2
Explorando dados na Web Semântica

A principal motivação para o desenvolvimento dessa dissertação parte do fato de que os dados na Web Semântica serão expressos em RDF e novos mecanismos de recuperação da informação deverão levar isso em consideração no intuito de prover aos usuários ferramentas mais eficientes para a exploração das informações (Marchionini, 2006), que permitam os usuários encontrarem um item de informação mais rápido e precisamente, com o menor número de passos. Portanto, nosso objetivo é desenvolver um modelo de exploração de dados RDF que permita o usuário explorar uma base semi-estruturada, sem conhecimento prévio do domínio dos dados.

2.1.
O que é exploração de dados?

Na área de hipertexto, busca, navegação e *browsing* descrevem processos distintos de recuperação de informação. Carmel et al. (1992), fizeram um vasto estudo sobre o processo cognitivo de *browsing*, e baseado nesse estudo faremos as seguintes distinções.

- Busca é o processo de procurar um item de informação específico;
- Browsing é o processo de investigar uma vasta coleção de itens de informação de forma superficial e não orientado a encontrar um item de informação especifico ou conhecido;
- Navegação é o processo de acessar, selecionar ou visualizar um conjunto de itens de informações, orientado a encontrar um item de informação especifico ou conhecido.

É importante fazermos tais distinções para que possamos definir qual ferramenta auxilia qual tarefa durante um processo de exploração. Vale ressaltar que os processos de browsing, busca e navegação alternam-se durante a realização de um tarefa de exploração, no entanto são processos totalmente distintos. De

forma prática, quando digitamos uma palavra-chave no Google estamos realizando uma busca, quando examinamos o resultado da consulta, no intuito de determinar qual link clicaremos, estamos fazendo *browsing* e quando clicamos em um resultado da consulta estamos navegando.

Chamamos de exploração de informação o processo de pesquisar, aprender e investigar um conjunto de itens de informação, através de busca, *browsing* ou navegação, mas não excluindo outras formas, no intuito de se descobrir algo novo. Usaremos o termo exploração no decorrer dessa dissertação para nos referirmos indistintamente à busca, *browsing*, ou navegação.

2.2. Trabalhos Relacionados

Abaixo, faremos uma revisão dos principais mecanismos de exploração de informação existentes e destinados a Web Semântica.

2.2.1. Navegadores ou Browsers

Navegadores, ou *browsers* - o termo em inglês - são aplicações que permitem os usuários visualizarem e interagirem com texto, imagens, vídeo, e outras informações que são encontradas em páginas Web. Texto, imagens e vídeos podem conter hyperlinks para outras páginas e documentos localizados na WWW (World Wide Web)[17] e é através desses hyperlinks que os usuários acessam as páginas da WWW.

Atualmente, algumas soluções foram propostas para navegação na Web Semântica. Ferramentas tais como o Tabulator[18], Disco[19], Marbles[20], Zitgist data viewer[21] e OpenLink Data Explorer[22] são navegadores de dados RDF que nos permitem visualizar um sub-grafo (ou conjunto de triplas) RDF em uma interface

[17] http://www.w3.org/WWW/

[18] http://www.w3.org/2005/ajar/tab

[19] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/

[20] http://beckr.org/marbles

[21] http://dataviewer.zitgist.com/

[22] http://demo.openlinksw.com/rdfbrowser2/

HTML (*HyperText Markup Language*)[23]. Tais mecanismos permitem a exibição das propriedades de um dado recurso (identificado por uma URI), e a navegação deste recurso para outros, através de suas propriedades (cujo valor são URIs de outros recursos), de forma análoga à navegação entre páginas HTML. Em geral, eles exibem um grafo RDF em uma interface HTML e permitem a navegação seqüencial baseada na ligação entre os recursos ou nodos do grafo.

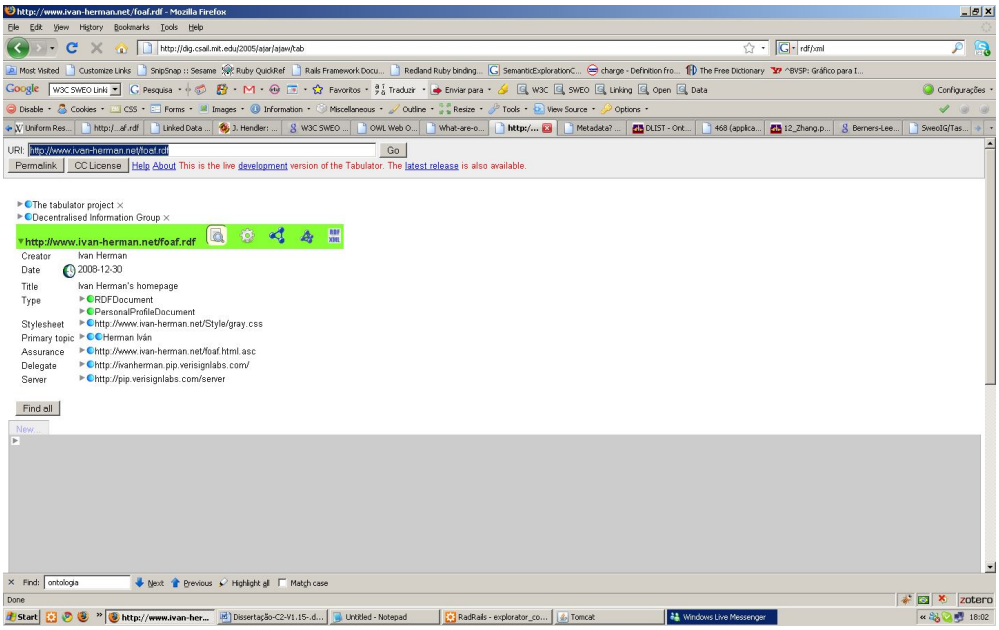

Figura 3 – Exemplo da visualização das triplas do arquivo RDF http://www.ivan-herman.net/foaf.rdf na interface do Tabulator.

Nesse processo de navegação utilizamos o mecanismo de de-referenciar[24] (em inglês: *dereferecing*), que é ação de recuperar uma representação de um recurso denotado por uma URI. As URIs de-referenciadas retornam um conjunto de triplas RDF, em geral representadas na notação RDF/XML. Alguns *browsers* também são capazes de manipular outros formatos de arquivo tais como NT e N3. Outra característica desses navegadores é que eles mostram as relações entre os recursos exibidos e as URIs de-referenciadas, dessa forma, os usuários são

[23] http://www.w3.org/MarkUp/

[24] http://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14

capazes de identificar a qual sub-grafo da Web Semântica aquele recurso pertence.

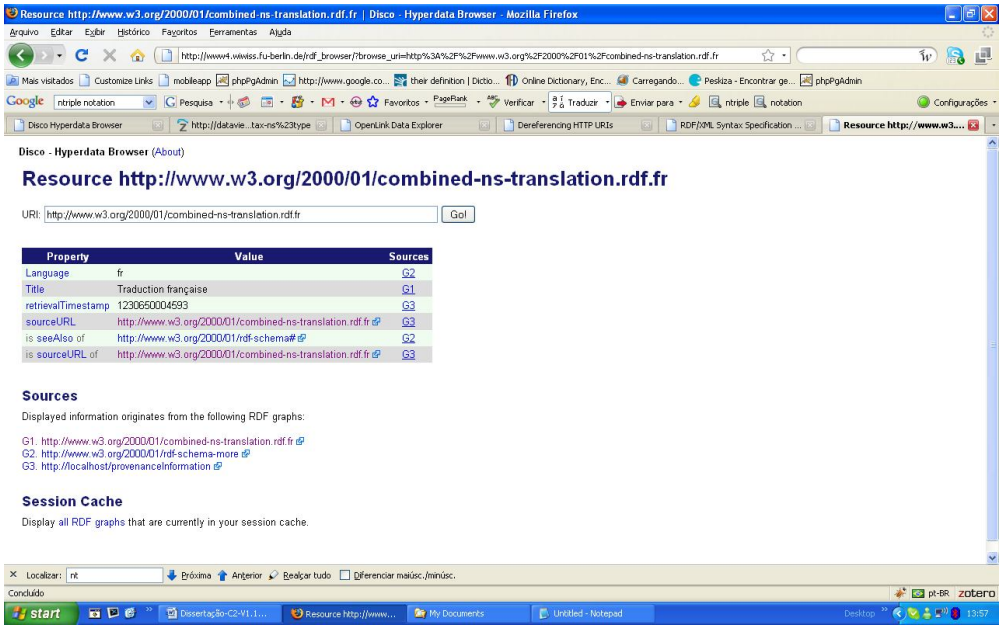

Figura 4 – Exemplo da de-referenciação da URI http://www.w3.org/2000/01/combined-ns-translation.rdf.fr na interface do Disco.

Tais navegadores são mais aplicáveis às tarefas executadas sobre uma base RDF com o domínio conhecido e com um volume reduzido de informações. É fácil notar que explorar uma base RDF, única e exclusivamente seguindo os elos entre os recursos, é um processo lento e pouco eficiente. Seria análogo a procurar por qualquer coisa na Web sem utilizar uma ferramenta de busca e apenas utilizando a navegação entre os elos HTML.

2.2.1.1. Sparql Endpoint

É importante ressaltar que nos navegadores HTML padrão, acessamos os recursos na Web, ou seja, as páginas HTML, diretamente na barra de endereço do *browser*, acessando uma URI ou seguindo os *hyperlinks* entre as páginas. Já nos navegadores semânticos, acessamos os recursos, ou seja, dados RDF, de duas formas distintas: através de uma URI, como vimos no tópico anterior, ou através

de um *Sparql*[25] *Endpoint*[26] que também é uma URI, porém com semântica diferenciada.

Um *Sparql Endpoint* é um serviço que implementa o protocolo SPARQL[27]. Ele permite ao usuário (humano ou máquina) fazer uma consulta a uma base de conhecimento usando a linguagem SPARQL. O resultado é retornado em um formato processável por máquina, por exemplo, um arquivo RDF. Sendo assim, podemos fornecer para um *browser* RDF (ex.: o Marbles) uma URI, que referencia um arquivo RDF diretamente, ou a URI de um *Sparql Endpoint*, que demanda uma consulta SPARQL e retorna um conjunto de triplas RDF.

Um exemplo prático de uso de um *Sparql Endpoint* para explorar a Web Semântica é a iniciativa LinkingOpenData[28]. Esta iniciativa engloba um conjunto de bases RDF interconectadas, que podem ser acessadas através de diversos *Sparql Endpoints*.

2.2.2. Navegação Facetada

Outra forma de exploração que vem sendo amplamente utilizada na área de buscas na Web é a navegação facetada - um paradigma de navegação introduzido pela área de recuperação de informação. A idéia principal é prover um mecanismo de indexação da informação que possa ser acessado pelo usuário, através de um índice de categorias, representados por um conjunto ortogonal de taxonomias, denominadas facetas. Do ponto de vista do usuário, a tarefa de encontrar informações se resume a selecionar arbitrariamente um conjunto de facetas na interface que restringem o conjunto inicial pesquisado a um subconjunto finito, progressivamente. Por exemplo, no domínio de aparelhos celulares poderíamos ter as facetas: Fabricantes (Nokia, Siemens, Motorola), Rede (GSM, TDMA, CDMA), Idioma (Inglês, Português, Chinês, Finlandês), etc. Cada faceta representa uma propriedade no domínio de dados, sendo assim, selecionando o

[25] http://www.w3.org/TR/rdf-sparql-query/

[26] Sparql é uma linguagem de consulta sobre dados RDF e um Endpoint indica um local específico de acesso a um serviço Web, usando um protocolo e formato de dados específico.

[27] Protocolo SPARQL: http://www.w3.org/TR/rdf-sparql-protocol/

[28] http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

valor de uma faceta o usuário estaria automaticamente selecionando todos os elementos que possuam tal propriedade com aquele valor selecionado. Os sistemas Flamenco[29] e FacetMap[30] são exemplos típicos da aplicação desse paradigma. O principal problema de tais ferramentas é que são dependentes do domínio, e as facetas são construídas manualmente.

Figura 5 - Interface do FacetMap exibindo uma lista de elemento e suas facetas.

Já o sistema BrowseRDF[31] (Oren et al. 2006) propõe um modelo estendido de navegação facetada, independente de domínio e que suporta a navegação dirigida em uma grafo RDF. Tal sistema permite aos usuários navegarem sobre um grafo RDF utilizando os predicados das triplas RDF como índices facetados. Através de uma interface visual, o usuário pode selecionar uma propriedade ou seu valor para obter o conjunto de recursos possuidores da propriedade (ou propriedade/valor) selecionada. Apesar de Oren et al. (2006) proporem um

[29] http://flamenco.berkeley.edu/

[30] http://www.facetmap.com/

[31] http://browserdf.org/

modelo de exploração que extrapola as interfaces facetadas existentes, o modelo de interface do BrowseRDF não suporta as operações propostas no modelo, falhando em prover ao usuário uma ferramenta eficiente de exploração. Por exemplo, não seria possível pesquisar na interface do BrowseRDF por telefones celular que sejam GSM e TDMA ao mesmo tempo. Além das limitações ligadas às operações que a interface desse sistema suporta, também há limitações quanto aos cenários de uso dessa interface. No capítulo 3 descreveremos um modelo formal de representação da navegação facetada para dados RDF e faremos um estudo mais detalhado de algumas ferramentas de navegação facetada.

Apesar de agregarem maior eficiência à tarefa de exploração do usuário do que ferramentas de busca e navegação RDF (Oren et al., 2006), o paradigma de navegação facetada, quando aplicado a bases RDF, com todas suas as variações existentes, não é suficiente para suportar a busca exploratória. Existem tarefas em que o usuário deseja comparar recursos, combiná-los, realizar interseção e diferença entre conjuntos, para se obter um conjunto de recursos de interesse. Além disso, ontologias que possuem um volume elevado de classes e propriedades são mais difíceis de serem exploradas em uma interface facetada, por ser mais complexo dispor em uma interface cerca 30 ou mais facetas. Mesmo utilizando-se de mecanismos de paginação ou sistemas de ordenação baseados na relevância das facetas, ainda assim é mais simples explorar o domínio manipulando diretamente seus elementos. Suponha por exemplo, que você queira conhecer todas as instâncias da classe Animal. É mais simples clicar em um link que retorne essa informação do que criar um conjunto de todas as instâncias e facetá-lo. Neste caso, teríamos a faceta Classe com os valores Pessoa, Animais, Plantas, etc., onde teria que ser selecionado o valor Animal para obter suas instâncias, ou seja, filtrar o conjunto inicial pelos que tenham a propriedade *type* igual a Animal.

Resumindo, o paradigma de navegação facetada, com todas as suas variações propostas, não é o mais adequado a todas as tarefas de exploração. Por exemplo, tarefas em que o usuário necessita buscar por um item conhecido é mais eficiente utilizar um mecanismo de busca por palavra-chave do que uma interface mais elaborada, como a navegação facetada (Marchionini G, 2006). No entanto, a navegação facetada não deve ser descartada como um modelo de navegação para

Web Semântica, dado que permite aos usuários fazerem consultas mais elaboradas do que a simples navegação entre recursos.

2.2.3. Linguagem de Consulta

Linguagens de consulta, tais como SQL[32], SPARQL, SERQL[33], dentre outras, são o mecanismo mais comum utilizado na recuperação de informação em banco de dados. Na Web Semântica a linguagem de consulta adotada para recuperar informação, em uma base RDF, tem sido o SPARQL, que é um padrão[34] do W3C. Segue abaixo um exemplo de uso da linguagem SPARQL:

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE
  { ?x foaf:name ?name .
    ?x foaf:mbox ?mbox }
```

Quadro 5 - conjunto de dados RDF representados na sintaxe N3.

Johnny Lee Outlaw	mailto:jlow@example.com
Peter Goodguy	mailto:peter@example.org

Quadro 6 - Exemplo de consulta na linguagem SPARQL.

[32] http://en.wikipedia.org/wiki/SQL

[33] http://www.openrdf.org/doc/sesame/users/ch06.html

[34] 15 de janeiro de 2008, SPARQL tornou-se oficialmente uma recomendação do W3C.

```
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
_:a  foaf:name   "Johnny Lee Outlaw" .
_:a  foaf:mbox   <mailto:jlow@example.com> .
_:b  foaf:name   "Peter Goodguy" .
_:b  foaf:mbox   <mailto:peter@example.org> .
_:c  foaf:mbox   <mailto:carol@example.org> .
```

Quadro 7 - Resultado da consulta SPARQL.

No fundo, qualquer sistema de exploração é baseado em uma linguagem de consulta. Porém, tais linguagens também são utilizadas diretamente através de interfaces, que nada mais são do que uma caixa de texto onde o usuário digita sua consulta na sintaxe especifica de uma dessas linguagens. No entanto, para formular uma consulta os usuários precisam deter total compreensão da estrutura dos dados e conhecimento da sintaxe da linguagem de consulta utilizada, o que torna a formulação de uma consulta uma tarefa árdua, devido à elevada carga mental demandada pelo formalismo de tais linguagens (Russel et al., 2008). Um simples erro de datilografia pode forçar o usuário a perder um tempo considerável na identificação do erro e reformulação de sua consulta.

É evidente que explorar uma base de dados RDF através de uma linguagem de consulta, seja ela qual for, não é uma tarefa amigável. No entanto, sistemas de consulta visual (VQS – Visual Query System) (Cartaci et al., 1997) visam melhorar a comunicação entre o homem e a máquina, facilitando aos usuários a tarefa de expressarem suas consultas e permitindo-lhes obter uma resposta satisfatória. Tais sistemas tornam mais ágil o uso das linguagens de consulta, forçando os usuários a formularem somente consultas válidas eximindo-os de conhecer as idiossincrasias sintáticas das linguagens.

A seguir faremos um resumo dos VQS baseados na linguagem SPARQL. O iSPARQL[35] é uma ferramenta poderosa para especificação de consultas SPARQL. No entanto, o conceito visual da interface ainda está longe do modelo mental do usuário final, pois para formular até mesmo uma consulta simples, o usuário deve deter o conhecimento de conceitos técnicos, tais como variáveis e *datatype*

[35] http://demo.openlinksw.com/isparql/

properties. Outra ferramenta de consulta visual para dados RDF é o NITELIGHT (Russell et al., 2008), que é bastante similar ao iSPARQL. A principal diferença entre as duas ferramentas reside nas notações visuais adotadas por cada uma delas, ambas praticamente suportam as mesmas funções da linguagem SPARQL. Apesar de toda a expressividade destas ferramentas, as notações visuais, por mais simples que sejam, exigem do usuário o conhecimento da sintaxe da linguagem de consulta subjacente.

O SEWASIE (Catarci et al., 2004) usa uma interface mais próxima do modelo mental dos usuários, porém permitindo-lhes fazer um conjunto limitado de consultas dentro do universo de consultas SPARQL possíveis. Nesta ferramenta, o usuário dispõe de um sistema de recomendação baseado na ontologia do domínio dos dados, que pode ser utilizada para guiá-lo na construção de sua consulta. A desvantagem dessa abordagem, é que o usuário fica limitado a um conjunto reduzido de funções da linguagem SPARQL, limitando assim sua capacidade de exploração.

Catarci et al. (Catarci et al., 1997) definiu um modelo formal de classificação de sistemas visuais de consulta, onde foram identificadas 4 categorias: sistemas baseados em formulários, sistemas baseados em gráficos, sistemas baseados em ícones e sistemas híbridos. Apesar da falta de um estudo empírico que compare as vantagens das diversas abordagens, todas elas caem no mesmo dilema: para fornecer expressividade ao usuário é demandado um elevado conhecimento técnico da linguagem subjacente.

É fácil notar que uma interface simplificada não é suficiente para expressar uma linguagem complexa como o SPARQL, RQL ou SERQL, e o meio termo entre expressividade e simplicidade não garante ao usuário o poder necessário para completar sua tarefa satisfatoriamente. O que vimos nas ferramentas existentes é praticamente uma interface de edição de consulta que demanda não só conhecimento do modelo RDF, mas também da linguagem de consulta em si.

2.2.4. Modelos de navegação RDF

Alguns autores propuseram modelos de navegação para dados RDF. No BrowserRDF, Oren et al. (2006), propuseram um modelo de navegação baseado em operações sobre grafos RDF. Tal modelo define um conjunto de operações

para seleção de recursos RDF, que simulam a navegação do usuário em uma estrutura em grafo. Oren et al.(2006) também fornecem uma interpretação e aplicação deste modelo para navegação facetada. Descreveremos mais sobre as limitações desta interpretação no capítulo 3, onde abordaremos em detalhes o tema de navegação facetada.

Szundy (2004), revisou e propôs a especificação da navegação para o método SHDM, uma modelagem para implementação de aplicações hipermídia governadas por ontologias. Dentro deste método, ele propôs o modelo navegacional que define quais informações poderão ser acessadas pelos usuários da aplicação e como estas informações poderão ser exploradas. Para tanto, ele define classes navegacionais, nós, contextos, estruturas de acesso, índices, dentre outras estruturas, que seriam os elementos de acesso aos conjuntos de recursos ou triplas de uma base RDF. Fazendo uso dessa metodologia é possível controlar como as informações serão acessadas, definindo de forma explícita ordem, agrupamento e vínculo entre elas. No entanto, a navegação neste modelo é limitada aos contextos e índices pré-definidos, não permitindo o usuário explorar as informações disponíveis numa base arbitrária, de forma flexível.

2.3. Modelo de Exploração para Dados RDF

Levando em consideração os sistemas de exploração existentes e suas limitações, propomos um modelo de exploração sobre dados RDF.

Um modelo de exploração é entendido aqui como sendo um conjunto de operações que suportam tarefas de exploração de uma base RDF. Tais tarefas envolvem a extração da informação, e ainda sua manipulação, incluindo aí a composição de informações.

O SHDM possui um modelo de consulta pouco flexível, além de não definir operações que nos permitam manipular as informações obtidas. Por exemplo, não podemos realizar uma operação de união, interseção ou diferença entre os elementos incluídos em um contexto ou índice definido neste modelo.

Modelos de navegação facetada limitam até as operações de consulta, predefinindo um conjunto de consultas que podem ser utilizadas pelos usuários.

O modelo proposto por Oren et al. falha ao limitar a manipulação da informação. As primitivas definidas neste modelo não possibilitam determinar a

união entre dois conjuntos de triplas ou recursos. Tal operação de união suporta diversas tarefas do usuário. Suponha, por exemplo, que você tenha um conjunto de amigos do trabalho e outro de familiares e deseja obter a união de tais recursos, para enviar uma mensagem eletrônica a todos. Operações de composição da informação são fundamentais na realização desse tipo de tarefa.

Acreditamos que um modelo de manipulação da informação deve ser visto como um modelo de operação sobre conjuntos. Tal abordagem leva em consideração que, ao exploramos um espaço de dados, estamos formando e manipulando conjuntos de itens de informação. Tal padrão de navegação foi definido em Rossi (Rossi et al., 1998) como *set-based navigation*. Na formação de um conjunto utilizamos uma consulta sobre a base ou a operação de união, interseção ou diferença sobre os conjuntos existentes. No nosso caso, os conjuntos que manipularemos são conjuntos de triplas e recursos RDF.

O modelo de exploração aqui proposto permite a construções de consultas sobre uma base RDF. Tal modelo será baseado em operações sobre conjuntos e operações de consulta sobre as linguagens SPARQL e suas extensões.

2.3.1. Conjuntos

A base de nosso modelo reside na teoria dos conjuntos. No domínio RDF, existem dois tipos de conjuntos relevantes: o conjunto de recursos, e o conjunto de triplas RDF.

Considerando o conjunto como sendo um conjunto de recursos RDF, estamos na verdade tratando de um conjunto de URIs, literais ou nodos brancos. Neste caso, operações sobre conjuntos, tais como união, interseção e diferença resumem-se a unir ou comparar tais recursos. No entanto, quando manipulamos conjunto de triplas, existem duas abordagens para a aplicação de tais operações. Ou comparamos as triplas em si, ou consideramos que uma tripla faz parte de uma entidade definida pelo sujeito da tripla – i.e., estas triplas são afirmações sobre esta entidade, que é o item de interesse - e as operações são aplicadas sobre tal sujeito, e não sobre as triplas em si.

No caso de conjunto de recursos, um caso especial aparece quando os temos nodos brancos[36] (*bnode* – sigla em inglês). Nodos brancos são utilizados para representar uma coleção de recursos. São comumente utilizados juntamente com os elementos RDF SEQ, BAG, ou LIST. No modelo RDF, apesar de nodos brancos possuírem URIs, eles, em teoria, não são identificáveis como os recursos RDF. A linguagem SPARQL, por exemplo, não suporta realizarmos uma pesquisa por um nodo branco. O uso de nodos brancos é desencorajado sendo mais recomendado o uso de URIs (Bizer et al., 2008). No modelo aqui proposto iremos tratá-los como URIs, e somente extensões do SPARQL que suportem operações com nodos brancos serão capazes de implementar tal característica dos dados.

O modelo aqui proposto utilizará tanto conjunto de recursos quanto conjunto de triplas RDF.

Uma tripla é denotada por (s,p,o) onde s, p e o são recursos. Tome A com sendo um conjunto de triplas. O conjunto R de recursos de A, pode ser definido por:

$$R = \text{conjunto de todas as URIs}$$

Quadro 8 – Definição de conjunto de recursos.

Dado o conjunto de triplas A, também temos as seguintes funções sobre A:

$$S = R_s(A) = \{x \in R \mid (x,p,o) \in A\}$$
$$P = R_p(A) = \{x \in R \mid (s,x,o) \in A\}$$
$$O = R_o(A) = \{x \in R \mid (s,p,x) \in A\}$$

Quadro 9 – Definição de conjunto de sujeito, predicado e objeto.

Onde, S é o conjunto de sujeitos, P o conjunto de predicados e O o conjunto de objetos das triplas de A.

A figura abaixo mostra as relações entre os conjuntos de recursos S, P e O e o conjunto de triplas A. A tracejado mais externo delimita todas as triplas; cada

[36] http://www.w3.org/TR/2003/WD-rdf-concepts-20031010/

linha representa uma única tripla, como está em evidência na terceira linha; e cada tracejado interno delimita o conjunto de sujeitos, predicados ou objetos.

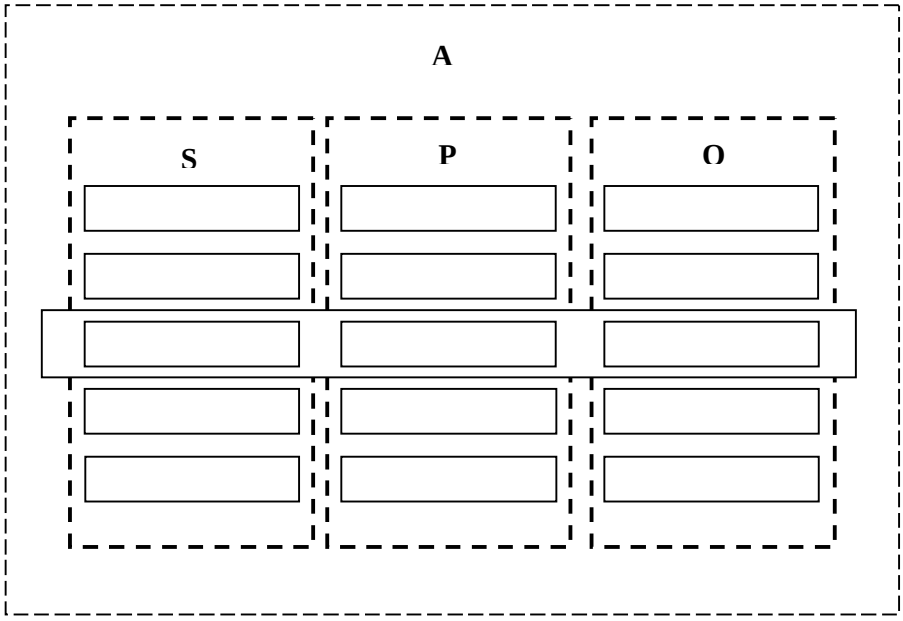

Figura 6 – Esquema representando os conjuntos de triplas, sujeitos, predicados e objetos.

2.3.2. Operações de Consulta

Denominaremos de operações de consulta as operações realizadas sobre o grafo RDF que visem explorar as relações entre os recursos RDF. No mais baixo nível de implementação, uma operação de consulta é uma consulta SPARQL na base. Na verdade, nosso modelo define uma única operação de consulta que chamaremos de *SPO*, definida a seguir.

Dado um conjunto de triplas A, um conjunto de recursos R e os subconjuntos de R, S, P, e O, podemos definir a função SPO como:

O conjunto de todas as triplas de A:

$SPO(\varnothing,\varnothing,\varnothing) = A$

O conjunto com apenas as triplas de A em que o sujeito está em S:

$SPO(S,\varnothing,\varnothing) = \{(s,p,o) \in A \mid s \in S \}$

O conjunto de triplas de A em que o predicado está em P:

$SPO(\varnothing,P,\varnothing) = \{(s,p,o) \in A \mid p \in P \}$

O conjunto de triplas de A em que o objeto está em O:

SPO(∅,∅,O) = {(s,p,o) ∈ A | o ∈ O}

O conjunto de triplas de A em que o sujeito está em S e o predicado está em P:

SPO(S,P,∅) = {(s,p,o) ∈ A | s ∈ S e p ∈ P }

O conjunto de triplas de A em que o sujeito está em S e o objeto está em O:

SPO(S,∅,O) = {(s, p,o) ∈ A | s ∈ S e o ∈ O}

O conjunto de triplas de A em que o predicado está em P e o objeto está em O:

SPO(∅,P,O) = {(s, p,o) ∈ A | p ∈ P e o ∈ O}

O conjunto de triplas de A em que o sujeito está em S, o predicado está em P e o objeto está em O:

SPO(S,P,O) = {(s, p,o) ∈ A | s ∈ S e p ∈ P e o ∈ O}

Fazendo uso das definições acima, a função SPO(∅,∅,∅) pode ser traduzida na seguinte consulta SPARQL:

```
SELECT ?s ?p ?o WHERE  {?s ?p ?o} .
```

Para os dados RDF abaixo expressos em N3, esta consulta retorna todas as triplas existentes.

```
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
_:a  foaf:name   "Johnny Lee Outlaw" .
_:a  foaf:mbox   <mailto:jlow@example.com> .
_:b  foaf:name   "Peter Goodguy" .
_:b  foaf:mbox   <mailto:peter@example.org> .
_:c  foaf:mbox   <mailto:carol@example.org> .
```

Já a função SPO(∅,{foaf:mbox}, ∅) retorna todas que triplas que tenham a propriedade foaf:mbox:

```
_:a  foaf:mbox   <mailto:jlow@example.com> .
_:b  foaf:mbox   <mailto:peter@example.org> .
_:c  foaf:mbox   <mailto:carol@example.org> .
```

E pode ser traduzida na seguinte consulta SPARQL:

```
SELECT ?s ?p ?o WHERE  { ?s ? p ?o. Filter (p = foaf:mbox)} .
```

É importante notar que, embora em SPARQL não possamos trabalhar com o conceito de conjunto de recursos e defini-los como parâmetro da consulta, nossa operação SPO suporta tal abordagem. A vantagem do uso dessa abordagem pode ser evidenciada no seguinte cenário: suponha que você deseje obter dentre um conjunto de recursos do tipo Telefone Celular, aqueles que possuem *browser* Wap e reproduzem MP3. Note que nestes cenários estamos trabalhando com dois conjuntos, um de aparelhos celulares e outro de características desses aparelhos. Uma consulta SPO para encontrá-los poderia ser formulada da seguinte forma:

```
SPO({Aparelho1,Aparelho2, Aparelho3},{Browser,Reproduz},{Wap,MP3})
```

2.3.3. Operações sobre conjuntos

No nosso modelo definiremos operações sobre conjuntos. Considerando que nossos conjuntos são conjuntos de recursos ou de triplas, definiremos as seguintes operações:

Seja M e N sendo conjuntos de triplas.

Seja $U_R = \{x \in R \mid x \in R_v(M) \text{ ou } x \in R_{v'}(N)\}$, onde v e v' podem ser s, p ou o
$U = SPO(U_R,\varnothing,\varnothing)$

Seja $I_R = \{x \in R \mid x \in R_v(M) \text{ e } x \in R_{v'}(N)\}$, onde v e v' podem ser s, p ou o
$I = SPO(I_R,\varnothing,\varnothing)$

Seja $D_{\underline{R}} = \{x \in R \mid x \in R_v(M) \text{ e } x \notin R_{v'}(N)\}$, onde v e v' podem ser s, p ou o
$D = SPO(D_R,\varnothing,\varnothing)$

As operações de união, interseção e diferença são calculadas sobre os conjuntos de recursos (sujeitos, predicados, ou objetos) de um conjunto de triplas. Por exemplo, para a operação (M, o) ∩ (N,s), onde (M,o) = R_O (M) representam todos os recursos que são objetos no conjunto triplas M e (N,s) = R_S(N) representam todos recursos que são sujeitos no conjunto triplas N, as operações são calculadas sobre estes dois conjuntos de recursos, e resultado final é o conjunto de triplas em que os recursos resultantes das operações são sujeitos.

Dito de outra forma, as operações de união, interseção e diferença formam conjunto de triplas operando o sujeito, predicado ou objeto das triplas.

Por exemplo, dados os conjuntos N e M abaixo:

N	M
Russia capital Moscou	China capital Pequim
Russia continente Ásia	China continente Ásia
Russia idioma Russo	China fronteiraCom Russia

Temos os seguintes exemplos de operações sobre o conjunto dos sujeitos de N e objetos de M:

- (N,s) ∩ (M,o) = SPO({Russia} ∩ {Pequim, Ásia, Russia},∅,∅) = SPO({Russia},∅,∅)
- (N,s) ∪ (M,o) = SPO({Russia} ∪ {Pequim, Ásia, Russia },∅,∅) = SPO({Russia, Ásia, Pequim },∅,∅)
- (N,s) - (M,o) = SPO({Russia} - { Pequim, Ásia, Russia },∅,∅) = SPO({},∅,∅)

2.3.4. Exemplo de uso

Descreveremos um cenário de exploração para exemplificar como a tarefa se traduz para o conjunto de operações deste modelo. O cenário a seguir foi definido sobre a base RDF Mondial[37], que contém informações geográficas.

Cenário: Devido à possibilidade de falta de água potável no mundo, a ONU resolveu propor ao governo russo que despolua e preserve os lagos contidos em

[37] http://www.dbis.informatik.uni-goettingen.de/Mondial/

seu território. Para isso, um analista da ONU precisa conhecer o nome dos lagos que estejam inteiramente contidos em território russo para formular um relatório.

Tarefa: Forme um conjunto com o nome dos lagos russos que estejam localizados exclusivamente em território russo.

Tal tarefa pode ser decomposta nos seguintes passos:

1. Encontre todos os lagos na base de dados, que denominaremos de TL;
2. Encontre todos os lagos da Rússia, obtendo um conjunto que chamaremos de LR;
3. Encontre os países que fazem fronteira com a Rússia, que denominaremos de VR;
4. Encontre os lagos que estão localizados nos países vizinhos da Rússia, obtendo um conjunto que chamaremos de LV, e;
5. Construa o conjunto dos lagos contidos exclusivamente na Rússia, calculando a diferença entre LR e LV (LR - LV) = LIR.

Abaixo seguem as expressões definidas de acordo com as operações de nosso modelo. As URIs utilizadas são as mesmas existentes na base Mondial. Utilizaremos o prefixo mondial: para representar a URI http://www.semwebtech.org/mondial/10/.

1. Obtendo todos os lagos:
   TL = SPO (∅,∅,{mondial:meta#lake})
2. Obtendo todos os lagos da Rússia:
   LR = SPO(R_S (TL), {mondial:meta#locatedIn},{mondial:countries/R/})
3. Obtendo países que fazem fronteira com a Rússia
   VR = SPO({mondial:countries/R/}, {mondial:meta#neighbor},∅)
4. Obtendo lagos contidos nos vizinhos da Rússia:
   LV = SPO(R_S (LR), {mondial:meta#locatedin}, R_S (VR))
5. Obtendo lagos contidos inteiramente na Rússia:
   LIR = R_S (LR) - R_S (VR)