



# The DCMI Abstract Model and DC Kernel

DCMI Kernel Working Group  
DC-2006: Metadata for Knowledge & Learning, Manzanillo, Mexico

Pete Johnston, Eduserv Foundation  
[pete.johnston@eduserv.org.uk](mailto:pete.johnston@eduserv.org.uk)  
[www.eduserv.org.uk/foundation](http://www.eduserv.org.uk/foundation)





# The DCMI Abstract Model and DC Kernel

- Quick overview of DCMI Abstract Model (DCAM)
- Comparison of DCAM features and Kernel features
  - Very much an initial survey
  - Not the final word!!



# DCMI Abstract Model: Overview

- *DCMI Abstract Model*
  - DCMI Recommendation March 2005
- Formalises and refines “DCMI Grammatical Principles”
- DCAM describes
  - Components and constructs that make up an information structure (“DC description set”)
  - How that information structure is to be interpreted
- Similar to RDF model
- DCAM does **not** describe how to represent DC description set in concrete form



## DCAM Resource Model

- Each *resource* has zero or more *property/value pairs*
- Each *property/value pair* is made up of one *property* and one *value*
- Each *value* is a *resource* (the physical or conceptual entity that is associated with a *property* when it is used to describe a *resource*)
- Each *resource* is an instance of one or more *classes*.
- Essentially, binary relationships between resources
  - Resource -- Property → Value
  - -- Property → Value
  - -- Property → Value
- Compatible with Kernel attribute/value model



# DCAM Vocabulary Model

- Describes the types of terms that are referenced in DC metadata description sets
  - Properties (elements)
  - Classes
  - Vocabulary Encoding Schemes (Enumerated Set of Resources)
  - Syntax Encoding Schemes (Datatype)
- Describes the types of relationships that exist between terms
  - essentially, RDF Schema concepts
  - subpropertyOf
  - subclassOf
  - (future) range
  - (future) domain
- DCAM does not specify set of terms to be used
- All terms are identified by & referenced using URIs



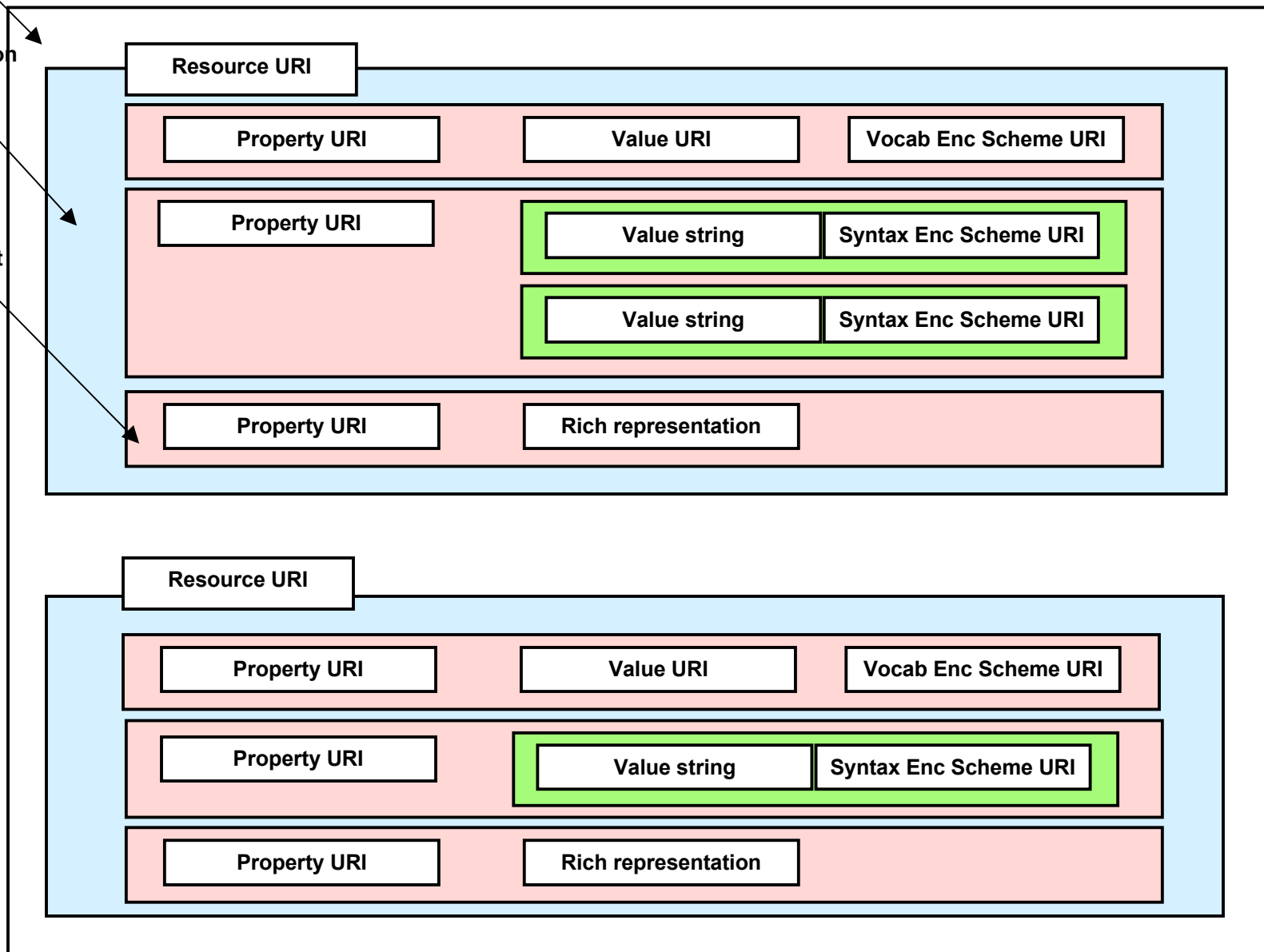
# DCAM Description Model

- a *description set* is made up of one or more *descriptions*
- a *description* is made up of
  - zero or one *resource URI* and
  - one or more *statements*
- a *statement* is made up of
  - exactly one *property URI* and
  - zero or one *reference to a value* in the form of a *value URI*
  - zero or more *representations of a value*, each in the form of a *value representation*
  - zero or one *vocabulary encoding scheme URI*
- a *value representation* is either
  - a *value string* or
  - a *rich representation*
- a *value string* may have an associated *value string language*
- a *value string* may have an associated *syntax encoding scheme URI*
- a *value* may be the subject of a *related description*

Description Set

Description

Statement





## DCAM Vocabulary Model & Kernel

- Kernel elements function as properties
- **But** Kernel elements not identified by URIs
  - Need label – URI mapping
  - (More later)
- No relations between Kernel elements?
  - No subproperty etc inferencing





# DCAM Description Model & Kernel

DCAM	Kernel
Description Set	Instance?
Description	Story see issues
Resource URI	
Statement	Element in Story
Property URI	Element label see issues
Value String	Value multiple value strings supported
Value String Language	



# DCAM Description Model & Kernel

DCAM	Kernel
Rich Representation	
Value URI	
Vocabulary Encoding Scheme URI	
Syntax Encoding Scheme URI	
Related Description	(Implicit in relations between contexts)



## Issues (1)

- Descriptions and Kernel Stories
  - A Description is a set of statements “about” a single described resource
  - Describes set of relationships between resource A and one or more other resources
  - Resource A
    - Is-related-in-way-P-to Value W
    - Is-related-in-way-Q-to Value X
  - Is this true for Kernel Stories?
    - e.g. Kernel note element describes Story not Expression, Content etc



## Issues (2)

- Properties and Kernel Elements
  - A Property is a type of relationship between two resources
  - Property URI always denotes same relationship type
  - **However**, Kernel element label denotes different relationship type depending on Story Context
    - erc where = has-identifier
    - erc-about where = has-spatial-coverage
  - So should map to **different** Property URIs depending on **context**?



## Issues (3)

- Contexts make Stories into Descriptions of Related Resources
  - Contexts may imply additional statements representing relationships between described resources?
  - Resource described by erc context story
  - is-expression-Of
  - Resource described by erc-about context story

**erc:**

**who: Lederberg, Joshua**

**what: Studies of Human Families for Genetic Linkage**

**when: 1974**

**where: <http://profiles.nlm.nih.gov/BB/AA/TT/tt.pdf>**

**note: This is an example of a record for an expression.**

**erc-about:**

**who: Smith, John**

**what: Genetics**

**when: 1965-1972**

**where: California, USA**

**note: This is an example of a record for the content.**

```
@prefix erc <http://kernel.example.org/elements/erc/> .
@prefix erc-about <http://kernel.example.org/elements/erc-about/> .
DescriptionSet (
  Description (
    Statement (
      PropertyURI ( erc:who )
      ValueString ( "Lederberg, Joshua" )
    )
    Statement (
      PropertyURI ( erc:what )
      ValueString ( "Studies of Human Families for Genetic Linkage" )
    )
    Statement (
      PropertyURI ( erc:when )
      ValueString ( "1974" )
    )
    Statement (
      PropertyURI ( erc:where )
      ValueString ( "http://profiles.nlm.nih.gov/BB/AA/TT/tt.pdf" )
    )
    Statement (
      PropertyURI ( erc:isExpressionOf )
      DescriptionRef ( content1 )
    )
  )
)
```

```
Description (  
  DescriptionId ( content1 )  
  Statement (  
    PropertyURI ( erc-about:who )  
    ValueString ( "Smith, John" )  
  )  
  Statement (  
    PropertyURI ( erc-about:what )  
    ValueString ( "Genetics" )  
  )  
  Statement (  
    PropertyURI ( erc-about:when )  
    ValueString ( "1965-1972" )  
  )  
  Statement (  
    PropertyURI ( erc-about:where )  
    ValueString ( "California, USA" )  
  )  
)  
)
```





## Summary

- Kernel resource model is compatible with DCAM resource model (I think!)
- Kernel vocabulary model is small subset of DCAM vocabulary model
  - Need mapping of element labels to URIs
  - Different properties for different contexts
- Kernel description model is **not** (I think!) strictly a subset of DCAM description model
  - Kernel instances may be mapped to DCAM description sets but may require additional term-specific knowledge (e.g. note element)



# The DCMI Abstract Model and DC Kernel

DCMI Kernel Working Group  
DC-2006: Metadata for Knowledge & Learning, Manzanillo, Mexico

Pete Johnston, Eduserv Foundation  
[pete.johnston@eduserv.org.uk](mailto:pete.johnston@eduserv.org.uk)  
[www.eduserv.org.uk/foundation](http://www.eduserv.org.uk/foundation)

