Ruby Semantic Web Pipes

Semantic Web Pipes just got easier

Daniel Hahn[2] and Michele Barbera[2]

NET7 Internet Open Solutions - Pisa
(hahn|barbera)@netseven.it

Ruby SemPipes are based on the *Semantic Web Pipes* proposed by Giovanni Tummarello, Christian Morbidoni et. al. [1]. Semantic web pipes provide a mechanism similar to Yahoo! Pipes[1], but work on semantic web data. They allow users to filter and recombine semantic web data on the fly (more information can be found in the cited work). The original Semantic Pipes also have some advanced features that are not touched here, such as the revocation of triples.

Ruby SemPipes[2]. are an intentionally simple reimplementation of the Semantic Web Pipes. They provide an easy "pipe description language" . Pipes can easily be created and modified from a script or application, the definitions are painless to read and they don't require the pipe developer to manually edit XML files or SPARQL queries.

To make things simple, *Ruby SemPipes* treat all RDF graphs as enumerations of triples. A pipe itself is also an enumeration over it's result set and any enumeration of triples (including other pipes) can be used as an input for a pipe.

Using Eyal Oren's *ActiveRDF* [2] library, it will be quite easy to provide input adapters that connect to "real" RDF endpoints and perform SPARQL queries. These adapters could then be used as input to a pipe. Conversely, output adapters could re-write the result of a pipe to a common storage format, such as RDF/XML.

The first implementation of *Ruby SemPipes* contains a *merge*, *filter* and *rewrite* operator; a *smushing* operator is in the pipeline. The *filter* and *rewrite* operators allow a simple but powerful filtering (or rewriting) of triples using regular expression. In addition to the operators, pipes may also contain *actions* which allow the developer to manipulate triples programmatically. All built-in operators are themselves pipes written in the same way as user-created pipes.

Each pipe takes on ore more sets of RDF triples as an input; the pipe itself will operate on the union (including duplicates) of the input sets – the *merge* operator will actually just remove duplicates from the input.

Listing 1.1. Sample Pipe setup

```
module SemPipe
  # The first pipe
  define_pipe :new_pipe do
    merged_triples = merge(input)
```

[1] http://pipes.yahoo.com/

[2] The first experimental version is available via public svn from http://svn.talia.discovery-project.eu/talia-intern/playground/ruby_pipes/trunk

```
    # Rewrite Tim Berners-Lee entries
    rewritten_triples = rewrite(merged_triples) do
      rewrite :all,
      "<http:\/\/dbpedia.org\/resource\/Tim_Berners-Lee>",
      "<http://www.w3.org/People/Berners-Lee/card#i>"
    end

    # Remove all triples dealing with Alice
    self.output = filter(rewritten_triples) do
      reject :all "<http:\/\/strangeworld.com\\/Alice"
    end
  end

  # Define another pipe
  define_pipe :another_pipe do
    temp = new_pipe(input)
    self.output = action(temp) do |input|
      input.to_a << [@subject, @predicate, @object]
    end
  end

  # Use the pipe
  AnotherPipe.new(load_some_input).each { |triple| puts triple }
end
```

Listing 1.1 shows a sample setup of two *Ruby SemPipe*s. The first pipe merges all input sets, then rewrites Tim Berners-Lee's dbpedia URL with his own URL and finally filters out all entries dealing with Alice. Pipes can be easily connected using variables; the special variable `input` is used in the pipe description to denote the pipe's input. The `self.output` variable is used to assign a value to the current pipe's result set.

The second pipe shows that other pipes can be used just as easily as the built-in operators. The pipe also contains a generic action that adds a (complete random) triple to the graph.

Acknowledgements

This work has been supported by Discovery, an ECP 2005 CULT 038206 project under the EC eContentplus programme. Of course we also like to acknowledge the original work of Christian and Giovanni.

References

1. Morbidoni, C., Polleres, A., Phuoc, D.L., Tummarello, G.: Semantic web pipes. Technical report, DERI (2007)
2. Oren, E., Debru, R., Gerke, S., Haller, A., Decker, S.: ActiveRDF: Object-Oriented Semantic Web Programming. In: 16th International World Wide Web Conference (WWW2007), Banff, Alberta, Canada. (8-12 May, 2007) 817–823