PROCEEDINGS OF TDWG

Biodiversity Information Science and Standards 1: e20379
doi: 10.3897/tdwgproceedings.1.20379

OPEN ACCESS

Conference Abstract

Fitness-for-Use-Framework-aware Data Quality workflows in Kurator

Paul J. Morris[‡], James Hanken[‡], David B. Lowery[‡,§], Bertram Ludäscher[|], James Macklin[¶], Timothy McPhillips[|], Robert A. Morris[§,‡], John Wieczorek[‡], Qian Zhang[|]

‡ Museum of Comparative Zoology, Harvard University, Cambridge, United States of America
§ University of Massachusetts, Boston, Boston, United States of America
| University of Illinois Urbana-Champaign, Champaign, United States of America
¶ Agriculture and Agri-Food Canada, Ottawa, Canada

Corresponding author: Paul J. Morris (mole@morris.net)

Received: 16 Aug 2017 | Published: 16 Aug 2017

Citation: Morris P, Hanken J, Lowery D, Ludäscher B, Macklin J, McPhillips T, Morris R, Wieczorek J, Zhang Q (2017) Fitness-for-Use-Framework-aware Data Quality workflows in Kurator. Proceedings of TDWG 1: e20379. https://doi.org/10.3897/tdwgproceedings.1.20379

Abstract

In the Kurator project, we are developing libraries of small modules, each designed to address a particular data quality test. These libraries, which can be run on single computers or scalable architecture, can be incorporated into data management processes in the form of customizable data quality scripts. A script composed of these modules can be incorporated into other software, run as command-line programs, or provided as a suite of “canned” workflows through a web interface.

In some of these modules, we have implemented a subset of the standard tests under development by Task Group 2 (TG2) of the Data Quality Interest Group. We have also been exploring use of the fitness-for-use-framework (Veiga et al. 2017) produced by Task Group 1 (TG1) of the Data Quality Interest Group. Our goals have been to explore use of the framework to describe capabilities of atomic modules of code, how we can use concepts in the framework to produce data quality reports, and what lessons can be learned from implementing data quality control code in the context of the framework. We have focused on the Data Quality Reports level of the framework; in particular, the representation of Data Quality Measures (measurements on some data quality dimension),

Validations (tests for compliance with quality needs), and Amendments (proposals to improve data quality).

At the implementation level, we have developed a set of Java annotations to mark methods as providing specific tests from the test suite. In terms of the framework, the annotations can also be used to mark methods as providing Measures, Validations, or Amendments and to associate method parameters with Information Elements by linking them to the Darwin Core terms that were either "acted upon" or "consulted" (Lowery et al. 2016).

These annotations can be used by a consumer to identify and run Measures and Validations in two phases: a Pre-Amendment phase, before the Amendments are run; and a Post-Amendment phase, after the changes proposed by the Amendments have been applied. Capturing the test results across both stages allows us to report on how much accepting the amendments would improve the quality of the dataset as a whole, data in some quality dimension, or data for some specific purpose.

We have found it important to be able to render data quality reports that identify which Darwin Core terms are the Valuable Information elements involved in a specific test, and, further, to identify which terms are acted upon and which are consulted. Identifying this information allows us, for example, to render tabular reports highlighting cells where amendments have proposed a change.

We have also found reporting of error and failure conditions to be important, and have been working on implementing the TG1 suggestion that report elements consist of a result (containing only appropriate values), status (containing a controlled vocabulary term such as completed, or data_prerequisites_not_met), and a human readable message (metadata about why the conclusion that was drawn was drawn, or error messages).

We have developed a stake-in-the-ground vocabulary for status values to describe failure conditions including the following concepts: Ambiguous (there was a result, but it has ambiguity, e.g., an event date inferred from the verbatim event date 04/05/1954), Internal Prerequisites Not Met (not able to run the test on the data provided, e.g., day was not an integer), and External Prerequisites Not Met (some external resource that this test consults was unavailable at runtime).

In implementing tests in the context of the framework, we have seen the value of identifying Measures, Validations, and Amendments in forcing us to develop small, focused tests, and in allowing us group assertions within data quality reports based on data quality needs.

Keywords

Biodiversity Informatics, Workflows, Data Quality

Presenting author

Paul J. Morris

Funding program

NSF DBI 1356438 and 1356751

References

- Lowery D, Morris P, Veiga AK (2016) Kurator-Org/Kurator-Ffdq: Initial Release Of Kurator-Ffdq Library Version 1.0.0. Zenodo https://doi.org/10.5281/ZENODO.192186
- Veiga AK, Saraiva AM, Chapman AD, Morris PJ, Gendreau C, Schigel D, Robertson TJ (2017) A conceptual framework for quality assessment and management of biodiversity data. PLOS ONE 12 (6): e0178731. https://doi.org/10.1371/journal.pone.0178731