21.01.2016 Arbeitstreffen in Frankfurt am Main

Ort: DNB Frankfurt, Raum 436

Zeit: 10:30 - 15:30 Uhr

Notizen/Protokoll: http://etherpad.lobid.org/p/20160121_dini-titeldaten (Urfassung. [22.01.2016 Jana Hentschke] *Übertragen in die jeweiligen Arbeitsseiten und unten auf dieser Seite*)

Teilnehmer:

Adrian Pohl	hbz
Thomas Striffler	HeBIS
Carsten Klee	ZDB
Cornelia Katz	BSZ
Andreas Kahl	BVB
Lars Svensson	DNB
Reinhold Heuvelmann	DNB
Sarah Hartmann	DNB
Jana Hentschke	DNB
Barbara Block	GBV
Stefanie Rühle	SUB Göttingen

Protokoll

- Protokoll
  - Wiederkehrendes Problem "Informationen sowohl als Verlinkung als auch als Literal vorhanden"
  - Offene Modellierungsfragen
  - Veröffentlichung der Empfehlungen als Profil/Schema/Shape in einer Constraint-Language
  - Weiteres Vorgehen, Zeitplan, Zusammenfassung der ToDos

Wiederkehrendes Problem "Informationen sowohl als Verlinkung als auch als Literal vorhanden"

Grundsätzlich klären, da es am saubersten immer gleich gelöst würde.

Optionen

- 2 Properties
- Object-Property und bNode für Literal
- Object-Property und generierten URI für Literal
- Literal zur Ausgabe nicht empfehlen
- 1 Property die nicht explizit als owl:Datatypeproperty oder owl:Objectproperty deklariert ist für beides benutzten (rdau-Properties machen diesbezüglich keine Angabe)

aktuelle Anwendungsfälle

- Verantwortlichkeit (aktuelle Empfehlung: dc:/dcterms:)
- Werktitel ab RDA (Diskussion läuft, s.u.)
- Zielgruppe (erweiterte Empfehlungen nach RDA: Literale zur Ausgabe nicht empfehlen)

Andreas und Carsten sind nicht vehement gegen bnodes. Frage ist, wie die Datennutzer das sehen. (Carsten: Wer sind überhaupt die Nutzer?)
DNB-interne Diskussion: für SPARQL-Abfragen ist bnode nützlich, weil eine Abfrage für beide Fälle (mit/ohne URI) funktioniert. (So hat auch Konstantin Baierer auf der LDS-Liste argumentiert.) Die beste Lösung für alle Fälle gibt es nicht.
hbz bietet JSON-LD an: da sind bnodes nicht so hässlich wie in anderen Serialisierungen, es fehlt schlicht eine "@id"-Property (vorausgesetzt, man verarbeitet die Daten als JSON, statt als Tripel)
Andreas berichtet vom praktischen Problem, dass bei der Serialisierung der Dumps in mehreren Durchgängen gearbeitet wird und dann schwer sichergestellt werden kann, dass die node-IDs eindeutig ist. Eine vollständige Umsetzung des bnode-Ansatzes im b3kat wäre somit höchstwahrscheinlich 2016 nicht möglich.
Zu Hash-URI-Option: Nachteil: IDs müssen erhalten bleiben (wenn man einen Hash-URI verwendet, wäre das aber ja eigentlich nur ein Teil der Bibliographischen Resource, auf die man verweist)
Carsten: Kann es denn nur _eine_ Empfehlung geben? Oder auch alternative Umsetzungen? Adrian pflichtet bei, dass in diesem Fall und evtl. auch bei anderen das Aufzeigen von Alternativen in der Implementierung sinnvoll wären.

Empfehlung: konsequent eine Lösung für alle Anwendungsfälle (alle "Elemente", wo URI oder Literal vorkommen können)
Reinhold: Ist die diskutierte Problematik RDF-spezifisch oder gibt es mit MARC21 ähnliche Probleme in der Verarbeitung und beim Aufbau von Endnutzerfunktionen?
Lars: Können wir davon ausgehen, dass unsere Kunden die Daten in erster Linie als RDF konsumieren? Oder nutzen sie XML mit XML-Tools, N-Triples mit grep etc.?
Andreas: Ziel: rdf anbieten und nicht auf "nutzer" eingehen, die xml auswerten
alle: rdf ist rdf, Empfehlung soll auf rdf ausgerichtet sein
Vorteil 2 properties: ohne bnode, bisherige Umsetzung, objectproperty sollte mehr Gewicht haben, um die Vision auf eines Tages flächendeckende Links zu unterstreichen
Lars: wenn Ziel rdf ist, dann pro bnode
(https://github.com/hbz/lobid-rdf-to-json/issues/17 bnode auch für gnd nutzen?)
Empfehlung: bnode? alle einverstanden?
bnode/hash-URI Generierung: hash-URI dereferenzierung? Rücklieferung: "Haupt-Ressource"
DDB-Lösung: Events generieren mit Hash-URIs .../...#event124 ?
Carsten: generierte Code als #URI
Jana-Vorschlag Werk: URIRessource#work
Zusammenstellungen: URIRessource#work1; URIRessource#work2
Hash-URIs könnten evtl. auch Probleme mit sich bringen, wenn sie durch GND-URIs ersetzt werden und somit nicht mehr existieren.
Stefanie: auch wenn zu der mit Hash-URI referenzierten Resource eines Tages keine Informationen mehr ausgeliefert werden (sobald ein GND-Verlinkung hergestellt wurde), ist der Nutzer dann immerhin an der Stelle, an der die neuen Informationen stehen.
Der Datenkonsument ist dann nicht mit HTTP Status Code 404 konfrontiert.
Carsten: eine 404-Meldung gäbe allerdings die Chance, die betreffenden Tripel als veraltet zu erkennen und z.B. im eigenen Triplestore zu löschen.
Empfehlung: bnode oder falls vom Datenprovider nicht verwendet werden soll/kann auch bnode+Hash-URI verwenden --> objectproperty verwenden --> im Zuge der Vereinheitlichung fällt dann auch die aktuelle 2-Property-Lösung dc:creator und dc:contributor weg

Anderes Thema, das sich aus dieser Diskussion ergab:
Wenn möglich: DC Elements komplett rausnehmen und einheitlich DC Terms verwenden
⚠️ **ToDo** Stefanie: checkt ob DC elements komplett rausgenommen werden kann und nur dcterms verwendet werden kann in den Empfehlungen (z. B. dc:identifier)
Beispiel für eine Hash-URI aus BSBs RISM: http://data.rism.info/id/rismid/452010020?format=rdf
Hash-URI darin: http://data.rism.info/id/rismid/452010020?format=rdf#ex-ks30000655-Musms1280

Keine Empfehlung für die genaue Konstruktion der Hash-URIs abgeben, aber ein Beispiel in den Empfehlungen nennen.
Fazit: Es soll ein allgemeingültiger Abschnitt in die Empfehlungen eingefügt werden, in dem die empfohlenen Optionen für die "Literal-statt-Verknüpfung" ssituation erläutert werden: bNode oder, wenn technisch nicht möglich, Hash-URIs bilden (mit Beispiel für Hash-URI aber ohne konkrete Implementierungsvorgaben). Dabei soll deutlich gemacht werden, dass sich die empfohlenen Optionen auf alle Stellen beziehen an denen Object-Properties mit URI empfohlen werden, man aber in der Praxis auch Literale vorliegen haben kann.
Die dc:/dct:creator/contributor Lösung der aktuellen Empfehlungen soll entfernt werden

Offene Modellierungsfragen

- Werke (Protokoll Diskussion siehe dort)
- Zusammenstellungen (Protokoll Diskussion siehe dort)
- Sonstiges (Protokoll Diskussion siehe dort)
    - Hochschulschriften
    - Sprachliche Anpassungen
- Erweiterte Empfehlungen (NEU: Kandidaten aus der Praxis)
  => wird vertragt, ⚠️ BVB und BSZ tragen ihre Erweiterungen nach unter https://wiki.dnb.de/x/VwaxBg

Veröffentlichung der Empfehlungen als Profil/Schema/Shape in einer Constraint-Language

Constraint Languages können a) Datenstruktur beschreiben und b) zur Datenvalidierung verwendet werden.
W3C Working Group arbeitet an etwas wie ein Schema für RDF-Daten um Validieren zu können.
Offen: Damit sollte man eigentlich auch Content Negotiation machen können. -> Vorschlag Lars Svensson für neuen HTTP-Header läuft
Als Constraint Language konkurieren gerade noch die beiden Lösungen *SHACL* und *ShEx*
Praktiker, die heute etwas zum Validieren brauchen, benutzen owl (Stardog Graph Database), json-schema (hbz) ... auch im JSON-LD Context könnte man schon teilweise etwas ausdrücken (z.B. ob Objekt URI)
Wenn jemand einen Anwendungsfall hat und das KIM-AP - in welcher Form auch immer - formalisiert, wird es natürlich geteilt. Ansonsten informieren Stefanie und Lars, die beide in der DCMI RDF Application Profiles Task Force sind, wenn sich ein Gewinner für die Constraint Language abzeichnet.
Einigkeit herrscht darüber, dass die Empfehlungen formalisiert werden sollen, sobald das möglich ist.
Fazit: sobald sich eine Constraint Language durchsetzt, sollen die KIM-Empfehlungen formalisiert werden.
⚠️ ToDo Stefanie Rühle und Lars Svensson aktivieren die Gruppe, sobald es soweit ist

Revisit: Ziele der Titeldatengruppe/der Empfehlungen

Anregung Adrian Pohl, Folien

Motivation in der Einleitung der Empfehlungen neu formulieren?
Fazit: das ist unbedingt nötig, da nicht mehr dieselbe Ausgangssituation besteht, wie bei den ersten Empfehlungen
⚠️ ToDo Adrian Pohl: neue Einleitung entwerfen

Weiteres Vorgehen, Zeitplan, Zusammenfassung der ToDos

- verbleibende Liste der Kandidaten für erweiterte Empfehlungen soll in TelKo im März durchgesprochen werden
    - Jana Hentschke organisiert Termin

    - Andreas Kahl ergänzt Erweiterungen im b3kat-RDF
- Zeitnah ins Wiki schon mal die Reinschrift der RDA-Anpassungsergebnisse (Jana Hentschke)
    - in dnbt neue Unterklassen von bibo:Thesis für die sieben Typen von Hochschulschriften anlegen, damit beschlossene Hochschulschriftenmodellierung umgesetzt werden kann (Lars Svensson)
- Neufassung der Empfehlungen (kein großer Zeitdruck. Bis ~Mitte 2016?) Finales Durchgehen in TelKo. Vorher:
    - es soll geprüft werden, ob DC elements komplett rausgenommen werden kann und nur dcterms verwendet werden kann (z. B. dc: identifier) (Stefanie Rühle)
    - Neue Einleitung (Adrian Pohl)
    - Entwürfe der Neufassungen (Jana Hentschke) (mit den sprachlichen Anpassungsoptionen von Cornelia Katz)
        - Einleitung, Vorgehen etc.
        - Empfehlungen (konzeptuell)
        - Erweiterte Empfehlungen
        - MARC-Mapping