3 Estado da arte

Nesse capítulo serão discutidas ferramentas, tecnologias e soluções existentes na área da web semântica.

Na seção 3.1 e 3.2 deste capítulo serão discutidas abordagens alternativas à solução proposta, que podem ser utilizadas na resolução dos problemas que foram apresentados anteriormente. As principais vantagens e problemas relacionados com essas alternativas também serão apresentados. As duas principais abordagens solucionam os problemas realizando consultas diretamente na base de dados ou via aplicações pré-definidas e construídas de forma específica. Existem poucas ferramentas genéricas que se propõem a resolver exatamente os problemas discutidos no capítulo anterior.

Na seção 3.3 deste capítulo, algumas ferramentas que possuem objetivos similares ou que estão parcialmente relacionadas com a área do trabalho em questão serão apresentadas e discutidas. Nesse caso, serão apresentadas as contribuições dessas ferramentas para o estado da arte da web semântica.

3.1 A linguagem de consultas SPARQL

Uma das abordagens possíveis para resolver problemas de consultas a bases de dados semânticos é através da linguagem SPARQL. Trata-se de uma linguagem específica para realizar consultas a bases de dados RDF.

SPARQL é uma linguagem de consulta de baixo nível, é possível fazer uma analogia entre SPARQL e a linguagem de consultas a bases relacionais SQL [Eisenberg et al, 2004]. Para utilizar diretamente a linguagem de consultas SPARQL é necessário conhecer sua sintaxe e as ontologias em que os dados estão armazenados. É possível utilizar consultas SPARQL para compreender as ontologias em que os dados estão representados, mas essa pode se tornar uma tarefa bastante trabalhosa e pouco eficiente.

Realizar consultas simples utilizando SPARQL não é uma tarefa difícil para profissionais da área. Por exemplo, utilizando uma base que possui uma ontologia simples para árvores genealógicas, para realizar uma consulta que retorne o nome dos filhos de uma determinada pessoa, podemos utilizar a seguinte consulta:

```
PREFIX family: <http://family-ontology/predicates/>
PREFIX people: <http://example-domain.com/people/>
PREFIX rfds: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?name
WHERE {
  people:carla family:isMotherOf ?daughter .
  ?daughter rfds:label ?name .
}
```

As três primeiras linhas são utilizadas para definir prefixos, de forma a tornar a consulta mais compacta e menos propensa a erros. Ou seja, "family:" se torna um acrônimo de "<http://family-ontology/predicates/>" e assim por diante. Na quarta linha é definido que a consulta deve retornar o resultado da variável "name" (o ponto de interrogação antes de uma palavra indica que se trata de uma variável). Dentro do bloco "WHERE", na quinta linha, estão as restrições. A primeira define que existe uma variável chamada "daughter" que possui o valor de todos os recursos que estão na posição de objeto em uma tripla que possua "people:carla" como sujeito e "`family:isMotherOf`" como predicado. A segunda restrição fixa os recursos da variável "daughter" como sujeito, o predicado como "`rfds:label`" e define o valor da variável "name", que é o retorno da consulta. Realizando essa consulta em um banco RDF que possua as triplas apresentadas na introdução, o retorno da consulta seria o nome "Patrícia".

Embora uma consulta de pouca complexidade seja relativamente simples de ser montada, uma mais complexa que relacione e filtre diversos dados, se mostra uma tarefa difícil até para profissionais da área. Por exemplo, em uma base que possui dados sobre publicações em conferências científicas, a tarefa de descobrir pesquisadores de uma dada instituição que publicaram pelo menos um trabalho em cada conferência da base pode se mostrar bastante trabalhosa.

A realização de consultas utilizando SPARQL para usuários leigos acessarem dados diretamente em bases RDF se mostra inviável, devido à

complexidade e os conhecimentos específicos necessários para a montagem das consultas. Existem aplicações que possibilitam a montagem de consultas a bases de forma visual [Catarci et al, 1997], para facilitar a exploração dos dados, mas como essas aplicações não conhecem a estrutura da base, a ajuda fornecida é limitada. A linguagem SPARQL é uma boa forma para aplicações acessarem bases RDF a fim de retornar para usuários os resultados de forma mais amigável, abstraindo as complexidades envolvidas nas consultas.

3.2 O acesso a bases RDF através de aplicações pré-definidas

Uma forma de prover acesso ao conteúdo semântico contido em bases RDF é através do desenvolvimento de aplicações específicas, uma forma muito comum de navegação em dados armazenados em bases relacionais. Nesse método tradicional de desenvolvimento, uma aplicação é projetada e implementada em alguma linguagem de programação para uma base, ou ontologia específica.

Existem alguns frameworks que auxiliam o desenvolvimento de aplicações que pretendem acessar bases em RDF. Em alguns casos esses frameworks provêem uma abstração do modelo RDF e em outros casos somente auxiliam no acesso e na manipulação dos dados. O objetivo desses frameworks é minimizar ao máximo o retrabalho do desenvolvedor nas partes comuns de acesso aos dados. Entre os frameworks mais relevantes, podemos destacar o Jena [Carroll et al, 2004], desenvolvido para a linguagem Java [Arnold, 2005] e o ActiveRDF [Oren et al, 2007], que provê uma camada de persistência para bases RDF para a linguagem Ruby [Flanagan et al, 2008].

O framework Jena provê diversas facilidades para a manipulação de dados em RDF, além de possuir suporte para RDFS e OWL. Através do Jena é possível realizar consultas em bases RDF utilizando SPARQL e também escrever e ler triplas RDF em diversos formatos, como RDF/XML, N3 e N-Triples. No entanto a funcionalidade que mais se destaca no Jena é chamada de Jena Rules[3]. Trata-se de uma poderosa ferramenta de inferência sobre dados semânticos. Utilizando Jena Rules é possível, por exemplo, validar consistência de dados e criar novas

[3] Jena 2 Inference support - http://jena.sourceforge.net/inference/

triplas, derivadas da aplicação de regras de inferência sobre informações contidas na base.

O framework ActiveRDF cria uma espécie de abstração do modelo RDF/RDFS para Ruby. Como Ruby é uma linguagem dinâmica, o framework consegue criar classes e métodos que refletem as classes e atributos existentes no modelo RDFS das bases acessadas. O objetivo é tornar mais fácil e ágil o desenvolvimento e as manipulações dos dados da base.

Outro framework que provê funcionalidades semelhantes é o RDFLib[4], desenvolvido para a linguagem Python [Sanner, 1999]. Além de facilitar a manipulação de triplas RDF, RDFLib permite o acesso a triplas no formato XML e também suporta a criação de consultas para diversos tipos de bases utilizando SPARQL.

Apesar da existência de frameworks que auxiliam o desenvolvimento de aplicações pré-definidas para bases RDF, esse método tradicional de desenvolver aplicações apresenta algumas falhas em pontos que fazem com que a sua utilização seja pouco eficiente para o problema específico da web semântica. O formato RDF prevê a fácil evolução das bases de dados. O uso de ontologias padronizadas visa o cruzamento de informações. A larga gama de usuários na web traz um forte requisito de customização na forma que o conhecimento é consumido. Uma ferramenta genérica que permita a construção de aplicações específicas de forma fácil e extensível pode ser uma solução mais adequada à realidade da web semântica.

3.3 Ferramentas semânticas

Existem algumas ferramentas que possibilitam a navegação e a manipulação de dados semânticos. A maioria permite somente a consulta a bases RDF de forma visual para facilitar a exploração dos dados, outras permitem transformações em cima dos dados das bases.

O primeiro tipo de aplicação, que só permite a navegação pelos dados de bases RDF, geralmente exibe as informações da base através de um grafo, formado pelos recursos e seus atributos e relações, sendo que na maioria das

[4] RDFLib - http://www.rdflib.net/

aplicações a navegação é realizada através dos recursos. Quando um usuário está visualizando um recurso, são exibidos seus atributos e relacionamentos. É possível então navegar para outro recurso e assim por diante. Como essas aplicações são genéricas, isto é, não conhecem a estrutura da base, a ajuda fornecida é limitada. O Paged Graph Visualization (PGV) [Deligiannidis et al, 2007] e o gFacet [Heim et al, 2008] são dois exemplos, sendo que o gFacet combina a visualização da base através de um grafo com um filtro facetado.

O segundo tipo de ferramenta permite a manipulação e transformação de dados semânticos, possivelmente vindos de diferentes fontes. Existem algumas linhas de pesquisa que estudam esse tipo de ferramenta semântica. Algumas dessas ferramentas se propõem a resolver problemas relacionados com os apresentados nesse trabalho.

Um dos grupos que pesquisam soluções para problemas relacionados é o DERI, que realiza estudos sobre Semantic Web Pipes [Morbidoni et al, 2007] e possui uma ferramenta chamada DERI Web Data Pipes. A ferramenta, baseada no Yahoo Pipes[5], possui uma interface que possibilita a usuários extrair dados semânticos representados em RDF, combinar dados de diferentes bases, realizar certas transformações pré-definidas e aplicar filtros para disponibilizar resultados que os interessam. Essa seqüência de transformações pode ser armazenada e compartilhada, possibilitando também sua evolução. O nome Pipe vem da semelhança com um operador existente na plataforma UNIX [Coffin, 1990] que provê um fluxo de dados entre aplicações (no caso do DERI pipes, os operadores pré-definidos). Embora a criação de um fluxo (pipe) possa ser feita facilmente com o auxílio de uma boa ferramenta visual, existe a necessidade do usuário ter conhecimento prévio das ontologias utilizadas pela base a ser explorada e da linguagem padrão de consulta a bases RDF, a SPARQL. Devido a essas limitações, a utilização por usuários com pouca experiência em programação é difícil.

Outra solução proposta para exploração de dados semânticos é o Explorator [Araujo et al, 2009]. Trata-se de uma ferramenta de exploração de bases RDF de forma visual proposta por Samur Araújo em sua dissertação de mestrado (PUC, 2009). O Explorator, assim como algumas ferramentas mencionadas acima, possui

[5] Yahoo! Pipes - http://pipes.yahoo.com/pipes/

o objetivo de permitir a exploração de uma base RDF para usuários que não possuem nenhum conhecimento prévio do domínio. A diferença do Explorator para as outras ferramentas mencionadas é a forma de visualização e navegação nos dados da base. O modelo de consultas do Explorator se baseia em manipulação direta, buscas e query-by-example [Zloof, 1977], além de prover uma estrutura para navegação facetada sobre os dados RDF.

A proposta desse trabalho é estender o Explorator, transformando-o em um gerador de aplicações para consultas a bases RDF/RDFS. No próximo capítulo essa proposta será apresentada e o Explorator discutido com mais detalhes, assim como a sua transformação no ambiente de desenvolvimento de aplicações do Excelplorator, a ferramenta que esse trabalho propõe.