Informationspraxis Bd. 4, Nr. 1 (2018)

DOI: https://doi.org/10.11588/ip.2018.1.49357

Christian HAUSCHKE, Tatiana WALTHER & Graham TRIGGS

Vitro – ein universell einsetzbarer Editor für Ontologien und Instanzen

Zusammenfassung
In diesem Artikel wird die Open-Source-Software Vitro beschrieben. Vitro ist ein universeller Instanz- und Ontologie-Editor, der hauptsächlich von der VIVO-Community entwickelt und gepflegt wird. Es wird ein Überblick über die Hauptmerkmale der Software geschaffen. Dann werden verschiedene Anwendungen der Software beispielhaft erläutert. Der Artikel schließt mit einem Ausblick auf zukünftige Anwendungsfälle und Entwicklungen von VitroU+002e

Schlüsselwörter
Linked Data, Ontologie-Entwicklung

Vitro – a general purpose editor for ontologies and instances

Abstract
In this article the open source software Vitro is described. Vitro is a general purpose instance- and ontology editor mainly developed and maintained by VIVO community. Following an overview of its main features, different uses of the software are explained as examples. The article concludes with an outlook about future use cases and developments of Vitro.

Keywords
Linked data, ontology development

Inhaltsverzeichnis

1 Einleitung

Vitro ist eine universell einsetzbare Webanwendung mit integriertem Ontologie- und Instanz-Editor und anpassbarem öffentlichen Browsing und wurde ursprünglich an der Cornell University entwickelt. Die Software wird als Kernstück des community-basierten Forschungsinformationssystems VIVO verwendet, ist jedoch nicht auf dieses Anwendungsgebiet beschränkt. Vielmehr kann Vitro auch als eigenständige Applikation eingesetzt werden. Durch die generische und universell erweiterbare Wissensbasis kann Vitro jeden beliebigen Gegenstandsbereich abbilden.

Vitro baut auf Linked-Data-Technologien und operiert mit formalisierten Ontologien und Vokabularen nach der Web-Ontology-Language-Spezifikation (OWL, W3C 2012). Vitro ist auch in der Lage, mit dem Simple Knowledge Organisation System (SKOS, Miles et al. 2005) umzugehen.

2 Features

Initial enthält eine Vitro-Instanz keine eigenen Ontologien außer einer internen Ontologie für die Konfiguration und Anzeige von Daten. Für das Befüllen und Anpassen des Datenmodells einer Vitro-Applikation ist im Backend der Ontologie-Editor vorhanden. Der Ontologie-Editor, wie Abbildungen 1 und 2 darstellen, umfasst Tools für die Anzeige, das Erstellen, Hochladen, Verändern und Exportieren von Ontologien.

Darüber hinaus bietet Vitro Instrumente für die Verwaltung von Klassen und Beziehungen (Object Properties) und Attribute (Datatype Properties). So können Hierarchien und Gruppierungen von Klassen, Beziehungen und Attributen erstellt, angezeigt und verändert werden.

Die Waterloosäule, gezeigt in Abbildung 3, ist beispielsweise eine Instanz der Klasse „Bauwerk oder Denkmal“ aus der Gemeinsame Normdatei (GND) Ontologie (Haffner 2015). Diese Instanzen – reale Objekte einer Klasse – können entweder in der Anwendung angelegt oder über verschiedene Mechanismen in Vitro importiert werden.

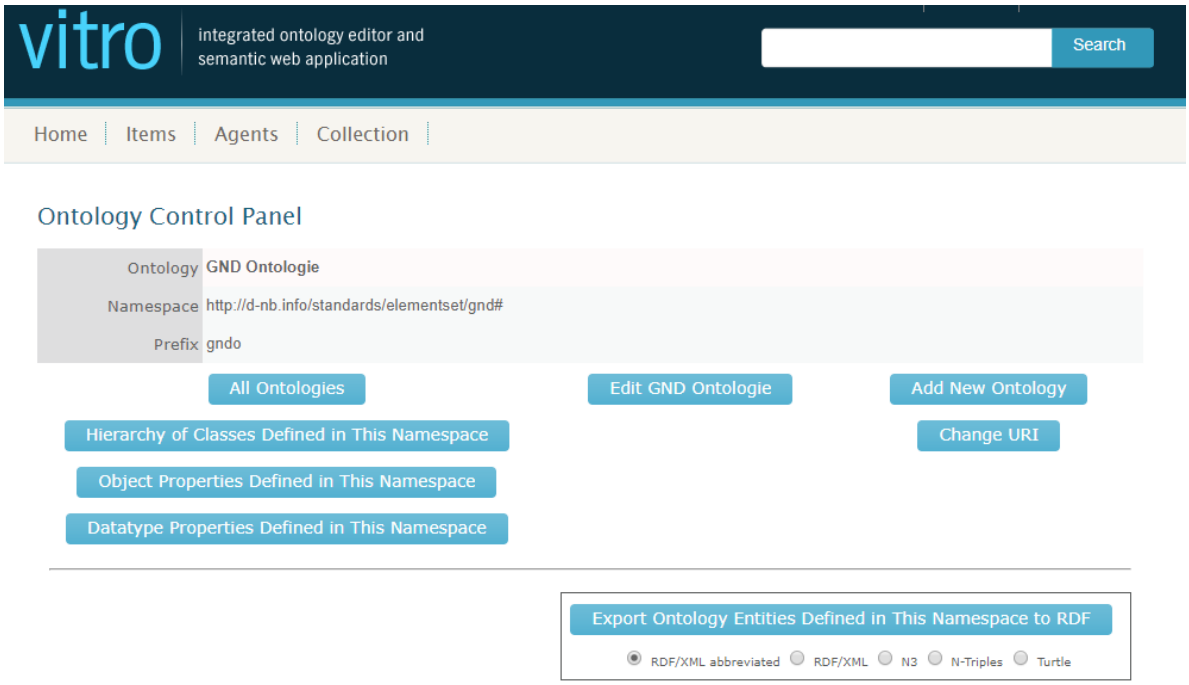

Abb. 1: GND-Ontologie im Ontologie-Editor

Jede Instanz bekommt dabei eine Profilseite und einen eindeutigen Identifikator – Unified Resource Identifier (URI). Die erfassten Daten werden via Apache Solr durchsuchbar gemacht. Durch die Darstellung von Informationen in Profilen ist Websichtbarkeit und -suchbarkeit für die ganze Vitro-Applikation als Web-Portal oder auch für einzelne Profilseiten gegeben.

Neben den Metadaten kann ein Profil mit einer Abbildung ausgestattet werden. Zurzeit müssen Abbildungen den Standard-Formaten – Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF) oder Portable Network Graphics (PNG) – und vorgegebenen Größen entsprechen.

Jedes Objekt in Vitro ist mit RDF-Triples beschrieben, also mit Aussagen oder Fakten. Die OWL-Ontologien für die Beschreibung dieser Objekte können bestimmte Regel enthalten, so genannte Axiome. Fakten und Axiome werden von einer in Vitro integrierten Reasoning-Komponente gelesen und interpretiert. Dazu wird der Reasoner Fact++ (chttp://owl.man.ac.uk/factplusplus/) eingesetzt, einer Java-Implementierung von Fact++ (Tsarkov & Horrocks 2006). Durch Interpretationen und Schlussfolgerungen (Inferencing oder Reasoning) entstehen zusätzliche Triples. Auf diese Weise wird in Vitro implizites Wissen explizit gemacht. So ist zum Beispiel die Klasse *gndo:BuildingOrMemorial* der Klasse *gndo:PlaceOrGeographicName* untergeordnet. Legt man nun eine Instanz der Klasse *gndo:BuildingOrMemorial* an, wird diese durch das Reasoning automatisch auch der übergeordneten Klasse zugeordnet.

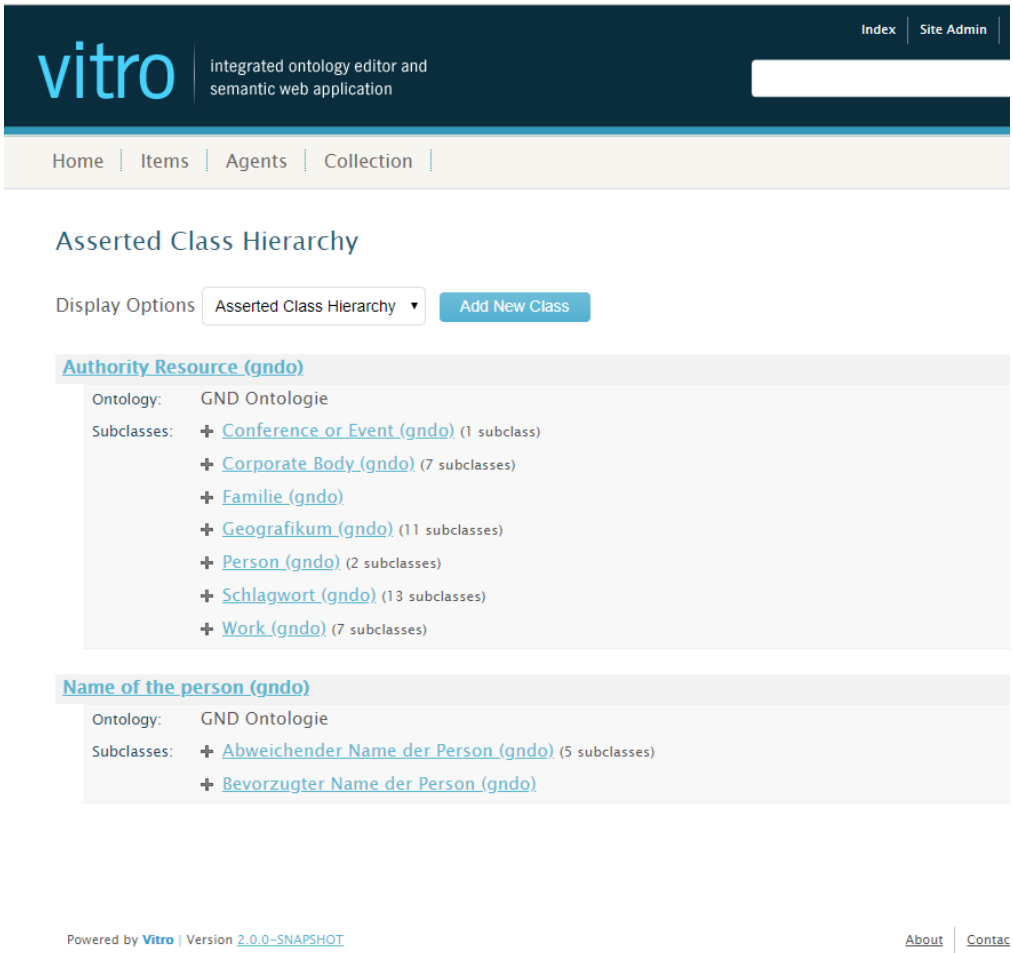

Abb. 2: Klassen aus der GND-Ontologie im Ontologie-Editor

3 Architektur

Vitro ist eine native Triplestore-Anwendung. Alle Ontologien sowie die gesamte Konfiguration, wie die Anwendung mit Eigenschaften, Benutzerkonten usw. umgehen soll, werden in einem von zwei verwendeten Triplestores gespeichert.

Vitro ist in Java geschrieben und nutzt Apache-Jena-Bibliotheken (https://jena.apache.org/) für die Interaktion mit den Triple-Stores. Standardmäßig verwendet es native Jena-Triple-Stores: TDB (Dateisystem) für die Konfigurationselemente und SDB (MySQL) für den Content Store. Der Content Store kann jedoch so konfiguriert werden, dass er TDB verwendet oder über SPARQL-Schnittstellen (W3C 2013) auf einen externen Triple-Store zugreift. Der Hauptzugriff auf den Content Store erfolgt daher über SPARQL-Abfragen. Jena-Modell-APIs werden nur für In-Memory-Inhalte verwendet – z.B. für Daten, die aus dem Triple-Store konstruiert oder aus Formularen erstellt wurden.

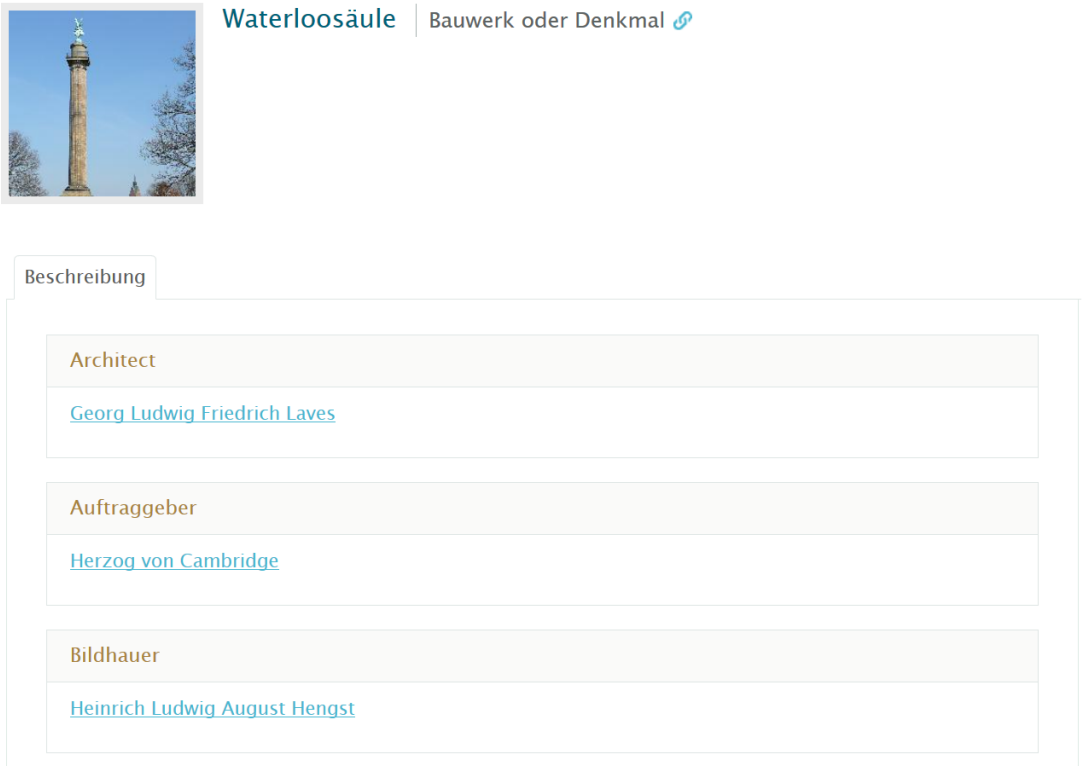

Abb. 3: Beispiel einer Profilseite in Vitro für die Beschreibung der Waterloosäule in Hannover

Die Benutzeroberfläche besteht hauptsächlich aus Freemarker-Templates (http://freemarker.sourceforge.net/), die über Makros auf Daten aus Jena Models zugreifen. Als universell einsetzbare Linked-Data-Anwendung wird der Inhalt und das Layout der Seiten durch Aussagen in diesen Jena-Modellen bestimmt. Dabei lässt sich unter anderem konfigurieren, ob bestimmte Properties gruppiert werden sollen und in welcher Reihenfolge sie angezeigt werden sollen.

<http://d-nb.info/standards/elementset/gnd#BuildingOrMemorial>
vitro:displayLimitAnnot "-1"^^xsd:int ;
vitro:displayRankAnnot "4"^^xsd:int ;
vitro:inClassGroup <http://vivoweb.org/ontology#vitroClassGroupitems> .

Das oben aufgeführte Beispiel zeigt einen Ausschnitt der Konfiguration für die Klasse *gndo:BuildingOrMemorial* aus der GND-Ontologie mit Hilfe von Vitro-Annotationen. Die Properties *vitro:displayLimitAnnot* und *vitro:displayRankAnnot* regulieren die Anzeige und Reihenfolge der Anzeige von Instanzen dieser Klasse. Durch das Attribut *vitro:inClassGroup* werden die Instanzen der Klasse innerhalb der angegebenen Klassen-Gruppe „Items“ angezeigt. Über diesen Mechanismus wird die gemeinsame Darstellung dieser Items in den systemeigenen Such- und Browse-Funktionen realisiert.

Um Anwendungen aus einer bestimmten Domäne zu unterstützen (wie zum Beispiel VIVO, siehe unten), können Properties so konfiguriert werden, dass eine SPARQL-Abfrage verwendet wird, um

mehr Daten zu erhalten und dazu ein benutzerdefiniertes Freemarker-Template, um diese anzuzeigen.
Zusätzlich werden die Datensätze in Apache Solr (http://lucene.apache.org/solr/) indiziert. Solr wird in Vitro als Grundlage für die anwendungsinterne Suche verwendet.

4 Vitro in der Praxis

4.1 Vitro als Fundament für VIVO

VIVO (http://vivoweb.org/) , vgl. Krafft et al. 2010) ist eine community-basierte Open-Source-Software zur Erfassung und Darstellung von Forschungsinformationen. VIVO basiert dabei vollständig auf Vitro, dem verschiedene, auf den Bereich Forschungsinformationen zugeschnittene Funktionen hinzugefügt werden. Dies umfasst unter anderem spezielle Eingabemasken zur Erfassung von Publikationen oder akademischen Institutionen, aber auch Visualisierungen zur Darstellung des Publikationsoutputs von Organisationen oder für Koautorschaften. Dazu gibt es eine Bandbreite von Anwendungen, die zum Beispiel zum Import von Daten aus Fachdatenbanken oder anderem dienen. Inhaltlicher Kern von VIVO ist die VIVO-ISF-Ontologie (https://github.com/openrif/vivo-isf-ontology), mit der die akademische Welt dargestellt wird.

4.2 DataStaR

Im Projekt DataStaR (Data Staging Repository) sollte eine Plattform entwickelt werden, die es Forschenden erlaubt, Forschungsdaten und minimale Metadaten hochzuladen und mit Kollegen zu tauschen. Optional sollte die Möglichkeit geschaffen werden, detaillierte Metadaten zu erfassen und die Forschungsdaten an externe Forschungsdatenrepositories zu übertragen. Die Veröffentlichung und Nutzung von Forschungsdaten sollte vereinfacht, die Erstellung und Pflege der dazugehörigen Metadaten dabei ohne zusätzlichen Aufwand ermöglicht werden.

Um dies zu realisieren, wurde Vitro angepasst und mit OWL-Ontologien ausgestattet, um Beziehungen zwischen Datasets, Personen und Organisationen herzustellen. Eingabeformulare für die Metadaten wurden auf Basis dieser Ontologien erstellt. Die dazugehörigen Datensätze wurden in einem Fedora-Repository gespeichert (Khan et al. 2011). Das Projekt wurde zu einem späteren Zeitpunkt aufgrund eines zu hohen Implementierungsaufwandes in Bezug auf die erforderlichen Ontologien beendet (Wright et al. 2013).

4.3 VitroLib

Als Beispiel für den Einsatz von Vitro im bibliothekarischen Bereich sind die Projekte Linked Data For Libraries Labs (LD4L Labs) und Linked Data For Production (LD4P) zu nennen. Im Rahmen dieser Projekte wurde VitroLib – ein Tool, dessen Basis Vitro bildet – für Katalogisierung von bibliothekarischen Metadaten als Linked Data entwickelt. Dabei erweitert VitroLib die Funktionalität von Vitro, indem es die Inhaltsanzeige und -bearbeitung, basierend auf dem Bibliographic Framework Initiative Datenmodell (BIBFRAME) (http://bibframe.org/), dessen Erweiterung Bibliotek-o und verwandten Ontologien ermöglicht (Khan, Rayle & Younes 2017). Abbildung 4 zeigt die für die Katalogisierung nach Bibliotek-o mit VitroLib zusätzlich programmierten Eingabeformulare.

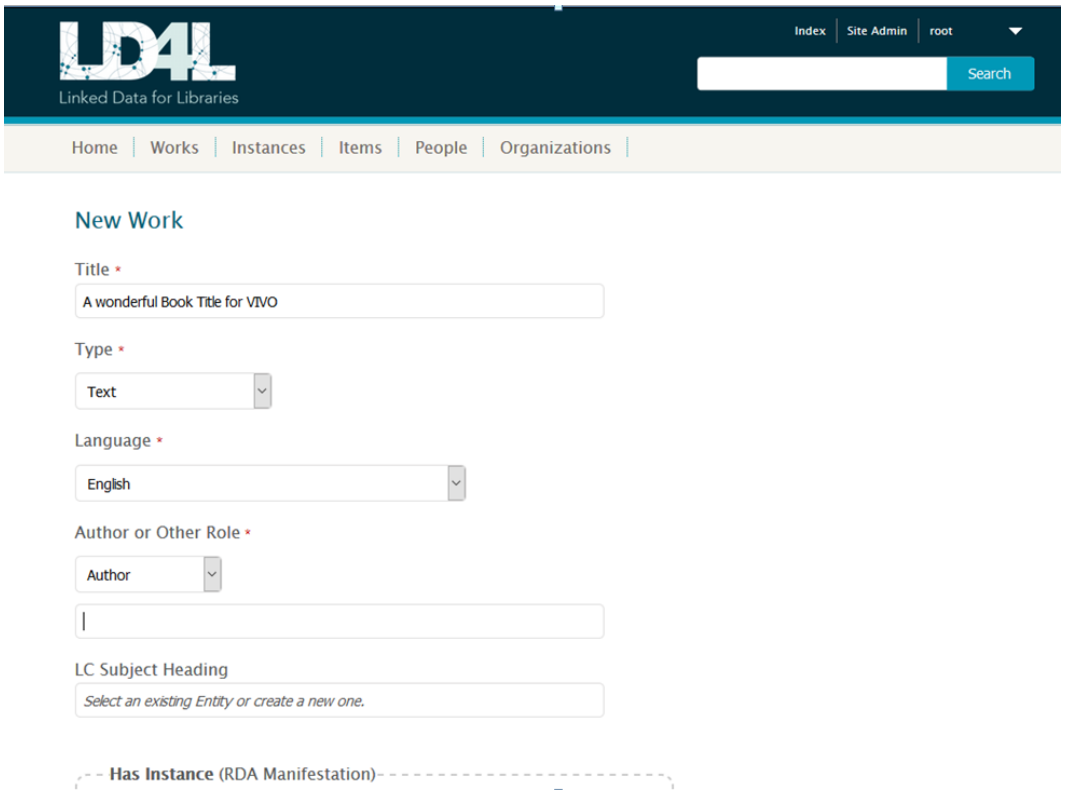

Abb. 4: Erfassung von bibliothekarischen Daten mit VitroLib

Neben der Standard-Erfassung der bibliothekarischen Metadaten bietet VitroLib auf Vitro basierende Lookup-Komponenten für externe Vokabularen und Normdaten für die inhaltliche Erschließung und Anreicherung von Daten.

5 Ausblick

Um Vitro an neue Herausforderungen anzupassen, wurde an der TIB das Advanced Role Management (ARM) entwickelt. Das ARM bietet die Möglichkeit, granulare Rechte für selbst zu definierende Rollen festzulegen. Damit lassen sich nun zum Beispiel Redaktionsgruppen für bestimmte inhaltliche Bereiche einrichten, was die Flexibilität des Systems deutlich erhöht.

Weiterhin wird derzeit am User Action Tracking (UAT) gearbeitet. Das UAT erlaubt die Aufzeichnung einzelner über die Benutzeroberfläche vorgenommener Änderungen an Daten. Diese Änderungen können dem ändernden Benutzer zugeordnet werden. Das UAT erlaubt es, die Provenienz von Daten aufzuzeichnen und sie anschließend weiterzuverarbeiten.

Ein wichtiges Feature, an dem seitens der TIB zur Zeit gearbeitet wird, ist das sogenannte Vitro Query Tool (VQT, Arbeitstitel). Das VQT soll die Möglichkeiten der bisherigen SPARQL-Queries erweitern. Zwar sind dort auch SPARQL-Queries vorgesehen, jedoch sollen diese gespeichert, importiert und exportiert werden können. Dies erleichtert nicht nur die institutionsinterne Verwaltung von Queries, auch der Austausch zwischen verschiedenen Einrichtungen wird ermöglicht. Dazu soll es die Möglichkeit geben, wiederholbare und terminierte Anfragen zu

gestalten. Ziel ist es, Abfragen erstellen zu können, deren Ergebnisse in regelmäßigen Abständen und in verschiedenen Ausgabeformaten per Mail an bestimmte Nutzer geschickt werden können. Dies ermöglicht die Etablierung einfacher Workflows, indem z.B. wöchentlich ein Report über alle Entitäten, die bestimmte Bedingungen erfüllen, an die Mitglieder einer Redaktion verschickt werden kann.

Für verschiedene Anwendungsgebiete sind viele weitere Anpassungen denkbar. So könnten zum Beispiel Tools für das Anzeigen hochauflösender Bilder in Vitro integriert werden. Denkbar wäre ein IIIF-Viewer (International Image Interoperability Framework, siehe Snydman, Sanderson & Cramer 2015). Dies würde das Erfassen von Digitalisaten in Vitro ermöglichen, zum Beispiel von Musealien.

Aus unserer Sicht ist das Potenzial von Vitro längst nicht ausgeschöpft. Überall, wo es Erfassung und Repräsentation von Metadaten stattfindet, kann Vitro genutzt werden. Wir sehen den zukünftigen Einsatz von Vitro unter anderem in musealen oder bibliothekarischen Bereichen, wo die Anwendung neben der einfachen Erfassung auch die semantische Verknüpfung und web-gerechte Veröffentlichung von Objekten als Linked (Open) Data im Web gewährleisten kann und auf diese Weise wertvolle Bestände zugänglich gemacht werden können. Auch für den Aufbau prototypischer Anwendungen im Linked-Data-Bereich ist Vitro durch seine Flexibilität sehr gut geeignet.

Literatur

Engel, Felix, Schlager, Stefan & Wittwer-Backofen, Ursula 2015. An Infrastructure for Digital Standardisation in Physical Anthropology: 11. Internationaler Kongress der Deutschen Gesellschaft

für Anthropologie e.V.: Evolutionäre und moderne Herausforderungen für Homo sapiens - eine anthropologische Spurensuche.

Haffner, Alexander 2015. GND Ontology: Namespace Document. Deutsche Nationalbibliothek. URL: https://d-nb.info/standards/elementset/gnd.

Khan, Huda, u.a. 2011. DataStaR: Using the Semantic Web approach for Data Curation. IJDC 6(2), 209–221. DOI: https://doi.org/10.2218/ijdc.v6i2.197.

Khan, Huda, Rayle, Lynette & Younes, Rebecca 2017. VitroLib: From an ontology and instance editor to a linked data cataloging editor. International Conference on Dublin Core and Metadata Applications 0, 91. Online im Internet: URL: http://dcpapers.dublincore.org/pubs/article/download/3873/2058.

Khan, Huda 2017. When Vitro goes rogue: The VitroLib application for library cataloging using linked data: VIVO Conference 2017. DOI: https://doi.org/10.6084/m9.figshare.5277187.v1

Krafft, Dean B., u.a. 2010. VIVO: Enabling National Networking of Scientists: Proceedings of the WebSci10: Extending the Frontiers of Society On-Line, 1310-1313.

Lowe, Brian, u.a. 2011. The Vitro Integrated Ontology Editor and Semantic Web Application, in Bodenreider, Olivier, Martone, Maryann E. & Ruttenberg, Alan (Hg.): ICBO-2011: International Conference on Biomedical Ontology: Proceedings of the 2nd International Conference on Biomedical Ontology: CEUR. (CEUR Workshop Proceedings, 833). Online im Internet: URL: http://ceur-ws.org/Vol-833/paper54.pdf .

Miles, Alistair, u.a. 2005. SKOS Core: Simple knowledge organisation for the Web. International Conference on Dublin Core and Metadata Applications, 3–10. Online im Internet: URL: http://dcpapers.dublincore.org/pubs/article/download/798/794.

Snydman, Stuart, Sanderson, Robert & Cramer, Tom 2015. The International Image Interoperability Framework (IIIF): A community & technology approach for web-based images: Archiving 2015: Final Program and Proceedings: Society for Imaging Science and Technology, 16–21.

Tsarkov, Dmitry & Horrocks, Ian 2006. FaCT++ Description Logic Reasoner: System Description, in Furbach, Ulrich (Hg.): Automated reasoning: Third international joint conference, IJCAR 2006, Seattle, WA, USA, August 17 - 20, 2006 ; proceedings. Berlin: Springer. (Lecture notes in computer science Lecture notes in artificial intelligence, 4130), 292–297. DOI: https://doi.org/10.1007/11814771_26.

W3C 2012. OWL 2 Web Ontology Language Document Overview: W3C Recommendation. Second Edition. URL: https://www.w3.org/TR/owl2-overview/ [Stand 2018-07-06].

W3C 2013. SPARQL 1.1 Overview: W3C Recommendation: World Wide Web Consortium. URL: https://www.w3.org/TR/sparql11-overview/.

Wright, Sarah J., u.a. 2013. Using Data Curation Profiles to Design the Datastar Dataset Registry. D-Lib Magazine 19(7/8). DOI: https://doi.org/10.1045/july2013-wright.

AutorInnen

Christian HAUSCHKE
Technische Informationsbibliothek (TIB)
Welfengarten 1B
30167 Hannover
https://www.tib.eu
christian.hauschke@tib.eu

Tatiana WALTHER
Technische Informationsbibliothek (TIB)
Welfengarten 1B
30167 Hannover
https://www.tib.eu
tatiana.walther@tib.eu

Graham TRIGGS
Technische Informationsbibliothek (TIB)
Welfengarten 1B
30167 Hannover
https://www.tib.eu
graham.triggs@tib.eu