Towards Automated Workflow Design and Analysis of Declarative Curation Workflows

Tianhong Song¹ Sven Köhler¹ Bertram Ludäscher*¹
James Hanken² Maureen Kelly² David Lowery²
James Macklin⁴ Paul Morris³ Robert Morris³,⁵

¹ Dept. of Computer Science, UC Davis, CA, USA *contact: ludaesch@ucdavis.edu
² Museum of Comparative Zoology, Harvard University, MA, USA
³ Harvard University Herbaria, 22 Divinity Avenue, Cambridge, MA, USA
⁴ Agriculture and Agri-Food Canada, Ottawa, Ontario, Canada
⁵ Computer Science Dept, University of Massachusetts, Boston, MA, USA
The Problem: Data Quality (here: in Biodiversity Data)

- **Collections & occurrence data** is “all over the map”
  - … literally *(off the map!)*

- **Issues**, e.g.:
  - Lat/Long transposition, coordinate & projection issues
  - Data entry/creation, “fuzzy” data, naming issues, bit rot, data conversions and transformations, schema mappings, … *(you name it!)*

- **Filtered-Push Collaboration**
Data Curation Workflows in Filtered Push

- **Curator, Collection-Manager:**
  - clean your local collections DB
  - “pull experts” into the workflow

- **Researcher (data consumer):**
  - subscribe to relevant streams
  - filter out your local collections DB

- **Researcher (QC/QI):**
  - clean community DBs by pushing annotations back to the source
Scientific Workflow Aims: ASAP!

- **Automation**
  - wfs to *automate* computational aspects of science
  - batch processing, scripting

- **Scaling** *(optimize machine cycles)*
  - wfs should make use of *parallel compute resources*
    - dataflow-orientation avoids von Neumann bottleneck
    - exploit parallel MoCs when deploying on cluster, cloud
  - wfs should be able handle *large data*

- **Abstraction, Evolution, Reuse** *(optimize human cycles)*
  - wfs should be easy to *change, evolve, share, reuse*

- **Provenance**
  - wfs should capture *processing history, data lineage*
    - traceable data- and wf-evolution
  - Reproducible Science
Workflow and Data Curation

- Curation:
  - actively manage, maintain long-term value, reuse, preserve

- (Semi-)automated Curation Workflows
  - automate curation pipeline with existing components
  - keep track of the data processing history
  - modify and rerun curation steps
  - reuse existing workflows
  - share workflow with other scientists
Filtered-Push Kuration (Kepler+Curation) Package
Provenance Browser: Replaying workflow execution, analyzing data & process dependencies

1. Execute curation workflow

2. Step forward & backward in the provenance trace
Scientific/Curation Workflow Challenges

- **Scalability** ("machine cycles")
  - remote services and large collections
  - even test runs can be time-consuming, incomplete

- **Workflow design** ("human cycles")
  - time-consuming and error-prone!

- Deep knowledge of actors and underlying system are required to develop efficient workflow
- Apply automatic analysis & design methods
- Workflow design should be more like story-telling! (and less like programming)
Automated Design

- User provides **workflow** or
- User provides input **data schema** and data items in need of quality control

The system will
- **select** relevant curation actors, then
- **assemble** a workflow that covers the to-be-checked data items
Static (= “compile-time”) Analysis

- Analysis is done before a workflow run and anticipates how a workflow might behave during runtime.

- The system informs designer where the problems are and suggests ways to fix them
Examples: Detect Design Errors

(a) Scientific Name Validator “starves” due to config-error (typo)

(b) … due to design-error (2\textsuperscript{nd} actor works on validated data)
Example: Exploit Parallelism

(Scientific-Name-Validator ; Flowering-Time-Validator) || Event-Date-Validator
Prototypical Implementation

- **Answer Set Programming** (DLV, Potassco) ➔ workflow design and analysis system
- **Akka** ([http://akka.io](http://akka.io)) ➔ parallel execution platform
Future work

- Improved analysis & optimization techniques
  - exploiting provenance during workflow life cycle
- Integrate with existing FilteredPush System.
References


- http://kepler_project.org

- Morris, P.J., et al. 2013. Improving Natural Science Collections data through quality control for research using Kepler workflows embedded in a FilteredPush network. SPNHC:

- Song, Tianhong et al. Towards Automated Workflow Design and Analysis of Declarative Curation Workflows. under revision