# Why URI Declarations?
## A comparison of architectural approaches

David Booth, Ph.D.

HP Software

European Semantic Web Conference 2008
Workshop on Identity and Reference

Latest version of these slides:
http://dbooth.org/2008/irsw/slides.ppt
Related paper: http://dbooth.org/2008/irsw/

Views expressed herein are those of the author and do not necessarily reflect those of HP.

# Three important roles

- URI owner:
  Mints a URI, e.g., http://example#dbooth
- Statement author:
  Uses the URI to make an assertion, e.g.,
  :s :v <http://example#dbooth> .
- Consuming application:
  Reads the assertion and needs to determine what it means.

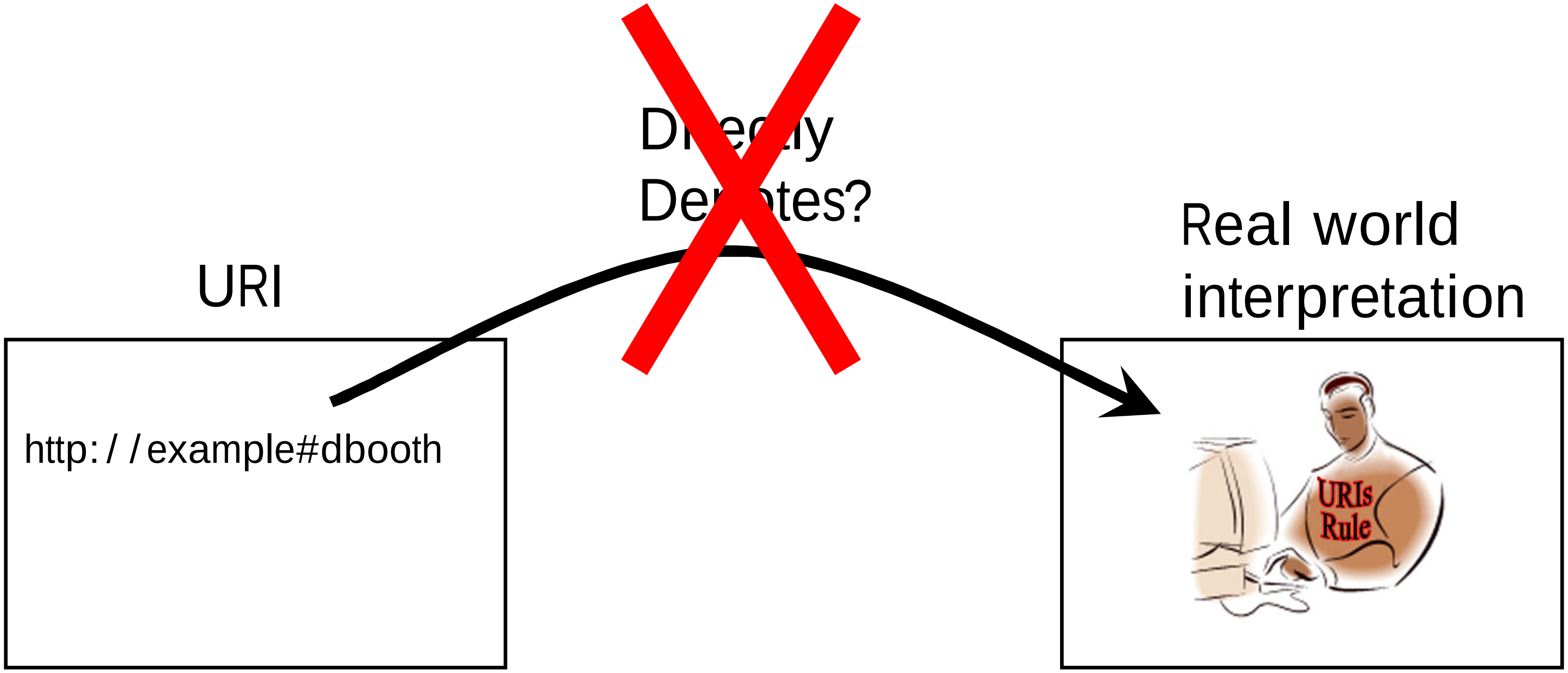
# The problem

Given an RDF statement, what does it mean?

```
:s   :v   <http://example#dbooth> .
```

What resource does <http://example#dbooth> denote?

# URI denotes a resource . . . but how?



URI

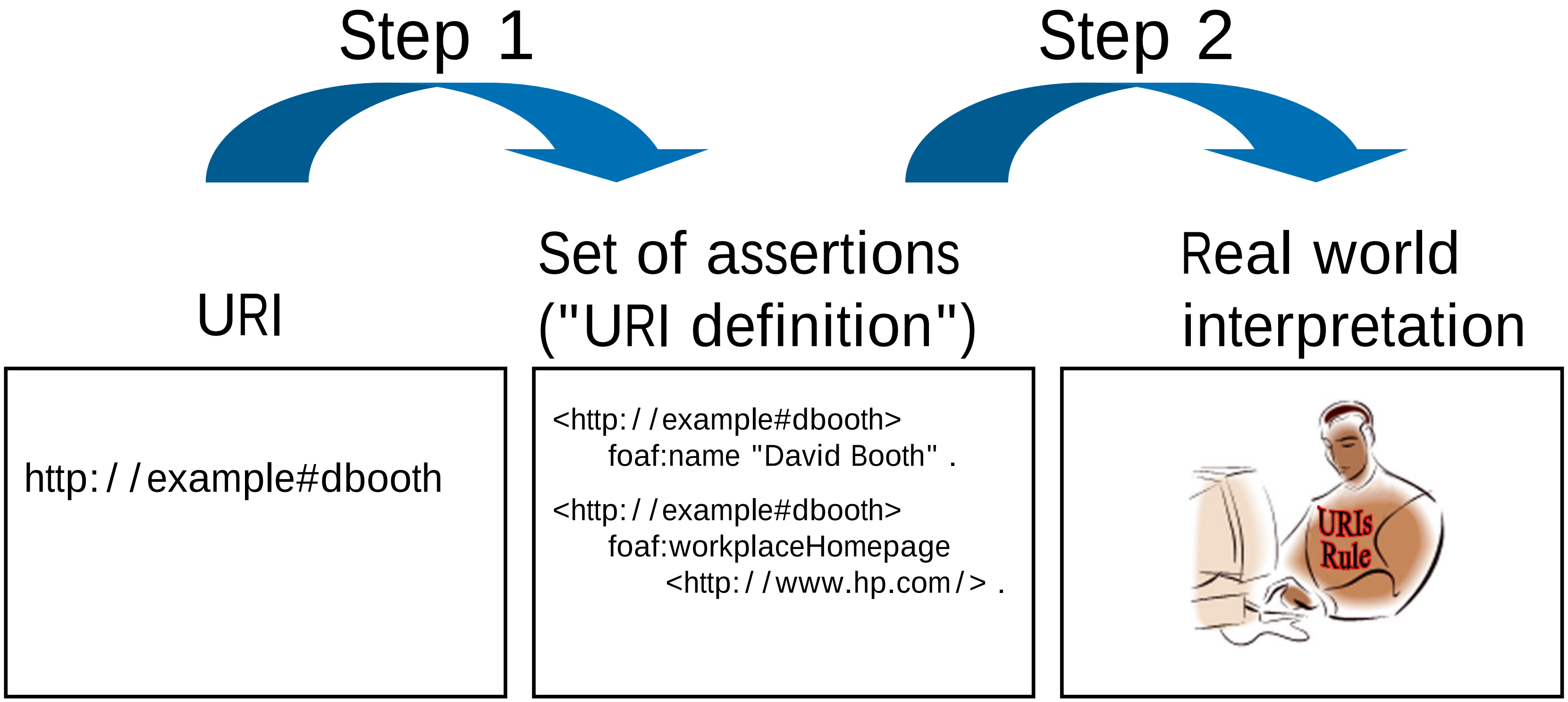
http://example#dbooth

~~Directly Denotes?~~

Real world interpretation

(A person)

# Two-step mapping from URI to resource



URI

http://example#dbooth

Set of assertions ("URI definition")

```
<http://example#dbooth>
     foaf:name "David Booth" .
<http://example#dbooth>
     foaf:workplaceHomepage
          <http://www.hp.com/> .
```

Real world interpretation

# Two-step mapping from URI to resource

```
<http://example#dbooth>
     foaf:name "David Booth" .

<http://example#dbooth>
     foaf:workplaceHomepage
          <http://www.hp.com/> .
```

# Two-step mapping from URI to resource

[redacted]

Step 2

[redacted]

Set of assertions ("URI definition")

```
<http://example#dbooth>
     foaf:name "David Booth" .
<http://example#dbooth>
     foaf:workplaceHomepage
          <http://www.hp.com/> .
```

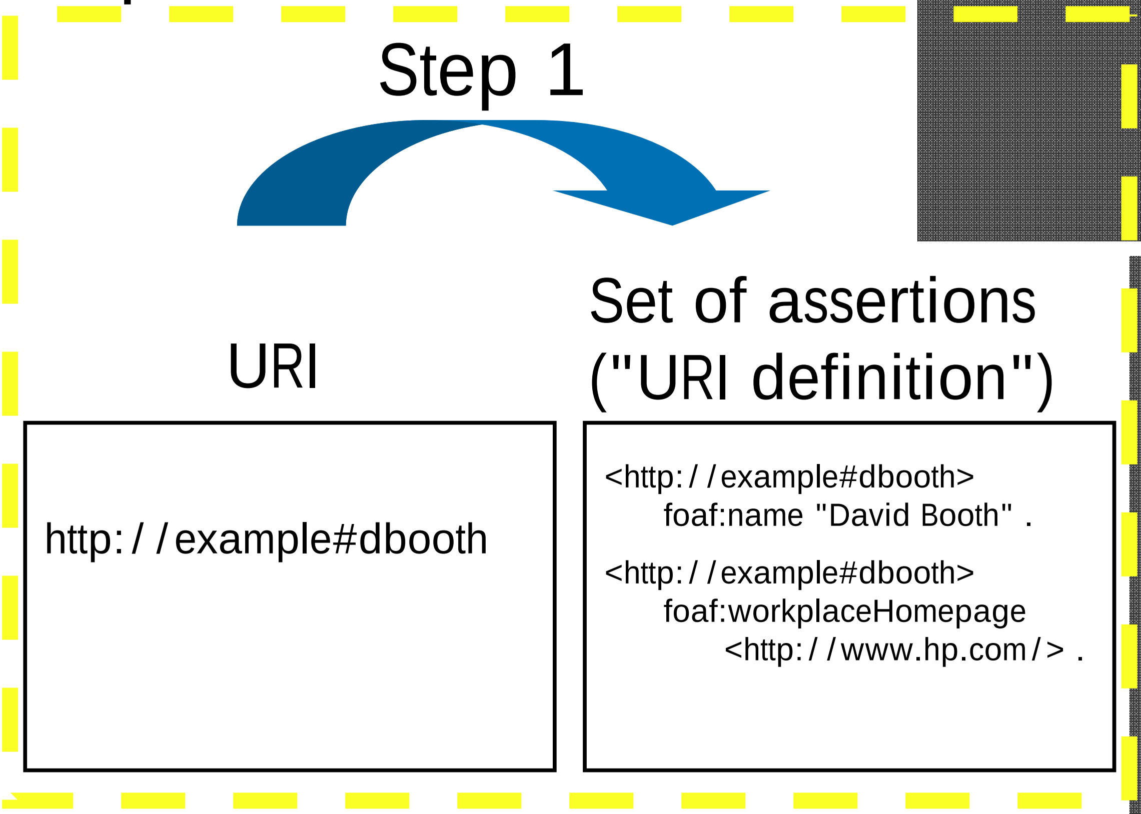
Real world interpretation

Step 2 interpretation:

- <http://example#dbooth> denotes whatever resource satisfies these assertions.

# Two-step mapping from URI to resource

Scope of semantic web architecture



Step 1

URI

Set of assertions ("URI definition")

http://example#dbooth

```
<http://example#dbooth>
     foaf:name "David Booth" .
<http://example#dbooth>
     foaf:workplaceHomepage
          <http://www.hp.com/> .
```

[redacted]

[redacted]

- Semantic web architecture can only govern step 1!

# URI definition determines meaning

Given

```
:s   :v   <http://example#dbooth> .
```

What URI definition did the statement author intend?

# Two architectural approaches

Competing definitions:

- All assertions are created equal.
- Statement author decides which definition to use.

URI declarations:

- Use of a URI implies agreement with its follow-your-nose* definition**.
  - If you disagree with the f-y-n definition, then use a different URI (and relate it to the original URI)
- Statement author decides which URI to use.

*Via 303-redirect or removing #fragID from the URI

**At the time the statement involving the URI was made

# Competing definitions approach

1. How can the statement author indicate what definition was used?
   - rdf:isDefinedBy or owl:imports do not necessarily have this meaning.
   - Hence, consuming app cannot be guaranteed of getting the correct definition.
   - This problem could be addressed by standardization.

2. Alternate URI definitions cause URI collision
   - Same URI has different meaning in different contexts
   - Complicates data merging

# URI declarations approach

1. What if the f-y-n definition is clearly erroneous? E.g., domain was hijacked
    - Statement authors can use a new URI that deprecates the old URI
    - Consuming apps can use a URI translating proxy to get an alternate definition

# Problem: How to indicate a relationship to a URI while disagreeing with its definition

Scenario:

- Alice has published a URI: http://alice.example#foo
- Bob disagrees with one assertion in Alice's URI definition: cos:pluto a cos:planet .
- Bob publishes a new URI: http://bob.example#foo
- Bob's URI definition is similar to Alice's except that it omits the offending assertion. For this reason it is broader (less constraining) than Alice's definition.

Question:

- How can Bob indicate the relationship between his URI and Alice's URI?

Observation:

- Bob's URI definition is skos:broader than Alice's
  - It omits the offending assertion
- The following statement would indicate the relationship:

  <http://alice.example#foo> skos:broader* <http://bob.example#foo> . # WRONG!
- But it would also indicate agreement with the offending assertion!

- *Update 16-Jan-2009: This example is also incorrect because skos:broader is being asserted between the resources denoted by http://alice.exalmple#foo and http://bob.example#foo rather than between their URI definitions. For more explanation see http://dbooth.org/2007/splitting/

# Solution

To avoid indicating agreement with Alice's assertions, Bob can instead write:

```
_:aliceFoo log:uri "http://alice.example#foo" .
_:aliceFoo skos:broader* <http://bob.example#foo> .
```

Explanation:

- log:uri relates a resource to a URI that denotes it.
- For any URI u, if u is used to denote a resource, then the following relationship is implied:

```
<u> log:uri "u" .
```

* Correction 16-Jan-2009: The skos:broader assertion above is between the two resources denoted by http://alice.example#foo and http://bob.example#foo when it should have been between their URI definitions. As described in http://dbooth.org/2007/splitting/ , the assertion between their URI definitions can be made as follows without indicating agreement with Alice's assertions:

```
"http://alice.example#foo" s:isNarrowerThan "http://bob.example#foo" .
```

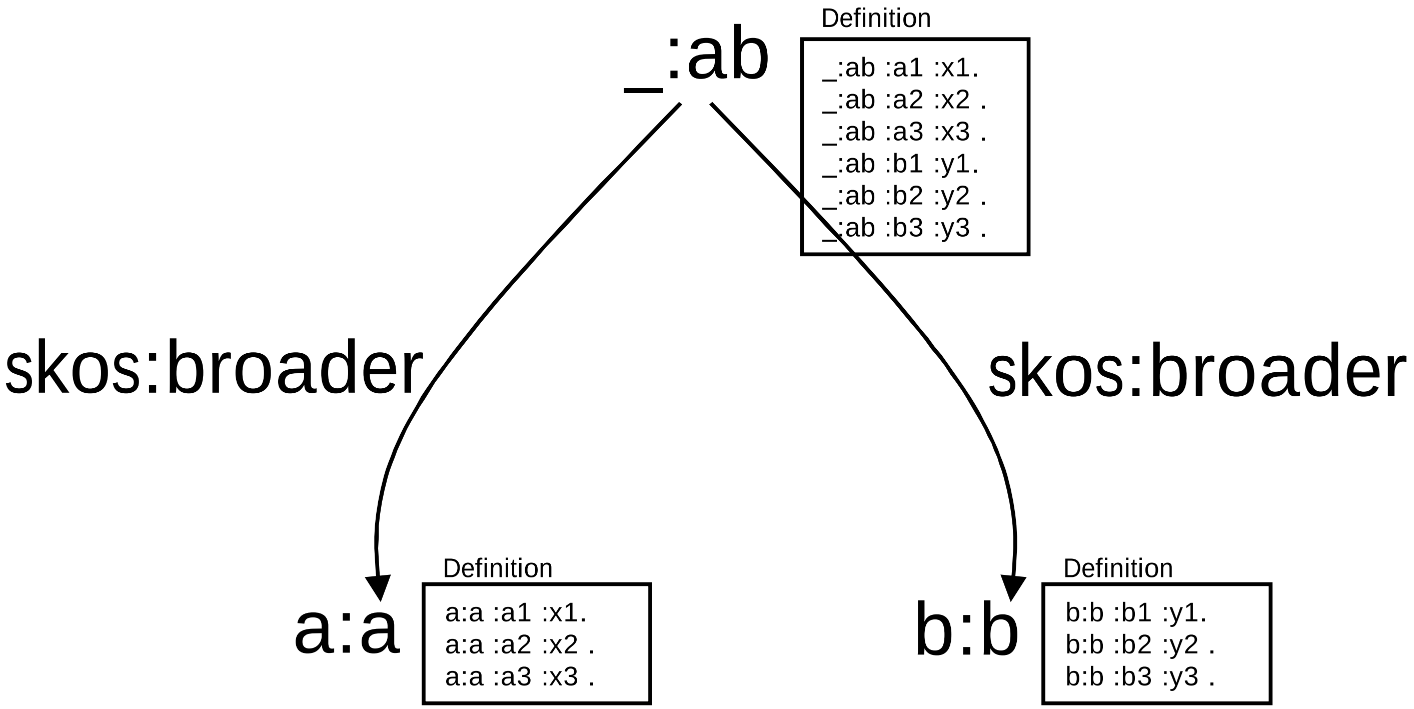
# owl:sameAs

- Creates value
  - Permits data to be merged.  Good!
- Also creates problems when combining data
- Contradiction does not mean that the data is wrong!
  - Models may be okay for one context, but inadequate for another
    - E.g., modeling the earth as flat is good enough for driving directions
- This problem will never go away!
  - Avoid it when possible
  - But be prepared when it happens

# Scenario (owl:sameAs)

- File1 says a:a owl:sameAs b:b .
- File2 says b:b owl:sameAs c:c .
- File1 and File2 each work fine by themselves, but cause a contradiction when used together. How can we deal with this?

# owl:sameAs effectively creates a new node



_:ab

Definition

```
_:ab :a1 :x1.
_:ab :a2 :x2 .
_:ab :a3 :x3 .
_:ab :b1 :y1.
_:ab :b2 :y2 .
_:ab :b3 :y3 .
```

skos:broader

skos:broader

a:a

Definition

```
a:a :a1 :x1.
a:a :a2 :x2 .
a:a :a3 :x3 .
```

b:b

Definition

```
b:b :b1 :y1.
b:b :b2 :y2 .
b:b :b3 :y3 .
```

- _:ab definition combines assertions from URI definitions of a:a and b:b
- Data that merged a:a and b:b effectively used _:ab
- To confine a conflict, s/a:a/_:ab/g in File1 (or analogously in File2)

- Clarification 16-Jan-2009: The skos:broader relation shown above should be between the URI definitions -- not between the resources denoted by _:ab and a:a or _:ab and b:b.

# Ambiguity and owl:sameAs

- Ambiguity is undesirable but unavoidable
  - An identity that was good enough for one app may be insufficiently precise for another
    - Pat Hayes the physical body?
    - At what point in time?
    - Pat Hayes the legal entity?
- Uses of owl:sameAs would be vanishingly few if limited to cases of identical URI definitions
- owl:sameAs can be viewed as an expression of belief: for this app/context, these two URIs denote the same resource.

# AKT example

- AKT protein is discovered
  - URI is minted: :akt
- Years later, research determines that there are actually three distinct proteins: AKT1, AKT2, AKT3
  - New URIs are minted: :akt1, :akt2, :akt3
- :akt is good enough for many applications
  - Do not change it!
  - Indicate that it has been deprecated by :akt1, :akt2, :akt3
- :akt skos:narrower* :akt1, :akt2, :akt3 .

*Correction 16-Jan-2009: skos:narrower is making an assertion between the resources denoted by :akt, :akt1, :akt2 and :akt3, when the assertion should have been between their associated URI definitions (though the example does not show their URIs). Using s:isBroaderThanResource as described in http://dbooth.org/2007/splitting/ the above line could have been correctly written as:

:akt s:isBroaderThanResource :akt1, :akt2, :akt3 .

However, it would have been better to express the relationship through URIs, using s:isBroaderThan, such as:

```
"http://jann.example#akt" s:isBroaderThan
        "http://luke.example#akt1" , "http://luke.example#akt2" , "http://luke.example#akt3" .
```

# httpRange-14 implications

- http://markbaker.ca/ denotes both:
    - Mark Baker the person
    - Mark Baker's web page
- Architecturally, this is no different from AKT example!

THEREFORE:

- Using the same URI to denote both a person and a web page:
    - is not a violation of web architecture, but
    - is a violation of good practice

# Good practice guidelines for minting URIs

- Support follow-your-nose dereferencing to URI declaration
  - http URIs
  - Use #hash or 303-redirect URIs
- Put only defining properties in the URI declaration
- Make distinctions that users are likely to need
  - e.g., distinguish a person from his/her web page
- URI declaration should also link to a page describing:
  - Known sources of "ancillary" assertions -- including yours
  - Related URIs (skos:broader/skos:narrower/skos:related)
- Clearly indicate the change policy for your URI declaration
- Indicate the date last modified
- Do not change your URI declaration
  - Except in accordance with its change policy
- Use persistent URLs (PURLs)

# Questions?