Programming with Linked Data and Linked Data Services

Andreas Harth
Institute AIFB
Karlsruhe Institute of Technology (KIT)

May 3, 2012

> In der Beschränkung zeigt sich
> erst der Meister
>
> Johann Wolfgang von Goethe,
> Das Sonett, 1815

1 Introduction

Systems in nature and technology can exhibit complex behaviour even if they are based on simple principles. The web is built on a set of basic principles, yet is an extremely complex artefact that grew out of decentralised publishing of content (hypertext documents but increasingly also data). In addition, the web provides uniform interfaces to functionality (via web forms but increasingly also via APIs). Much of the content and functionality are provided by legacy systems or technologies that predate the web. As such, the web provides a prime scenario for data integration and system interoperation.

2 Web Architecture

The web's basic building block is a resource identified via URIs, following a loosely object-oriented view. Resources may have multiple representations – multiple file formats depending on user agent preference. For example, a user agent can request an image in PNG or GIF given the server provides these options.

Resource representations may contain references (links) other resources. The use of identifiers across sources (the so-called "Hypermedia" principle) allows for decentralised publishing and unilateral interlinking between sources.

The protocol that underlies the web - HTTP - provides a restricted set of operations (HTTP "verbs" applied to the URI "objects"). User agents (web browsers or mobile applications, say) can create, read, update and delete resources (PUT, GET, POST, DELETE) via HTTP. While defining new operations is possible in principles, systems should restrict themselves to the handful of standard HTTP operations.

3 Linked Data and Linked Data Services

Linked Data adds means for basic knowledge representation to traditional web architecture. Linked Data makes use of a distinction between resource and documents (information resource) and of RDF, a format for encoding graph-structured data. The atomic construct in RDF are triples consisting of subject-predicate-object (e.g., stating that Anna knows Bert:

```
<http://xmpl.org/doc.rdf#anna> foaf:knows <http://xmpl.org/doc.rdf#bert> .
```

Triples can also contain literals on the object position (either untyped or typed using XML Schema Datatypes).

RDF triples occur in documents (e.g. in `http://xmpl.org/doc.rdf`). Linked Data principles mandate a correspondence between the resource (e.g., `http://xmpl.org/doc.rdf#anna`) and the information resource/document (e.g., `http://xmpl.org/doc.rdf`) which contains a description of the resource.

Linked Data Services (LIDS) [1] cover a method to constructing URIs for resources including parameters. Also, LIDS include a description of input and output, and the relation between input and output.

4 Data Access and Manipulation

One can view Linked Data resources in object-oriented terminology. Linked Data resources are objects and can be members of classes. Classes can be defined using ontology languages such as RDFS and OWL (we only consider a simple object model, upon which methods for more elaborate class descriptions can be layered).

Linked Data resources can contain arbitrary many attributes which are dynamically typed (literals can be integer, date, string or other datatypes). Linked Data resources can have relationships to arbitrary many other Linked Data resources.

Operations are those supported via basic web architecture. One can create (HTTP PUT) or delete (HTTP DELETE) a Linked Data resource. One can read all attributes and relationships data (HTTP GET). One can add/remove (update) data (HTTP POST). HTTP operations may lead to either success (status codes 2xx) or failure (status codes 4xx and 5xx).

Such a restricted interface facilitates the collaborative construction of elaborate systems in complex information ecosystems. The principles behind web architecture are simple and cater for the requirements in large distributed systems (e.g., stateless communication, concurrent access, caching, heterogeneity of data and systems). Linked Data and Linked Data Services build on web architecture and semantic technologies and allow for accessing information and functionality.

5 Conclusion

The success of the web shows that basic web architecture principles can provide the foundations for very large decentralised systems which enable data integration and system interoperation. Information-rich programming should support data manipulation in such a web scenario.

References

[1] S. Speiser and A. Harth. Integrating Linked Data and Services with Linked Data Services. In *Ith Extended Semantic Web Conference (ESWC 2011)*, 2011.