A Laboratory Notebook System

EuroPython 2012 (05.07.2012, Florence, Italy)

Andreas Schreiber <Andreas.Schreiber@dlr.de>
German Aerospace Center (DLR)
Overview

- Background
  - Good Laboratory Practice
  - Scientific Workflows
  - Laboratory Notebooks
  - DataFinder

- DataFinder-based Laboratory Notebook
  - Data model
  - Process documentation
  - Evidential preservation
  - Signing data

- Future Work
Background
Background
Good Laboratory Practice

The principles of Good Laboratory Practice (GLP) have been developed to promote the quality and validity of test data used for determining the safety of chemicals and chemicals products.

OECD Principles on Good Laboratory Practice (as revised in 1997)

[The recommendations] are designed to provide a framework for the deliberations and measures which each institution will have to conduct for itself according to its constitution and its mission.

Deutsche Forschungsgemeinschaft:
Sicherung guter wissenschaftlicher Praxis (Safeguarding good scientific practice) 1998 (p.50).
Background
Scientific Workflow

1. Planning, Design: Literature, Data of other projects
2. Execution: Apparatus, Computer
3. Evaluation: Computer, (self developed) Software
4. Interpretation, Publication
5. Archiving: subsequent use, display of research results

Picture adapted from: www.belab-forschung.de
Background
Laboratory Notebooks

“The laboratory notebook is the diary of the experimenting scientist“

(Schreiben und Publizieren in den Naturwissenschaften
Von Hans F. Ebel, Claus Bliefert, Walter Greulich; chapter 1.3 - page 16)
Background

DataFinder

- Data management system: DataFinder
  - Developed by DLR
  - Open Source Project (BSD License)
  - Implemented in Python
  - Data management and work flow management
  - Supports meta data handling
DataFinder User Interface

File System

Shared Data Repository

Output
DataFinder – Connected to Repository
DataFinder
Structuring Data

- Structuring of data in a standardized way through a data model
- Restricting the user to a layout
- Forcing the user to enter meta data
DataFinder
Heterogeneous Storage Resources

- Using heterogeneous storage backend for data
  - Best fitting storage solution depending on data
  - Existing solutions can be kept
  - Using offline storage is possible
DataFinder
Script Extensions

- DataFinder is extendable by Python scripts
  - Integration with existing environment
  - Automation of data processing steps
DataFinder-based Laboratory Notebook
# Laboratory Notebook

## Requirements for Good Scientific Documentation

<table>
<thead>
<tr>
<th>Requirements:</th>
<th>Realization:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Data structure</td>
<td>- Data model</td>
</tr>
<tr>
<td>- Traceability</td>
<td>- Process documentation</td>
</tr>
<tr>
<td>- Durability</td>
<td>- Evidential preservation</td>
</tr>
<tr>
<td>- Credibility</td>
<td>- Signing data</td>
</tr>
</tbody>
</table>
Realization

Data Model

- Meta data type = preparation
  - Start date = Date
  - End date = Date

- Study XYZ

- User

- Preparation
  - Meta data
    - Type = input
    - Signature = Hash
    - .txt
    - .pdf
    - .c
    - .doc

- Study Plan
  - Standard Procedure

- Experiment
  - Preparation
    - Environmental parameter
    - .cpp
    - .xls
    - .xml
  - Calibration Data
    - evaluated Data
  - Interpretation
    - Notes
    - Interpreted data
  - Study Report
  - Archiving

- Archiving
  - Achieves
Realization
Process Documentation

- Process documentation: Recording the *Provenance* of that process

- Provenance (lat. provenire = to come from): origin of data, source

- Provenance of process gives *traceability* and *credibility*

- Steps to add Provenance recording to software (i.e., DataFinder)

  1. Developing a provenance model for the „Good Laboratory Practice“

  2. Provide Provenance storing system

  3. Integration into DataFinder
Process Documentation

Provenance Data Model

- Apply methodology to define a Provenance model
- Representation of the real world’s process
Process Documentation
Provenance Data Model

User

Controlled by

Name (M)

Preparation

Generated by

Derived from

Manuals

- identifier (M)
- version (M)
- signature (O)

Derived from

used

Standard Procedure

- identifier (M)
- version (M)
- signature (O)

Derived from

used

Study Plan

- identifier (M)
- version (M)
- signature (O)
Process Documentation
Provenance Storing System

Provenance Store *prO0st*
- Java Implementation
- Server: Jetty
- Graph Database: Neo4j

- Interfaces
  - Storing Provenance (REST)
  - Extracting Provenance (REST)
  - Extracting Provenance (Servlet)

- Open Source (Apache License 2.0)
  - https://proost.sourceforge.net
Process Documentation
Integration Into DataFinder

- User actions on files are recorded in the provenance store
- Dialog for asking additional questions
Realization
Evidential Preservation

„Recommendation 7: Primary data as the basis for publications shall be securely stored for ten years in a durable form in the institution of their origin.“

Deutsche Forschungsgemeinschaft: Sicherung guter wissenschaftlicher Praxis (Safeguarding good scientific practice) 1998 (p.55).

- Steps to add evidential preservation to software (i.e., DataFinder)

1. Create an archive with all relevant data (e.g., for a publication)

2. Integration of a preservation service
Evidential Preservation
Create an Archive With All Relevant Data

Extraction of data relevant for the preservation process
Evidential Preservation
Create an Archive With All Relevant Data

In DataFinder

- User chooses report (publication etc.)
- Python script queries relevant files from the Provenance store
- Relevant files are added to an archive
- Archive is stored in DataFinder
Evidential Preservation
Integration of a Preservation Service

We use the BeLab service (Beweissicheres Laborbuch Project)

- DFG Project (http://www.belab-forschung.de):
  - Physikalisches Bundesanstalt Braunschweig
  - Karlsruher Institute of Technology
  - Universität Kassel

- The BeLab service
  - characterizes the preservation time of an item
  - characterizes the legal trustworthiness of an item
  - stores the archive securely
Evidential Preservation
Integration of a Preservation Service

In DataFinder

- User chooses an archive and activates script
- Script sends the archive to BeLab service via WS-Secure
- The service processes the archive
- Service returns preservation information, which is stored
Realization
Signing Data

- Authenticity in general

- Attesting authentication

- Steps to add data signing to software (i.e., DataFinder)

1. Concept:
   - Signing files: signature stored as meta meta item
   - Meta data: Extraction as XML file, then signed

2. Integration into DataFinder
Signing Data Integration Into DataFinder

Signature of the data (files) as separate file

- User chooses a file and executes script
- A signature file is generated (PKCS #7)
- Signature file is stored in the DataFinder
Future Work
Future Work
Enhanced User Interface

- User interface for taking notes
  - Annotation of data

- Doing calculations and data analysis (similar to MATLAB or Mathematica Notebooks)
  - Integration of The Larch Environment
  - Integration of NumPy/IPython

- Exploring Provenance data
  - Insights and understanding of processes

- Tablet version
  - Entering data
  - Synchronization for offline use
Questions?

Summary
- DataFinder-based Electronic Lab Notebook
- Traceability, Durability, and Credibility for data
- Documentation, evidential preservation, and data signing

Andreas Schreiber
Andreas.Schreiber@dlr.de
http://www.dlr.de/sc