REPORT ON OpenDreamKit DELIVERABLE D4.6

Tools for collaborating on notebooks via version-control

BENJAMIN RAGAN-KELLEY & MARTIN SANDVE ALNÆS & VIDAR FAUSKE

Due on	31/08/2016 (M12)
Delivered on	31/08/2016
Lead	Simula Research Laboratory (Simula)
Progress on and finalization of this deliverable has been tracked publicly at: `https://github.com/OpenDreamKit/OpenDreamKit/issues/95`	

DELIVERABLE DESCRIPTION, AS TAKEN FROM GITHUB ISSUE #95 ON 2016-09-07

- **WP4:** User Interfaces
- **Lead Institution:** Simula Research Laboratory
- **Due:** 2016-08-31 (month 12)
- **Nature:** Other
- **Task:** T4.2 (#70)
- **Proposal**: p.48
- **Final report**

Version control tools, such as Git and Mercurial, have become an integral part of open and collaborative science and software. Version control tools allow proposed changes to be reviewed ('diffing') and resolve conflicts through combination of changes ('merging'). Jupyter notebook documents are stored in text files as JSON formatted data. This makes them well suited to tracking in version control, but the JSON structure can make diffing and merging difficult.

For this deliverable, we have developed the `nbdime` tool for diffing and merging notebook documents with awareness of the structured nature of the documents, allowing a significantly improved experience over naïve text-file diffing and merging tools. `nbdime` can be integrated into the popular git version control system, maximizing impact on common developer workflows.

CONTENTS

1. Collaborating on Jupyter notebooks with nbdime

Version control tools, such as Git and Mercurial, have become an integral part of open, collaborative, and reproducible science and software. Version control tools allow proposed changes to be reviewed ('diffing') and resolve conflicts through combination of changes ('merging'). This is particularly relevant for Jupyter notebooks, which allow scientists to write and share documents that mix live code, equations, visualizations and explanatory text.

Such documents are stored in text files as JSON formatted data which makes them well suited to tracking in version control. However the JSON structure can make diffing and merging difficult. We have developed tools for diffing and merging notebook documents with awareness of the structured nature of the documents, allowing a significantly improved experience over naïve text-file tools. These tools also provide integration with the git version control system.

We have built a new Python package called nbdime, for **N**ote**b**ook **Di**ffing and **Me**rging. nbdime makes use of the nbformat package, part of the Jupyter project. nbdime aims to improve the experience of scientists collaborating on notebooks, in particular addressing difficulties in the diffing and merging stages of collaboration. We submitted the project to Jupyter via the Jupyter Enhancement Proposal process, and it has been accepted as an official Jupyter project.

nbdime is available on GitHub as `https://github.com/jupyter/nbdime`

2. Diffing notebooks

nbdime provides two mechanisms for comparing notebooks. First is a command-line diff of notebooks via the nbdiff command, enabling use in a text-based terminal where developers spend much of their time. The key for diffing notebooks is that not all information in a notebook has equal importance, and not all of it is sensibly viewable as text, such as embedded images. The second important task of nbdime is displaying text content in a more legible form than the raw JSON. JSON includes additional structured markup, such as quotation marks, indentation, and escaped characters, which make the text difficult to read, and appear different from the text a user might see when they are editing the notebook. By recognizing the structure of the notebook, nbdime is able to make intelligent decisions about how to display text without additional markup and deemphasize information that is not viewable in a text-only environment. nbdime includes the structure and metadata with minimal markup, designed for humans to read. When images or other embedded data is seen, it is replaced by an indication that there is binary data that change, rather than the unintelligible difference between the binary data.

nbdime also provides a web-based diff viewer via the nbdiff-web command. Notebooks are most commonly viewed and produced in a web-based environment, with the ability to view rendered images and HTML. The goal of the web-based viewer is to show the reviewer the actual changes in the notebook document, including changes in output images and rendered HTML. This allows the most natural experience of reviewing changes in a document, where the review environment is as similar to the interactive notebook environment as possible.

3. Merging notebooks

One challenge for merging two notebooks using text is the structured JSON information, where text-based merge strategies can result in a syntactically invalid document. This is a problem for merging all structured documents, including source code in programming languages such as C, Python, or LaTeX. However, for tools where the editor is not plain-text, such as notebooks, preventing the creation of these invalid documents is particularly important. Another challenge for merging is the fact that some of the content in notebooks is generated, and thus should be treated differently when considering conflicts between two change sets.

```
      },
      "metadata": {},
      "output_type": "display_data"
@@ -1080,7 +334,10 @@
    "cell_type": "markdown",
    "metadata": {},
    "source": [
-    "When you are directly writing your own classes, you can adapt them f
 can't easily modify. We now illustrate how to add rich output capabilitie
ssion."
+    "When you are writing your own classes, you can adapt them for displa
+    "But in practice, you often need to work with existing classes that y
+    "We now illustrate how to add rich output capabilities to existing ob
+    "We will use the NumPy polynomials and change their default represent
    ]
   },
   {
@@ -1100,7 +357,7 @@
     {
      "data": {
       "text/plain": [
-       "Polynomial([ 1.,  2.,  3.], [-10.,  10.], [-1,  1])"
+       "Polynomial([ 3.,  2.,  1.], [-10.,  10.], [-1,  1])"
       ]
      },
      "execution_count": 9,
@@ -1109,7 +366,7 @@
     }
    ],
    "source": [
-    "p = np.polynomial.Polynomial([1,2,3], [-10, 10])\n",
+    "p = np.polynomial.Polynomial([3, 2, 1], [-10, 10])\n",
     "p"
.■
```

FIGURE 1. Text-based diff of a notebook as JSON, showing additional markup impeding readability.

NBDIME addresses the issue of conflicts with its own merge mechanism, always ensuring that a valid notebook is the result, even if there are conflicts. The second task of NBDIME's merge tool is to allow users to ignore conflicts on certain fields of the notebook. There is a hierarchy of significance in the content of notebooks, and not all changes should be considered conflicting. For instance, if two change sets both alter certain metadata, the changes are not significant and automatic resolution of conflicts can be carried out aggressively by clearing conflicting values. Similarly for generated output, conflicts can be automatically resolved without the consequences of automatic resolution in the source code. It remains possible to be strict about differences in output, especially as it pertains to evaluating the reproducibility of notebooks. In the worst case scenario, a notebook can be re-executed to produce new output.

When merge conflicts do occur, the NBMERGE web tool can be used to view and resolve the conflicting changes one by one.

```
minrk[13:58]~/Dropbox/Documents/Jupyter/pres/scipy2016 $ nbdiff base.ipynb local.ipynb
nbdiff base.ipynb local.ipynb
--- a: base.ipynb
+++ b: local.ipynb

patch a/cells/1/outputs/0/text:
@@ -1 +1 @@
matplotlib-1.5.2matplotlib-2.0.0b2.post1762+gcf24e05
patch a/cells/2/source:
@@ -1,4 +1,5 @@
from numpy.random import uniform, seed
from matplotlib.mlab import griddata
import matplotlib.pyplot as plt
import numpy as np
plt.rc('image', cmap='viridis')
replace at a/cells/3/outputs/0/data/image/png:
- iVBORw0KGgoAAAAN...<snip base64>...AAAAElFTkSuQmCC
+ iVBORw0KGgoAAAAN...<snip base64>...AA
+ AElFTkSuQmCC
patch a/cells/3/outputs/0/data/text/plain:
@@ -1 +1 @@
<matplotlib.figure.Figure at 0x105eeb588>0x1096b8e10>
replace at a/cells/3/outputs/0/metadata/image/png/height:
- 392
+ 382
replace at a/cells/5/outputs/0/data/image/png:
- iVBORw0KGgoAAAAN...<snip base64>...ABJRU5ErkJggg==
+ iVBORw0KGgoAAAAN...<snip base64>...ABJRU5ErkJggg==
patch a/cells/5/outputs/0/data/text/plain:
@@ -1 +1 @@
<matplotlib.figure.Figure at 0x105eeb2e8>0x10a53ef98>
replace at a/cells/5/outputs/0/metadata/image/png/height:
- 376
+ 367
patch a/metadata/kernelspec/display_name:
@@ -1 +1 @@
Python 3(conda:stable)
patch a/metadata/kernelspec/name:
@@ -1 +1 @@
stablepython3
```

FIGURE 2. NBDIFF output, including truncated image changes and clearer changes in source code compared to JSON.

4. VERSION CONTROL INTEGRATION

NBDIME provides integration with the git version control system for its diffing and merging tools. Git provides two mechanisms for integrating external software for performing the diff and merge operations on specific files.

The first is called a ”driver”, and is used internally by git when computing diffs and performing merges. NBDIME can register its command-line diff rendering as a diff driver for notebook files. This can be done as a system-wide configuration, or for individual git repositories. NBDIME also provides a merge driver for auto-resolving conflicts on notebooks, ensuring that a GIT MERGE never results in an invalid notebook document, and that conflicts can be resolved in the user's familiar notebook-editing environment.

The second git integration point is called a ”tool”, e.g. ”difftool” or ”mergetool”. These are like drivers, but allow external applications that may open windows to be launched. NBDIME provides tool entrypoints for diff and merge to launch the web interfaces for viewing diffs and manually resolving merge conflicts.

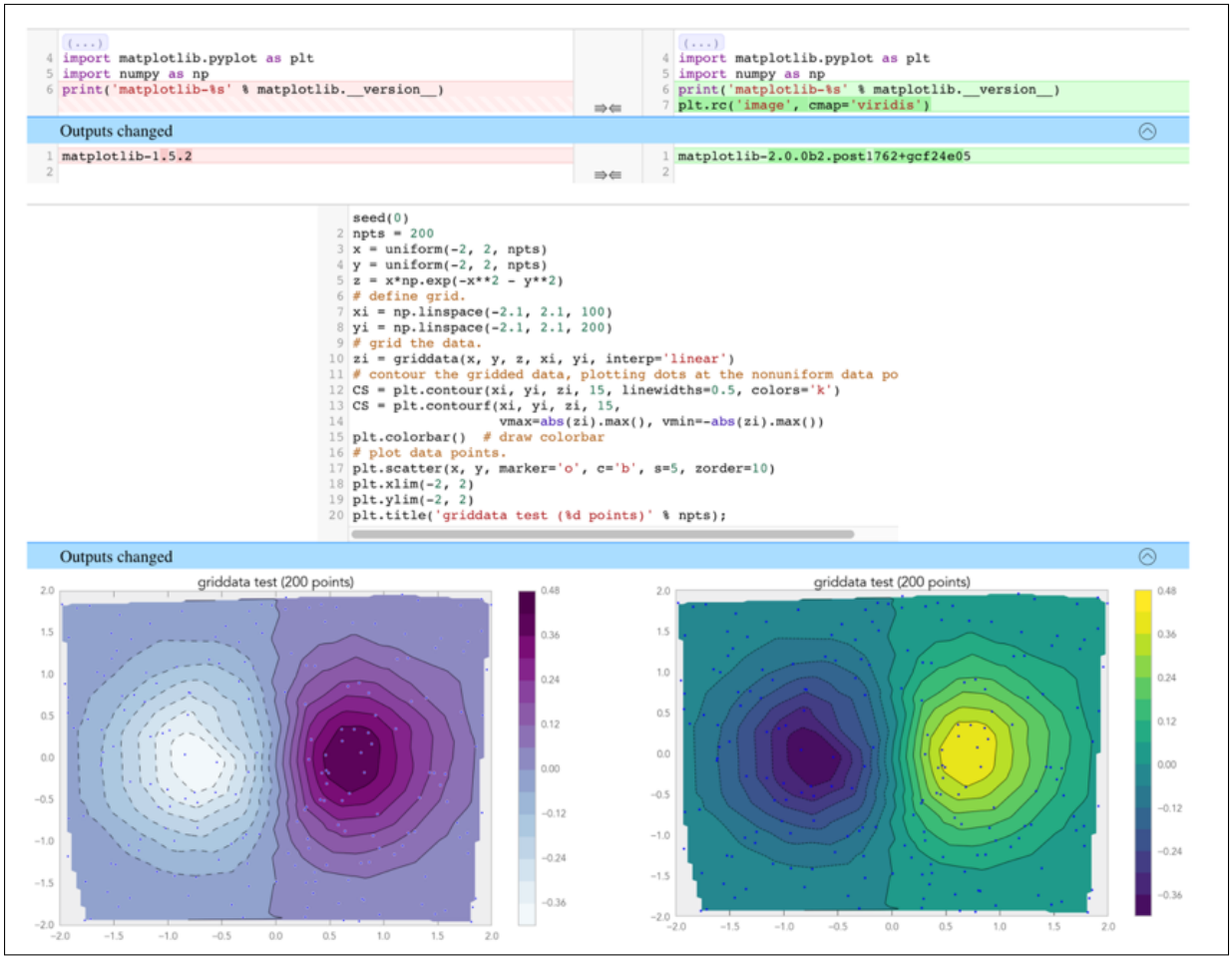

FIGURE 3. nbdiff-web output showing side-by-side comparison of output images.

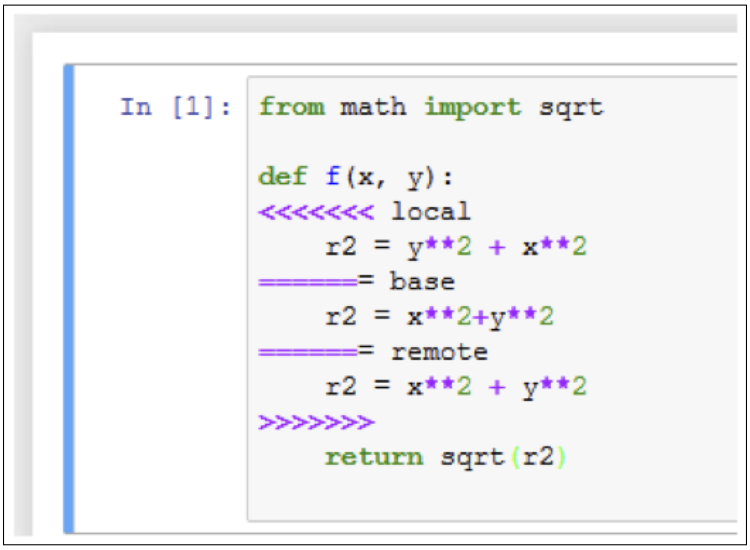

FIGURE 4. The result of NBMERGE with a conflict is still a valid notebook.

5. NBDIME AND REPRODUCIBILITY

Efficiently comparing notebooks and their output benefits the goals of OpenDreamKit, especially as it improves the ability of scientists and developers to follow reproducible workflows. By better integrating with version control systems, best practices in reproducibility are encouraged

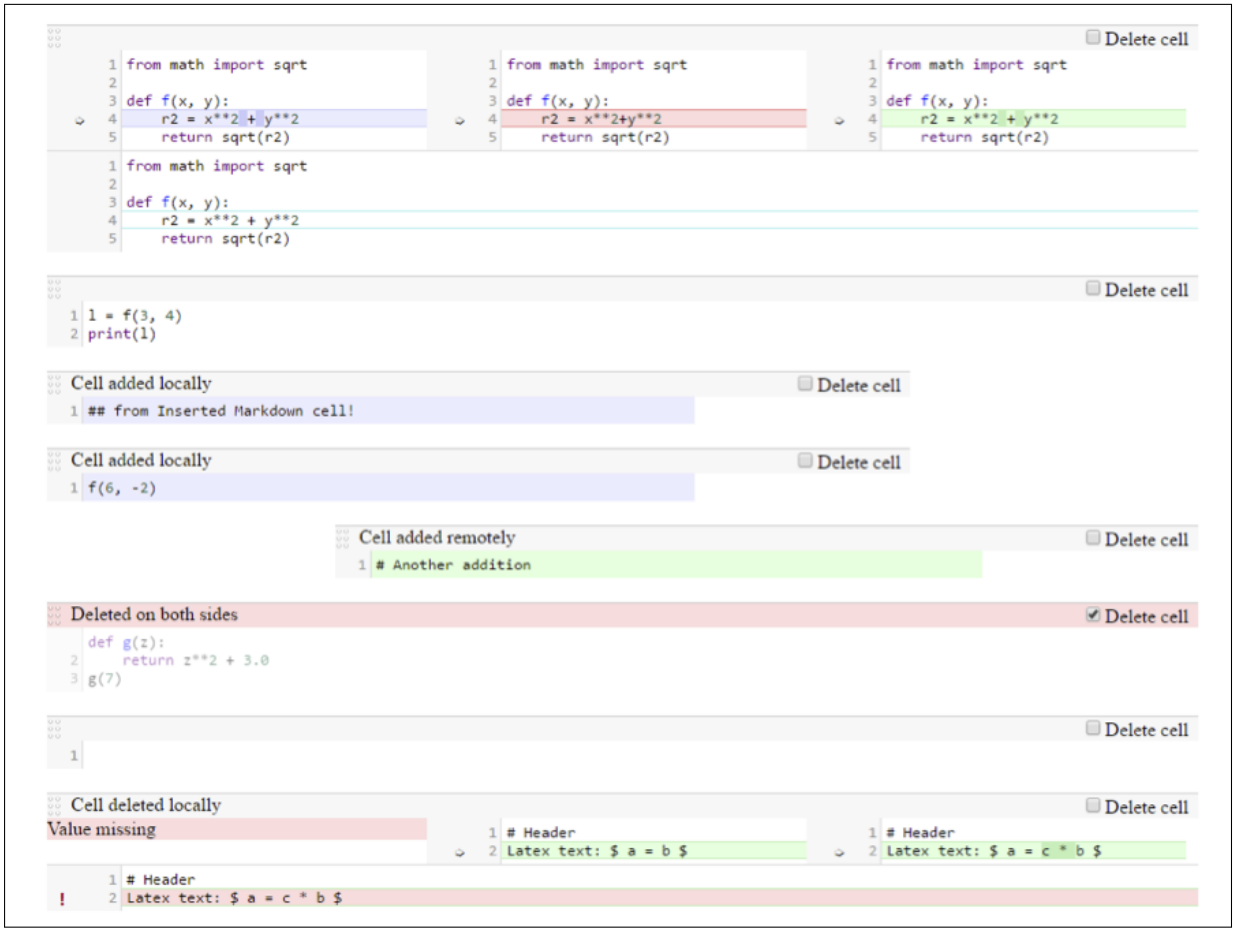

FIGURE 5. NBMERGE web-based merge conflict resolution tool allows individual conflict resolution.

and facilitated, and verification of changes in notebooks is enabled. Further work in D4.8 (Facilities for running notebooks as verification tests) can leverage NBDIME to trace and identify changes in notebooks, leading to verification tools that can efficiently evaluate the reproducibility of notebooks.

6. FUTURE WORK

The current version of NBDIME is a working prototype but we expect some more work to be necessary to make it a polished product. Designers from the wider JUPYTER community will be involved in polishing the visual presentation of the web interfaces, and users will become involved in further testing once the tool reaches a wider user base. Furthermore we expect to be able to improve the algorithms used for diffing and merging further, both for improving the diff quality and performance.

While NBDIME integrates well into the local git workflow of diffing and merging, much of the diffing and merging work of today's scientists and developers happens on public websites such as GitHub or Bitbucket, where the user cannot customize the diffing or merging behavior. However, these websites do support custom diff viewing of certain file types, such as images and GeoJSON files. After some maturation of the design of the NBDIME-WEB DIFF viewer, we will work with GitHub and others to integrate diff viewing of notebooks into their websites, as we have done in the past with rendering of single notebooks. By integrating NBDIME with remote notebook editing tools like JUPYTERLAB, comparing notebooks on remote servers should become much easier than today, and would integrate well with such a development process.

Disclaimer: this report, together with its annexes and the reports for the earlier deliverables, is self contained for auditing and reviewing purposes. Hyperlinks to external resources are meant as a convenience for casual readers wishing to follow our progress; such links have been checked for correctness at the time of submission of the deliverable, but there is no guarantee implied that they will remain valid.