implementada de modo que os arquivos *shapefile* e planilhas de dados importados pelos usuários sejam armazenados no *File System*, que representa o sistema de arquivos do servidor da aplicação. Os parâmetros de acesso a repositórios de dados, como serviços WFS e Bancos de Dados, também são armazenados no *File System* em formatos de arquivos.

O PostgreSQL representado na Figura 21 faz parte da arquitetura mas é utilizado por outras funcionalidades do SID, portanto, não possui nenhum papel relacionado à *Data Layer*.

Como parte de seu objetivo no escopo do projeto para o qual foi implementado, o SID permite que os dados geológicos sejam consultados interativamente na interface da *Web Application*. Os mapas disponíveis podem ser exportados em diversos formatos, como PNG (*Portable Network Graphics*), PDF (*Portable Document Format*) e arquivos shapefile. Com o uso do Geoserver, o SID ainda disponibiliza uma interface de serviços nos padrões OGC, como WFS e WMS[32] (*Web Map Service*), para que outras aplicações também possam consumir os dados publicados.

5.2 Contextualização da Implementação

O protótipo do *framework* foi implementado como parte do SID desenvolvido para o projeto Mapa Neotectônico do Brasil, realizado pelas universidades UNESP, UFPR, USP, UNICAMP, UFRRJ, UFES, UNB, UFBA, UFRN, UFAM e UNIFESP, em parceria com a Petrobras. O projeto tem como objetivo caracterizar o cenário neotectônico do Brasil, isto é, estudar os movimentos internos da crosta terrestre que

[30] http://jena.apache.org/documentation/serving_data/

[31] http://jena.apache.org/

[32] http://www.opengeospatial.org/standards/wms

ocorreram no passado recente e continuam até os dias atuais. A área explorada pelo projeto é restrita ao território brasileiro.

Cada universidade participante possui uma equipe de pesquisadores, formada por professores, profissionais e alunos da área de Geologia e relacionadas, como Geofísica e Sismologia, por exemplo. Cada uma destas equipes ficou responsável por coletar informações de uma determinada região do Brasil. As informações referem-se principalmente a falhas geológicas, lineamentos e o estudo de afloramentos.

O projeto conta com diversas etapas de obtenção de dados. A primeira etapa consistiu em um levantamento bibliográfico para determinar as publicações que já disponibilizavam conjuntos de dados de interesse do projeto. Em paralelo, novos estudos foram realizados pelas equipes participantes do projeto. Na etapa final, todo o conjunto de dados reunido pelo projeto deve estar integrado no Sistema de Integração de Dados NEOTEC[33], desenvolvido para esta finalidade.

Para realização do projeto, foi definido um conjunto básico de atributos para caracterizar cada tipo de dado. Os atributos foram divididos entre obrigatórios, geralmente utilizados para representar as feições nos mapas produzidos, e opcionais, que agregam informações importantes a cada feição, mas que por razões variadas podem não estar disponíveis.

Neste projeto, não foi definido o uso de nenhum padrão de metadados. O uso de um padrão poderia impactar no tempo de trabalho das equipes participantes do projeto, pois as descrições dos dados teriam que se adequar rigidamente aos padrões dos metadados. Além disso, houve diversas modificações nos atributos utilizados na descrição dos dados, e a adoção de metadados tornaria o processo de mudança mais complexo. Além disso, com a intenção futura de permitir a alimentação de dados por parte da comunidade, a ideia foi a de causar o menor impacto possível com relação à adequação de dados externos para serem inseridos no sistema. Assim, a organização dos atributos na composição dos dados produzidos e compilados pela equipe do projeto foi deixada livre para cada equipe definir. Apenas uma indicação dos atributos necessários e de interesse do projeto foi disponibilizada para as equipes. A ordem de apresentação e

[33] http://neotec.rc.unesp.br/

a abreviação dos nomes dos atributos nos arquivos *shapefile*, que são limitados a dez caracteres, foi definido por cada equipe. Os valores dos atributos, quando não definidos como texto livre, foram indicados como listas de valores possíveis. Com esta organização, foi gerado um conjunto de dados que, apesar de seguir orientações mínimas, pode ser considerado heterogêneo.

Com o objetivo de estabelecer um comparativo entre a abordagem de integração sintática e a integração semântica, foi realizado o uso do SID com a importação direta de dados em formato *shapefile* e planilha de dados. A importação foi realizada de maneira simples, sem modificação da estrutura dos dados importados, para gerar um exemplo de integração apenas em nível sintático. Os dados foram importados e inseridos em tabelas de um banco de dados PostgreSQL. Posteriormente, os dados das tabelas foram publicados no Geoserver, de modo a disponibilizar os dados nos padrões OGC, o que permitiu exibir os mapas na interface do sistema.

Entretanto, o nível de integração provido pelos serviços OGC, que são baseados em XML, não garante total integração se não houver um padrão de metadados definido. Desta forma, não foi possível realizar a verificação dos dados, como por exemplo, a existência de atributos obrigatórios, ou mesmo a validade dos valores definidos. Isto também tornou a interação do usuário com o sistema mais complicada. Por não haver padronização, a associação entre dados relacionados tornou-se complexa, uma vez que o nome indicado para atributos chave variava em cada conjunto de dados.

Sem o uso de metadados, o principal exemplo de dificuldade de manipulação uniforme dos dados é a definição de estilos para os mapas publicados nos padrões OGC. O formato utilizado para definir as cores e formas com que as feições devem ser apresentadas é o SLD[34] (*Styled Layer Descriptor*), um padrão de descrição baseado em XML no qual regras de estilo são definidas a partir dos atributos e seus valores. Na primeira experiência de importação, como os atributos eram heterogêneos, um XML teve que ser gerado para cada conjunto de dados, ou seja, não foi possível utilizar um único arquivo de estilo ou automatizar a definição de estilo.

[34] http://www.opengeospatial.org/standards/sld

O uso de metadados permite solucionar alguns destes problemas, como padronizar os atributos e definir algumas regras de estruturação. Entretanto, o nível de informação semântica neste caso não estaria disponível para processamento do sistema. Caberia ao usuário adequar seus dados à semântica definida pelos metadados. Além disto, não seria possível de definir no SID formas para permitir verificar inconsistências semânticas de relacionamento e valores de atributos, muito menos gerar novas informações a partir de inferência.

Com a utilização do protótipo do *framework*, os aspectos que seriam endereçados pelo uso de metadados foram solucionados pela anotação semântica, realizada por meio da transformação dos esquemas de dados em RDF e posterior alinhamento com as ontologias do SID. Por meio da descrição semântica dos dados, também foi possível realizar inferências para detecção de inconsistências na descrição dos dados, bem como possíveis conflitos, duplicações e/ou combinações entre os dados integrados. O processo de inferência ainda permitiu a expansão da descrição dos dados, no sentido de classificação do tipo de informação e na expressão de atributos inferidos. Esta expansão garante uma interação mais ampla com os dados no SID, conforme será evidenciado no Capítulo 6.

Nas seções seguintes, são apresentados detalhes das implementações de cada componente do *framework* no protótipo apresentado.

5.3 *Semantic Middleware*

O *Semantic Middleware* foi implementado como um processo WPS no Geoserver disponível na arquitetura do SID. A vantagem de utilizar o WPS é a capacidade de manipular dados geoespaciais que já estão disponíveis no ambiente do Geoserver.

O processo WPS é uma classe Java instanciada quando a execução do processo é requisitada. Esta classe possui um método de execução principal que recebe os parâmetro e retorna os resultados do processo. A execução do processo é realizada pelo Geoserver.

As funcionalidades de cada componente do *Semantic Middleware* são implementadas em classes individuais, ou seja, há uma classe para cada componente. Estas classes são instanciadas e utilizadas pelo método principal do processo WPS de acordo com os parâmetro definidos para sua execução. Desta forma, o processo que representa o *Semantic Middleware* implementa os papéis do *RDF Mapper*, *RDF Enhancer*, *Query Translator* e *SPARQL Builder*, bem como a interface definida pelo *Query Execution Engine* de acordo com a necessidade da etapa de integração.

5.4 *Semantic Repository*

As ontologias utilizadas como referência no protótipo (*Reference Ontologies*) foram desenvolvidas com base no escopo do domínio do Sistema de Integração de Dados. Estas ontologias, apresentadas na seção 3.3.1, foram divididas em três ontologias de domínio e uma ontologia de aplicação, respectivamente: Ontologia Neotectônica, que descreve conceitos relacionados à Geologia Estrutural; Ontologia Neotectônica de Referências Bibliográficas, que descreve conceitos relacionados à bibliografia, uma vez que as fontes de dados do SID são em sua maioria compilações da literatura do domínio; Ontologia Neotectônica Espacial, que descreve conceitos relacionados ao domínio geoespacial, como coordenadas e representações geométrica e espacial; e finalmente a Ontologia Neotectônica de Aplicação, uma ontologia de aplicação que importa as demais ontologias e complementa a descrição dos conceitos de domínio com conceitos e propriedades pertinentes ao escopo da Web Application do SID. A ontologia de aplicação também possui as regras utilizadas pela análise semântica realizada na etapa de alinhamento de instâncias.

Todas as ontologias foram desenvolvidas utilizando o software *Protégé*[35] e exportadas para arquivos no formato RDF/XML. Os arquivos gerados foram importados para o Apache Fuseki, a implementação do *Semantic Repository* do protótipo.

[35] http://protege.stanford.edu/

O Apache Fuseki garante que as ontologias fiquem disponíveis em todo o ambiente por meio do *Endpoint SPARQL*, que responde a *queries* de consulta e atualização.

5.5 *Alignment* Server

Conforme discutido na seção 4.4.4, o *Alignment Server* pode possuir diferentes métodos para alinhamento de esquemas e instâncias, ou apenas um método que seja capaz de separar as execuções destas diferentes etapas. No protótipo do *framework* foi desenvolvido apenas um método de alinhamento, pois desta forma, o algoritmo ficou concentrado em um único projeto de desenvolvimento. Isto permitiu com que alguns métodos fossem compartilhados pelas duas etapas, como, por exemplo, o acesso às *Reference Ontologies* e os algoritmos de comparação de termos. De qualquer forma, o método foi implementado de modo a permitir a execução independente do alinhamento de esquemas e alinhamento de instâncias.

Nas subseções seguintes, são apresentados maiores detalhes da implementação das duas etapas de alinhamento.

5.5.1 Implementação do *Schema Matcher*

De acordo com o que foi discutido na seção 2.3, um algoritmo de alinhamento de ontologias pode ser determinístico ou não determinístico. Nesta implementação do *Schema Matcher*, um algoritmo determinístico foi escolhido, pois é uma solução mais simples, com eficiência suficiente para alinhar os esquemas das fontes de dados (*RDF-sch*) com as ontologias (*Reference Ontologies*). No escopo do protótipo, os esquemas não possuem mais do que algumas dezenas de atributos, e as ontologias não chegam a atingir milhares de conceitos. Além disso, a menor complexidade do algoritmo determinístico diminui o tempo de execução, e consequentemente o tempo de espera do usuário, que deve validar o resultado gerado (*Schema Alignment*) para que o processo de

integração tenha continuidade. No Algoritmo 1 é apresentado o algoritmo geral do método para realizar o papel do *Schema Matcher*.

```
1  //Global list of alignment matches found between entities
2  global alignmentCellIterator;
3
4  /**
5  * ontology1: The source ontology - Usually the RDF-sch
6  * ontology2: The target ontology - Usually the Reference Ontologies
7  * params: The alignment method configuration parameters
8  */
9  matchSchema( ontology1, ontology2, params)  {
10
11     List sourceEntities = New list to store the source ontology entities;
12     List targetEntities = New list to store the target ontology entities;
13
14     if(params.SCHEMA_ALIGNMENT == true){
15         Add the Classes, DataPropeties and ObjectProperties from ontology1 to sourceEntities;
16         Add the Classes, DataPropeties and ObjectProperties from ontology2 to targetEntities;
17
18         //Test if the entities were really loaded from the ontologies.
19         if(sourceEntities.size() == 0 OR targetEntities.size() == 0){
20             Throw error and stop.
21         }
22
23         //Iterate over the entities of both ontologies, starting from the source entities.
24         FOR EACH sourceEntity IN sourceEntities {
25             //List to keep the possible matches found for the source entity
26             possibleMatches = New map struture to keep the possible matches found.
27
28             FOR EACH targetEntity IN targetEntities{
29                 //Perform String Matching
30                 double score = stringMatch(sourceEntity, targetEntity);
31                 //Perform Structural Matching and increment the score
32                 score += structureMatch(sourceEntity, targetEntity);
33
34                 //Test against the score threshold defined into alignment parameters
35                 if(score >= params.THRESHOLD){
36                     Add the targetEntity and score to possibleMatches;
37                 }
38             }
39             //Create an alignment correspondence for each match that satisfy the score threshold
40             FOR EACH match IN possibleMatches {
41                 alignCell = Alignment(sourceEntity, match.targetEntity, "equivalence", match.score );
42                 Add the alignCell to a alignmentCellIterator.
43             }
44         }
45     }
46 }
```

Algoritmo 1: Algoritmo do método de alinhamento de esquemas.

O primeiro requisito para o método é definir uma lista para armazenar todas as correspondências armazenadas (Linha 2). A lista é global, pois o método foi implementado com os padrões da *Alignment API*, que indica que esta lista deve ser acessível pelo *Alignment Server* fora do escopo do método.

Os parâmetros necessários para o *matchSchema* são as duas ontologias a serem alinhadas e os parâmetros de configuração da execução (Linhas 4 – 9). O *matchSchema* inicia a criação de duas listas para armazenar as entidades das duas ontologias (Linhas 11 e 12). Quando iniciadas, as listas de entidades são preenchidas com as classes e todos os tipos de propriedades das ontologias (Linhas 15-21). Com estas listas preenchidas, o algoritmo percorre as entidades da primeira ontologia para listar as possibilidades de equivalência com cada entidade da segunda ontologia (Linhas 23 – 28). Para cada par de entidades, o *matchSchema* executa uma comparação textual, ou Casamento Terminológico (Linha 30). No caso deste método todas as *labels* de mesmo idioma definidas pelas propriedades *rdfs:label*, *skos:prefLabel* e *skos:altLabel* são comparadas por meio do algoritmo SMOA (CROCHEMORE, 1992), cuja implementação é disponibilizada pela *Alignment API*. A medida de similaridade encontrada é armazenada na variável *score*.

Após a comparação textual, uma comparação estrutural, ou Casamento Baseado em Estrutura, é realizado (Linha 32), de forma que: se duas classes ou duas propriedades foram alinhadas na primeira etapa, o resultado da comparação estrutural é um valor arbitrário positivo. Porém, se uma classe foi alinhada com uma propriedade, o resultado da comparação estrutural é um valor arbitrário negativo. Este valor arbitrário pode ser modificado por um parâmetro de execução. O resultado é somado ao *score* da etapa anterior, o que reforça a equivalência se o resultado for positivo, e penaliza a equivalência se o resultado for negativo. Se o *score* final, sempre nivelado entre 0.0 e 1.0, estiver no limite de aceitação definido pelos parâmetros de execução, a equivalência é adicionada à lista de possíveis combinações para as entidades (Linhas 35 - 37). Ao final da iteração entre todas as entidades, uma lista de alinhamentos é gerada no formato definido pela *Aligment API* (Linhas 40 – 43).

O alinhamento de esquemas não requer um nível de precisão elevado, uma vez que o alinhamento gerado pelo *Schema Matcher* é validado pelo usuário.

5.5.2 Implementação do *Instance Matcher*

Conforme a seção 4.4.6, o processo de alinhamento de instâncias realizado pelo *Instance Matcher* tem como objetivo padronizar a descrição dos dados, e encontrar possíveis duplicações, combinações e conflitos entre as instâncias de dados geológicos.

O algoritmo implementado no protótipo segue a concepção da tarefa definida pelo *framework*, ou seja, realiza duas etapas para atingir o objetivo de integrar instâncias de dados: 1) uma tradução da descrição das instâncias; e, 2) uma análise semântica. A tradução consiste em descrever as instâncias de dados com o vocabulário das *Reference Ontologies*. A análise semântica é executada por um mecanismo de inferência para detectar possíveis duplicações, combinações e/ou conflitos. No Algoritmo 2 é apresentado o algoritmo geral do método de alinhamento de instâncias.

```
1  //Global list of alignment matches found between entities
2  global alignmentCellIterator;
3
4  /**
5  * ontology1: The source ontology - Usually the RDF-data
6  * ontology2: The target ontology - Usually the Reference Ontologies
7  * params: The alignment method configuration parameters
8  */
9  matchInstances( ontology1, ontology2, params) {
10
11     if(params.INSTANCES_ALIGNMENT == true){
12
13         translatedInstances = new OntModel to keep the translated instances;
14
15         sourceModel = Get the ontology1 model; //Usually with instances
16         targetModel = Get the ontology2 model;
17
18         if(params.INSTANCE_ANALYSIS_DIRECTION == "SOURCE_TO_TARGET"){
19             translatedInstances = translateInstances(sourceModel, targetModel);
20         }else if(params.INSTANCE_ANALYSIS_DIRECTION == "TARGET_TO_SOURCE"){
21             //If the RDF-data ontology is set as the target ontology (or ontology2)
22             translatedInstances = translateInstances(targetModel, sourceModel);
23         }
24
25         //Annotations like the source of data, the timestamp of the translation
26         Add annotations to the translatedInstances model;
27
28         if(params.INSTANCES_SEMANTIC_ANALYSIS == true){
29             Add the Reference Ontologies model to the translatedInstances model;
30             //Run the inference to check inconsistency
31             inferredInstances = GeoSWRLReasoning.execute(translatedInstances);
32
33             if(inferredInstances == null){ //Something went wrong
34                 inconsistency = Get the GeoSWRLReasoning inconsistency message;
35                 Throw error with the inconsistency message;
36             }
37
38             Run SPARQL query to get combinations, duplications and conflicts relations;
39
40             FOR EACH CONFLICT|DUPLICATION|COMBINATION {
41                 alignCell = Alignment(individual1, individual2, "relation", null );
42                 Add the alignCell to a alignmentCellIterator;
43             }
44
45             Store the inferredInstances into the Semantic Repository;
46         }else {
47             Store the translatedInstances into the Semantic Repository;
48         }
49 }
```

Algoritmo 2: Algoritmo do método de alinhamento de instâncias.

O método de alinhamento de instâncias também segue o padrão definido pela *Alignment API* e armazena seus resultados em uma lista global (Linha 1). Os parâmetros necessários para os métodos são as duas ontologias a serem alinhadas e os parâmetros de configuração para execução (Linhas 4 – 9). Na implementação realizada, a primeira ontologia (*ontology1* ou *source ontology*) será sempre a descrição dos dados a serem integrados (*RDF-data*) e a segunda ontologia (*ontology2* ou *target ontology*) serão sempre as *Reference Ontologies*. O algoritmo deve ser sempre executado de modo a

traduzir as instâncias do *RDF-data* para o vocabulário das *Reference Ontologies*, então um parâmetro de configuração é utilizado para definir a ontologia *RDF-data* e as *Reference Ontologies* (Linhas 18 – 23). O próximo passo executado é a tradução, implementada em um método separado do principal, o *translateInstances* (Linhas 19 e 22).

A tradução da descrição das instâncias de dados utiliza como base o *Schema Alignment* gerado pelo *Schema Matcher*. Esta tradução pode parecer, inicialmente, o simples processo de acessar as equivalências definidas no *Schema Alignment*, mas alguns cuidados devem ser tomados.

No processo de tradução, todos os indivíduos do *RDF-data* são percorridos e novos indivíduos são criados com as propriedades e classes das *Reference Ontologies* que são equivalentes ao tipo e atributos definidos originalmente para os dados. Entretanto, os valores destes atributos podem variar de acordo com os tipos de propriedades identificadas como equivalentes no *Schema Alignment*. Desta forma, os valores textuais dos atributos que são equivalentes à *DatatypeProperties* permanecem inalterados, já os valores dos atributos que são equivalentes à *ObjectProperties* podem sofrer alterações dependendo do tipo de valor encontrado:

- Se o valor for um URI, um indivíduo pré-existente é buscado e associado como valor da propriedade.
- Se o valor for uma unidade de medida, esta informação é decomposta em valor numérico e unidade e é representada como um indivíduo do tipo de unidade de medida indicado pelo atributo.
- Se o valor for textual, indivíduos que representam aquele conceito são testados para encontrar uma equivalência, utilizando os algoritmos de comparação de termos, com o cuidado de verificar se o indivíduo pertence ao *range* da *ObjectProperty* mapeada.

Outro cuidado com relação à tradução para *ObjectProperties* são os atributos indiretos, isto é, atributos que não pertencem diretamente ao conceito descrito, mas

pertencem a uma parte constituinte ou relacionada ao conceito. Na Figura 22, um exemplo deste tipo de relação indireta é apresentado.

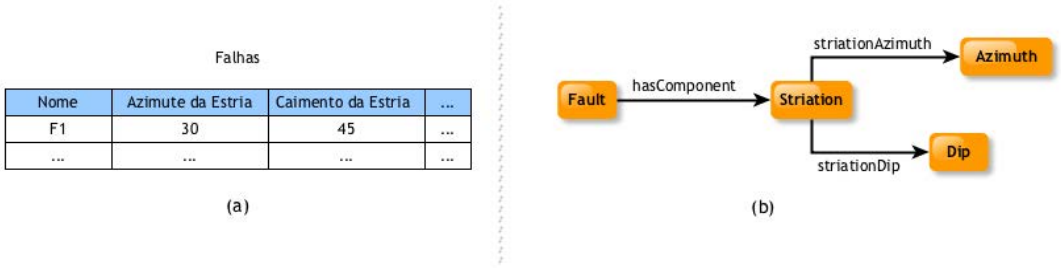

Figura 22: Relação indireta de propriedade.

No exemplo da Figura 22 (a), os atributos "Azimute da Estria" e "Caimento da Estria" são associados diretamente a uma falha quando esta é representada em uma tabela. Porém, estes atributos indicam a atitude das estrias que fazem parte do plano de falha, e não são relacionados diretamente à falha. Esta relação é detalhada na Figura 22 (b) que representa a descrição destes conceitos nas *Reference Ontologies*.

O algoritmo de tradução verifica o domínio definido para o *ObjectProperty* e aplica correções nos relacionamentos quando necessário, para garantir a descrição de acordo com a definição conceitual das *Reference Ontologies*.

Uma vez que os indivíduos estejam traduzidos, isto é, descritos com um único vocabulário, o método *matchInstances* do Algoritmo 2, executa a análise semântica de acordo com o valor de um parâmetro de configuração. Se o valor for falso, o conjunto de indivíduos traduzidos é armazenado no *Semantic Repository* (Linha 47) sem nenhum processamento adicional. Neste caso a análise pode ser realizada posteriormente. Se estiver definido o contrário, a análise semântica é realiza.

O método *execute* do *GeoSWRLReasoning* (Linha 31), que executa a análise semântica, deve receber os indivíduos traduzidos combinados com as *Reference Ontologies* (Linha 29), para que o processo de inferência seja realizado. O *GeoSWRLReasoning* utiliza o raciocinador Pellet (SIRIN et al., 2007) para gerar inferências sobre esta ontologia combinada. O processo de inferência verifica a consistência dos indivíduos no que diz respeito à definição de tipos e relações de

atributos. Se houver algum erro, isto é, alguma inconsistência semântica na descrição dos indivíduos, a análise é interrompida e um aviso é enviado para a aplicação que solicitou o processo de integração (Linhas 33 – 36). Se não houver erro, a inferência expande classificações dos indivíduos e infere valores de atributos. Além disso, todas as regras SWRL descritas nas *Reference Ontologies* são aplicadas durante a inferência e, neste momento, o raciocinador define relações entre pares de instâncias de dados, que são detectadas como casos de possíveis combinações, duplicações ou conflitos, dependendo da regra ativada pelas instâncias. O resultado da inferência é acrescentado à descrição pré-existente, que retorna ao método *matchInstances*.

Com base na inferência gerada, o *matchInstances* consulta todos os casos de duplicações, combinações e conflitos e constrói uma lista de alinhamentos no formato definido pela *Aligment API* (Linhas 38 – 43). Este alinhamento (*Instance Alignment*) é armazenado no *Alignment Server* para ser posteriormente recuperado pela *Web Application*. O conjunto de dados gerados na tradução, junto com as descrições inferidas, é armazenado no *Semantic Repository*.

Na próxima seção é apresentada a extensão que foi desenvolvida para permitir ao raciocinador Pellet executar as funções de manipulação de dados espaciais, quando estas funções são utilizadas em regras SWRL. Esta capacidade é necessária porque algumas regras são utilizadas para detectar conflitos na relação espacial de indivíduos.

5.5.3 GeoSWRL

A linguagem SWRL, utilizada para compor as regras de análise de instâncias, permite a utilização de algumas funções na descrição das regras. Estas funções, denominadas *Built-Ins*, permitem realizar operações com indivíduos e suas propriedades. Há Built-Ins disponíveis para comparações, operações matemáticas, manipular tipos de dados como *boolean*, *strings*, e datas, além de manipular URIs e listas. Entretanto, não há suporte para dados geoespaciais. Este suporte é necessário para permitir a construção de regras que levem em consideração a relação espacial dos dados geológicos para inferir novas informações.

Para analisar um caso de possível combinação de falhas geológicas, por exemplo, o primeiro passo é verificar se as estruturas estão alinhadas próximas uma da outra. A seguinte sentença explicita o conhecimento utilizado nesta análise em linguagem natural:

"Se duas falhas de mesmo tipo, F1 e F2, estiverem alinhadas muito próximas uma da outra, mas não estiverem sobrepostas, então é possível que seja definida uma combinação entre F1 e F2"

Com o vocabulário e funções padrões do SWRL, não é possível codificar esta sentença em uma regra, pois não existem funções para determinar se duas entidades estão alinhadas, muito próximas ou sobrepostas, ou seja, não é possível determinar, a partir de operações com as informações espaciais, qual a relação topológica de duas entidades.

A solução adotada foi utilizar um recurso disponível no raciocinador Pellet que permite definir funções personalizadas para serem utilizadas em regras SWRL. Assim, foi desenvolvida uma extensão da linguagem SWRL, denominada GeoSWRL.

A extensão GeoSWRL foi desenvolvida com o intuito de fornecer à linguagem SWRL funções capazes de manipular dados geoespaciais. Esta extensão foi desenvolvida para ser utilizada com o Pellet, que é um dos poucos raciocinadores de código aberto e licença livre com suporte adequado à linguagem SWRL. A extensão foi desenvolvida com a linguagem Java, assim como o Pellet. Desta forma, para realizar operações com dados espaciais, foram utilizados métodos fornecidas pelo GeoTools[36], uma biblioteca Java que oferece ferramentas para manipular dados espaciais[37].

A extensão GeoSWRL implementa basicamente algumas das funções do GeoTools para manipular de dados espaciais, ou utiliza as funcionalidades disponíveis na biblioteca para implementar as funções. Todas as funções recebem pelo menos dois parâmetros principais, (*?wkt1* e *?wkt2*). O GeoSWRL pressupõe o uso do vocabulário

[36] http://www.geotools.org/

[37] O GeoTools é a base do servidor de mapas Geoserver

GEOSPARQL para descrever geometrias e outras informações geoespaciais, portanto, os dois parâmetros principais das funções devem ser serializações WKT[38] (*Well Known Text*) de geometrias que estejam relacionadas aos indivíduos que representam dados geológicos. A relação das funções implementadas é apresentada a seguir:

Função *covers*: Verifica se uma geometria cobre a outra.

geoswrl:covers(?wkt1,?wkt2)

Retorna VERDADEIRO se *?wkt1* cobre *?wkt2*.

Função *crosses*: Verifica se uma geometria cruza com a outra.

geoswrl:crosses(?wkt1,?wkt2)

Retorna VERDADEIRO se *?wkt1* cruza *?wkt2*.

Função *disjoint*: Verifica se uma geometria é completamente separada da outra.

geoswrl:disjoint(?wkt1,?wkt2)

Retorna VERDADEIRO se *?wkt1* é completamente separada de *?wkt2*.

Função *isWithinDistance*: Verifica se uma geometria está a uma distância máxima de outra.

geoswrl:isWithinDistance(?wkt1,?wkt2,?limit)

Retorna VERDADEIRO se *?wkt1* está a uma distância *?limit* de *?wkt2*.

Função *intersects*: Verifica se uma geometria intercepta a outra.

geoswrl:intersects(?wkt1,?wkt2)

Retorna VERDADEIRO se *?wkt1* intercepta *?wkt2*.

Função overlaps: Verifica se uma geometria se sobrepõe a outra.

geoswrl:overlaps(?wkt1,?wkt2)

Retorna VERDADEIRO se *?wkt1* sobrepõe *?wkt2*.

Função *touches*: Verifica se uma geometria apenas toca a outra.

geoswrl:touches(?wkt1,?wkt2)

[38] http://www.opengeospatial.org/standards/sfa

Retorna VERDADEIRO se *?wkt1* apenas toca *?wkt2*.

Função *within*: Verifica se uma geometria está completamente dentro da outra.

geoswrl:within(?wkt1,?wkt2)

Retorna VERDADEIRO se *?wkt1* está completamente dentro de *?wkt2*.

Função *angleBetween*: Verifica se o ângulo entre duas linhas está dentro de um valor máximo.

geoswrl:angleBetween(?wkt1,?wkt2,?limit)

Retorna VERDADEIRO se *?wkt1* e *?wkt2* são linhas e o ângulo entre elas é no máximo *?limit*.

Com o uso das funções criadas no GeoSWRL, é possível expressar a sentença apresentada anteriormente como a seguinte regra SWRL:

```
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>
PREFIX swrl: <http://www.w3.org/2003/11/swrl#>
PREFIX geoswrl: <http://neotec.rc.unesp.br/resource/geoswrl/>
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>

neotec:Normal(?f1), neotec:Normal(?f2),
geosparql:hasGeometry(?f1, ?geom1),
geosparql:hasGeometry(?f2,?geom2), geosparql:asWKT(?geom1,
?wkt1), geosparql:asWKT(?geom2,?wkt2),
swrl:DifferentFrom(?f1,?f2), geoswrl:disjoint(?wkt1,?wkt2),
geoswrl:angleBetween(?f1,?f2,0.0),
 geoswrl:isWithinDistance(?f1,?f2, 0.01) ->
neotec:possiblyCombinesWith(?f1,?f2)
```

A regra acima verifica os seguinte critérios:

SE

`neotec:Normal(?f1)`, *f1* é uma falha normal

`neotec:Normal(?f2)`, *f2* é uma falha normal

`geosparql:hasGeometry(?f1, ?geom1)`, *f1* possui uma geometria *geom1*

`geosparql:hasGeometry(?f2,?geom2)`, *f1* possui uma geometria *geom2*

`geosparql:asWKT(?geom1, ?wkt1)`, O WKT de *geom1* é *wkt1*

`geosparql:asWKT(?geom2,?wkt2)`, O WKT de *geom2* é *wkt2*

`swrl:DifferentFrom(?f1,?f2)`, *f1* e *f2* são distintas

`geoswrl:disjoint(?wkt1,?wkt2)`, *f1* e *f2* não são sobrepostas

`geoswrl:angleBetween(?f1,?f2,0.0)`, *f1* e *f2* são paralelas

`geoswrl:isWithinDistance(?f1,?f2, 0.01)` *f1* e *f2* estão dentro de uma distância de 1Km[39]

ENTÃO

`neotec:possiblyCombinesWith(?f1,?f2)` *f1* possivelmente combina com *f2*.

Devido a limitações do SWRL, não é possível definir uma única regra que verifique se dois indivíduos pertencem a um mesmo tipo (classe) de maneira genérica, portanto, para cada tipo de falha, uma regra semelhante deve ser definida.

Exemplos de aplicação das regras implementadas no protótipo apresentado neste capítulo são exibidos no Capítulo 6.

5.6 Conclusão do Capítulo

Neste capítulo foi descrita a implementação de um protótipo de acordo com as definições apresentadas do *Framework para Integração de Dados Geoespaciais*. Esta implementação foi desenvolvida dentro do escopo de um projeto de pesquisa, em caráter de protótipo, com a finalidade de verificar a viabilidade real da organização proposta conceitualmente pelo *framework*.

A implementação do protótipo permitiu analisar diversos aspectos específicos de implementação de cada parte do *framework* proposto neste trabalho. Algumas tecnologias e ferramentas foram utilizadas sem modificações, e outras precisaram de desenvolvimento, como o caso do GeoSWRL.

[39] A unidade utilizada pela função *isWithinDistance* é o grau, 1 grau é aproximadamente 111Km.

A extensão GeoSWRL é uma importante contribuição deste trabalho, e sua necessidade foi identificada devido à implementação do *framework*, já que a limitação da linguagem SWRL com relação a dados espaciais não foi identificada na etapa de concepção conceitual.

No Capítulo 6, alguns estudos de caso são apresentados como forma de validar a implementação apresentada neste capítulo e identificar, de maneira objetiva, as principais contribuições do *Framework para Integração de Dados Geoespaciais*.

6 Estudos de Caso

Neste capítulo são apresentados alguns estudos de casos realizados com base na implementação do *Framework para Integração de Dados Geoespaciais* apresentada no Capítulo 5. Estes estudos de caso, caracterizados como testes de funcionalidades, têm por finalidade realizar experimentos de integração de dados com a abordagem semântica e avaliar as contribuições que este tipo de integração pode apresentar.

Os estudos de caso estão organizados de modo a destacar o objetivo do teste e os dados utilizados. Em seguida, é apresentada a descrição do caso e dos resultados. As discussões gerais sobre os resultados de todos os casos são apresentadas na seção de conclusão do capítulo.

6.1 Processo de Integração

O primeiro estudo de caso envolve a realização de testes de integração de dados geológicos no SID apresentado nas seções 5.1 e 5.2 do capítulo anterior. O processo de integração de dados neste sistema em questão é realizado pelo protótipo do *Framework* para Integração de Dados Geoespaciais.

Objetivo

O objetivo deste estudo de caso é demonstrar o uso da integração de dados geológicos por meio da abordagem semântica proposta pelo *framework.* Os resultados apresentados devem demonstrar a solução de algumas questões como a heterogeneidade dos dados e as possibilidades de processar os dados quando a semântica está explícita em suas descrições.

Dados Utilizados

O conjunto de dados utilizado para este teste são *shapefiles* gerados por equipes participantes do projeto Neotectônica (seção 5.2). Os arquivos contêm informações a respeito de falhas geológicas, tais como a geometria dos traços que representam as

falhas, e os atributos referentes a cada traço. Os atributos das falhas foram definidos pelas equipes que geraram os dados, com base na lista de recomendações presente no sistema.

Descrição do Estudo de Caso

Utilizado as funcionalidades do sistema Web, um acesso autenticado é feito pelo usuário. Para iniciar o processo de integração é selecionada, no menu "Gerência de Mapas", a opção "Integrar Dados". Na interface Web de integração, exibida na Figura 23, o usuário tem uma breve descrição da funcionalidade e a relação de opções disponíveis. Neste exemplo, a importação ocorre a partir de um conjunto de arquivos *shapefile*. Cabe ao usuário selecionar os arquivos a serem integrados.

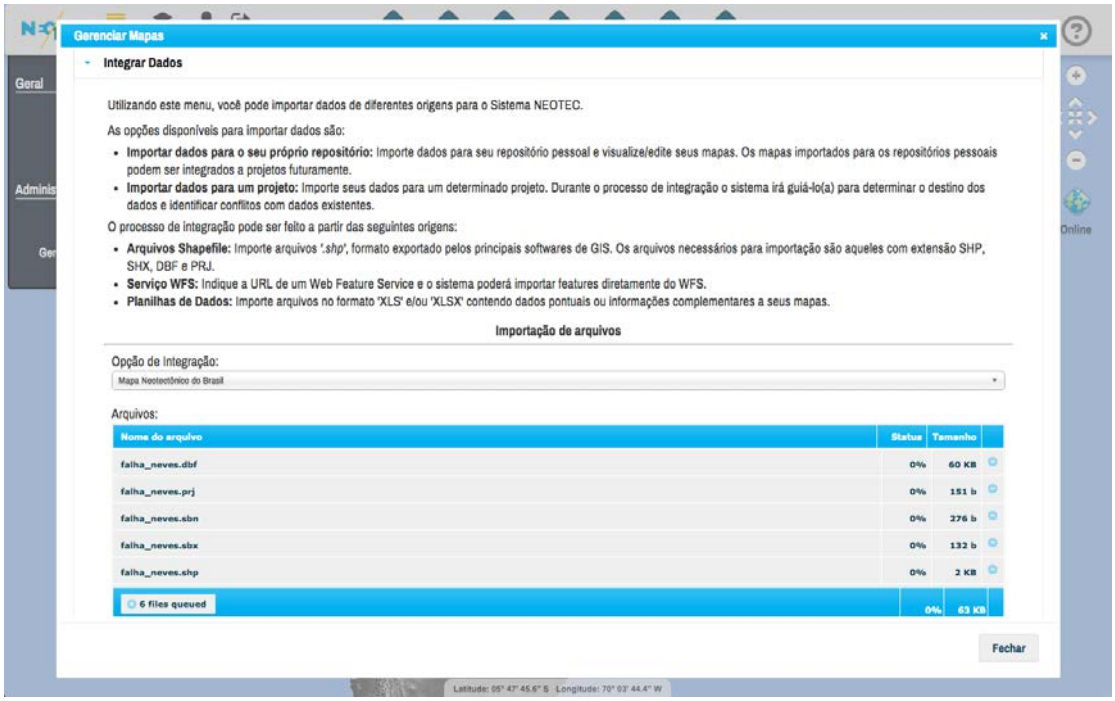

Figura 23: Interface do primeiro passo da Integração.

Quando os arquivos são enviados, o processo de integração, descrito nos Capítulos 4 e 5, é iniciado. A próxima interação do usuário, apresentada na Figura 24, é para a confirmação do alinhamento de esquemas. As correspondências encontradas são

divididas em “Tipo de Dado” e “Mapeamento de Atributos”. No primeiro, encontra-se o mapeamento do nome do arquivo com o conceito que indica o tipo de dado representado – no caso, Falha. No segundo estão listados, à esquerda, os nomes dos atributos definidos originalmente no arquivo, e à direita, uma lista de conceitos correspondentes. Ao centro está indicado o grau de correspondência, que pode ser no máximo 1.0 (totalmente equivalente), e no mínimo 0.75 (parcialmente correspondente), um valor arbitrário definido na configuração do sistema. As linhas marcadas em verde indicam as correções realizadas pelo usuário. Há ainda a possibilidade de ignorar atributos, que serão excluídos no mapeamento de esquema, e posteriormente ignorados na integração de instâncias.

Na confirmação do alinhamento de esquema é apresentado, para cada atributo, mais de uma opção de correspondência. Desta forma o usuário pode optar por alterar a indicação inicial do sistema. Nesta implementação é possível ao usuário visualizar detalhes dos conceitos, ou ainda selecionar outras opções de conceitos que não foram apresentadas inicialmente, ao selecionar a opção “Outro...” na lista de correspondências de um atributo.

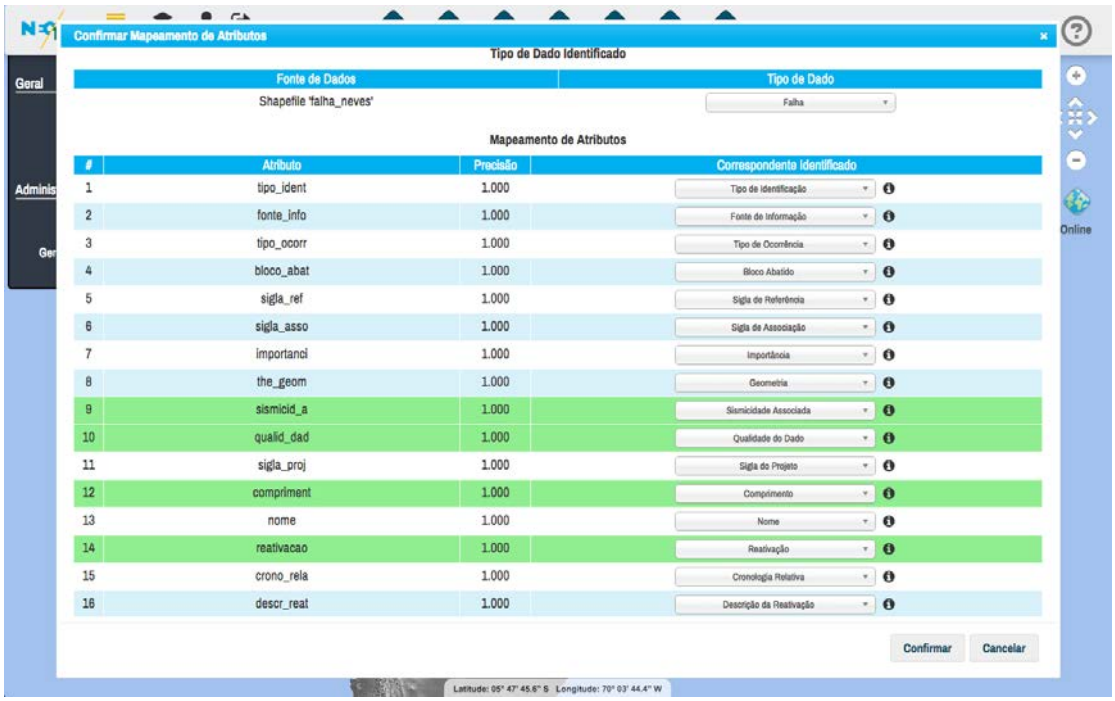

Figura 24: Interface para confirmação do alinhamento de esquema.

Após o usuário confirmar a avaliação do alinhamento de esquema, o processo continua internamente e são realizados os passos necessários para integração das instâncias de dados.

O processo de integração de instâncias pode gerar uma lista de inconsistências detectadas pela análise semântica realizadas nos dados. Estas inconsistências possuem caráter sugestivo, e fica sobre a responsabilidade do usuário validar ou rejeitar a sugestão feita pelo sistema. O usuário pode decidir em que momento analisar a lista de inconsistências gerada pelo processo de integração.

Os tipos de inconsistências que podem ser encontradas são: possíveis combinações; possíveis duplicações; e possíveis conflitos. Os tipos de inconsistências são detalhadas a seguir.

a) Possíveis Combinações

O primeiro tipo de inconsistência é definido como a possibilidade de combinar duas instâncias de dados que se enquadrem em determinados critérios. No caso de duas instâncias falhas geológicas, o critério foi estabelecido da seguinte forma:

"Se duas falhas de mesmo tipo, estiverem alinhadas muito próximas uma da outra, e não estiverem sobrepostas, então é possível que seja definida uma combinação entre estas falhas"

Esta definição conceitual foi utilizada para definir a seguinte regra SWRL:

```
PREFIX gs: <http://www.opengis.net/ont/geosparql#>
PREFIX gn: <http://www.geonames.org/ontology#>
PREFIX swrl: <http://www.w3.org/2003/11/swrl#>
PREFIX geoswrl: <http://neotec.rc.unesp.br/resource/geoswrl/>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>
PREFIX neotecapp:
<http://neotec.rc.unesp.br/resource/NeotectonicsApplication/>
```

```
neotec:Reverse(?f1), neotec:Reverse(?f2),
gs:hasGeometry(?f1,?g1), gs:hasGeometry(?f2,?g2), gs:asWKT(?g1,
?wkt1), gs:asWKT(?g2, ?wkt2), geoswrl:isWithinDistance(?wkt1,
?wkt2, 0.01), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:disjoint(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:possiblyCombinesWith(?f1, ?f2)
```

Esta regra estabelece uma relação de possível combinação entre duas falhas inversas que: não se sobrepõem, estejam a menos de 1Km uma da outra e cujo ângulo entre elas seja menor do que 30 graus. Neste caso, o conceito "alinhadas" é traduzido como sendo de linhas que possuem um determinado ângulo entre si, considerando um erro - no exemplo, 30 graus. O conceito "muito próximas" é traduzido como uma distância inferior a 1Km.

Todas as regras codificadas para identificar combinações estão listadas no APÊNDICE A.

b) Possíveis Duplicações

Outro tipo de inconsistência entre os dados integrados é a ocorrência de duplicações. Esta inconsistência é comum de ocorrer quando se trata de fontes de dados distintas que podem conter exatamente a mesma informação de forma duplicada, ou até mesmo a mesma representação de um objeto real, com aspectos diferentes observados sobre ele.

No caso de falhas geológicas, o critério estabelecido para identificar uma duplicação foi o seguinte:

"Se duas falhas do mesmo tipo estiverem alinhadas, de modo que ocorra sobreposição entre elas, então estas falhas podem ser uma duplicação"

Esta definição conceitual foi utilizada para definir a seguinte regra SWRL:

```
PREFIX gs: <http://www.opengis.net/ont/geosparql#>
```

```
PREFIX swrl: <http://www.w3.org/2003/11/swrl#>
PREFIX geoswrl: <http://neotec.rc.unesp.br/resource/geoswrl/>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>
PREFIX neotecapp:
<http://neotec.rc.unesp.br/resource/NeotectonicsApplication/>

neotec:Normal(?f1), neotec:Normal(?f2), gs:hasGeometry(?f1,?g1),
gs:hasGeometry(?f2,?g2), gs:asWKT(?g1, ?wkt1), gs:asWKT(?g2,
?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) -
> neotecapp:isPossibleDuplicationOf(?f1, ?f2)
```

Esta regra estabelece uma relação de possível duplicação entre duas falhas normais que se interceptam ao menos uma vez e o ângulo entre elas é menor do que 30 graus. Neste caso, o conceito "alinhadas" é traduzido como linhas que possuem um determinado ângulo entre si, considerando um erro - no exemplo, 30 graus. A ocorrência de sobreposição é traduzida como a ocorrência de intersecção entre os traços de falhas.

Todas as regras codificadas para identificar duplicações estão listadas no APÊNDICE A.

c) Possíveis Conflitos

O último tipo de inconsistência entre os dados integrados é a ocorrência de conflitos. Os conflitos no contexto de integração de dados geológicos são caracterizados por divergências na definição de atributos de uma mesma entidade. Como consequência, há também divergências na classificação dos dados.

No caso de falhas geológicas, o critério para caracterizar um conflito em relação à classificação foi definido da seguinte forma:

"*Se duas falhas de tipos diferentes estiverem alinhadas, de modo que ocorra sobreposição entre elas, então ocorre um conflito entre as falhas.*"

Esta definição foi traduzida para a seguinte regra SWRL:

```
PREFIX gs: <http://www.opengis.net/ont/geosparql#>
PREFIX gn: <http://www.geonames.org/ontology#>
PREFIX swrl: <http://www.w3.org/2003/11/swrl#>
PREFIX geoswrl: <http://neotec.rc.unesp.br/resource/geoswrl/>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>
PREFIX neotecapp:
<http://neotec.rc.unesp.br/resource/NeotectonicsApplication/>

neotec:Normal(?f1), (not neotec:Normal)(?f2), Feature(?f2),
gs:hasGeometry(?f1,?g1), gs:hasGeometry(?f2,?g2), gs:asWKT(?g1,
?wkt1), gs:asWKT(?g2, ?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2,
30.0), geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1,
?f2) -> neotecapp:possiblyConflictsWith(?f1, ?f2)
```

Esta regra estabelece uma relação de possível conflito entre duas falhas, uma falha normal e outra de qualquer outro tipo que se interceptam ao menos uma vez e cujo ângulo entre elas é menor do que 30 graus. Neste caso, o conceito “alinhadas” é traduzido como linhas que possuem um determinado ângulo entre si, considerando um erro (30 grau)s. A ocorrência de sobreposição é traduzida como a ocorrência de intersecção entre os traços de falhas.

Todas as regras codificadas para identificar conflitos estão listadas no APÊNDICE A.

Em todas as traduções para regras são realizadas aproximações devido à característica dos dados, que são linhas irregulares, muitas vezes com algum grau de curvatura. Os valores de distância e o ângulo nas regras são arbitrários e podem ser redefinidos de acordo com a necessidade. Por exemplo, o ângulo entre as geometrias pode ser ajustado para zero (ou próximo de zero) quando se desejar identificar linhas paralelas.

As regras utilizam um tipo específico de falha, pois, devido a limitações da linguagem SWRL, não é possível generalizar a identificação de tipos de dois indivíduos.

Em qualquer momento é possível modificar as regras existentes ou adicionar novas regras para cada tipo de inconsistência.

O processo de integração é finalizado completamente quando a integração de instâncias de dados termina. Como no exemplo na Figura 25, o usuário pode visualizar as falhas no mapa e começar a interagir com cada feição.

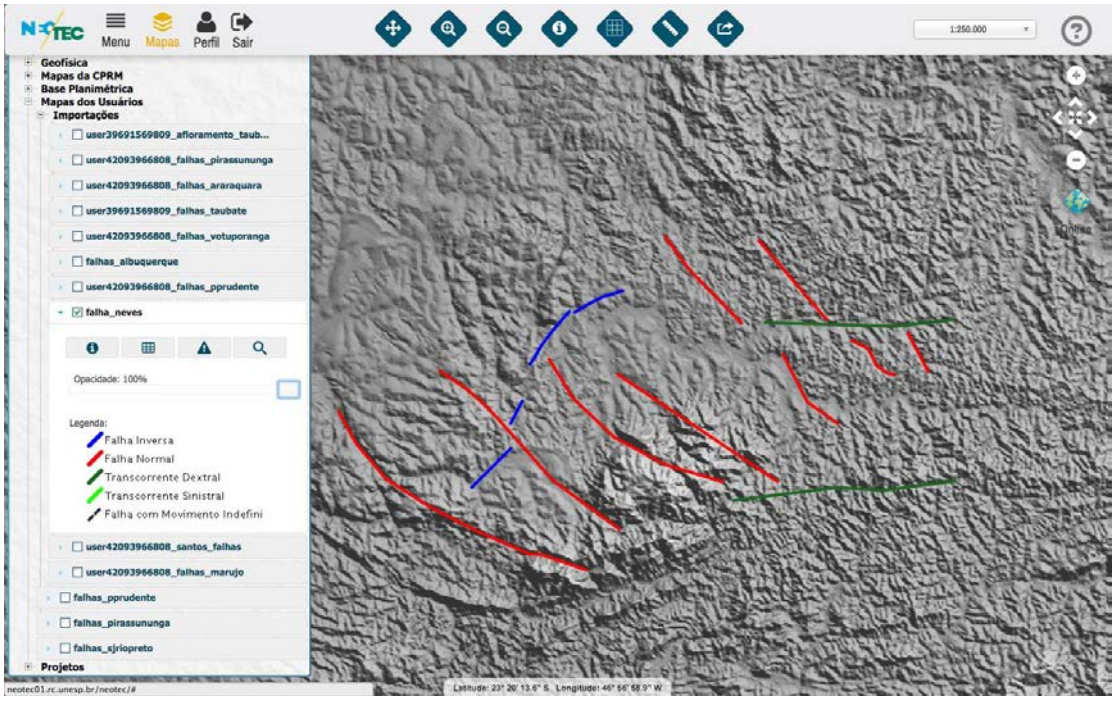

Figura 25: Interface com o resultado da integração de dados.

A classificação indicada na legenda do mapa, na Figura 25, foi gerada pelo processo de tradução e análise semântica, e permite ao SID aplicar um único estilo SLD pré-definido para qualquer conjunto de dados de falhas, de modo a exibir cada tipo de falha na sua cor correspondente.

Um dos resultados do processo de integração é a padronização da descrição dos dados integrados. Tanto os atributos quanto seus valores seguem os conceitos definidos nas *Reference Ontologies* do SID. Com base nesta padronização, é possível processar os atributos dos dados de maneira uniforme. Quando o usuário acessa os atributos de uma falha, por exemplo, um processamento é feito para criar links entre entidades, referências bibliográficas e outros, conforme exibido na Figura 26. O usuário também pode acessar as definições conceituais dos atributos e dos valores para entender seus significados.

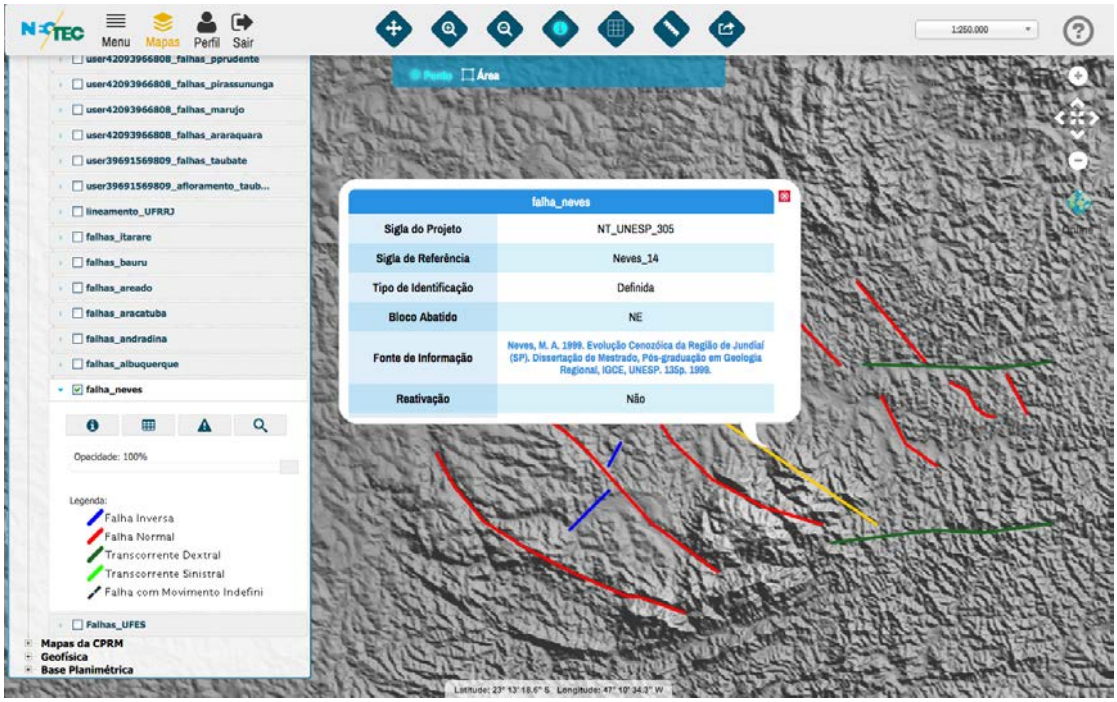

Figura 26: Atributos dos dados padronizados e processados.

O resultado da detecção de inconsistências é exibido para o usuário na forma de uma lista, conforme apresentado na Figura 27. Nesta lista, o usuário pode selecionar os casos para serem destacados no mapa (em amarelo) e também decidir o que deseja: combinar os traços de falha ou ignorar a sugestão. Se optar por combinar os dados, o usuário é guiado pelo sistema a modificar a geometria unificada e a decidir sobre a combinação de cada um dos atributos das falhas.

Os traços de falhas exibidos na Figura 27 como um caso de combinação, são considerados alinhados dentro da tolerância indicada no último parâmetro da função *geoswrl:angleBetween* porque esta função utiliza apenas os pontos iniciais e finais das geometrias para calcular o ângulo entre elas.

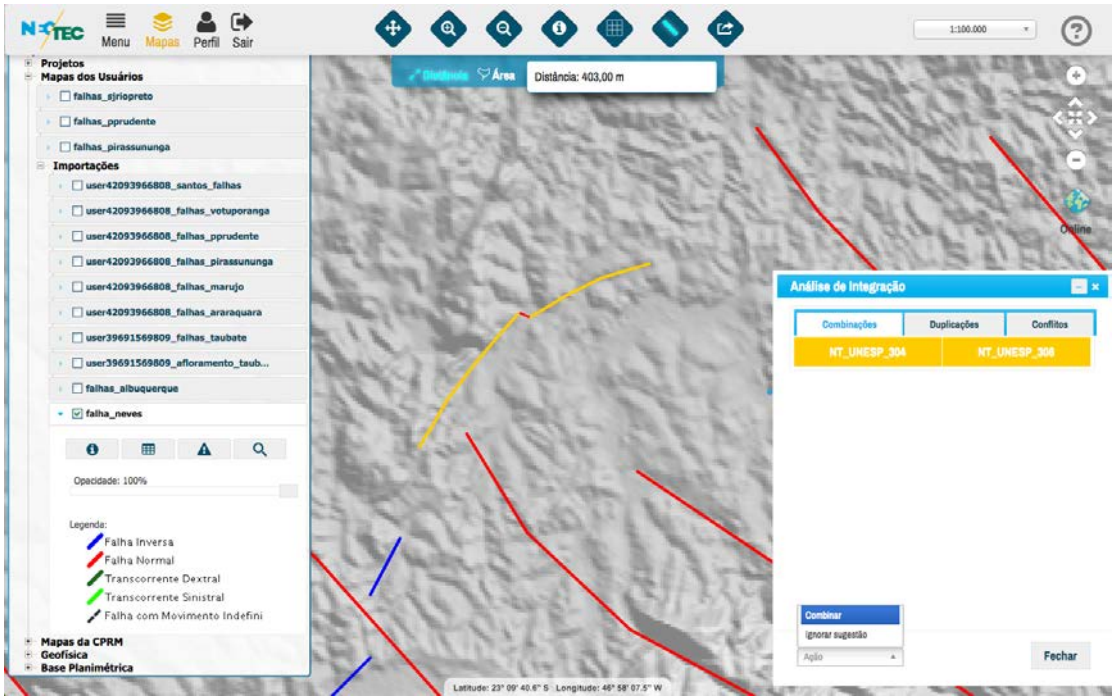

Figura 27: Exemplo de uma possível combinação indicada pelo sistema.

O processo de importação também foi realizado com outros conjuntos de dados. Estes testes foram realizados com o objetivo de verificar a ocorrência de outros casos de inconsistências.

O exemplo na Figura 28 é um caso de duplicação. Este problema é mais comum de ocorrer quando a produção dos dados integrados não foi realizada em um mesmo escopo de trabalho. Desta forma, para certificar a identificação de duplicações no processo de integração, o mesmo conjunto de dados foi importado duas vezes no SID. No exemplo da Figura 28 é possível ver apenas um conjunto de traços, pois os dados duplicados se sobrepõem. O destaque em amarelo mostra um traço mais fino e opaco e um traço exagerado com uma transparência para evidenciar que existem duas instâncias na mesma posição.

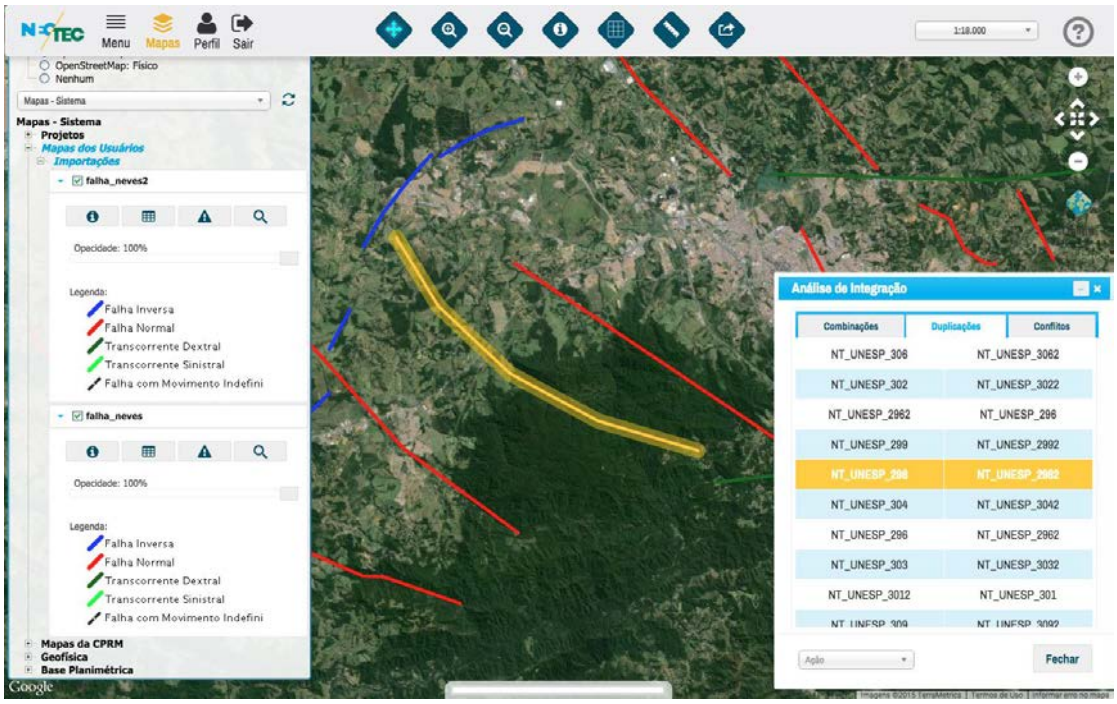

Figura 28: Exemplo de duplicações indicadas pelo sistema.

O exemplo na Figura 29 é um caso de conflito. Este conflito foi identificado quando dois conjuntos de dados, produzidos por equipes diferentes, foram integrados. Em cada conjunto, havia traços de falhas muito próximos que possuíam classificações diferentes, ou seja, cada equipe determinou um traço e uma classificação para o que possivelmente poderiam ser as mesmas falhas.

A falha destaca em amarelo na Figura 29 é uma falha de empurrão. É possível ver um traço vermelho, que indica uma falha normal, um pouco mais curto e praticamente alinhado ao traço destacado, caracterizando assim um conflito de classificações.

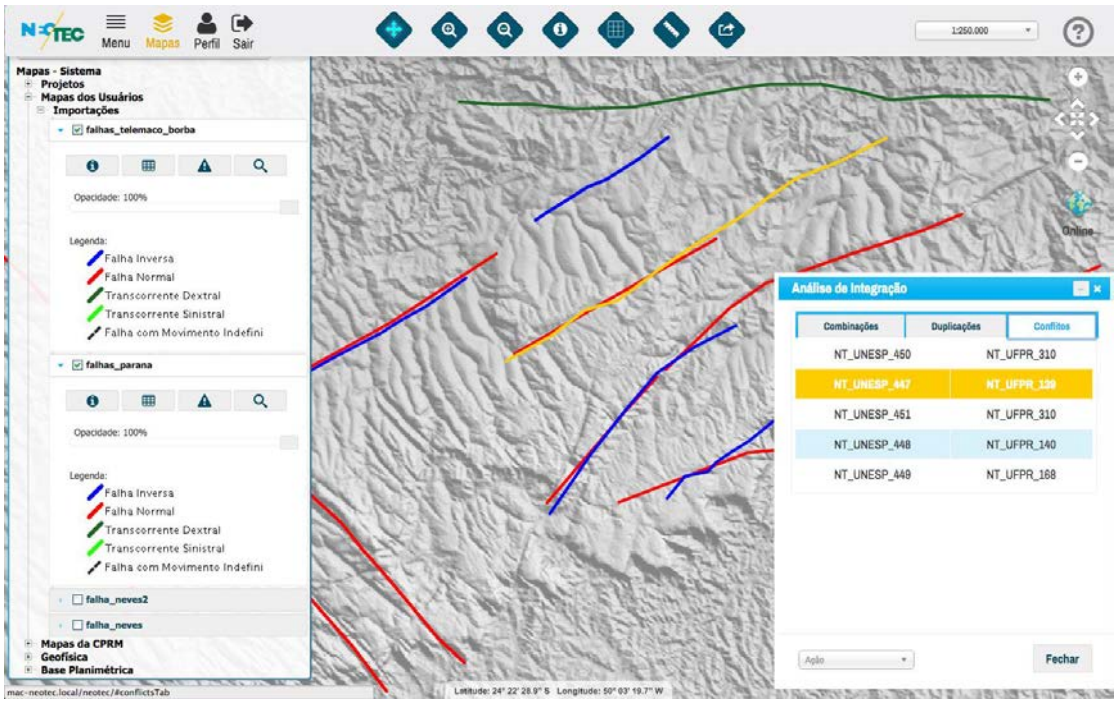

Figura 29: Exemplo de conflitos indicados pelo sistema.

O principal resultado do processo de integração é permitir a manipulação integrada dos dados. O caso de uso a seguir apresenta exemplos de integração de s dados utilizando a abordagem semântica.

6.2 Consulta Local

Uma vez que os dados estejam armazenados no *Semantic Repository* do SID, além de visualizá-los de maneira padronizada, é possível realizar consultas semânticas sobre eles. As consultas geralmente são realizadas para exibir os dados no mapa da interface do SID.

Neste estudo de caso, são apresentados alguns exemplos de consultas que podem ser realizadas.

Objetivo

O objetivo deste estudo de caso é demonstrar o acesso uniforme aos dados originados de diferentes fontes após a realização do processo de integração semântica proposta neste trabalho. Esta demonstração é apresentada na forma de algumas consultas SPARQL. Estas consultas, por sua vez, evidenciam as possibilidades de manipular os dados semanticamente.

Dados Utilizados

O conjunto de dados utilizado para este teste também são *shapefiles* gerados por equipes participantes do projeto Neotec (seção 5.2). Os arquivos, que contêm informações a respeito de falhas geológicas, foram integrados no SID seguindo o mesmo processo descrito no estudo de caso descrito na seção anterior.

Descrição do Estudo de Caso

No exemplo apresentado neste estudo de caso, alguns mapas, que são disponibilizados na interface do SID, são definidos a partir de consultas SPARQL aos dados armazenados no *Semantic Repository*. As consultas associadas às definições dos mapas também são armazenadas no *Semantic Repositor*y.

A seguir, são apresentados alguns mapas gerados através das consultas semânticas realizadas no conjunto de dados integrados no SID.

a) Mapa de todas as falhas

Exibe todas as falhas existentes no *Semantic Repository*. O mapa é construído a partir da seguinte consulta:

```
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>

DESCRIBE ?fault WHERE {
   ?fault a geosparql:Feature .
   ?fault a [rdfs:subClassOf* neotec:Fault]
```

```
}
```

O resultado da consulta, que exibe todos os testes com dados de falhas importados no sistema, é exibido na Figura 30.

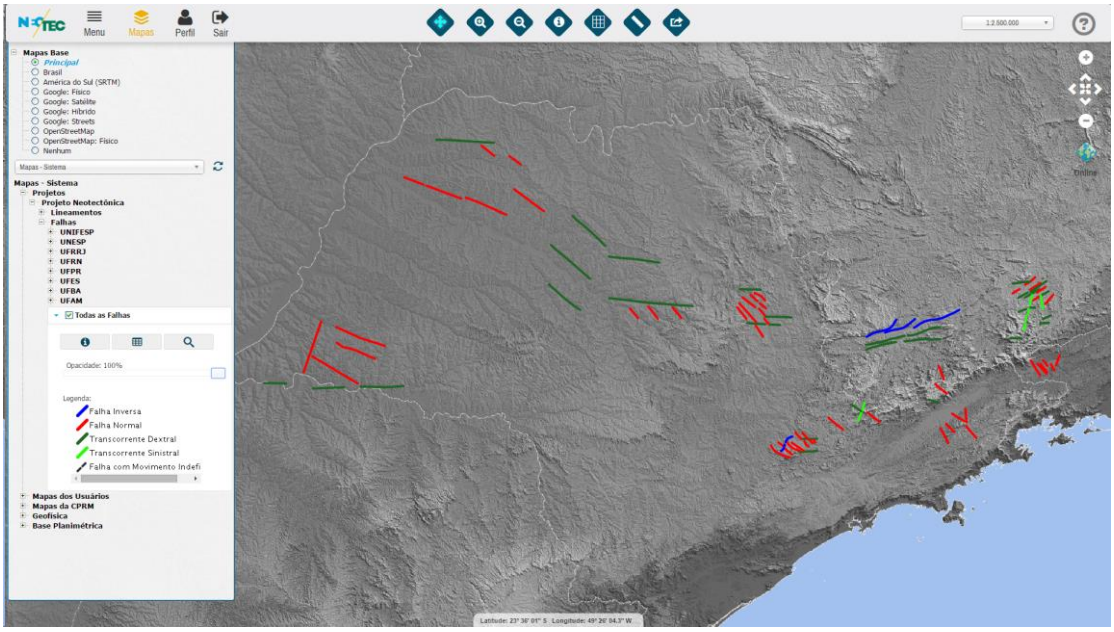

Figura 30: Exemplo de mapa com todas as falhas integradas.

b) Mapa de Falhas Normais

Exibe todas as falhas normais existentes no *Semantic Repository*. O mapa é construído a partir da seguinte consulta:

```
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>

DESCRIBE ?feature WHERE {
    ?feature a geosparql:Feature .
    ?feature a neotec:Normal
}
```

O resultado da consulta, exibindo todas as falhas normais que foram integradas no sistema é exibido na Figura 31:

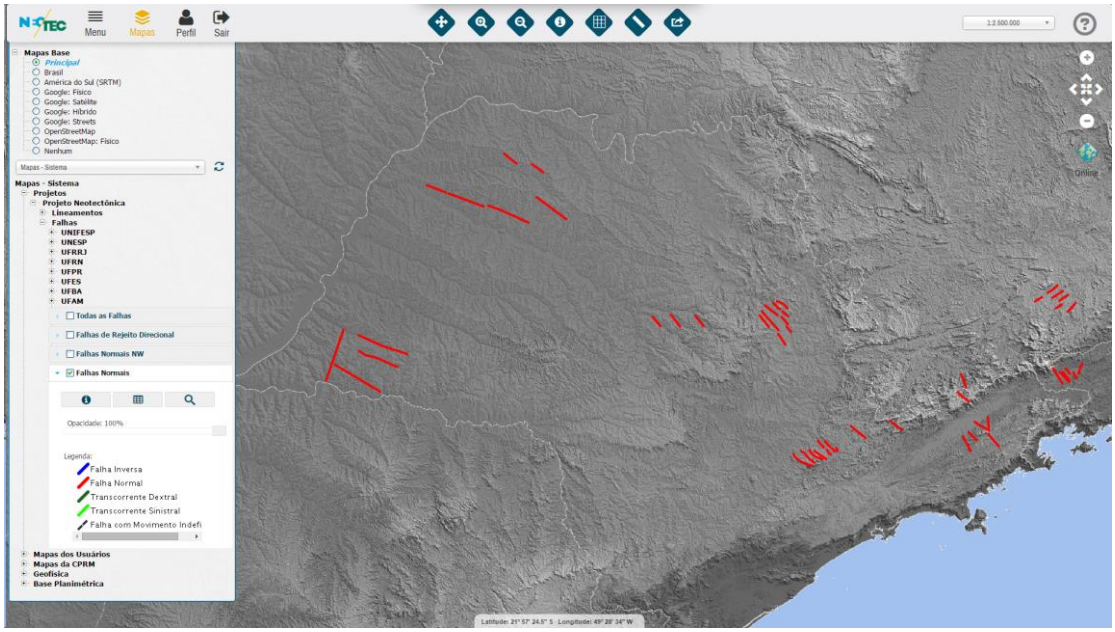

Figura 31: Mapa de Falhas Normais integradas.

c) Mapas de Falhas de Rejeito Direcional

Exibe todas as falhas de rejeito direcional existentes no *Semantic Repository*. Construído a partir da seguinte consulta:

```
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

DESCRIBE ?feature WHERE {
      ?feature a geosparql:Feature .
      ?feature a neotec:Fault .
      {
            ?feature neotec:hasClassification neotec:Dextral
      } UNION {
            ?feature neotec:hasClassification neotec:Sinistral
```

```
        }
}
```

Neste caso, a consulta explora um atributo (*neotec:hasClassification*) que determina a classificação das falhas. O critério de seleção busca por tipos específicos de falhas de rejeito direcional, ou seja, falhas transcorrentes dextrais (*neotec:Dextral*) e transcorrentes sinistrais (*neotec:Sinistral*). A consulta utiliza o operador UNION para selecionar as duas classificações. O resultado da consulta é exibido na Figura 32:

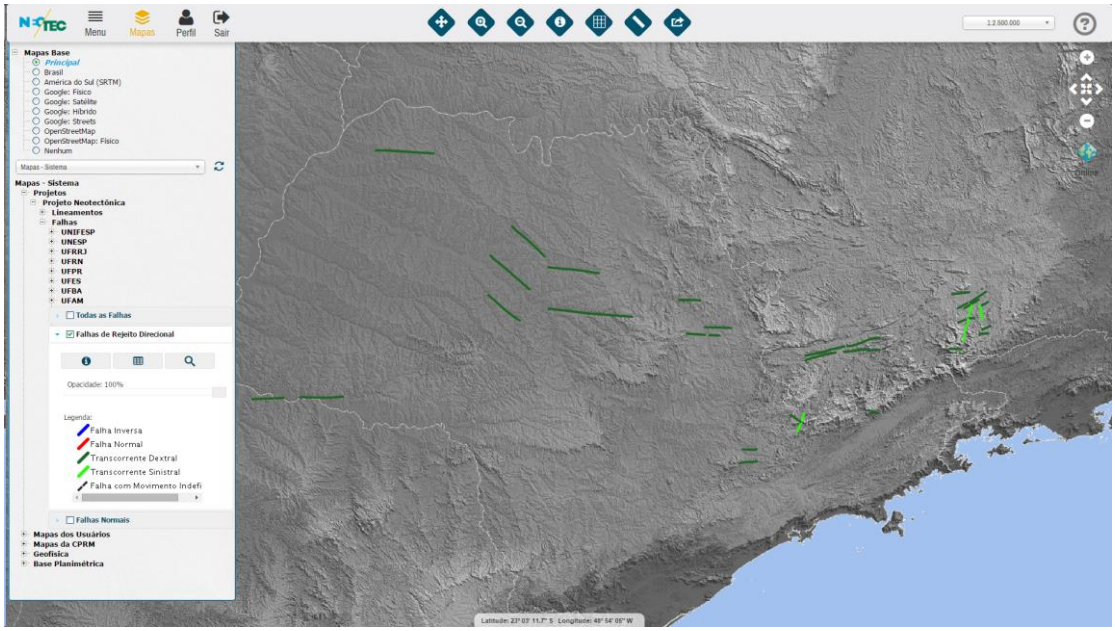

Figura 32: Mapa de Falhas de Rejeito Direcional.

O mesmo mapa ainda poderia ser gerado por outras consultas. No exemplo a seguir, pode ser utilizado o tipo específico das falhas:

```
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

DESCRIBE ?feature WHERE {
```

```
        ?feature a geosparql:Feature .
        ?feature a neotec:Fault .
        {
              ?feature a neotec:Dextral
        } UNION {
              ?feature a neotec:Sinistral
        }
}
```

Outra consulta mais simples para selecionar diretamente os indivíduos do tipo "Falha de Rejeito Direcional" (*neotec:StrikeSlip*):

```
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

DESCRIBE ?feature WHERE {
      ?feature a geosparql:Feature .
      ?feature a neotec:StrikeSlip
}
```

Neste exemplo pode-se verificar a flexibilidade de acesso aos dados que as consultas semânticas possibilitam. Os mesmos dados podem ser recuperados pelas diferentes consultas, pois o processo de inferência na etapa de integração realiza a expansão da classificação e das propriedades das instâncias de dados.

d) Mapa de Falhas Normais com mergulho a nordeste

A consulta para exibir todas as falhas normais existentes no *Semantic Repository* cujo bloco abatido esteja a nordeste. O mapa é construído a partir da seguinte consulta:

```
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX neotec:
<http://neotec.rc.unesp.br/resource/Neotectonics/>
```

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>\

DESCRIBE ?feature WHERE {
      ?feature a geosparql:Feature .
      ?feature a neotec:Normal .
      ?feature neotec:downedBlock neotec:Northeast
}
```

Neste exemplo é apresentado como os dados podem ser filtrados de acordo com seu tipo e também com seus atributos. O resultado da consulta é exibido na Figura 33:

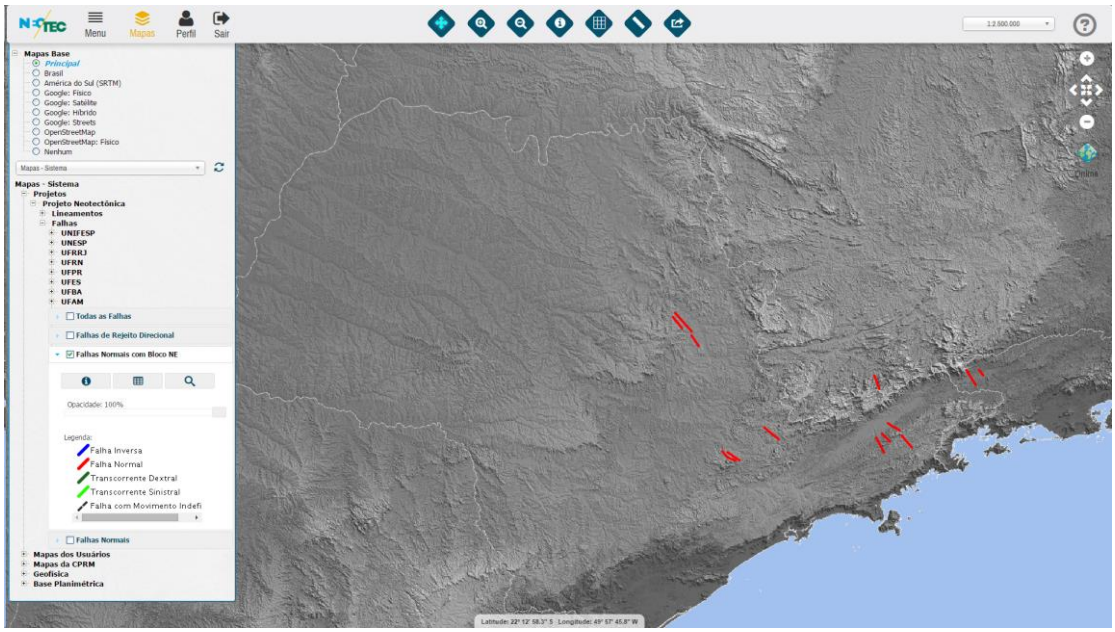

Figura 33: Mapa de Falhas Normais com Bloco Abatido a NE.

Todos os exemplos exibidos nesta seção são demonstrações de tipos de consultas que podem ser executadas com os dados integrados semanticamente no SID.

Há diversas outras interações possíveis, que podem ser realizadas por meio das diferentes formas de consultas disponíveis no SPARQL[40].

[40] http://www.w3.org/TR/rdf-sparql-query/#QueryForms

6.3 Conclusão do Capítulo

Neste capítulo foram apresentados estudos de casos baseados na implementação do *Framework para Integração de Dados Geoespaciais*. Os estudos apresentados caracterizam exemplos de uso prático do *framework* proposto neste trabalho, como solução para integração de dados geológicos em nível semântico.

Os resultados apresentados nestes estudos de casos permitem demonstrar os objetivos para os quais o *framework* é proposto. Mostrar que os dados geológicos, provenientes de diferentes fontes podem ser exibidos, consultados e analisados de maneira uniforme quando passam pelo processo de integração semântica.

A análise semântica realizada com base em inferência e regras eleva o nível de integração de dados geológicos, no sentido que, além de verificar aspectos semânticos da padronização dos dados, também identifica possíveis combinações, duplicações e conflitos. As inconsistências identificadas quando novos dados são adicionados ao SID, facilitam a tarefa dos usuários de manter os dados integrados de maneira coerente, sem ocorrência de repetições ou inconsistência de informações. Neste contexto, o uso de regras também contribui para facilitar a interação do usuário, ou seja, a aplicação poderia conter uma interface para os usuários definirem as regras de acordo com sua necessidade. Como a linguagem de regras trabalha com conceitos de alto nível, esta é mais próxima da linguagem natural do que uma linguagem de programação, e isto permite uma maior facilidade de entendimento e capacidade dos usuários em definir ou modificar as regras. Outro fator importante é que o uso de regras codificadas nas ontologias permite com que este conhecimento seja compartilhado, e não fique totalmente atrelado à implementação em nível de código ou algoritmo. Desta forma, qualquer sistema com suporte à linguagem das regras, pode utilizá-las no processo de inferência. A extensão GeoSWRL entretanto, por não pertencer ao conjunto de funções padrão do SWRL, precisa ser adicionada ao raciocinador. Até o momento, somente o Pellet possui esta capacidade de extensão.

Na implementação utilizada nos estudos de caso, os dados integrados e armazenados localmente puderam ser acessados por meio de conceitos e propriedades

descritas nas ontologias do SID (*Reference Ontologies*). As inferências geradas na análise de instâncias permite com que os dados sejam acessados a partir de diferentes consultas, como demonstrado no exemplo de consultas a falhas de rejeito de mergulho, que puderam ser consultadas pelo conceito propriamente dito, ou pela associação de falhas transcorrentes dextrais e sinistrais, que são dois tipos de falhas de rejeito de mergulho. Além disso, a consulta por falhas transcorrentes pôde ser realizada tanto pela definição de um atributo (*neotec:hasClassification*), quanto pela declaração explícita do tipo das falhas (*?feature a neotec:Dextral*, por exemplo). Este tipo de interação só é possível devido ao fato da semântica estar explícita nas ontologias. São as hierarquias de classes, por exemplo, que permitem ao raciocinador concluir que tanto uma falha transcorrente dextral quanto uma falha transcorrente sinistral são falhas de rejeito direcional.

7 Considerações Finais

Neste trabalho, foram investigados diversos aspectos relacionados à integração de dados geológicos, especialmente os aspectos relacionados às informações geoespaciais associadas a estes dados. Além disso, a integração semântica, baseada em ontologias, foi evidenciada por trazer diversas vantagens para automatizar o processo de integração de dados geológicos. Neste contexto, um *Framework para Integração de Dados Geoespaciais* foi proposto como solução para automatizar a integração de dados geológicos em nível semântico.

O *Framework para Integração de Dados Geoespaciais* é baseado na utilização de ontologias e no processo de alinhamento de ontologias para realizar a integração de dados geológicos provenientes de diferentes fontes e organizados com diferentes esquemas. O *framework* ainda considera a presença ou ausência de metadados associados aos dados geológicos, ao permitir que, em ambos os casos, os dados sejam integrados em nível semântico. Esta integração é possível devido à abordagem de transformar os esquemas de dados em ontologias e realizar o alinhamento destas com ontologias de referência mantidas pelo *framework*.

Outro aspecto levado em consideração pelo *Framework para Integração de Dados Geoespaciais* é a opção de configurar a integração para ser realizada a partir de consultas distribuídas, geralmente aplicável quando se manipulam dados públicos; ou configurar o armazenamento local dos dados integrados, geralmente aplicável quando se manipulam dados privados. Estas estratégias ainda podem ser utilizadas simultaneamente.

A análise semântica realizada durante a integração de instâncias garante que as inferências realizadas por um raciocinador sejam utilizadas para garantir a consistência dos dados integrados. Permite que partes da tarefa de integração feitas por usuários sejam automatizadas, tais como: a identificação de duplicações e conflitos e as possibilidades de combinação dos dados. Geralmente esta tarefa precisa da análise de aspectos semânticos da informação. Um outro aspecto a ser destacado foi a codificação das regras para a execução da tarefa de integração. Na codificação das regras foi utilizada a linaguagem SWRL, que pode ser facilmente interpretada devido a sua

proximidade com a linguagem natural, torna as definições utilizadas na análise dos dados mais acessíveis para os usuários. Os usuários não precisam codificar estas definições em algoritmos e linguagens de programação, geralmente mais complexas do que a SWRL.

Outras contribuições positivas da solução para integração de dados geológicos apresentada neste trabalho são as Ontologias Neotectônicas, desenvolvidas para serem utilizadas como referência no *Framework para Integração de Dados Geoespaciais*, e o GeoSWRL, a extensão da linguagem SWRL que permite a manipulação de dados geoespaciais na definição de regras.

O *framework* proposto neste trabalho foi implementado e integrado a um Sistema de Integração de Dados, em um contexto de um projeto real. De acordo com os resultados obtidos nos estudos de casos, esta implementação trouxe benefícios ao SID, no sentido de padronizar o acesso a dados geológicos heterogêneos, e permitir a manipulação destes dados em nível semântico.

Durante a implementação do *framework*, alguns aspectos que precisam de mais investigação foram identificados, como por exemplo o desempenho do raciocinador durante a integração a análise de instâncias. Com um volume de algumas dezenas de instâncias para analisar o desempenho do raciocinador mostrou uma degradação considerável. Uma abordagem mais adequada para atender a um grande volume de dados deve ser investigada.

Alguns dos aspectos relacionados às informações geoespaciais discutidos no Capítulo 3, tais como escala representação e o fator tempo, podem ser mais bem analisados dentro do processo de integração. Neste aspecto, também podem ser mais bem investigados o uso e a definição de certos padrões de metadados.

A extensão GeoSWRL pode ser expandida para incluir outras operações com dados geoespaciais e permitir a elaboração de diferentes tipos de regras.

Finalmente, devido ao fato do *Framework* para Integração de Dados Geoespaciais ser baseado em ontologias e possuir uma arquitetura distribuída em serviços, a solução de integração pode ser adaptada para ser utilizada em outros subdomínios das Geociências.

REFERÊNCIAS

ACAMPORA, G.; LOIA, V.; SALERNO, S. A hybrid evolutionary approach for solving the ontology alignment problem. **International Journal of Intelligent Systems** v. 27, p. 189–216 , 2012. Disponível em: <http://onlinelibrary.wiley.com/doi/10.1002/int.20517/full>. Acesso em: 30 set. 2013.

ANTONIOU, G.; HARMELEN, F. VAN. **A Semantic Web Primer**. 2. ed. Londres: The MIT Press, 2008. p. 264

BECK, H.; PINTO, H. S. **Overview of Approach, Methodologies, Standards, and Tools for Ontologies**, 2002.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The Semantic Web. **Scientific American**, v. 284, p. 34–43, 2001.

BERNERS-LEE, T. **WWW Past & Future**. 2003. Disponível em: <http://www.w3.org/2003/Talks/0922-rsoc-tbl/>. Acesso em: 08 dez. 2013.

BERNERS-LEE, Tim. **Linked Data:** Design Issues. 2006. Disponível em: <http://www.w3.org/DesignIssues/LinkedData.html>. Acesso em: 01 jun. 2015.

BOCK, J.; HETTENHAUSEN, J. Discrete particle swarm optimisation for ontology alignment. **Information Sciences**, v. 192, p. 152–173, jun. 2012.

CÂMARA, G.; DAVIS, C.; MONTEIRO, Antoˆn o Miguel Vieira. Introdução à Ciência da Geoinformação. São José dos Campos: Inpe, 2001.

CROCHEMORE, Maxime. String-matching on ordered alphabets. **Theoretical Computer Science**. Paris, p. 33-47. 06 jan. 1992. Disponível em: <http://www.sciencedirect.com/science/article/pii/0304397592901342>. Acesso em: 25 fev. 2015.

CRUZ, I. F. et al. GIVA: A Semantic *Framework* for Geospatial and Temporal Data Integration, Visualization, and Analytics. 2013, New York, New York, USA: ACM Press, 2013. p.534–537. Disponível em: <http://dl.acm.org/citation.cfm?id=2525324>

DAVID, J. et al. The *Alignment API* 4.0, **Semantic web journal.** 2(1):3-10, 2011

DJEDDI, W. E.; KHADIR, M. T. Introducing Artificial Neural Network in Ontologies Alignment Process. **New Trends in Databases & Inform. Sys**, p. 175–186, 2013.

DU, H. et al. Matching Formal and Informal Geospatial Ontologies. **Geographic Information Science at the Heart of Europe**, p. 155–171, 2013.

EUZENAT, J.; SHVAIKO, P. **Ontology Matching**. Heidelberg: Springer-Verlag, 2007. p. 341

MANOLA, F. Association For Computing Machinery (Ed.). **RDF Primer**. Disponível em: <http://www.w3.org/TR/rdf-primer/>. Acesso em: 20 nov. 2013.

GASPAR, J. A. **Cartas e Projecções Cartográficas**: 3a Edição Actualizada e Aumentada. 3. ed. Lisboa: Lidel Edições Técnicas, 2005. 352 p.

GIANNOPOULOS, G. et al. FAGI-tr: A tool for aligning geospatial RDF vocabularies ESWC 2014. 2014, London: IMIS Institute, "Athena" Research Center, 2014. p.5. Disponível em: <http://2014.eswc-conferences.org/sites/default/files/eswc2014pd_submission_35.pdf>. Acesso em: 4 jun. 2014.

GRUBER, T. R. A translation approach to portable ontology specifications. **Knowledge Acquisition**, v. 5, p. 199–220, 1993.

HWANG, J.; NAM, K. W.; RYU, K. H. Designing and implementing a geologic information system using a spatiotemporal ontology model for a geologic map of Korea. **Computers & Geosciences**, v. 48, p. 173–186 , 2012.

ISELE, R.; BIZER, C. Learning Expressive Linkage Rules using Genetic Programming . **Proceedings of the VLDB Endowment**, v. 5, 2012.

JANOWICZ, K. et al. Geospatial semantics and linked spatiotemporal data–Past, present, and future. **Semantic Web**, v. 3, n. 4, p. 321-332, 2012.

JEAN-MARY, Y. R.; SHIRONOSHITA, E. P.; KABUKA, M. R. Ontology Matching with Semantic Verification. **Web Semantics: Science, Services and Agents on the World Wide Web**, v. 7, n. 3, p. 235–251, 1 set. 2009.

KHETARPAUL, S.; GUPTA, S. K.; CHAUHAN, R. SR-Match : A Novel Schema Matcher Based on. Computer Networks and Intelligent Computing, v. 157, p. 93–102, 2011.

KLIEN, E. **Semantic Annotation of Geographic Information**. 2008. 159 f. Tese (Doutorado) - Curso de Computer Science, Departamento de Institute For Geoinformatics, University Of Muenster, Muenster, 2008. Disponível em: <http://ifgi.uni-muenster.de/~klien/publications/Klien_PhDThesis_full.pdf>. Acesso em: 24 jan. 2014.

LIN, K.; LUDÄSCHER, B. A system for semantic integration of geologic maps via ontologies. **Semantic Web Technologies for Searching and Retrieving Scientific Data (SCIS), ISWC Workshop**, v. 83, p. 6, 2003.

LIU, J.; QIN, L.; WANG, H. An Ontology Mapping Method Based on Support Vector Machine. **disi.unitn.it**, p. 2–3, 2013.

MA, X.; FOX, P. Recent progress on geologic time ontologies and considerations for future works. **Earth Science Informatics**, v. 6, n. 1, p. 31–46, 9 fev. 2013.

MACARIO, C. G. N. **Anotação Semântica de Dados Geoespaciais**. 2009. 114 f. Tese (Doutorado) - Curso de Ciência da Computação, Departamento de Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2009. Disponível em: <http://www.bibliotecadigital.unicamp.br/document/?code=000476654>. Acesso em: 17 jul. 2014.

MASCARDI, V.; LOCORO, A.; ROSSO, P. Automatic Ontology Matching via Upper Ontologies: A Systematic Evaluation. **IEEE Transactions on Knowledge and Data Engineering**, v. 22, 2010.

MELNIK, S.; GARCIA-MOLINA, H.; RAHM, E. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. **Proceedings 18th International Conference on Data Engineering**, 2002.

NAGY, M.; VARGAS-VERA, M.; MOTTA, E. DSSim-ontology mapping with uncertainty. **Proceedings of the 1st International Workshop on Ontology Matching**, v. 225, n. 10, 2006.

NGO, D.; BELLAHSENE, Z.; TODOROV, K. Opening the Black Box of Ontology Matching. **The Semantic Web: Semantics and Big Data**, v. 7882, 2013.

NGOMO, A.; LYKO, K.. **Unsupervised Learning of Link Specifications: Deterministic vs. Non-Deterministic**. 2013. Disponível em: <http://disi.unitn.it/~p2p/OM-2013/om2013_Tpaper3.pdf>. Acesso em: 23 fev. 2015.

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

RUSSEL, S. J.; NORVIG, P.; Artificial Intelligence: A Modern Approach. Upper Saddle River, New Jersey. Prentice Hall, p. 111–114, 2003.

SHVAIKO, P.; EUZENAT, J. Ontology Matching: State of the Art and Future Challenges. **IEEE Transactions on Knowledge and Data Engineering**, v. 25, n. 1, 2013.

SIRIN, Evren et al. Pellet: A practical owl-dl reasoner. **Web Semantics: science, services and agents on the World Wide Web**, v. 5, n. 2, p. 51-53, 2007.

W3C OWL WORKING GROUP (Ed.). **OWL 2 Web Ontology Language Document**: Overview (Second Edition). Disponível em: <http://www.w3.org/TR/owl2-overview/>. Acesso em: 20 nov. 2013.

WERLANG, Ricardo. **Ontology-based approach for standard formats integration in reservoir modeling**. 2015. 93 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015.

Disponível em: <http://www.lume.ufrgs.br/handle/10183/115196>. Acesso em: 01 maio 2015.

WORLD WIDE WEB CONSORTIUM. **A direct mapping of relational data to rdf.** , 2012. Disponível em: <http://www.w3.org/TR/rdb-direct-mapping/>. Acesso em: 21 jul. 2014.

WORLD WIDE WEB CONSORTIUM. **W3C XML Schema Definition Language (XSD) 1.1**: Part 1: Structures. 2012. Disponível em: <http://www.w3.org/TR/xmlschema11-1/>. Acesso em: 08 dez. 2013.

WORLD WIDE WEB CONSORTIUM. **RDF Vocabulary Description Language 1.0**: RDF Schema. 2004. Disponível em: <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>. Acesso em: 08 dez. 2013.

ZHONG, J.; AYDINA, A.; MCGUINNESS, D. L. Ontology of fractures. **Journal of Structural Geology**, v. 31, n. 3, p. 251–259, mar. 2009.

APÊNDICE A

Regras SWRL codificadas nas Ontologias Neotectônica

São apresentadas as regras SWRL codificadas nas Ontologias Neotectônicas que utilizam as funções desenvolvidas na extensão GeoSWRL.

Prefixos utilizados nas regras:

```
PREFIX neotec: <http://neotec.rc.unesp.br/resource/Neotectonics/>
PREFIX neotecapp:
<http://neotec.rc.unesp.br/resource/Neotectonics/Application>
PREFIX geosparql: <http://www.opengis.net/ont/geosparql#>
PREFIX swrl: <http://www.w3.org/2003/11/swrl#>
PREFIX geoswrl: < http://neotec.rc.unesp.br/resource/geoswrl>
```

Regras para detectar possíveis combinações:

Se duas falhas normais distintas, forem disjuntas e estiverem alinhadas com uma diferença menor do que 30 graus a uma distância inferior a 1Km uma da outra, então estas falhas possivelmente combinam.

```
neotec:Normal(?f1), neotec:Normal(?f2), geosparql:hasGeometry(?f1,
?g1), geosparql:hasGeometry(?f2, ?g2), geosparql:asWKT(?g1, ?wkt1),
geosparql:asWKT(?g2, ?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:disjoint(?wkt1, ?wkt2), geoswrl:isWithinDistance(?wkt1, ?wkt2,
0.01),  swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:possiblyCombinesWith(?f1, ?f2)
```

Se duas falhas transcorrentes dextrais distintas, forem disjuntas e estiverem alinhadas com uma diferença menor do que 30 graus a uma distância inferior a 1Km uma da outra, então estas falhas possivelmente combinam.

```
neotec:Dextral(?f1), neotec:Dextral(?f2), geosparql:hasGeometry(?f1,
?g1), geosparql:hasGeometry(?f2, ?g2), geosparql:asWKT(?g1, ?wkt1),
geosparql:asWKT(?g2, ?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:disjoint(?wkt1, ?wkt2), geoswrl:isWithinDistance(?wkt1, ?wkt2,
0.01),  swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:possiblyCombinesWith(?f1, ?f2)
```

Se duas falhas transcorrentes sinistrais distintas, forem disjuntas e estiverem alinhadas com uma diferença menor do que 30 graus a uma distância inferior a 1Km uma da outra, então estas falhas possivelmente combinam.

```
neotec:Sinistral(?f1), neotec:Sinistral(?f2),
geosparql:hasGeometry(?f1, ?g1), geosparql:hasGeometry(?f2, ?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0), geoswrl:disjoint(?wkt1,
?wkt2), geoswrl:isWithinDistance(?wkt1, ?wkt2, 0.01),
swrl:DifferentFrom (?f1, ?f2) -> neotecapp:possiblyCombinesWith(?f1,
?f2)
```

Se duas falhas inversas distintas, forem disjuntas e estiverem alinhadas com uma diferença menor do que 30 graus a uma distância inferior a 1Km uma da outra, então estas falhas possivelmente combinam.

```
neotec:Reverse(?f1), neotec:Reverse(?f2), geosparql:hasGeometry(?f1,
?g1), geosparql:hasGeometry(?f2, ?g2), geosparql:asWKT(?g1, ?wkt1),
geosparql:asWKT(?g2, ?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:disjoint(?wkt1, ?wkt2), geoswrl:isWithinDistance(?wkt1, ?wkt2,
0.01),  swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:possiblyCombinesWith(?f1, ?f2)
```

Se duas falhas de empurrão distintas, forem disjuntas e estiverem alinhadas com uma diferença menor do que 30 graus a uma distância inferior a 1Km uma da outra, então estas falhas possivelmente combinam.

```
neotec:Thrust(?f1), neotec:Thrust(?f2), geosparql:hasGeometry(?f1,
?g1), geosparql:hasGeometry(?f2, ?g2), geosparql:asWKT(?g1, ?wkt1),
geosparql:asWKT(?g2, ?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:disjoint(?wkt1, ?wkt2), geoswrl:isWithinDistance(?wkt1, ?wkt2,
0.01),  swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:possiblyCombinesWith(?f1, ?f2)
```

Se uma falha inversa e uma de empurrão, distintas, forem disjuntas e estiverem alinhadas com uma diferença menor do que 30 graus a uma distância inferior a 1Km uma da outra, então estas falhas possivelmente combinam.

```
neotec:Reverse(?f1), Thrust(?f2), geosparql:hasGeometry(?f1, ?g1),
geosparql:hasGeometry(?f2, ?g2), geosparql:asWKT(?g1, ?wkt1),
geosparql:asWKT(?g2, ?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:disjoint(?wkt1, ?wkt2), geoswrl:isWithinDistance(?wkt1, ?wkt2,
0.01),  swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:possiblyCombinesWith(?f1, ?f2)
```

Se duas falhas transtensionais distintas, forem disjuntas e estiverem alinhadas com uma diferença menor do que 30 graus a uma distância inferior a 1Km uma da outra, então estas falhas possivelmente combinam.

```
neotec:Transtensional(?f1), neotec:Transtensional(?f2),
geosparql:hasGeometry(?f1, ?g1), geosparql:hasGeometry(?f2, ?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0), geoswrl:disjoint(?wkt1,
?wkt2), geoswrl:isWithinDistance(?wkt1, ?wkt2, 0.01),
swrl:DifferentFrom (?f1, ?f2) -> neotecapp:possiblyCombinesWith(?f1,
?f2)
```

Se duas falhas transpressionais distintas, forem disjuntas e estiverem alinhadas com uma diferença menor do que 30 graus a uma distância inferior a 1Km uma da outra, então estas falhas possivelmente combinam.

```
neotec:Transpressional(?f1), neotec:Transpressional(?f2),
geosparql:hasGeometry(?f1, ?g1), geosparql:hasGeometry(?f2, ?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0), geoswrl:disjoint(?wkt1,
?wkt2), geoswrl:isWithinDistance(?wkt1, ?wkt2, 0.01),
swrl:DifferentFrom (?f1, ?f2) -> neotecapp:possiblyCombinesWith(?f1,
?f2)
```

Regras para possíveis duplicações

Se duas falhas normais distintas interceptam-se com um ângulo máximo de 30 graus entre si, então estas falhas são uma possível duplicação uma da outra.

```
neotec:Normal(?f1), neotec:Normal(?f2),
geosparql:hasGeometry(?f1,?g1), geosparql:hasGeometry(?f2,?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:isPossibleDuplicationOf(?f1, ?f2)
```

Se duas falhas transcorrentes dextrais distintas interceptam-se com um ângulo máximo de 30 graus entre si, então estas falhas são uma possível duplicação uma da outra.

```
neotec:Dextral(?f1), neotec:Dextral(?f2),
geosparql:hasGeometry(?f1,?g1), geosparql:hasGeometry(?f2,?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:isPossibleDuplicationOf(?f1, ?f2)
```

Se duas falhas transcorrentes sinistrais distintas interceptam-se com um ângulo máximo de 30 graus entre si, então estas falhas são uma possível duplicação uma da outra.

```
neotec:Sinistral(?f1), neotec:Sinistral(?f2),
geosparql:hasGeometry(?f1,?g1), geosparql:hasGeometry(?f2,?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:isPossibleDuplicationOf(?f1, ?f2)
```

Se duas falhas inversas distintas interceptam-se com um ângulo máximo de 30 graus entre si, então estas falhas são uma possível duplicação uma da outra.

```
neotec:Reverse(?f1), neotec:Reverse(?f2),
geosparql:hasGeometry(?f1,?g1), geosparql:hasGeometry(?f2,?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:isPossibleDuplicationOf(?f1, ?f2)
```

Se duas falhas de empurrão distintas interceptam-se com um ângulo máximo de 30 graus entre si, então estas falhas são uma possível duplicação uma da outra.

```
neotec:Thrust(?f1), neotec:Thrust(?f2),
geosparql:hasGeometry(?f1,?g1), geosparql:hasGeometry(?f2,?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:isPossibleDuplicationOf(?f1, ?f2)
```

Se uma falha inversa e uma de empurrão, distintas, interceptam-se com um ângulo máximo de 30 graus entre si, então estas falhas são uma possível duplicação uma da outra.

```
neotec:Thrust(?f1), neotec:Reverse(?f2),
geosparql:hasGeometry(?f1,?g1), geosparql:hasGeometry(?f2,?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
```

```
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:isPossibleDuplicationOf(?f1, ?f2),
```

Se duas falhas transtensionais distintas interceptam-se com um ângulo máximo de 30 graus entre si, então estas falhas são uma possível duplicação uma da outra.

```
neotec:Transtensional(?f1), neotec:Transtensional(?f2),
geosparql:hasGeometry(?f1,?g1), geosparql:hasGeometry(?f2,?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:isPossibleDuplicationOf(?f1, ?f2)
```

Se duas falhas transpressionais distintas interceptam-se com um ângulo máximo de 30 graus entre si, então estas falhas são uma possível duplicação uma da outra.

```
neotec:Transpressional(?f1), neotec:Transpressional(?f2),
geosparql:hasGeometry(?f1,?g1), geosparql:hasGeometry(?f2,?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1,?wkt2), swrl:DifferentFrom (?f1, ?f2) ->
neotecapp:isPossibleDuplicationOf(?f1, ?f2)
```

Regras para possíveis conflitos

Se uma falha normal e uma que não seja normal interceptarem-se com um ângulo máximo de 30 graus entre si, então estas falhas conflitam uma com a outra.

```
neotec:Normal(?f1), (not(neotec:Normal))(?f2), Fault(?f2),
geosparql:hasGeometry(?f1, ?g1), geosparql:hasGeometry(?f2, ?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0), geoswrl:intersects(?wkt1,
?wkt2),  swrl:DifferentFrom(?f1, ?f2) ->
neotecapp:possiblyConflictsWith(?f1, ?f2)
```

Se uma falha transcorrente dextral e uma que não seja transcorrente dextral interceptarem-se com um ângulo máximo de 30 graus entre si, então estas falhas conflitam uma com a outra.

```
neotec:Dextral(?f1), (not(neotec:Dextral))(?f2), Fault(?f2),
geosparql:hasGeometry(?f1, ?g1), geosparql:hasGeometry(?f2, ?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0), geoswrl:intersects(?wkt1,
?wkt2),  swrl:DifferentFrom(?f1, ?f2) ->
neotecapp:possiblyConflictsWith(?f1, ?f2)
```

Se uma falha transcorrente sinistral e uma que não seja transcorrente sinistral interceptarem-se com um ângulo máximo de 30 graus entre si, então estas falhas conflitam uma com a outra.

```
neotec:Sinistral(?f1), (not(neotec:Sinistral))(?f2), Fault(?f2),
geosparql:hasGeometry(?f1, ?g1), geosparql:hasGeometry(?f2, ?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0), geoswrl:intersects(?wkt1,
?wkt2),  swrl:DifferentFrom(?f1, ?f2) ->
neotecapp:possiblyConflictsWith(?f1, ?f2)
```

Se uma falha inversa e uma que não seja inversa interceptarem-se com um ângulo máximo de 30 graus entre si, então estas falhas conflitam uma com a outra.

```
neotec:Reverse(?f1), (not(neotec:Reverse))(?f2), Fault(?f2),
geosparql:hasGeometry(?f1, ?g1), geosparql:hasGeometry(?f2, ?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0), geoswrl:intersects(?wkt1,
?wkt2),  swrl:DifferentFrom(?f1, ?f2) ->
neotecapp:possiblyConflictsWith(?f1, ?f2)
```

Se uma falha de empurrão e uma que não seja de empurrão interceptarem-se com um ângulo máximo de 30 graus entre si, então estas falhas conflitam uma com a outra.

```
neotec:Thrust(?f1), (not(neotec:Thrust))(?f2), Fault(?f2),
geosparql:hasGeometry(?f1, ?g1), geosparql:hasGeometry(?f2, ?g2),
geosparql:asWKT(?g1, ?wkt1), geosparql:asWKT(?g2, ?wkt2),
geoswrl:angleBetween(?wkt1, ?wkt2, 30.0), geoswrl:intersects(?wkt1,
?wkt2),  swrl:DifferentFrom(?f1, ?f2) ->
neotecapp:possiblyConflictsWith(?f1, ?f2)
```

Se uma falha transpressional e uma que não seja transpressional interceptarem-se com um ângulo máximo de 30 graus entre si, então estas falhas conflitam uma com a outra.

```
neotec:Transpressional(?f1), (not(neotec:Transpressional))(?f2),
Fault(?f2), geosparql:hasGeometry(?f1, ?g1),
geosparql:hasGeometry(?f2, ?g2), geosparql:asWKT(?g1, ?wkt1),
geosparql:asWKT(?g2, ?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1, ?wkt2),  swrl:DifferentFrom(?f1, ?f2) ->
neotecapp:possiblyConflictsWith(?f1, ?f2)
```

Se uma falha transtensional e uma que não seja transtensional interceptarem-se com um ângulo máximo de 30 graus entre si, então estas falhas conflitam uma com a outra.

```
neotec:Transtensional(?f1), (not(neotec:Transtensional))(?f2),
Fault(?f2), geosparql:hasGeometry(?f1, ?g1),
geosparql:hasGeometry(?f2, ?g2), geosparql:asWKT(?g1, ?wkt1),
geosparql:asWKT(?g2, ?wkt2), geoswrl:angleBetween(?wkt1, ?wkt2, 30.0),
geoswrl:intersects(?wkt1, ?wkt2),  swrl:DifferentFrom(?f1, ?f2) ->
neotecapp:possiblyConflictsWith(?f1, ?f2)
```