

[Slide stating problem]

# Appropriateness – misusing DC to describe "people" or "places"

- Possible solution
  - Evaluate appropriateness element-by-element relative to entities described
  - At a minimum, Identifier and Type ("Person")?
    - Helps users decide utility of resources
  - Guidance, type-by-type, on appropriate usage
  - Use protocol to express preference for richer formats (see below)

# Quality of metadata content

- "Junk" records make junky indexes
- Responses
  - Junk metadata problem broader than just DC!
  - Better tools, interfaces, partitioned workflows for metadata creation and quality control
  - Better guidelines for use of DC in specific communities
  - DC useful not just for indexing, but listing and browsing results

# Quality – too-broad semantics

- Intrinsic problem with DC?
  - Broad, fuzzy buckets inevitably hold diversity
- Harvester must sometimes guess context
  - e.g., dc:identifier, dc:subject
    - Proliferation of more-specific elements/qualifiers would increase complexity
    - Rather, recommended constraints, e.g., "dc:identifier should be a URI"

# Effort – stopping short of providing richer metadata

- Problem
  - People with rich metadata dumbing down to DC
  - People with no metadata make DC then stop, even if DC is not ideal
- Responses
  - Promote richer metadata parallel to DC
  - Point to preferred metadata from **about** container for required DC format
  - Or consider DC "non-preferred" by default (most cases)

# Examples of richer formats

- LOM
- MARC
- Academic metadata format (AMF)
- RDF combination of multiple metadata formats
- OLAC

# Group mood – summary

- Keep Dublin Core mandatory
- Encourage use of richer formats
- Support indication of "preferred" formats
- Constrain too-broad elements with usage recommendations ("use a URI for dc:identifier")
- Explain how to use DC with non-document-like resources (just Identifier/Type?)

# Issues

- Constraint that no solution can invalidate current implementations
- Issue of empty DC records that are actually compliant