Abstract. This paper illustrates a technique allowing contextual identification of nodes and links in Named Graphs.

Keywords: Linked Data, Named Graph, eXtensible Resource Identifier (XRI).

1 Introduction

Named Graphs [1] rigidly bound an identifier to a given “copy” of an RDF graph. However, they do not provide the ability to identify nodes and links as parts of a graph. The main contribution of this paper is to illustrate a technique to do this, its benefits and its backward compatibility with Named Graph and Linked Data.

2 Structured and Semantically Aware Identifiers and Contexts

Let us consider the eXtensible Resource Identifier (XRI), in its application XRI Data Interchange (XDI). The following XDI statement:

```
```

containing three “XRrefs” (=John.Smith, +knows, =Alice.Doe) is the “X3” representation of an equivalent N3 sentence in a RDF graph:

```
```

At a first glance, XDI misses “namespaces” which are instead in RDF; however, it provides a native concept called “context”. XDI contexts work by concatenating existing XRrefs to form new “multiple subsegmented” XRrefs [2]. For example, property instance +knows under the context of subject =John.Smith is identified by the two subsegments XRref =John.Smith+knows. In turns, object =Alice.Doe under the context of =John.Smith+knows is identified by the three subsegments XRref =John.Smith+knows=Alice.Doe.

This approach caters for language element identification, a necessary condition to efficiently support reification. Additionally, it natively solves many use cases
proposed in [3]. For example, =Rachel.Stein+mother=Lynn.Stein and =Lynn.Stein @MIT+professor=Lynn.Stein, are three different XRIrefs. Being subjects of separate statements in different contexts their attributes do not mix; but, semantically, one could discover in them three different roles for the same referent\(^1\). Finally, a finer grained assignment of versioning, timestamps, access permissions, and metadata to nodes and links in Named Graphs is made possible.

3 Is XDI really compatible with Named Graphs and Linked Data?

Actually a XRIref is an abstract identifier and needs to be resolved into a URIref [4] by detecting a “resolution context” [2]. Consider the following Named Graph:

\[
\text{ex:myGraph\{ex:John.Smith foaf:based_near ex:Alice.Doe .\}}
\]

If the inner statement contained XRIrefs instead of URIrefs, one could assume the graph’s name http://example.org/res/myGraph as their resolution context; for example, the object in this statement, would be assigned the URIref http://example.org/res/myGraph/=John.Smith knows =Alice.Doe.

On the other hand “cross references” enable to encapsulate URIrefs into a XRIref, replacing single segment XRIrefs with corresponding “cross-referenced URIrefs”. By applying this transformation and by following the encoding rules in [2] the previous reference gets rid of XRIrefs, becoming the equivalent (yet less human appealing)

\[
\text{http://example.org/res/myGraph/(http%253A%252F%252Fexample.com%252Fresource%2523John.Smith)(http%253A%252F%252Fxmlns.com%252Ffoaf%252F0.1%252Fknows)(http%253A%252F%252Fexample.com%252Fresource%2523Alice.Doe)}
\]

References


\(^1\) We think that this feature could provide a formal support to codify the “follow your nose” approach. This is what “structured” and “semantically aware” stands for.