

International Conference on Dublin Core and Metadata Applications

Architecture Forum

Open session

Mikael Nilsson

<mikael@nilsson.name>



☼ Major goals of the Architecture Forum

- Document the DCMI metadata framework
- Develop technical specifications
- Provide feedback on technical issues
 - To DCMI users
 - To other DCMI communities
 - To the Usage Board
- Serve as a platform for discussions and evaluations of future technical directions



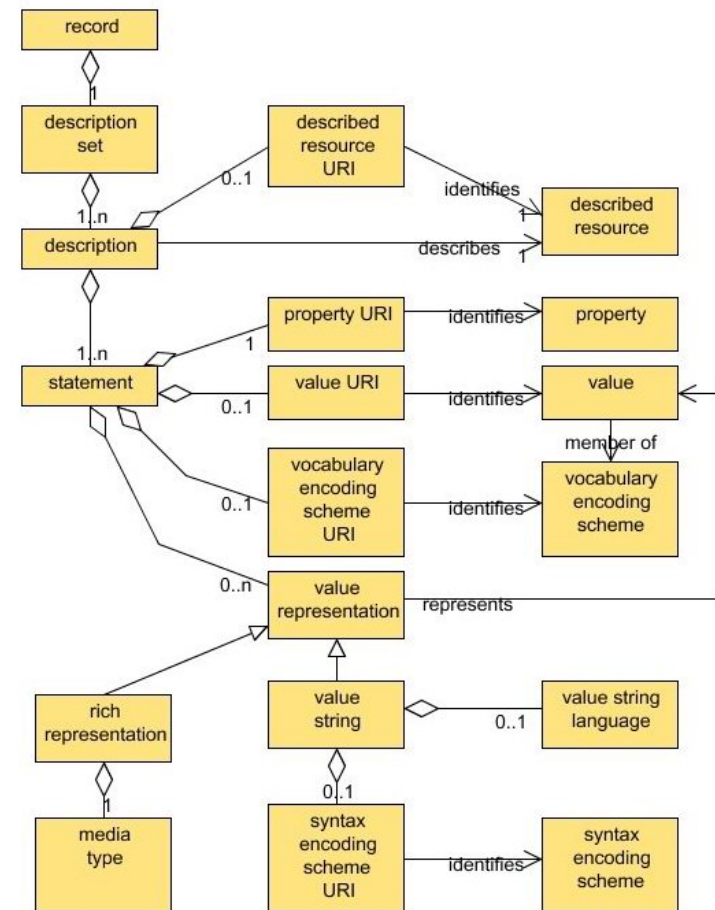
☼ Major specifications from DC-ARCH

- DCMI Abstract Model (2005, 2007)
 - Underlying model for DCMI metadata
 - Defines notions of
 - Property
 - Vocabulary Encoding Scheme
 - Syntax Encoding Scheme
 - Etc
 - ... and how they fit together
 - Used as basis for metadata expressions
 - Used as basis for how DCMI terms are defined



Understanding the model

- Literal and non-literal values
- “Description sets” vs “records”
- “Values” vs “value strings” and URIs
- “Domain model” of described entities



Who needs to understand what?

- Application architects (who implement)
 - To follow model, need to know all of the above!
 - Understand how formats express the model
- Non-expert users (who define requirements)
 - Examples of common design patterns
 - Simple question trees for narrowing choices
 - Palette of pre-defined well-engineered choices



Guidelines for profile designers

- Guidelines for Dublin Core Application Profiles
- Palette of profile snippets to cut-and-paste?
 - Wiki format convertible into XML
 - Library of usage templates?



☼ Major specifications from DC-ARCH

- DCMI Metadata expressions
- DC-RDF (2008)
 - Replaces DCQ-RDF-XML (2002) and DCMES-XML (2002)
- DC-HTML (2008)
 - Replaces DCQ-HTML (2003)
- DC-DS-XML (200?)
 - Might replace DC-XML (2003)
- DC-TEXT (2007)
 - A simple human-readable text format (not for machine processing)



☼ Major specifications from DC-ARCH

- Singapore Framework for DC Application Profiles (2008)
- Description set profiles (200?) - working draft
- Interoperability levels for Dublin Core metadata



☼ The Singapore Framework

- In Singapore, at DC2007, a new definition of a “Dublin Core Application Profile” was introduced
- A “DC Application Profile” is packet of documentation which consists of:
 - Functional requirements (mandatory)
 - Domain model (mandatory)
 - Description Set Profile (DSP) (mandatory)
 - Usage guidelines (optional)
 - Encoding syntax guidelines (optional)

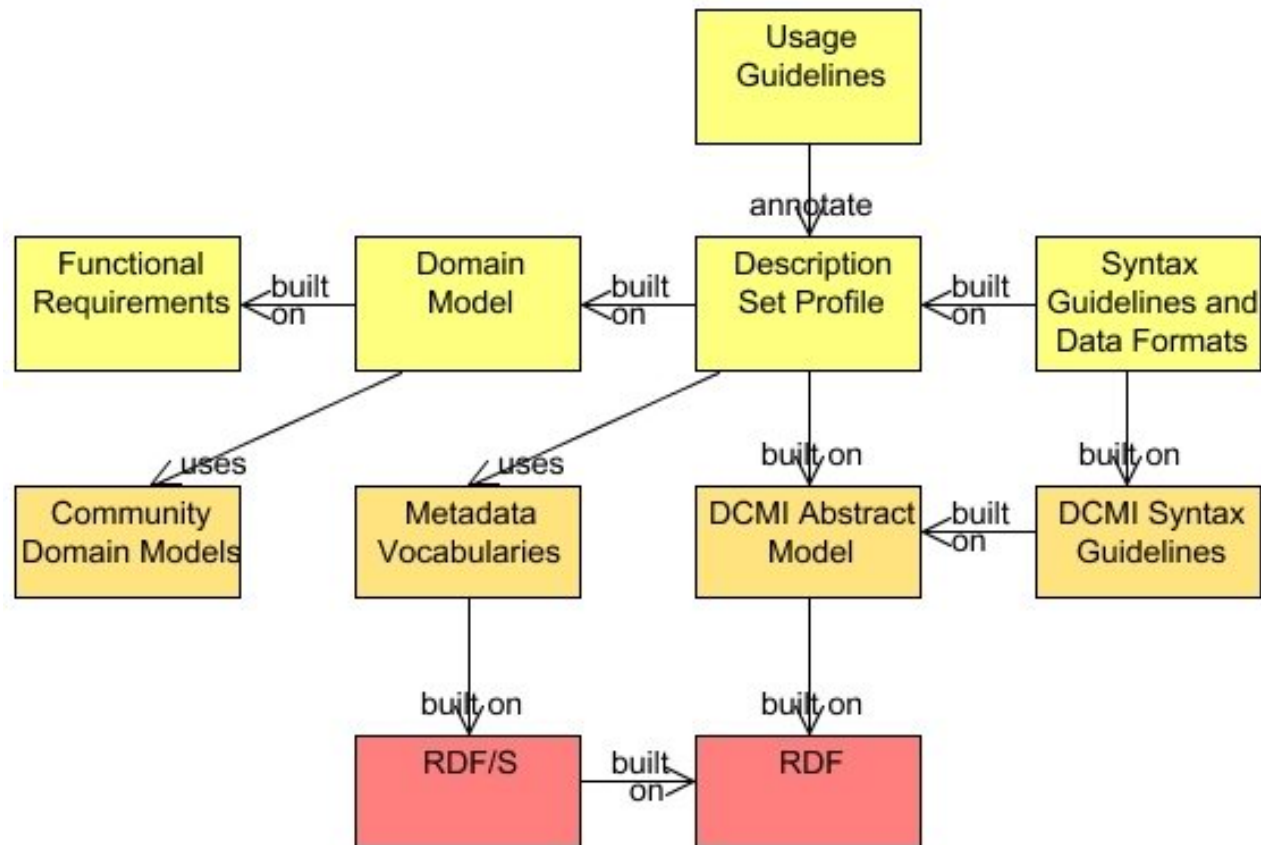


☼ Profiles and standards

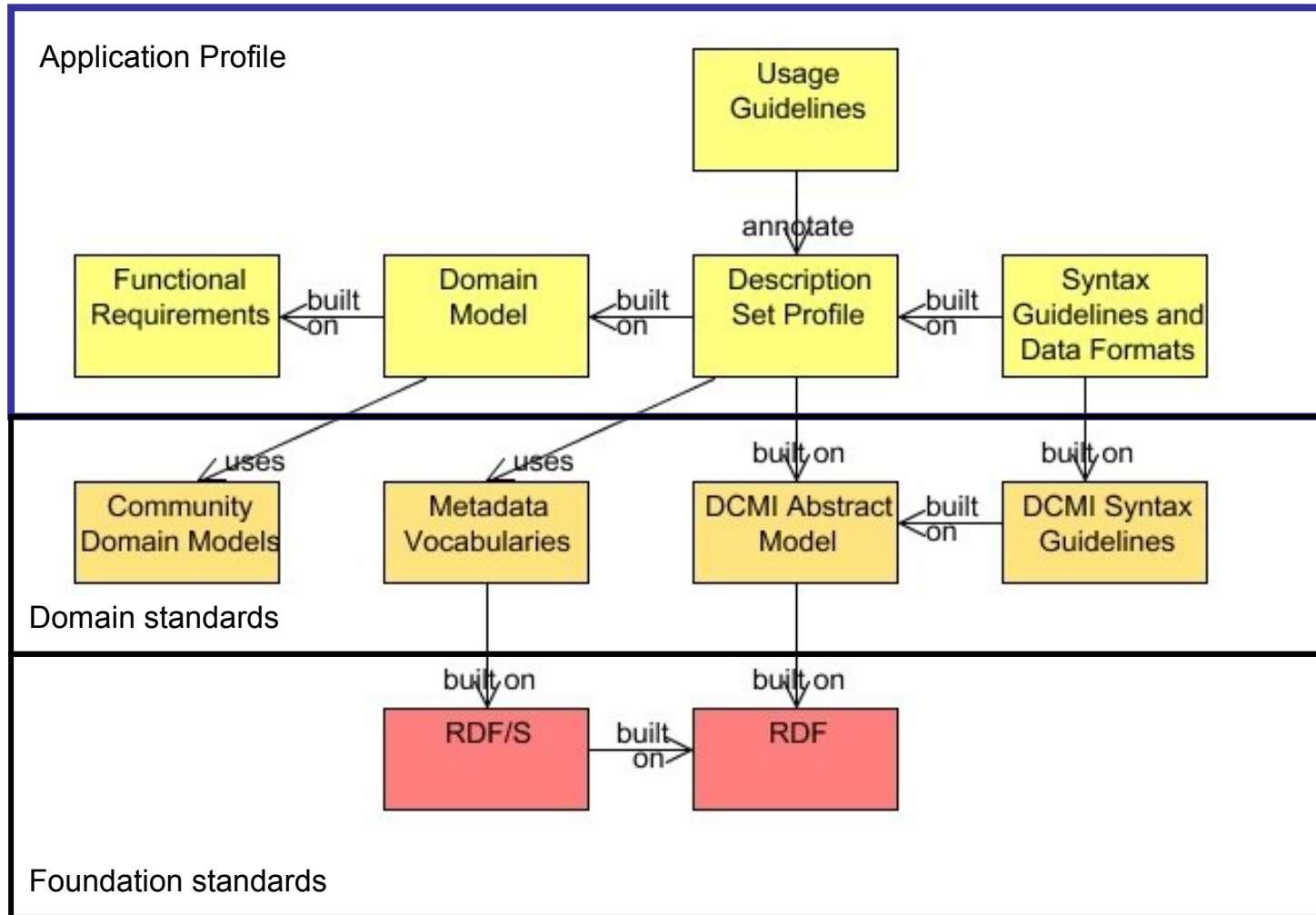
- Profiles are based on domain standards:
 - Standard metadata vocabularies (e.g., Dublin Core elements)
 - Standard domain models (e.g., FRBR)
 - Functional Requirements for Bibliographic Records
 - A standard record model (DCMI Abstract Model)
- Foundation is Resource Description Framework (“Semantic Web”)
 - RDF is the model underlying the DCMI Abstract Model
 - RDF Schema is the model underlying the machine processable definitions of terms



☼ The Singapore Framework



☀ The Singapore Framework



Interoperability Levels for Dublin Core metadata

- Level 1: Informal interoperability
 - Shared concepts with natural-language definitions
 - No use of formal models or term URIs
 - Test: Is there a mapping to shared elements?
 - Example: IEEE LOM reuses some definitions and maps to 15-element “Dublin Core” (ISO 15836)



Interoperability Levels for Dublin Core metadata

- Level 2: Semantic Interoperability
 - Correct use of formal RDF graph model with conformant vocabularies (eg DCMI Metadata terms)
 - Use of URIs and formal semantic relationships between terms (eg subproperties)
 - Test: Is there a mapping to RDF triples?
 - Examples:
 - All RDF data (by definition)
 - All RDF data extracted from non-RDF formats (eg via GRDDL transforms)
 - All XHTML or HTML data using RDFa or DC-HTML/2008.



Interoperability Levels for Dublin Core metadata

- Level 3: Description set syntactic interoperability
 - Level-2-compatible data packaged in bounded description sets (records) as per DCMI Abstract Model (DC-AM)
 - Conventions for citing vocabulary encoding schemes (controlled vocabularies)
 - Test: Is there a mapping to “Expressing Dublin Core metadata using the DC-Text format”?
 - Examples: All data using DC-AM-compatible specifications, such as DC-DS-XML.



Interoperability Levels for Dublin Core metadata

- Level 4: Description Set Profile Interoperability
 - Level-3-compatible data that follows the specification “Description Set Profiles: A constraint language for Dublin Core Application Profiles”
 - Additional interoperability via shared Functional Requirements and Domain Model (“Singapore Framework for Dublin Core Application Profiles”)
 - Test: Is there a mapping to DSP constraints?
 - Examples:
 - Scholarly Works Application Profile



Interoperability Levels for Dublin Core metadata

- Level 4: Description Set Profile Interoperability
- Level 3: Description Set syntactic interoperability
- Level 2: Semantic interoperability
- Level 1: Informal interoperability





Description Set Profiles

Mikael Nilsson <mikael@nilsson.name>

DC 2007, Singapore
Aug 27-31, 2007



DC Application Profiles

- traditional definition

“A Dublin Core Application Profile (DCAP) is a declaration specifying **which metadata terms** an organization, information provider, or user community uses in its metadata. By definition, a DCAP identifies the source of metadata terms used—whether they have been defined in formally maintained standards such as Dublin Core, in less formally defined element sets and vocabularies, or by the creator of the DCAP itself for local use in an application. Optionally, a DCAP may provide additional documentation on **how the terms are constrained, encoded or interpreted** for application-specific purposes.” --CEN CWA 14855:2003



Machine-readable constraints?

- XML Schema?
 - Not applicable to RDF
 - Not applicable to HTML
 - Not applicable to
- CEN Guidelines
 - Not based on DCAM
 - No support for Description Sets
- Needed something new
 - “Dublin Core Description Set Profile”



Envisioned usages

- as a formal representation of the constraints of a Dublin Core Application Profile
- as configuration for databases
- as configuration for metadata editing tools
- etc.



Scope of a DSP spec

- Information model: Structural constraints on a description set:
 - what descriptions may occur
 - what properties may be used
 - what ways might a value surrogate be given
- XML expression



Out of scope

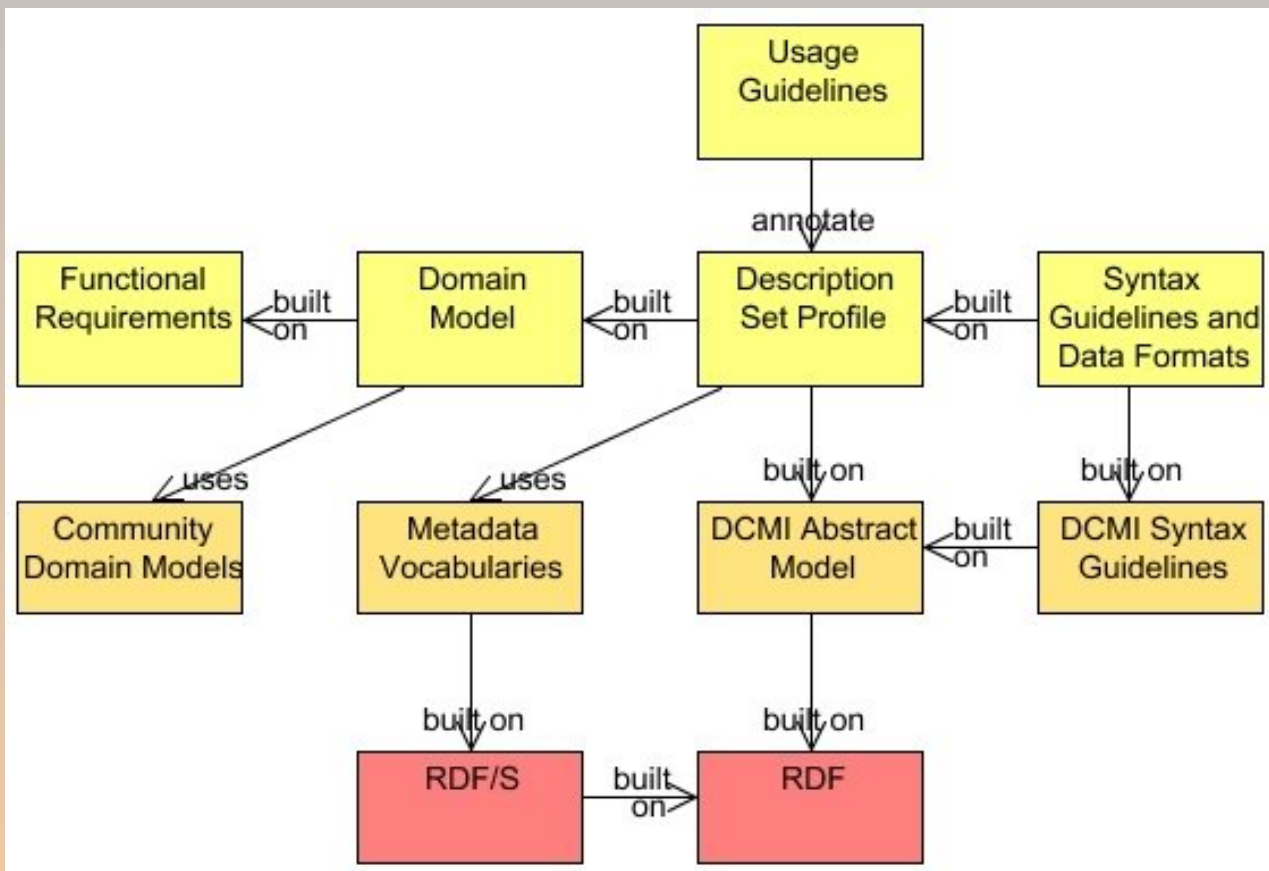
- Human-readable documentation
- Definition of vocabularies
- Version control
- etc.

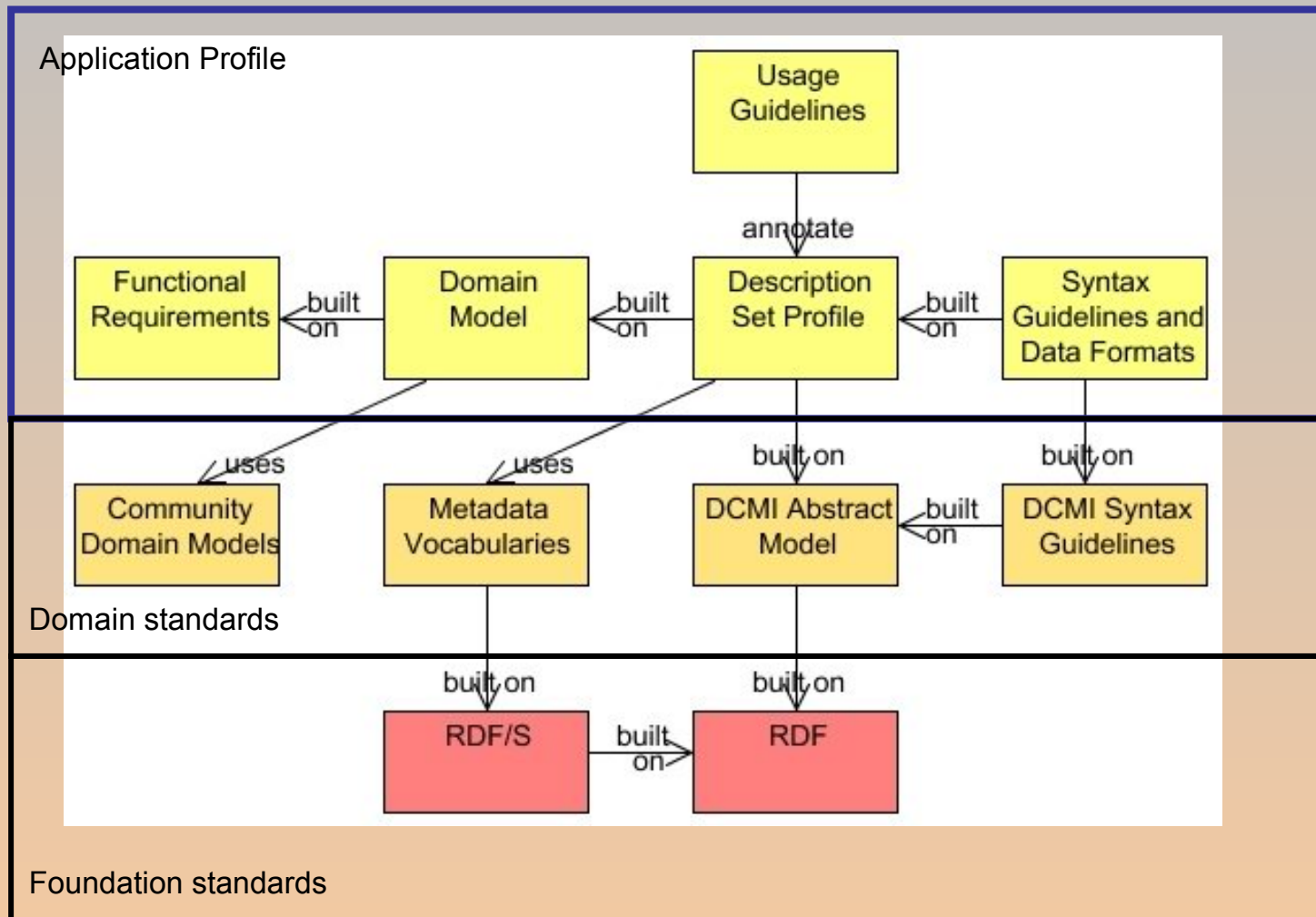


DC Application Profiles

- new definition (in progress)

- A DCAM-conformant Application Profile (“DC Application Profile”) is packet of documentation which consists of:
 - Functional requirements (mandatory)
 - Domain model (mandatory)
 - Description Set Profile (DSP) (mandatory)
 - Usage guidelines (optional)
 - Encoding syntax guidelines (optional)







First working draft

- <http://dublincore.org/architecturewiki/DescriptionSetProfile>
- Comments on DC-ARCHITECTURE
- Publication schedule not defined



Example: The book AP

- A book:
 - a literal title
 - a creator, described separately
- A creator
 - a literal name



Description Template: Book

Statement template: literal title

Property: `dcterms:title`

Literal value

Language

SES

Statement template: creator

Property: `dcterms:creator`

Description reference: Creator

Value URI

Vocabulary Encoding Scheme

Value string

Language

SES

Description Template: Creator

Statement template: literal name

Property: `foaf:name`

`standalone:no`

Literal value

Language

SES



<http://knowware.nada.kth.se/DCWiki/TheBookAP?action=raw>

<http://knowware.nada.kth.se/DCWiki/TheBookAP>

<http://knowware.nada.kth.se/DCWiki/TheBookAP?action=DSP2XML>

{{{#!DSP

== Description template: Book ==

DT=(min="1" max="1" standalone="yes" identifier="book")

=== Title ===

[Link](#)

ST=(max="1" type="literal" PC={http://purl.org/dc/terms/title})

|| Definition || A name given to the resource. ||

LC=(LangC=(occurrence="optional") SESConstraint=(occurrence="disallowed"))

=== Creator ===

ST=(max="1" type="nonliteral" PC={http://purl.org/dc/terms/creator})

|| Definition || An entity primarily responsible for making the resource. ||

NLC=(VURIConstraint=(occurrence="disallowed") VESConstraint=(occurrence="disallowed")

VStringConstraint=(max="1" LangC=(occurrence="disallowed")

SESConstraint=(occurrence="disallowed")) description="creator")

== Description template: Creator ==

DT=(min="1" max="1" standalone="no" identifier="creator")

=== Name ===

ST=(max="1" type="literal" PC={http://xmlns.com/foaf/0.1/name})

|| Definition || A name for some thing. ||

LC=(LangC=(occurrence="disallowed") SESConstraint=(occurrence="disallowed"))

}}}



<http://knowware.nada.kth.se/DCWiki/TheBookAP?action=raw>

<http://knowware.nada.kth.se/DCWiki/TheBookAP>

<http://knowware.nada.kth.se/DCWiki/TheBookAP?action=DSP2XML>

Description template: Book

Title

Property	http://purl.org/dc/terms/title
Max occurrence	1
Literal?	Yes
Definition	A name given to the resource.
Value (Literal)	Syntax Encoding Scheme: Occurrence disallowed Language Constraint: Occurrence optional

Creator

Property	http://purl.org/dc/terms/creator
Max occurrence	1
Literal?	No
Definition	An entity primarily responsible for making the resource.
Value (Non-Literal)	Description: creator Value URI Constraint: Occurrence disallowed Vocabulary Encoding Scheme Constraint Occurrence: disallowed Value String Constraint: Max occurrence 1 <i>Syntax Encoding Syntax Constraint:</i> Occurrence: disallowed <i>Language Constraint:</i> Occurrence disallowed

Description template: Creator

Name

Property	http://xmlns.com/foaf/0.1/name
Max occurrence	1
Literal?	Yes
Definition	A name for some thing.
Value (Literal)	Syntax Encoding Scheme: Occurrence disallowed Language Constraint: Occurrence disallowed



<http://knowware.nada.kth.se/DCWiki/TheBookAP?action=raw>

<http://knowware.nada.kth.se/DCWiki/TheBookAP>

<http://knowware.nada.kth.se/DCWiki/TheBookAP?action=DSP2XML>

```
<?xml version="1.0"?>
```

```
<DescriptionSetTemplate>
```

```
  <DescriptionTemplate maxOccur="1" minOccur="1">
```

```
    <StatementTemplate maxOccur="1" type="literal">
```

```
      <Property>http://purl.org/dc/terms/title</Property>
```

```
      <LiteralConstraint>
```

```
        <SyntaxEncodingSchemeOccurrence>disallowed</SyntaxEncodingSchemeOccurrence>
```

```
        <LanguageOccurrence>optional</LanguageOccurrence>
```

```
      </LiteralConstraint>
```

```
    </StatementTemplate>
```

```
    <StatementTemplate maxOccur="1" type="nonliteral">
```

```
      <Property>http://purl.org/dc/terms/creator</Property>
```

```
      <NonliteralConstraint descriptionTemplateID="creator">
```

```
        <ValueURIOccurrence>disallowed</ValueURIOccurrence>
```

```
        <VocabularyEncodingSchemeOccurrence>disallowed</VocabularyEncodingSchemeOccurrence>
```

```
        <ValueStringConstraint maxOccur="1">
```

```
          <SyntaxEncodingSchemeOccurrence>disallowed</SyntaxEncodingSchemeOccurrence>
```

```
          <LanguageOccurrence>disallowed</LanguageOccurrence>
```

```
        </ValueStringConstraint>
```

```
      </NonliteralConstraint>
```

```
    </StatementTemplate>
```

```
  </DescriptionTemplate>
```

```
<DescriptionTemplate maxOccur="1" minOccur="1">
```

```
  <StatementTemplate maxOccur="1" type="literal">
```

```
    <Property>http://xmlns.com/foaf/0.1/name</Property>
```

```
    <LiteralConstraint>
```

```
      <SyntaxEncodingSchemeOccurrence>disallowed</SyntaxEncodingSchemeOccurrence>
```

```
      <LanguageOccurrence>disallowed</LanguageOccurrence>
```

```
    </LiteralConstraint>
```

```
  </StatementTemplate>
```

```
</DescriptionTemplate>
```

```
</DescriptionSetTemplate>
```




Putting DSPs to work

- The SHAME demo
- Takes DSP-XML
- Generates an RDF editor on the fly
 - RDF conforms to DSP constraints

The speaker

- PhD student at Royal Inst. of Technology, Stockholm
 - Subject: Metadata Standardization and Interoperability
- Engaged in metadata within IEEE, ISO, DCMI
- Co-chair of
 - DC Architecture Forum
 - Joint DCMI / IEEE LTSC Taskforce
- Co-author of
 - DCMI Abstract Model
 - Expressing Dublin Core in RDF
 - Singapore Framework for DC Application Profiles



Overview of tutorial

- **Metadata specifications**
- Interoperability
- The human side of metadata
- The Semantic web
- Metadata records
- Application Profiles

☼ Metadata specifications outside of DCMI

- Many domains:
 - **E-Government**
 - **Education**
 - Geospatial information
 - **Libraries**
 - Business
 - Multimedia
 - Geospatial information
 - etc.



☼ Metadata specifications outside of DCMI

- Many technical environments
 - Low-level (file systems, protocols etc)
 - **File formats** (HTML, Atom, etc.)
 - Ontologies (OWL, etc.)
 - **Repositories**
 - **Harvesting**

☼ Metadata specifications outside of DCMI

- Many expressions
 - XML
 - RDF
 - SQL
 - HTML
 - ID3
 - EXIF
 -



Metadata specifications outside of DCMI

- Many complete “schemas”
 - MARC
 - METS
 - MODS
 - IEEE LOM
 - MPEG-7
 - CSDGM
 - etc.



☼ Metadata specifications outside of DCMI

- Many purposes
 - **Search**
 - **Describe**
 - **Administrate**
 - **Structure**
 - etc.



☼ “Interoperable with Dublin Core”?

- What does it take to be called interoperable?
 - Specific domain? **NO**
 - Specific technical environment? **NO**
 - Specific expression? **NO**
 - Specific complete “schema”? **Not really...**
 - Specific purpose? **NO**
- But seriously...?
- ...we obviously need to talk about the term

“interoperability”



Overview of tutorial

- Metadata specifications
- **Interoperability**
- The human side of metadata
- The Semantic web
- Metadata records
- Application Profiles



☼ The elusive notion of interoperability

- IEEE definition of interoperability:
 - “the ability of two or more systems or components to exchange information and to use the information that has been exchanged.”
- DCMI has four notions of “use the information”:
 - Use the **documented definition** of the DCMI terms
 - Use the **machine-processable semantics** of terms
 - Use the **structure of the metadata record**
 - Use the **constraints in an application profile**
- Each usage leads to a different notion of interoperability



☼ DCMI specifications for interoperability

- Textual definition of the DCMI terms:
 - DCMI Metadata Terms
- Machine-processable semantics of terms:
 - DCMI Abstract Model
 - RDF expression of Dublin Core metadata
 - RDF schemas for DCMI terms
- Metadata records
 - DCMI Abstract Model - “description sets”
 - DCMI expressions: XML, (X)HTML, etc.
- Application profiles
 - Singapore Framework
 - Description Set Profiles



Overview of tutorial

- Metadata specifications
- Interoperability
- **The human side of metadata**
- The Semantic web
- Metadata records
- Application Profiles

☼ Human semantics for Dublin Core

- The Dublin Core terms have carefully crafted definitions

Example:

Label: *Creator*

Definition: *An entity primarily responsible for making the resource.*

Comment: *Examples of a Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.*

- The meaning of these terms is well understood
- Therefore, we often see compatible definitions used in other specifications
- The 15 elements in the “Dublin Core Metadata Element Set” are most often reused



☼ Example: IEEE LOM

- IEEE LOM is the major metadata specification used for Learning resources
- 1.6 Coverage
 - The time, culture, geography or region to which this learning object applies.
 - The extent or scope of the content of the learning object. Coverage will typically include spatial location [...]
 - NOTE —This is the definition from the Dublin Core Metadata Element Set, version 1.1

☼ Example: The Atom Syndication Format

- Atom (like RSS) is used for content syndication (news feeds, podcasts etc.)
- The "**atom:rights**" element
 - is a Text construct that conveys information about rights held in and over an entry or feed. (RFC 4287)
- Cf. the Dublin Core term "Rights":
 - "Information about rights held in and over the resource."



☼ Uses of human semantics

- Compatible reuse of Dublin Core element semantics implies:
 - Well-understood definitions
 - Easier to input metadata
 - Easier to process metadata
 - Definitions interoperable with other metadata specifications
 - Less work to design a metadata specification
 - Easier to create crosswalks
 - Etc.
- Human semantics often sufficient for limited locally scoped projects



Overview of tutorial

- Metadata specifications
- Interoperability
- The human side of metadata
- **The Semantic web**
- Metadata records
- Application Profiles

Dublin Core in the wild: The Semantic Web

- The DCMI Terms carry machine-readable definitions as well:

```

<rdf:Description rdf:about="http://purl.org/dc/terms/creator">
  <rdfs:label xml:lang="en-US">Creator</rdfs:label>
  <rdfs:comment xml:lang="en-US">An entity primarily responsible for making the
    resource.</rdfs:comment>
  <dcterms:description xml:lang="en-US">Examples of a Creator include a person, an
    organization, or a service. Typically, the name of a Creator should be used to indicate the
    entity.</dcterms:description>
  <rdfs:isDefinedBy rdf:resource="http://purl.org/dc/terms"/>
  <dcterms:issued>2008-01-14</dcterms:issued>
  <dcterms:modified>2008-01-14</dcterms:modified>
  <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <dcterms:hasVersion rdf:resource="http://dublincore.org/usage/terms/history/#creatorT-001"/>
  <rdfs:range rdf:resource="http://purl.org/dc/terms/Agent"/>
  <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/elements/1.1/creator"/>
  <rdfs:subPropertyOf rdf:resource="http://purl.org/dc/terms/contributor"/>
</rdf:Description>
  
```



☼ What happens on the Semantic Web?

- Terms are referenced by URI
 - Unique identification part of the framework
- The meaning of terms is carried by the URI
 - No need for manual crosswalks
 - “Context-free” metadata
- Metadata can be merged, mixed and matched
 - No trouble combining metadata from several sources
- Usage of terms is restricted by domains and ranges
 - A “Creator” of a resource is always an Agent.
 - Improves processing possibilities
- “Refinements” are part of the framework
 - A “Vocalist” (marcrel:VOC) of a resource is understood to be “Contributor” (dc:contributor) of the same resource.



☼ Example: Adobe XMP

- Metadata part of media files (JPEG, PDF, RAW, etc.)
- Supported in a wide range of Adobe products
- Mixes Dublin Core properties with
 - EXIF properties
 - PDF properties
 - Etc.

```

<dc:publisher>
  <rdf:Bag>
    <rdf:li rdf:parseType="Resource">
      <rdf:value>James Bond</rdf:value>
      <ns:role>secret agent</ns:role>
    </rdf:li>
  </rdf:Bag>
</dc:publisher>
<tiff:ImageDescription>
  <rdf:Alt>
    <rdf:li xml:lang="x-default">TIFF image description</rdf:li>
    <rdf:li xml:lang="de-DE">TIFF Bildbeschreibung</rdf:li>
  </rdf:Alt>
</tiff:ImageDescription>
<xmpDM:videoFrameSize
  stDim:w="16"
  stDim:h="9"
  stDim:unit="inch"/>

```

☼ Example: Open Document Format

- ODF 1.2 (OASIS) introduced RDF-based metadata
- Supports metadata in
 - Manifest (metadata file)
 - Inline in text of document

```
<bibtex:Article rdf:about="info:pmid/17445913">
  <dc:title>Neuraminidase inhibitor susceptibility[...].</dc:title>
  <dcterms:abstract>As an intermediate host of avian and [...] .</dcterms:abstract>
  <dc:creator>K Bauer</dc:creator>
  <dc:creator>M Schmidtke</dc:creator>

  <foaf:maker>
    <foaf:Person rdf:about="http://purl.org/net/hubmed/ns/pmid/17445913/authors/Bauer,K">
      <foaf:name>K Bauer</foaf:name>
      <rdf:value>K Bauer</rdf:value>
      <foaf:givenname></foaf:givenname>
      <foaf:surname>Bauer</foaf:surname>
    </foaf:Person>
  </foaf:maker>
  <dc:identifier>10.1016/j.antiviral.2007.03.007</dc:identifier>
  <prism:publicationName>Antiviral Res</prism:publicationName>
  <prism:publicationDate>2007-09</prism:publicationDate>
  <prism:volume>75</prism:volume>
  <prism:number>3</prism:number>
  <prism:startingPage>219</prism:startingPage>
  <prism:endingPage>226</prism:endingPage>
  <prism:isPartOf rdf:resource="urn:issn:0166-3542"/>
</bibtex:Article>
```



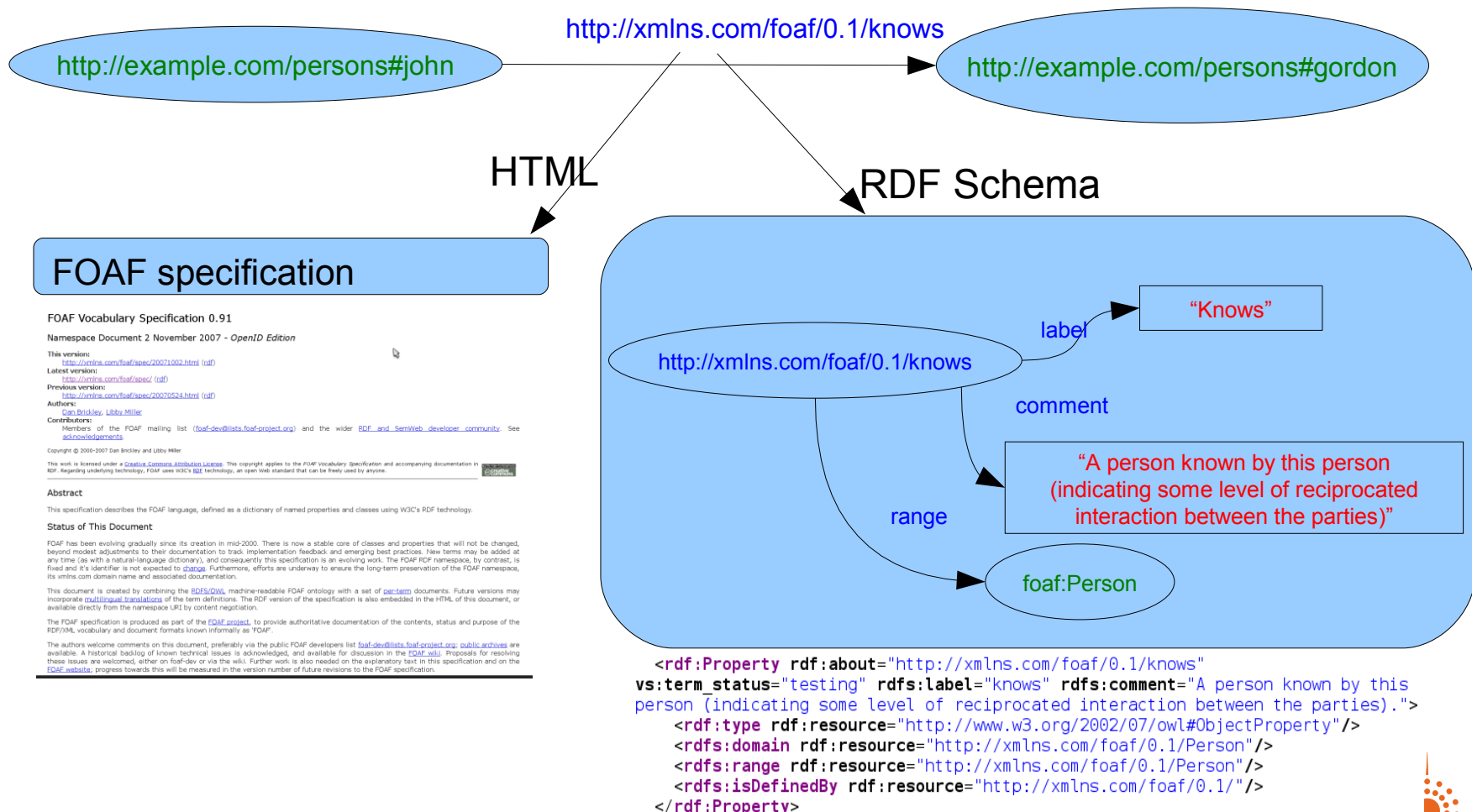
☼ Uses of machine semantics

- The use of a common framework means metadata...
 - from different domains
 - using different vocabularies
 - used in different technical environments...

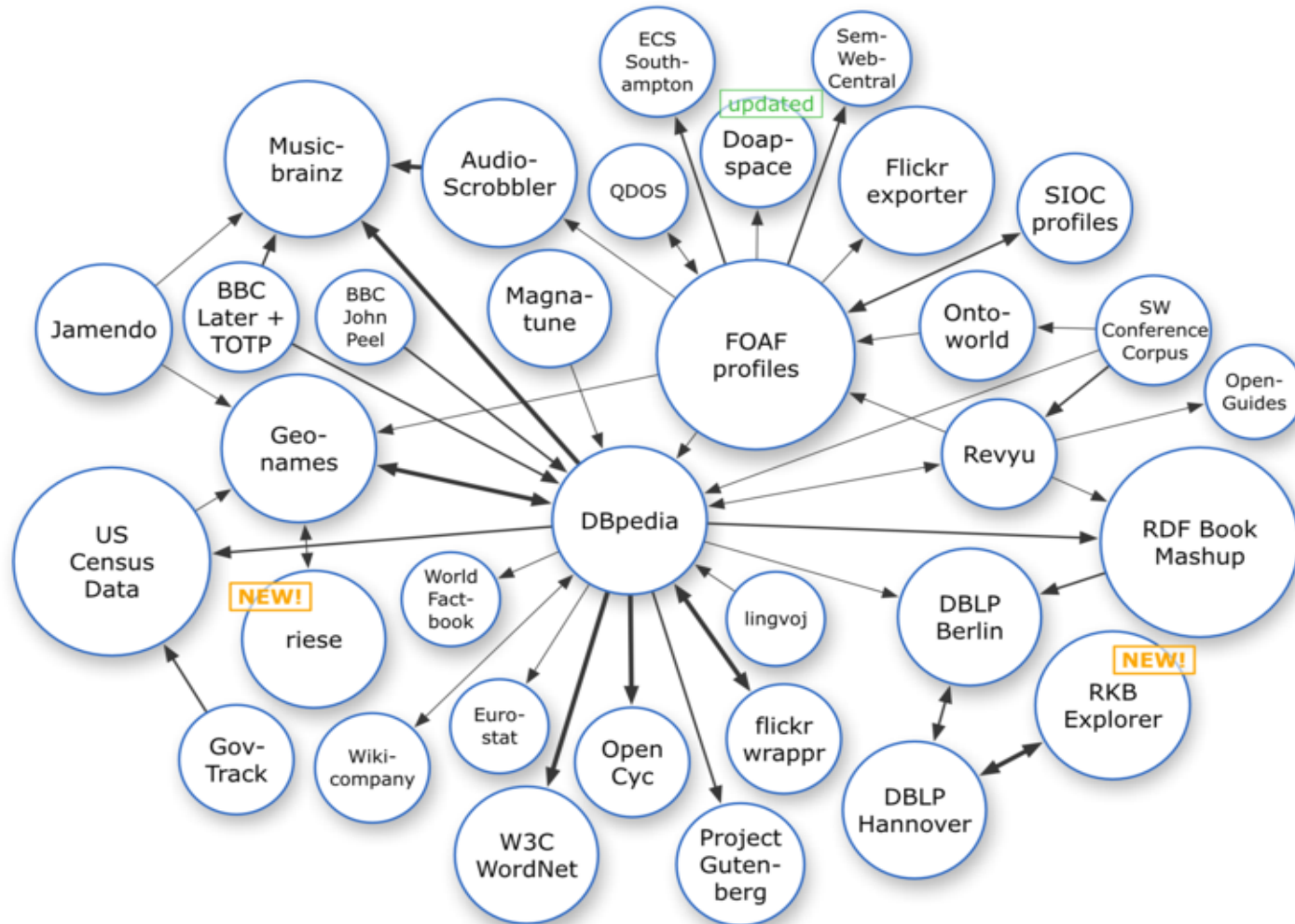
... can be combined without effort.

- Ontologies
 - Enable advanced processing of metadata
- Automatic discovery of term definitions
 - “Follow your nose”
- Linked open data
 - Giant global graph of metadata

Follow your nose



☼ Linked Open Data



- More than 2 billion RDF triples



☼ Support for machine semantics in DCMI

- DCMI Abstract Model
 - Lays the foundations for definition and usage of terms in Dublin Core metadata
 - Builds on RDF
- RDF schemas for DCMI terms
 - Available in “follow your nose”-compatible way
- RDF expression of Dublin Core
 - Defines how to express Dublin Core metadata using RDF



Overview of tutorial

- Metadata specifications
- Interoperability
- The human side of metadata
- The Semantic web
- **Metadata records**
- Application Profiles

☼ Metadata records: structure in the chaos

- Dublin Core builds on experiences from the library community
- A strong influence is the “library card”
 - A manageable “unit of metadata”
- On the other hand: the “one-to-one principle”
 - One resource, one description
 - A Book and its author are described separately
 - A Book and a digital copy are described separately
- The DCMI Abstract Model
 - A formalization of the “library card” for Dublin Core, first formalized in 2005



☼ Why the DCMI Abstract Model?

- A DCMI-specific definition of “metadata record”
 - A framework for designing metadata records
 - A basis for validation of records
 - A basis for exchange formats for records
- The DCAM provides
 - A notion of a “description set”(= record) as a collection of descriptions of individual resources.
 - A formalization of earlier practices in the DC community



☼ Example: DC-DS-XML

- A proposed XML language that provides a generic XML encoding for DC metadata
- Allows for validation with XML schema
- Any properties (even non-DC) can be used.

```
<?xml version="1.0" encoding="UTF-8" ?>
<dcds:descriptionSet
  xmlns:dcds="http://purl.org/dc/xmlns/2008/09/01/dc-ds-xml/">
  <dcds:description
    dcds:resourceURI="http://dublincore.org/pages/home">
      <dcds:statement dcds:propertyURI="http://purl.org/dc/terms/title">
        <dcds:literalValueString xml:lang="en-GB">DCMI Home Page</dcds:literalValueString>
      </dcds:statement>
      <dcds:statement dcds:propertyURI="http://purl.org/dc/terms/publisher">
        dcds:valueURI="http://example.org/agents/DCMI">
          <dcds:valueString>Dublin Core Metadata Initiative</dcds:valueString>
        </dcds:statement>
        <dcds:statement dcds:propertyURI="http://purl.org/dc/terms/date">
          <dcds:literalValueString>2005-05-05</dcds:literalValueString>
        </dcds:statement>
      </dcds:description>
    </dcds:descriptionSet>
```



☼ Example: DC-HTML

- HTML encoding of metadata records
- Does not support full version of DCAM
- Any properties (even non-DC) can be used.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head profile="http://dublincore.org/documents/2008/08/04/dc-html/">
    <title>Services to Government</title>
    <link rel="schema.DC" href="http://purl.org/dc/elements/1.1/" />
    <link rel="schema.MARCREL" href="http://www.loc.gov/loc.terms/relators/" />
    <meta name="DC.title" content="Services to Government" />
    <link rel="MARCREL.EDT" href="http://example.org/agents/DeptOfObfuscation" />
  </head>
```



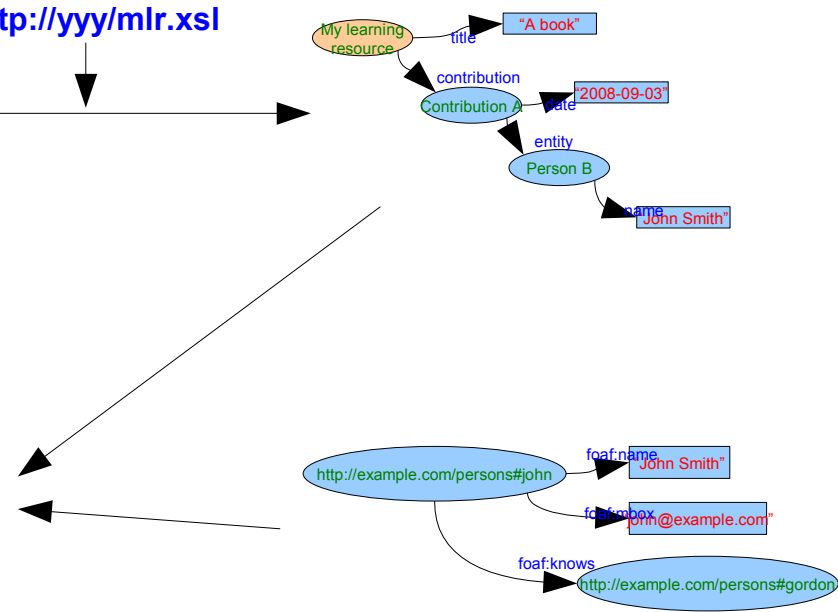
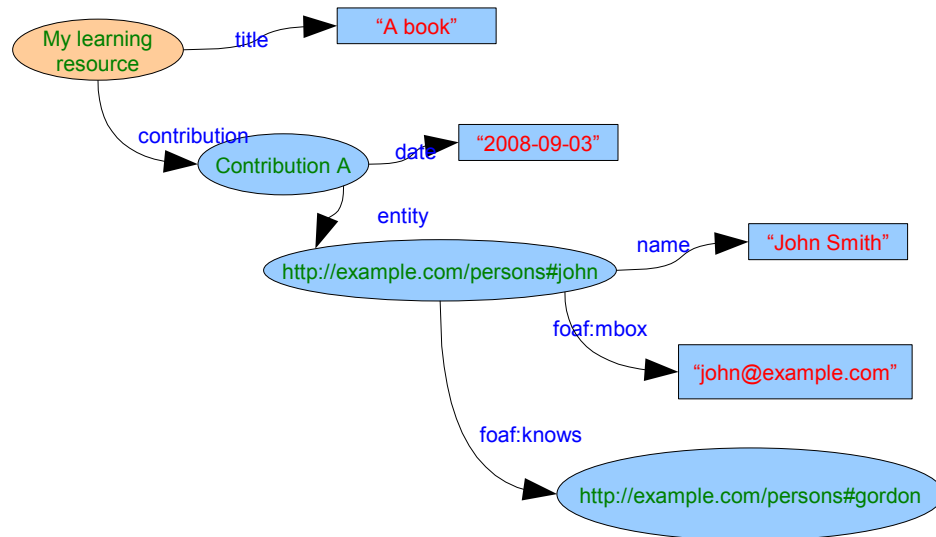
Connecting records and semantics

- Newer DCMI syntaxes support GRRDL from W3C
 - “Gleaning Resource Descriptions from Dialects of Languages”
- Idea: All XML-based languages can auto-generate RDF data
- For DCAM-based formats, it's straightforward.
- Other XML languages are starting to use GRDDL
 - In particular, the Microformats community

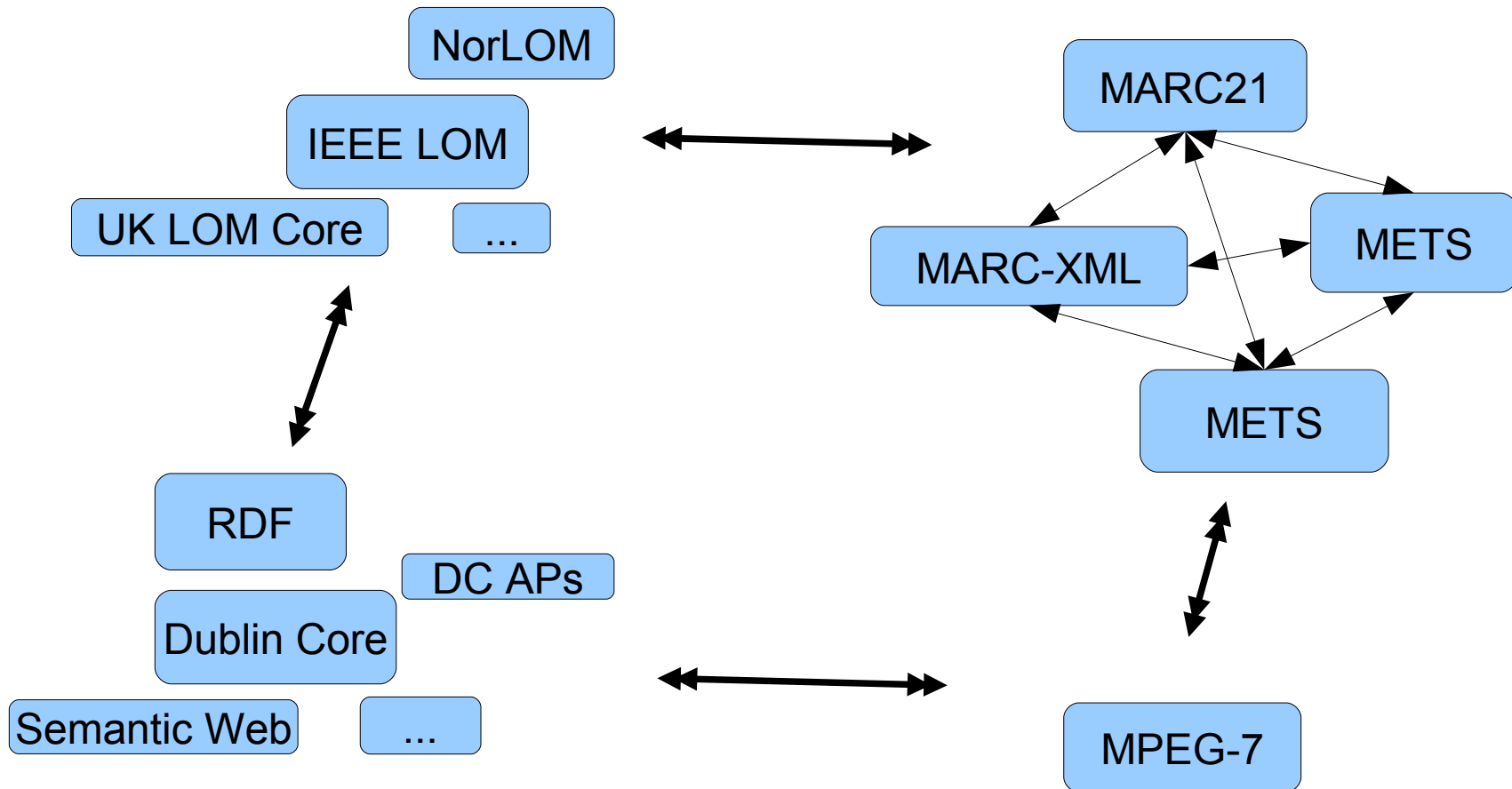
Example: From XML to graphs

```
<LearningResource grddl:transform="http://yyy/mlr.xsl">
  <Title>A book</title>
  <Contribution>
    <Date>2008-09-03</Date>
    <Entity>
      <Name>John Smith</Name>
    </Entity>
  </Contribution>
</LearningResource>
```

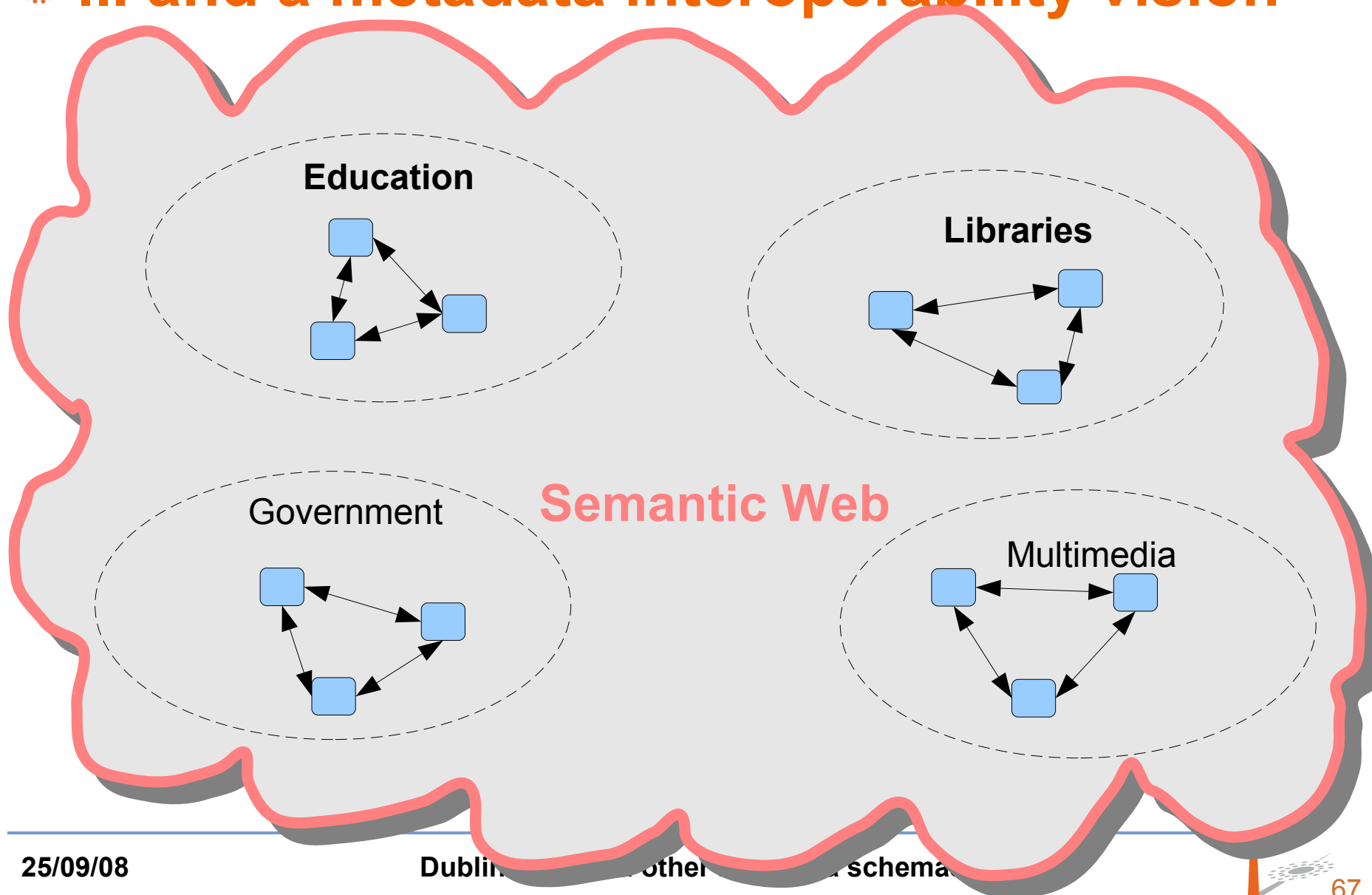
<http://yyy/mlr.xsl>



☼ Metadata islands...



☼ ... and a metadata interoperability vision



☼ Metadata records in DCMI specifications

- DCMI Abstract Model
 - Defines “description sets”, the DCMI notion of metadata records
- DC-DS-XML
 - Encodes general description sets in XML
- DC-HTML?
 - Supports a single description in HTML/XHTML
 - With a few limitations
- DC-RDF?
 - Does not represent “records” explicitly



Overview of tutorial

- Metadata specifications
- Interoperability
- The human side of metadata
- The Semantic web
- Metadata records
- **Application Profiles**



Dublin Core Application Profiles

- We have a definition of manageable records
- We have the option of global interoperability
- We have the widely used term definitions
- What's left?
 - Community interoperability!
- A particular application, domain or community may want
 - Better documentation of their metadata records
 - More support for quality control / validation
- This is usually described with Application Profiles



☼ Application Profiles

- Specifies a community's use of metadata records
- Defines
 - What things are described?
 - Why?
 - Which properties are used?
 - What kinds of values are used?
 - What vocabularies are referenced?
 - What guidelines for data entry are used?
 - etc.



A word of Warning

- DCMI uses “application profile” to mean:
 - A specification of which metadata terms are used
 - A specification of how those terms are constrained and interpreted in the local context
- Other communities use “profile” to mean
 - A customization of an existing schema
 - Starts from a “base” standard, adds context-specifics
- The two notions are not very interoperable!

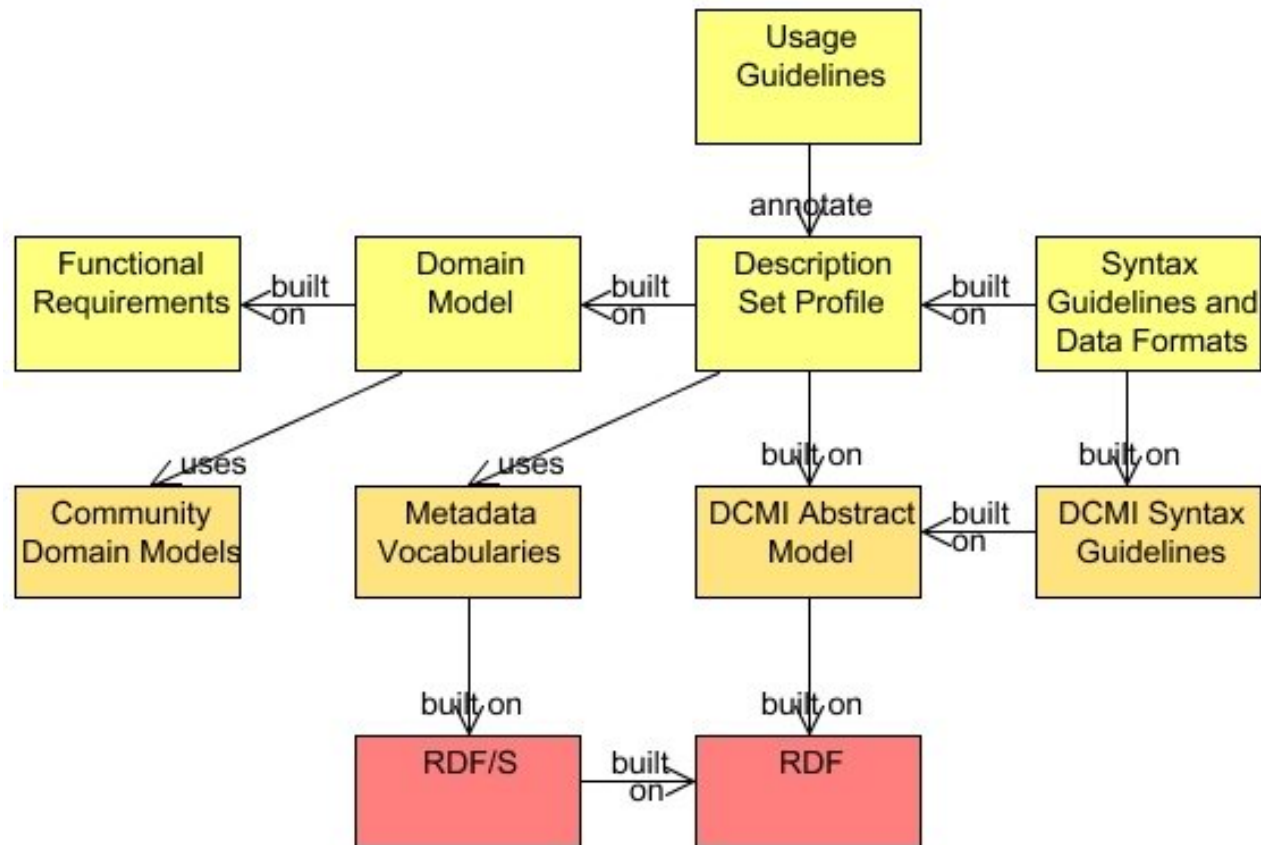
☼ The Singapore Framework

- In Singapore, at DC2007, a new definition of a “Dublin Core Application Profile” was introduced
- A “DC Application Profile” is packet of documentation which consists of:
 - Functional requirements (mandatory)
 - Domain model (mandatory)
 - Description Set Profile (DSP) (mandatory)
 - Usage guidelines (optional)
 - Encoding syntax guidelines (optional)

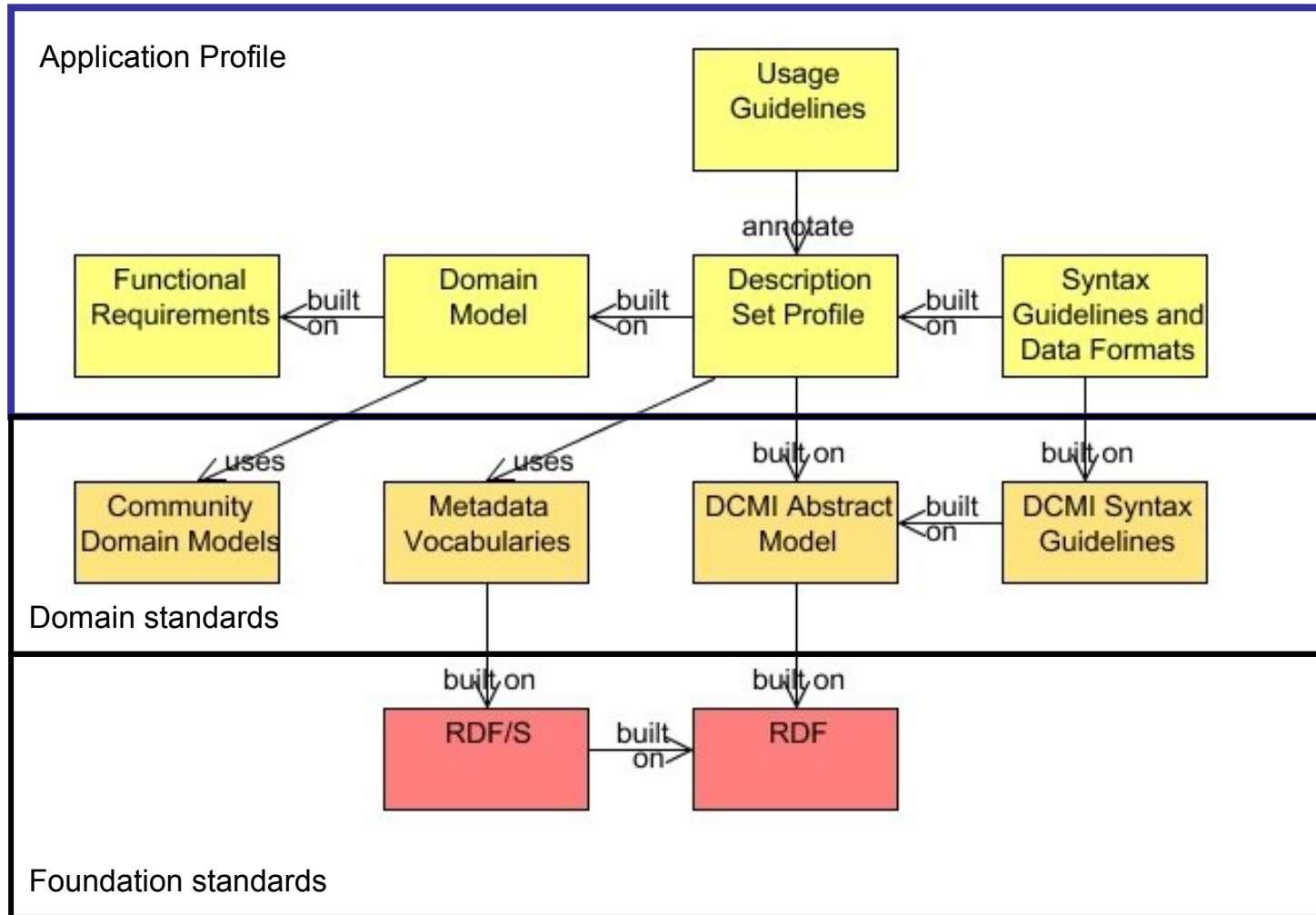
☼ Profiles and standards

- Profiles are based on domain standards:
 - Standard metadata vocabularies (e.g., Dublin Core elements)
 - Standard domain models (e.g., FRBR)
 - Functional Requirements for Bibliographic Records
 - A standard record model (DCMI Abstract Model)
- Foundation is Resource Description Framework (“Semantic Web”)
 - RDF is the model underlying the DCMI Abstract Model
 - RDF Schema is the model underlying the machine processable definitions of terms

☼ The Singapore Framework



☀ The Singapore Framework



Functional requirements

- Describe the functions that the application profile is designed to support
 - as well as functions that are out of scope.
- Form the basis of evaluating the application profile for internal consistency
- Gives guidance on the appropriateness of the application profile for a given use.

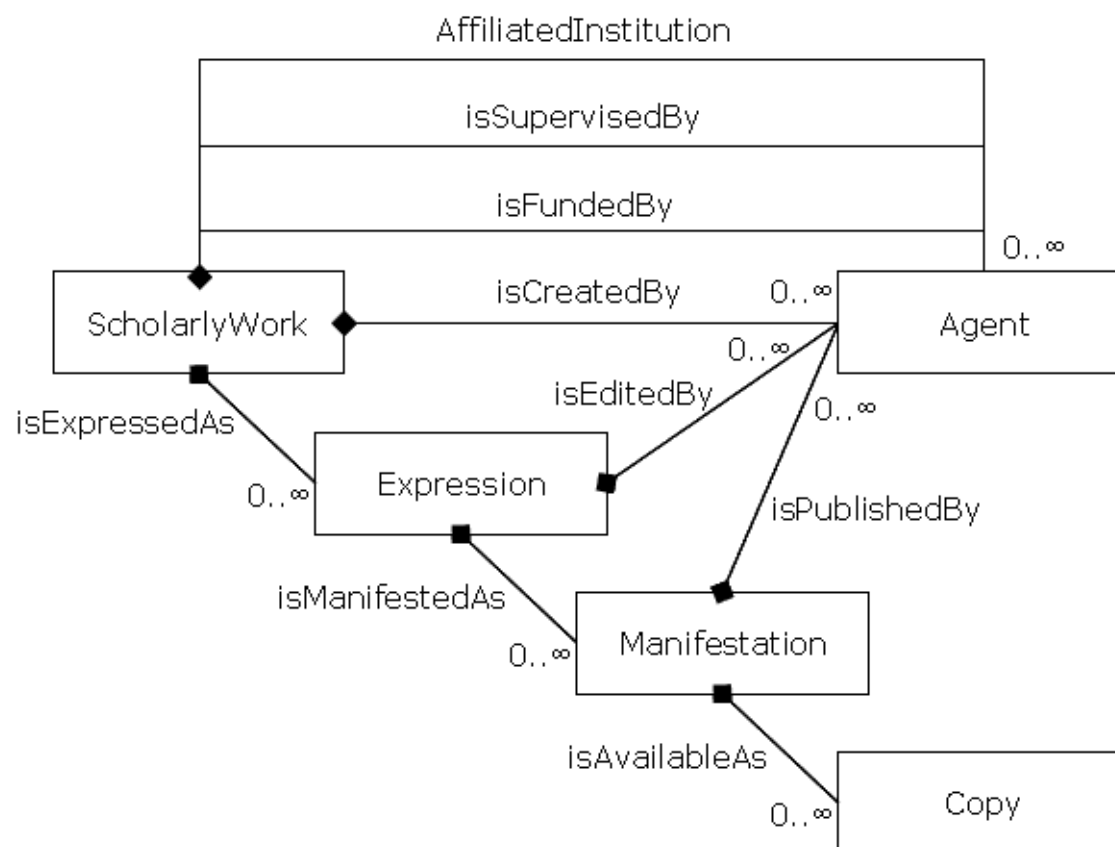
Example: ePrints Functional requirements

- Requirement: Provide a richer set of metadata than is currently possible with simple DC
- Requirement: Be compatible with preservation metadata approaches.
- Requirement: Support extensibility of the application profile for other types of material.
- Requirement: Implement an unambiguous method of identifying the full-text(s).
- Requirement: Support navigation between different 'versions' of the same eprint
- etc.

Domain models

- Defines the basic entities described by the application profile and their fundamental relationships.
- Concretizes the scope for the application profile.
- The domain model can be expressed using just text or using a more formal approach such as UML.
- Does NOT say what properties to use

Example: ePrints domain model



Description Set Profiles

- Defines a set of metadata records that are valid instances of an application profile.
- The Description Set Profile model is currently being developed within the Dublin Core Architecture Forum
- Designed to offer a simple constraint language for Dublin Core metadata, based on the DCMI Abstract Model
- A DSP constrains
 - the resources that may be described by descriptions in a description set conforming to the application profile,
 - the properties that may be used,
 - the ways a value may be referenced.



☼ Description Set Profiles

- A Description Set Profile can be used for purposes such as:
 - formally representing constraints used in metadata
 - configuring databases
 - configuring metadata editing tools

The screenshot shows a web browser window with the address bar displaying <http://www.dn.se/>. The form has a blue background and contains several input fields and buttons. The fields are organized into pairs, each with a 'Value' label and a 'Subject' label. The 'Value' fields contain the following text: 'DSP Presentation', 'Mikael Nilsson', 'application profiles', 'description set profiles', and 'urce description framework'. The 'Subject' fields are empty. Each pair of fields has a green '+' button and a red '-' button to its right. The language is set to 'English' for each pair. The form is titled 'Title' and 'Creator'.

Field	Value	Subject	Language
Title	DSP Presentation		English
Creator	Mikael Nilsson		
Subject	application profiles		English
Subject	description set profiles		English
Subject	urce description framework		English

☀ Example: ePrints DSP

Property	http://purl.org/dc/terms/abstract
Literal?	No
Definition	A summary of the content of the resource.
Eprint-specific recommendation	A summary of the important points of the eprint.

Identifier

Property	http://purl.org/dc/elements/1.1/identifier						
Min occurrence	1						
Literal?	Yes						
Definition	An unambiguous reference to the resource within a given context.						
Eprint-specific recommendation	A URI for the eprint.						
Value (Literal)	<table border="1"> <tr> <th colspan="2">Syntax Encoding Scheme:</th></tr> <tr> <td>Occurrence</td><td>mandatory</td></tr> <tr> <td>Choose from</td><td>http://purl.org/dc/terms/URI</td></tr> </table>	Syntax Encoding Scheme:		Occurrence	mandatory	Choose from	http://purl.org/dc/terms/URI
Syntax Encoding Scheme:							
Occurrence	mandatory						
Choose from	http://purl.org/dc/terms/URI						



Description Template: Book

Statement template: literal title

Property:	dcterms:title		
	Literal value	Language	SES

Statement template: creator

Property:	dcterms:creator	Description reference: Creator
	Value URI	
	Vocabulary Encoding Scheme	
	Value string	Language SES

Description Template: Creator

Statement template: literal name

Property:	foaf:name	standalone:no
	Literal value	Language SES



☼ Usage guidelines

- Usage guidelines describe how to apply the application profile
- How are the used properties intended to be used in the application context?
- What principles are used when gathering data?
- What other principles governs the implementation and use of the application profile

☼ Example: Collections description

- Guidelines for Title element:
 - Where an existing name is used, the value string should preserve the original wording, order and spelling of an existing name.
 - Punctuation need not reflect the usage of the original.
 - Subtitles should be separated from the title by a sequence of space-colon-space, for example:
 - Voices from the Dust Bowl: The Charles L. Todd and Robert Sonkin Migrant Worker Collection “



☼ DCMI specs for Application Profiles

- Singapore Framework
 - <http://dublincore.org/documents/singapore-framework/>
 - Documentation guidelines, January 2008
- Description Set Profiles
 - <http://dublincore.org/documents/dc-dsp/>
 - Formal (machine-readable) part of Singapore Framework
 - Working draft, March 2008
- Implementation experience still needed

Summary

- Dublin Core metadata defined by different forms of “interoperability”
 - For human understanding
 - Machine semantics
 - Metadata records
 - Application Profiles
- Projects need to place themselves on this map
 - What do I need? How can I achieve it?
- The Dublin Core community is open to implementers at all levels



International Conference on Dublin Core and Metadata Applications

Dublin Core and other metadata
schemas

Mikael Nilsson

[<mikael@nilsson.name>](mailto:mikael@nilsson.name)



Interoperability Levels for Dublin Core metadata

- Level 1: Informal interoperability
 - Shared concepts with natural-language definitions
 - No use of formal models or term URIs
 - Test: Is there a mapping to shared elements?
 - Example: IEEE LOM reuses some definitions and maps to 15-element “Dublin Core” (ISO 15836)



Interoperability Levels for Dublin Core metadata

- Level 2: Semantic Interoperability
 - Correct use of formal RDF graph model with conformant vocabularies (eg DCMI Metadata terms)
 - Use of URIs and formal semantic relationships between terms (eg subproperties)
 - Test: Is there a mapping to RDF triples?
 - Examples:
 - All RDF data (by definition)
 - All RDF data extracted from non-RDF formats (eg via GRDDL transforms)
 - All XHTML or HTML data using RDFa or DC-HTML/2008.



Interoperability Levels for Dublin Core metadata

- Level 3: Description set syntactic interoperability
 - Level-2-compatible data packaged in bounded description sets (records) as per DCMI Abstract Model (DC-AM)
 - Conventions for citing vocabulary encoding schemes (controlled vocabularies)
 - Test: Is there a mapping to “Expressing Dublin Core metadata using the DC-Text format”?
 - Examples: All data using DC-AM-compatible specifications, such as DC-DS-XML.



Interoperability Levels for Dublin Core metadata

- Level 4: Description Set Profile Interoperability
 - Level-3-compatible data that follows the specification “Description Set Profiles: A constraint language for Dublin Core Application Profiles”
 - Additional interoperability via shared Functional Requirements and Domain Model (“Singapore Framework for Dublin Core Application Profiles”)
 - Test: Is there a mapping to DSP constraints?
 - Examples:
 - Scholarly Works Application Profile



Interoperability Levels for Dublin Core metadata

- Level 4: Description Set Profile Interoperability
- Level 3: Description Set syntactic interoperability
- Level 2: Semantic interoperability
- Level 1: Informal interoperability

