Illuminating DSpace‘s Linked Data Support

Pascal-Nicolas Becker

https://lib-co.de/or16

The Library Code

DSpace Service Provider

About myself

- Computer scientist and developer
- Almost 10 years of experience with repositories
- DSpace Committer since July 2014
  - Automatic assignment of DOIs with DataCite
  - Linked Data support
  - Many improvements and bugfixes
- Working for Technische Universität Berlin
- Founded a DSpace service provider in 2015:
  The Library Code
- Diploma Thesis:
  Repositories and the Semantic Web
  http://doi.org/bd9k (German)

Motivation

The Semantic Web

i
know
what
you
mean...
THE SEMANTIC WEB
A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities
by TIM BERNERS-LEE, JAMES HENDLER and ORA LASSILA
PHOTOILLUSTRATIONS BY MIGUEL SALMERON

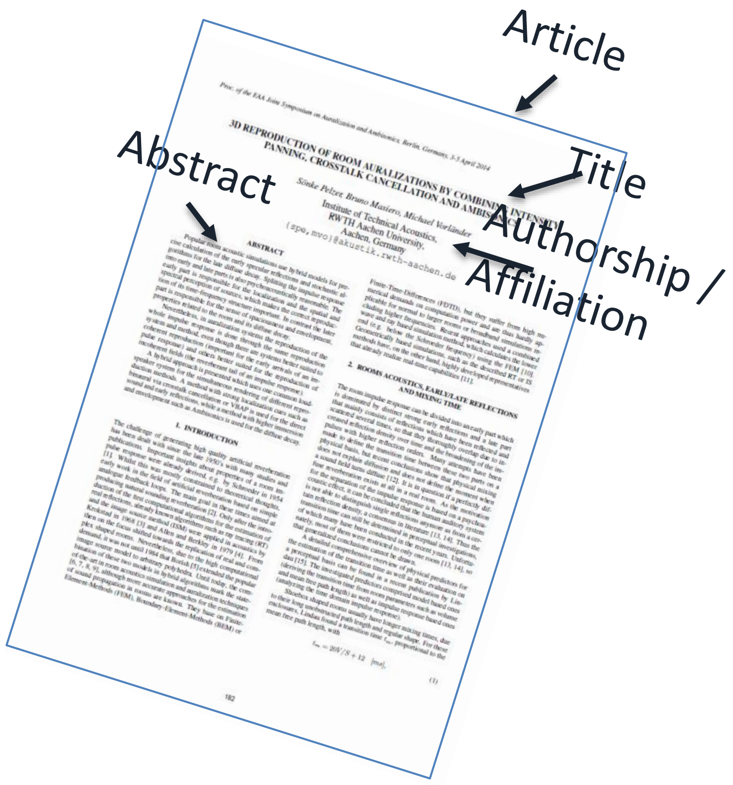
„Information varies along many axes. One of these is the difference between information produced primarily for human consumption and that produced mainly for machines. […]

To date, the Web has developed most rapidly as a medium of documents for people rather than for data and information that can be processed automatically.“

Berners-Lee, Handler, Lasilla:
The Semantic Web
In: Scientific American 284.5, 2001

Basic idea

- Humans recognize context
- Hard to build algorithms capable of doing so, even while AI is getting better
- Basic idea of the Semantic Web: make implicit information explicit
- Include explicit information in the web:
  This URI represents an article…
  The article was written by …
- Use vocabularies and ontologies to describe terms and relations between them


Proc. of the EAA Joint Symposium on Auralization and Ambisonics, Berlin, Germany, 3-5 April 2014

3D REPRODUCTION OF ROOM AURALIZATIONS BY COMBINING INTENSITY PANNING, CROSSTALK CANCELLATION AND AMBISONICS

Sönke Pelzer, Bruno Masiero, Michael Vorländer
Institute of Technical Acoustics, RWTH Aachen University, Aachen, Germany
{spe,mvo}@akustik.rwth-aachen.de

ABSTRACT

[illegible]

1. INTRODUCTION

[illegible]

2. ROOMS ACOUSTICS, EARLY/LATE REFLECTIONS AND MIXING TIME

[illegible]

182

The Semantic Web

The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

Berners-Lee, Handler, Lasilla:
The Semantic Web
In: Scientific American 284.5, 2001

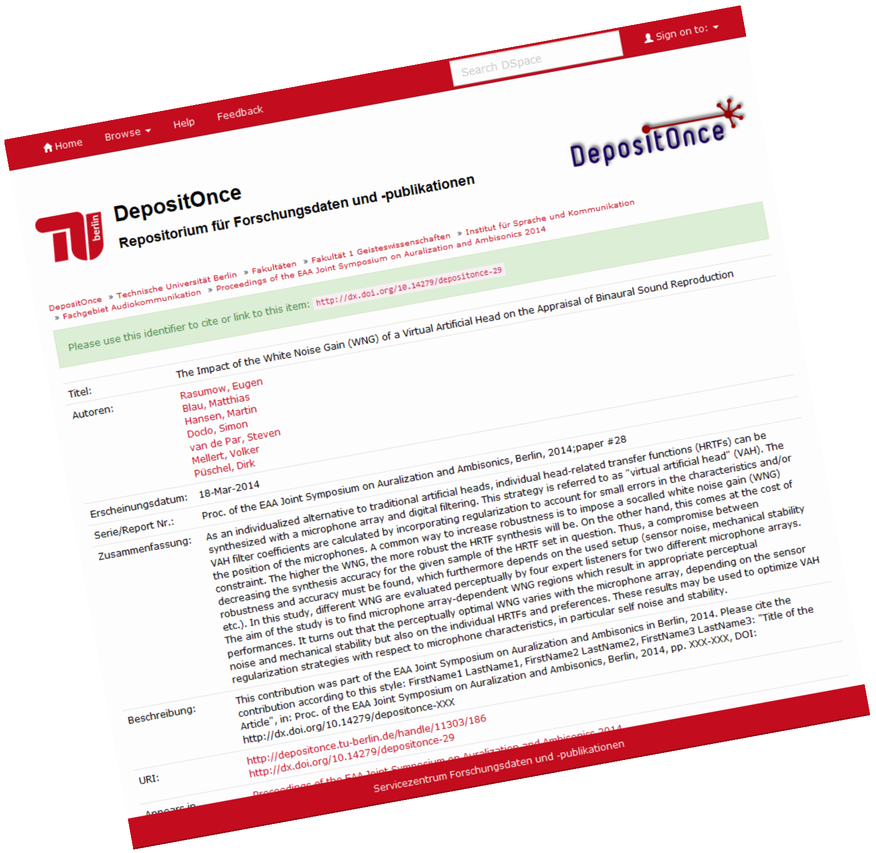
This creates what I call a Semantic Web – a web of data that can be processed directly or indirectly by machines.

Tim Berners-Lee with Marc Fischetti:
“Weaving the Web. The Past, Present and Future of the World Wide Web by its Initiator”
London 1999, ISBN: 0-7528-2090-7

Linked Data

- Linked Data is a term for conventions on how to publish data in the Semantic Web aka “Web of data”
- Linked Data can also be used as term for data published following those conventions
- The idea of automatic reasoning using Linked Data is part of the original Semantic Web idea but beyond the scope of this workshop

Repository contents is particularly suited



DepositOnce
Repositorium für Forschungsdaten und -publikationen

DepositOnce » Technische Universität Berlin » Fakultäten » Fakultät 1 Geisteswissenschaften » Institut für Sprache und Kommunikation » Fachgebiet Audiokommunikation » Proceedings of the EAA Joint Symposium on Auralization and Ambisonics 2014

Please use this identifier to cite or link to this item: http://dx.doi.org/10.14279/depositonce-29

Titel:	The Impact of the White Noise Gain (WNG) of a Virtual Artificial Head on the Appraisal of Binaural Sound Reproduction
Autoren:	Rasumow, Eugen Blau, Matthias Hansen, Martin Doclo, Simon van de Par, Steven Mellert, Volker Püschel, Dirk
Erscheinungsdatum:	18-Mar-2014
Serie/Report Nr.:	Proc. of the EAA Joint Symposium on Auralization and Ambisonics, Berlin, 2014;paper #28
Zusammenfassung:	As an individualized alternative to traditional artificial heads, individual head-related transfer functions (HRTFs) can be synthesized with a microphone array and digital filtering. This strategy is referred to as "virtual artificial head" (VAH). The VAH filter coefficients are calculated by incorporating regularization to account for small errors in the characteristics and/or the position of the microphones. A common way to increase robustness is to impose a socalled white noise gain (WNG) constraint. The higher the WNG, the more robust the HRTF synthesis will be. On the other hand, this comes at the cost of decreasing the synthesis accuracy for the given sample of the HRTF set in question. Thus, a compromise between robustness and accuracy must be found, which furthermore depends on the used setup (sensor noise, mechanical stability etc.). In this study, different WNG are evaluated perceptually by four expert listeners for two different microphone arrays. The aim of the study is to find microphone array-dependent WNG regions which result in appropriate perceptual performances. It turns out that the perceptually optimal WNG varies with the microphone array, depending on the sensor noise and mechanical stability but also on the individual HRTFs and preferences. These results may be used to optimize VAH regularization strategies with respect to microphone characteristics, in particular self noise and stability.
Beschreibung:	This contribution was part of the EAA Joint Symposium on Auralization and Ambisonics in Berlin, 2014. Please cite the contribution according to this style: FirstName1 LastName1, FirstName2 LastName2, FirstName3 LastName3: "Title of the Article", in: Proc. of the EAA Joint Symposium on Auralization and Ambisonics, Berlin, 2014, pp. XXX-XXX, DOI: http://dx.doi.org/10.14279/depositonce-XXX
URI:	http://depositonce.tu-berlin.de/handle/11303/186 http://dx.doi.org/10.14279/depositonce-29

Servicezentrum Forschungsdaten und -publikationen

The data stored in repositories are particularly suited to be published as Linked Data:

- Metadata already exist in a structured form
- They do not have to be generated or entered manually for publication as Linked Data
- Convert the data in RDF, add links and publish it respecting the Linked Data Principles
- Use Linked Data to provide repository contents as processible data

What about OAI-PMH?

- Open Archive Initiative – Protocol for Metadata Harvesting:
  de facto standard in the context of repositories
- Google retired support for OAI-PMH in 2008
- “Just” an interface, not a format
- The Library Way:
  Limited to the context of repositories

Why Linked Data?

- Linked Data is a generic, native way of data exchange
- Not limited to the field of repositories
- Data published following the Linked Data Principles is self-descriptive
- Linked Data simplifies data exchange with repositories for everyone outside of the repository environment
- SPARQL endpoints allow searches within the contents of foreign repositories (see later)

Simply spoken:
Linked Data for repositories can be seen as much wider supported OAI-PMH interface with better integration of foreign data and concepts (links, vocabularies and ontologies)*

*Limitations

- DSpace can harvest other repositories using OAI-PMH
- Linked Data support in DSpace currently is export oriented only
- OAI-PMH can harvest all documents changed within a specified time slot. To realize this in Linked Data we still have to agree on vocabularies and/or conventions

xxx.lanl.gov / arXiv.org

"Although the WorldWideWeb still represents only a small fraction of the overall usage, this access mode is expected to become dominant in the near future."

Paul Ginsparg, 1994

Paul Ginsparg: *First Steps Towards Electronic Research Communication*.
In: *Computer in Physics*, Vol. 8, No. 4, 1994, pp. 390-396.
Photo: Kindly provided by Paul Ginsparg

Linked Data –
A really short introduction

The Linked Data Principles

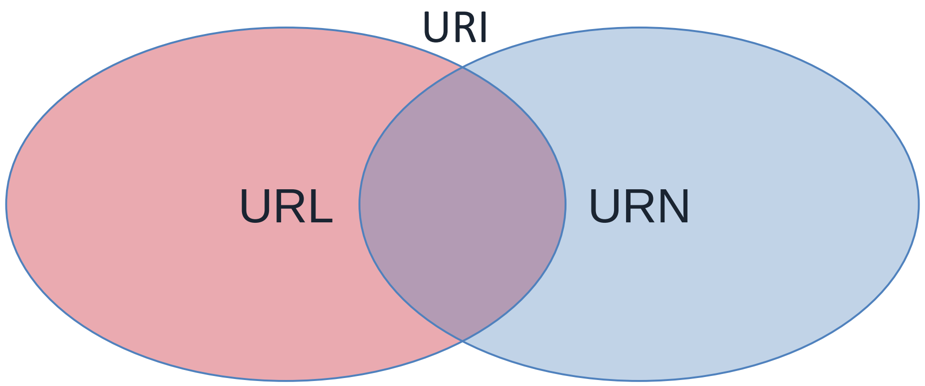
1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those things
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs, so they can discover more things

Tim Berners-Lee
http://www.w3.org/DesignIssues/LinkedData.html

What does that mean?

- Linked Data makes extensive use of URIs
- URIs are used as names and identifiers, not as locators only
- doi:10.14279/depositonce-5015 isn’t a HTTP URI and does not work in browsers (at least not without extensions)
- If an URI is requested, deliver data (use content negotiation)
- Use RDF as data model and one of it’s representations (RDF/XML, Turtle, …) as format
- Provide a SPARQL endpoint
- Link is the glue of the web, create links within your data!

URI = URL + URN



http://digital-repositories.org/ontologies/dspace#Repository
mailto:contact@the-library-code.de

- Uniformed Resource **Identifier** is an identifier of an abstract or physical resource
- An URI can be a Uniformed Resource **Name**, Uniformed Resource **Locator** or both
- An URN is a name, an URL is an address
- An URI is unique!

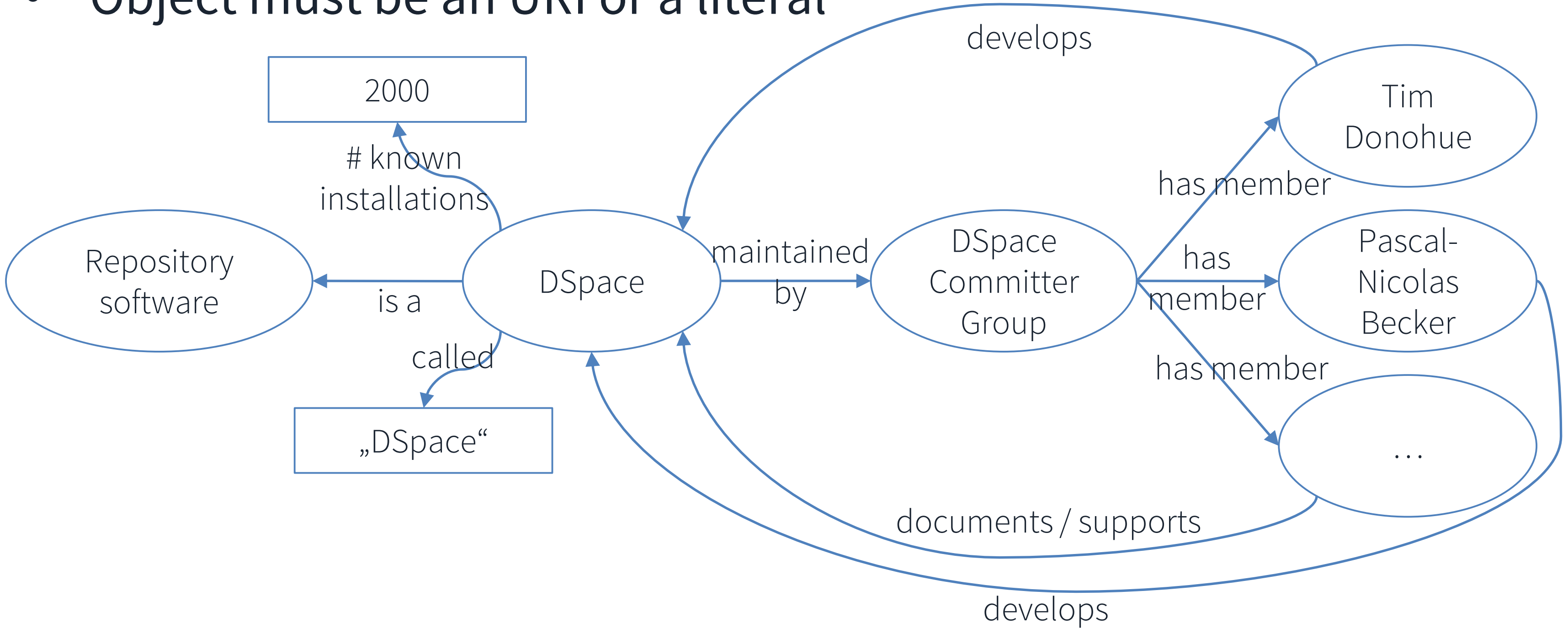
RFC 3986

Resource Description Framework

- Resource Description Framework (RDF) is a data model
- RDF makes extensive use of URIs and Links
- There are many formats (syntax) to express RDF: RDF/XML, Turtle, N-Triples, …
- Vocabularies and Ontologies express semantics of terms and relations between terms used in the Semantic Web
- The idea of automatic reasoning using Linked Data is part of the original Semantic Web idea but beyond the scope of this workshop

Ressource Description Framework

- Making statements about resources
- Structured in triples: **Subject – Predicate – Object**
- Subject and Predicate must be URIs
- Object must be an URI or a literal



Syntax – Semantics – Pragmatics

- Syntax: about form
  - Green, yellow, red
  - Bottom, middle, top
- Semantics:
  about meanings
  - Green = Go
  - Yellow = change
  - Red = stop
- Pragmatics: about use
  - If red and no traffic then allowed to turn right

„Traffic Light Tree" by Kevan
https://www.flickr.com/photos/kevandotorg/
Licensed under CC-BY 2.0

Syntax

- RDF is not a format, RDF is a data model
- There are several kinds of representations: graphical, RDF/XML, Turtle, N-Triples, N3, …
- Trutle is more user-friendly then RDF/XML
- Dspace‘s Linked Data support is configured using Turtle (more information to follow)

Turtle

- Simple: just write down Subject, Predicate and Object separated by spaces
- End Statements with an dot .
- Frame URIs by angle brackets <http://www.the-library-code.de>
- Put literals in “quotation marks“
- Prefixes at the document head declare short URIs within a document:

  ```
  @prefix dc: <http://purl.org/dc/elements/1.1/> .
  ```

  allows to use dc:title (without brackets!) within the document.
- Use semicolon instead of dot if next statement starts with the same subject as the previous one
- Few more rules exists (language tags, type declarations, …)
- Turtle quite readable and manually writeable

Example: DSpace Metadata RDF Mapping Vocabulary

```
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dc:    <http://purl.org/dc/elements/1.1/> .
@prefix :      <http://digital-repositories.org/ontologies/dspace-metadata-mapping/0.2.0#> .
...
<http://digital-repositories.org/ontologies/dspace-metadata-mapping/0.2.0>
    rdfs:comment   "Vocabulary for describing mappings of DSpace metadata to rdf." ;
    dc:contributor "Pascal-Nicolas Becker" ;
    dc:title       "DSpace Metadata RDF Mapping Spec" ;
    dc:description "Vocabulary for describing mappings of DSpace metadata to RDF. This
                    vocabulary is used to configure DSpace how to convert stored metadata
                    into RDF." ;
    dc:date        "2014-04-18".

:DSpaceMetadataRDFMapping
    a            rdfs:Class ;
    rdfs:label   "DSpace Metadata RDF Mapping" ;
    rdfs:comment "Represents the mapping of a DSpace metadata value to an RDF equivalent.".

:Result
    a               rdfs:Class ;
    rdfs:subClassOf rdf:Statement ;
    rdfs:label      "DSpace Metadata RDF Mapping Result" ;
    rdfs:comment    "A reified statement that describes the result of the
                     DSpaceMetadataRDFMapping.".
...
```

Semantics: Vocabularies and Ontologies

- To make statements in RDF we need URIs
- URIs should have a meaning (semantic)
- Vocabularies are used to associate semantics and URIs

- Logical statements can be used for reasoning
- Further knowledge additionally to vocabularies may be necessary
- Ontologies can be used to describe domain and background knowledge

- Ontologies and vocabularies are published as Linked Data and following the Linked Data Principles as RDF
- RDF-Schema (RDFS) and the Web Ontology Language (OWL) are used to describe vocabularies and ontologies

Example: DSpace Repository Ontology

```
@prefix : <http://digital-repositories.org/ontologies/dspace/#> .
...

<http://digital-repositories.org/ontologies/dspace/> rdf:type owl:Ontology ;
    rdfs:label "DSpace Repository Ontology"@en ;
    dc:creator "Pascal-Nicolas Becker" ;
    dc:date "2014-09-11" ;
    dc:description "DSpace Repository Ontology"@en ;
    dc:description "Ontology to describe a repository using DSpace. You can find further
                    information about dspace at http://www.dspace.org."@en .
...

:hasBitstream rdf:type owl:ObjectProperty ;
    rdfs:comment "Links from an item to a bitstream of the item. Bitstreams can be a
        representation of an item or a part of a representation composed of several bitstreams.
        Bitstreams are arbitrary files, e.g. documents, archives, images, ..."@en ;
    rdfs:domain :Item ;
    rdfs:subPropertyOf :hasPart .
...

###  http://digital-repositories.org/ontologies/dspace/#hasPart
:hasPart rdf:type owl:ObjectProperty ,
                  owl:TransitiveProperty ;
         rdfs:comment "Links top down between the structure of a DSpace repository."@en ;
         owl:inverseOf :isPartOf .
...
```

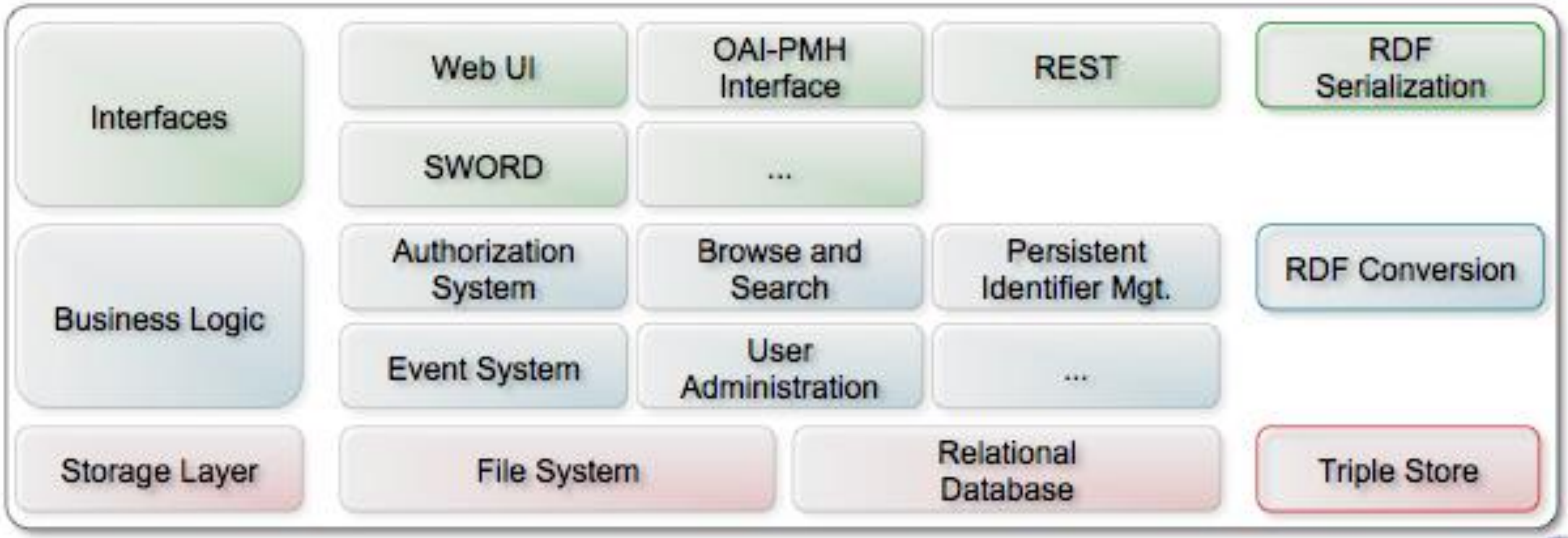
Summary

- RDF is a data model
- RDF use triples: **Subject – Predicate – Object**
- Subject and Predicate must be URIs
- Object must be an URI or a literal
- Turtle is simple to read and write
- Vocabularies and ontologies are used to define terms, to associate semantic and URIs and to describe background and domain knowledge
- A computer doesn’t “understand” the semantic of anything
- A computer can use logic to deduce
- Inference and deduction seams smart

DSpace‘s Linked Data support

Extend DSpace

- Install a Triple Store and switch Link Data support on
- The Triple Store is used as cache for the converted data and to provide a SPARQL endpoint
- Added methods to convert data into RDF and to add links
- Added a module to serve data as RDF serializations
- Added content negotiation



Characteristics of repositories

- Repository contents change rarely (to be citable and reliable)
  Conversion may be time intensive
  - ➢ Convert data and store converted data in a cache
- Persistent Identifiers (handle, DOI, …) may violate the Linked Data Principles
  - ➢ Use Persistent Identifiers in form of HTTP(S) URIs (http://dx.doi.org/...)
- Repositories generate URIs that shall be used to address their content
  - ➢ Reuse those URIs
  - ➢ Add content negotiation to them
  - ➢ Use Placeholders in configuration for those URIs
- Metadata may use already existing vocabularies (e.g. Dublin Core, LCSH, …)
  - ➢ Convert metadata values to URIs / links
  - ➢ Allow to use regular expressions to modify metadata values
- Different repositories may use different metadata schemas
  - ➢ Conversion must be highly configurable and extendable
  - ➢ Use RDF for the configuration so all of its features can be used easily

Linked Data within DSpace

- The Triple Store is just a cache
- You can recreate all the data in the triple store whenever necessary
- As the content of the Triple Store is public readable (over SPARQL) only archived and discoverable Items are converted
- `[dspace]/bin/dspace rdfizer --help`
  gives you an overview over the commands to maintain the triple store
- RDFizer can be used to delete all contents from the triple store, to re-convert all contents and to update contents
- Use RDFizer to initially convert all existing data when you switch the Linked Data support on
- Add "rdf" to the event.dispatcher.default.consumers (dspace.cfg) and DSpace converts data automatically whenever new DSpaceObjects are created or a DSpaceObject is changed (no need to run rdfizer periodically)

What do repositories store?

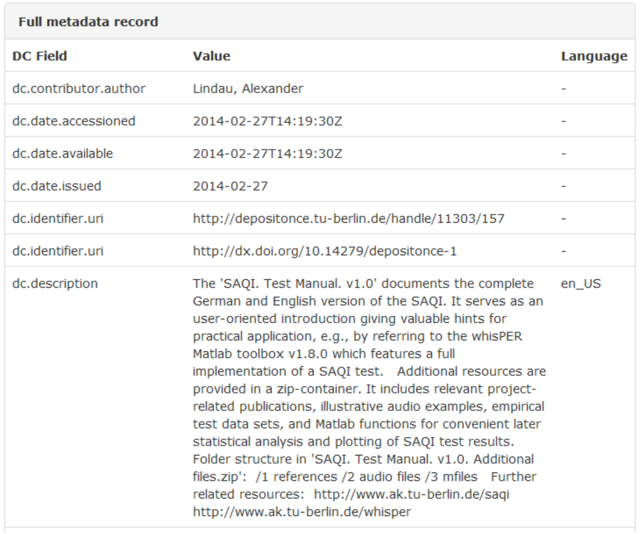
“Repositories are systems to safely store and publish digital objects and their descriptive metadata.”

Digital objects

One or several files:
Documents (PDF, Text, …), Tables (CSV, …), Images (PNG, Tiff …), Audio (Wave, …), Video, File Archives, …

Descriptive metadata

Structured metadata as key – value:
dc.title, dc.contributor.author, dc.description, dc.date.available, dc.subject.lcsh, dc.subject.ddc, …



Full metadata record

DC Field	Value	Language
dc.contributor.author	Lindau, Alexander	-
dc.date.accessioned	2014-02-27T14:19:30Z	-
dc.date.available	2014-02-27T14:19:30Z	-
dc.date.issued	2014-02-27	-
dc.identifier.uri	http://depositonce.tu-berlin.de/handle/11303/157	-
dc.identifier.uri	http://dx.doi.org/10.14279/depositonce-1	-
dc.description	The 'SAQI. Test Manual. v1.0' documents the complete German and English version of the SAQI. It serves as an user-oriented introduction giving valuable hints for practical application, e.g., by referring to the whisPER Matlab toolbox v1.8.0 which features a full implementation of a SAQI test. Additional resources are provided in a zip-container. It includes relevant project-related publications, illustrative audio examples, empirical test data sets, and Matlab functions for convenient later statistical analysis and plotting of SAQI test results. Folder structure in 'SAQI. Test Manual. v1.0. Additional files.zip': /1 references /2 audio files /3 mfiles Further related resources: http://www.ak.tu-berlin.de/saqi http://www.ak.tu-berlin.de/whisper	en_US

- We can‘t convert the files (technical problems, too much work, too diverse)
- But we can convert the metadata and link the files!

Configuring the conversion

- Conversion is highly configurable and easy extendible
- [dspace]/config/modules/rdf.cfg is the main configuration file
  - Triple Store connection details
  - Which URIs to use (DOI, Handle, local address)[1]
  - List of plugins used for the conversion[1]
  - Configuration of the plugins
  - Where to find further configuration files
- [dspace]/config/modules/rdf/metadata-rdf-mapping.ttl contains rules to convert the metadata
- [dspace]/config/modules/rdf/*-prefixes.ttl contains prefixes to use
- [dspace]/config/modules/rdf/fuseki-assembler.ttl contains a configuration for Apache Jena Fuseki 1.* Triple Store

[1] Moved in DSpace 6 to config/spring/api/rdf.xml

Connection details

- DSpace uses the SPARQL 1.1 Query Language and the SPARQL 1.1 Graph Store HTTP Protocol to query and manipulate the Triple Store
- config/modules/rdf.cfg contains the connection details for those endpoints and authentication credentials if necessary
- The address of the SPARQL endpoint as it should be provided to the public must be set in rdf.cfg too and can additionally be configured as part of the conversion (e.g. using void:sparqlEndpoint)

Generating URIs

- Different repositories use different URIs (DOIs, Handles, local addresses)
- All those should be used as HTTP URIs, following the Linked Data Principles
- The URIs used by the repository for its content Items should be reused
- An URIGenerator gets a DSpaceObject and returns the appropriate URI to use
- Currently DSpace provides out of the box:
  - LocalURIGenerator: Uses “local”URIs of the repository
    e.g. https://depsotionce.tu-berlin.de/handle/11303/5330
  - HandleURIGenerator: Uses Handles (may Fallback to local URIs)
    e.g. http://hdl.handle.net/11303/5330
  - DOIHanldeURIGenerator / DOIURIGenerator: Uses DOIs (may Fallback to URIs or Handles)
    e.g. http://dx.doi.org/10.14279/depositonce-5015
- Of course you may extend this with your own URIGenerator

Reuse, avoid URI generation

- Don‘t generate URIs if adequate ones exists
  - Example: For classifications like the Library of Congress Subject Headings (LCSH) or the Dewey Decimal Classification (DDC) appropriate URIs exists
- Reuse of URIs enables other to recognize the resources you used/linked
- Create new URIs only if all necessary information are available
  - Is the name of an author enough to create proper URIs?
  - How would you distinguish authors with the same name?
  - How would you recognize two publications from the same author?
  - Shouldn‘t authors be able to create their own URIs providing their own content?
- But: create URIs if you have the authority to do so
  - Create URIs for Objects of your repository, who should do it instead?
- Create links whenever possible

Converter Plugins

- DSpace currently contains the following plugins:
  - StaticDSOConverterPlugin to include static Linked Data
  - SimpleDSOReleationsConverterPlugin to create links between DSpaceObjects
  - MetadataConverterPlugin to convert Metadata of items
- RDF conversion is extendable by creating new plugins
- In DSpace 5 plugins used during the conversion can be configured in config/modules/rdf.cfg
- In DSpace 6 this moved in DSpace 6 to config/spring/api/rdf.xml

StaticDSOConveterPlugin

- [dspace]/config/modules/rdf/constant-data-*.ttl contains Linked Data that should be included in every DSpaceObject or every Item/Collection/Community or the information about the whole repository (Site)
- You can also use other RDF serializations than Turtle by changing the filename and its location in rdf.cfg

Example:

```
@prefix void: <http://rdfs.org/ns/void#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<> foaf:homepage <http://demo.dspace.org/>;
   void:sparqlEndpoint
         <http://dspace.pnjb.de/fuseki/demo.dspace.org/sparql> .
```

Relations between DSpaceObjects

- The SimpleDSORelationsConverterPlugins create links between different DSpaceObjects
- It links from the site object that represents the whole repository to the top level communities
- It links from communities to sub-communities
- It links from communities to collections
- It links from collections to items
- It links from items to **bitstreams**
- It create the same links in the opposite direction
- Every link is constructed with a DSpaceObject as subject, another DSpaceObject as object and a configurable predicate (see config/modules/rdf.cfg)
- The links are necessary to crawl the repository using linked data

MetadataConverterPlugin

- Converts metadata into RDF
- Currently supports items only (should be easy to extend now as DSpace supports metadata for all DSpaceObjects since version 5)
- Uses rules to convert metadata
- Default rules provided for the fields provided by DSpace out of the box
- Rules are written in RDF to be able to use all of RDF's feature
- The DSpace Metadata RDF Mapping Vocabulary was created to write these mappings

DSpace Metadata RDF Mapping

http://digital-repositories.org/ontologies/dspace-metadata-mapping/

- One Mapping describes how to convert one metadata field into RDF
- Can detect the metadata field by its name (key) and a regular expression used on its value
- Creates one or several triples
- Can use a placeholder for the URI of the object being converted currently
- Can create Literals or Resources as needed
- Can specify value types and language tags
- Can use the language tag DSpace stores for some metadata fields
- Can reuse the metadata value, of course
- May use regular expressions to modify metadata values used as Literals or Resource URIs

Example Mapping: dc.title

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dm: <http://digital-repositories.org/ontologies/
                                      dspace-metadata-mapping/0.2.0#> .
@prefix : <#> .

:title
  dm:metadataName "dc.title" ;
  dm:creates [
    dm:subject dm:DSpaceObjectIRI ;
    dm:predicate dcterms:title ;
    dm:object dm:DSpaceValue ;
  ] ;
.
```

Code	Explanation
`:title`	Internal „name“ of this rule
`dm:metadataName "dc.title" ;`	Converts „dc.title“ fields
`dm:creates [`	Initiate „Blank node“ describing RDF to create
`dm:subject dm:DSpaceObjectIRI ;`	Use DSpaceObject‘s URI as subject
`dm:predicate dcterms:title ;`	Use „dcterms:title“ as predicate
`dm:object dm:DSpaceValue ;`	Use the unchanged metadata value as (literal) object

Example Mapping: DOIs

```
:doi
    dm:metadataName „dc.identifier.doi" ;
    dm:condition „^doi:“ ;
    dm:creates [
        dm:subject dm:DSpaceObjectIRI ;
        dm:predicate dc:identifier;
        dm:object [
            a dm:ResourceGenerator ;
            dm:modifier [
                dm:matcher „^doi:(.*)$“ ;
                dm:replacement „http://dx.doi.org/$1“ ;
            ] ;
            dm:pattern „$DSpaceValue“ ;
        ] ;
    ] ;
.
```

Use this rule only if value starts with doi:

Create an URI as object

Regular Expression to match the part after doi:

Replace value with http://dx…/ and the matched part

Do this operation on the metadata value

dspace-rdf

- dspace-rdf is a webapplication
- Used to provide RDF serializations
- If a specific resource is requested, dspace-rdf loads the data from the Triple Store and converts it in the requested serialization
- Maybe the target of the Content Negotiation
- Deploy dspace-rdf as /data or /rdf when you enable DSpace's Linked Data support
- Add its location to configuration property content.path (rdf.cfg)
- http://demo.dspace.org/data/handle/10673/3/ttl?text,
  http://demo.dspace.org/data/handle/10673/3/nt?text,
  http://demo.dspace.org/data/handle/10673/3/ttl?text
  (The parameter text sets the mime-type so that the browser shows the file instead of downloading it)

Hands on!

Query demo.dspace.org

- Create your own item on demo.dspace.org
- Open the item and exchange the UI with „data“ (http://demo.dspace.org/xxxui/handle/10673/3 -> http://demo.dspace.org/data/handle/10673/3)
- Change the item using a UI and reload the data view
- Query dspace-rdf for another RDF serialization
- Use content negotiation: wget -O - --header=‘Accept: text/turtle’ http://demo.dspace.org/jspui/handle/10673/5

Install a Triple Store

- Install Apache Jena Fuseki Triple Store and start it:http://archive.apache.org/dist/jena/binaries/jena-fuseki1-1.1.2-distribution.zip
- If you use version 1.x, the configuration file provided by DSpace should work out of the box
- Fuseki >= 1.3.0 and 2.3.0 requires Java 8
- Enable RDF (rdf.cfg and as event consumer in dspace.cfg), enable content negotiation
- Deploy dspace-rdf
- Create Items and look what this does in your Triple Store

Change the URIGeneration between Handle and local URIs

- Create some contents and open the RDF representation
- Change the configuration to use the HandleURIGenerator
- Delete all data from the Triple Store and reconvert them
- Re-open the RDF representation in another tab and compare them

Describe your repository and institution

- Use the StaticDSOConverterPlugin to add constant data describing your institution and repository
- FOAF is a very well-known vocabulary to describe persons, that can be used to describe organizations as well: http://www.foaf-project.org
- Linked Datasets can be described by void: http://www.w3.org/TR/void/
- Look out for other vocabularies/ontologies that might be useful

Convert a locally added metadata field

- Add a local metadata schema
- Add a field
- Add a mapping and convert this field
- Look out for the converted field in the data output

Querying demo.dspace.org

SPARQL

- SPARQL Protocol And RDF Query Language is the Query language for RDF
- Comparable to SQL for databases
- A SPARQL interface is called SPARQL endpoint
- Since SPARQL 1.1 SPARQL can be used to manipulate data

Example:

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

http://dspace.pnjb.de/fuseki/demo.dspace.org/sparql?query=SELECT+*+WHERE+%7B+%3Fs+%3Fp+%3Fo+.+%7D&output=text

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?s WHERE { ?s dc:contributor "Becker, Pascal-
  Nicolas" . }
```

Named Graphs

- A Triple Store is a database for RDF
- As RDF can be serialized into graphs
- Following Triple Store can be seen as Graph databases
- It is possible to store several Graphs in one Triple Store
- DSpace creates one Graph per DSpaceObject
- dspace-rdf fetches all data of the appropriate graph if it is requested for rdf describing a specific DSpace Object
- rdfizer --delete-all queries all graphs and deletes one by one
- The default graph may contain all data of all named graphs if the Triple Store is configured accordingly

Querying Named Graphs

Request all data from all graphs (Limited to 100 triples):

`SELECT * WHERE { GRAPH ?g { ?s ?p ?o } } LIMIT 100`

`http://dspace.pnjb.de/fuseki/demo.dspace.org/sparql?query=SELECT+*+WHERE+%7B+GRAPH+%3Fg+%7B+%3Fs+%3Fp+%3Fo+%7D+%7D+LIMIT+100%0D%0A&output=text`

Request all Triples from one specific Graph:

`SELECT ?s ?p ?o WHERE { GRAPH` <http://demo.dspace.org/data/resource/10673/5> { ?s ?p ?o } }

http://dspace.pnjb.de/fuseki/demo.dspace.org/sparql?query=SELECT+%3Fs+%3Fp+%3Fo+WHERE+%7B+GRAPH+%3Chttp%3A%2F%2Fdemo.dspace.org%2Fdata%2Fresource%2F10673%2F5%3E+%7B+%3Fs+%3Fp+%3Fo+%7D+%7D%0D%0A&output=text

Search remote repositories

- SPARQL is quite powerful
- FILTER allows you to use regex while querying the Triple Store using SPARQL: https://jena.apache.org/tutorials/sparql_filters.html
- See SPARQL by example for further examples: http://www.cambridgesemantics.com/semantic-university/sparql-by-example
- If an repository offers a public readable SPARQL endpoint, you can use it like a powerful search interface

Thank you.

The Library Code

DSpace Service Provider