REPORT ON OpenDreamKit DELIVERABLE D2.17
Introduce OpenDreamKit to Researchers and Teachers as laid out in Task 2.6

MIKE CROUCHER, HANS FANGHOR AND NICOLAS M. THIÉRY

Due on	31/08/2019 (M48)
Delivered on	04/09/2019
Lead	University of Leeds (ULeeds)
Progress on and finalization of this deliverable has been tracked publicly at: `https://github.com/OpenDreamKit/OpenDreamKit/issues/252`	

1. Brainstorming on #D2.17

TODO's: Jeroen, Luca, Marcin, Alex, Tania

1.1. **Introduction**

Notebooks – interactive documents mixing prose, code, outputs, and visualization – have existed for decades; we may cite the Maple, Mathematica, or SageMath notebook. There is a long time experience in the community of using such notebooks for a variety of purposes in teaching and research, ranging from scratchpad for interactive computation to narratives such as interactive teaching documents or logbooks of computational research.

At the time of writing the OpenDreamKit Proposal in 2014, the Jupyter notebook was a an emerging technology which had recently evolved from a generalization of IPython notebook that supported a variety of programming languages or interactive systems. Based on modern web technologies, with a modular design informed by many former implementations, and a growing ecosystem of related tools, it showed strong potential as core user interface component for building Virtual Research Environments. Of particular interest are the Jupyter widgets that bridge the gap between interactive computation and interactive visualization and applications, in effect making for a smooth continuous learning curve from user to power user to developer, and enlarging the range of use-cases.

In Task 2.6, we have disseminated Jupyter based Virtual Research Environment to students, researchers, and teachers to act as multipliers of knowledge and dissemination. This covered the technology but also best practices. The notebook, despite all its benefits, also has pitfals, with which the participants have extensive experience.

OpenDreamKit actively organized or participated in dozens of events where they advertised and delivered training on OpenDreamKit technology. See D2.6, D2.11, D2.15 for details. Section XXX reviews other evaluation and dissemination activities that were carried out during the project.

In a second section, we briefly reflect on the lessons learned. Indeed, these activities were also first and foremost the occasion to gather first hand testimony of the technology on the battle field, this to continuously inform and motivate the project

1.2. **Dissemination and evaluation of activities**

1.2.1. *Dissemination events.* TODO: Check that this was reported on in the events deliverable and remove

In 2017, we organised a comprehensive workshop on Computational Mathematics With Jupyter in Edinburgh, targetting academics, researchers and teachers at the same time. Topics delivered by OpenDreamKit members included Jupyter, nbdime, nbval, Docker containers, JupyterHub. We managed to engage external contributors, such as Christian LawsonPerfect from Newcastle reporting on the Numbas web-based e-assessment system for mathematics and Mark Quinn discussing the use of SageMathCloud for teaching purposes.

We contributed a workshop on "Jupyter Notebooks for reproducible research" at the 2017 International Research Software Engineering conference in the UK. While a lot of this work may appear as UK -centric, this did have dissemination effects across Europe as the emergence of the research software engineering idea and community was driven in the UK; researchers across Europe were closely monitored and participating in these events.

1.2.2. *Teaching with Jupyter.* Many – if not most – of the OpenDreamKit participants engaged actively in using and testing Jupyter technologies in their daily teaching duties.

Here are a few striking examples:

- OpenDreamKit experts on Jupyter delivered multiple courses (topic?) at the NGCM Summer School 2016 and 2017 at Southampton, which attracted many PhD students and some researchers from the UK, Europe and some from overseas.
- Paris Sud participants used Jupyter in a variety of classes (graph theory, computational combinatorics, computational algebra, experimental mathematics, numerical analysis) throughout the math and computer science curriculum. An highlight is a 400 students introductory programming in C++ class. Having a uniform user interface across systems (C++, Python, SageMath, . . .) was a major selling point by enabling students to be immediately productive in their new environment. The teaching was supported by the deployment of a local JupyterHub server and the occasion to experiment with various tools including the notebook converters nbsphinx and nbconvert, the C++ interactive interpreter xeus-cling, Jupyter widgets, interactive web pages with ThebeLab, assignement handling and grading with nbgrader. This led to an invitation of the course leader to a workshop in Edinburgh to share experience and participate to an nbgrader coding sprints where several contributions were submitted and accepted.
- Gent (TODO: Jeroen)
- UVSQ (TODO: Luca)
- Silesia (TODO: Marcin + refer to interactive book)
- St Andrews (TODO: Alex)
- Sheffield OpenDreamKit participants spearheaded a transformation in the way computation was taught at The University of Sheffield across multiple subjects and at many levels. The Department of Physics, for example, now teaches *all* of its undergraduates how to program in Python using Jupyter notebooks and the CoCalc (Formerly SageMathCloud) cloud-based environemnt. Furthermore, many other physics modules at Sheffield now include some type of computation using the same technologies. Programming is no longer considered separately but as the integral part of modern science that it really is.

  Working with an Italian Marie-Curie fellow in Bioinformatics, Sheffield's OpenDreamKit participants developed a set of short postgraduate 'Bioinformatics Awareness Days' which used OpenDreamKit technology to introduce Bioinformatics workflows to clinicians at Sheffield's Institute for Translational Neuroscience. This further led to an OpenDreamKit teaching tutorial being held at University of Naples Parthenope.

  Other subject areas where Sheffield's OpenDreamKit participants assisted in developing enhanced lecture material using OpenDreamKit technologies include Machine Learning, Mathematics, Computer Science, Biology and High Performance Computing.

1.3. **Teaching material**

Interactive lecture notes are an area where commercial vendors such as MapleSoft and Wolfram Research are spending a lot of time and money developing material. Within the Jupyter ecosystem it has become possible to author interactive lecture notes and make them openly available (e.g. through BinderHub). We have created such interactive lecture materials, used them in university education, and made them openly available on the Internet. This includes four interactive textbooks (see also D2.9 and D2.14), but also Software Carpentry lessons, course notes, etc.

Anecdotal evidence and feedback from individual users (see D2.14) shows that they are used outside the OpenDreamKit partners, in and out of Europe, both by individual students and university lectures.

1.3.1. *Local consulting.* Across all the OpenDreamKit partner sites, OpenDreamKit staff and PIs have engaged with colleagues, students and decision makers to advocate the benefits of the Jupyter research environment; often effectively serving as consultants for best practice computational mathematics and science tools and workflows.

At Sheffield, taster seminar (1-2 hours) and follow-up short course (1-2 days) on Jupyter for lecturers and researcher were organised targeting and engaging science and engineering disciplines beyond mathematics as the Jupyter ecosystem of tools is of value for teaching and research in any discipline having to work with computational and data based research. These workshops were integrated into the research support services of the Sheffield IT department, and integrated with the research software engineering movement that originated in the UK over the previous years.

The OpenDreamKit funding was essential in forming Sheffield's Research Software Engineering (RSE) Group, one of the first such groups in the UK. Providing training, documentation and support in the use of OpenDreamKit technologies in both teaching and research helped the Sheffield RSE group demonstrate to the University how vital RSE support can be. This directly led to the fully-funded, diverse group that now exists at Sheffield which has served as a model for many other such groups around the UK, Europe and more, recently, the United States.

1.4. **Contributions**

At Southampton, OpenDreamKit developed and enhanced tools such as Jupyter, nbval and nbdime were integrated into the taught Masters programme of the national Centre for Doctoral Training in Next Generation Computational Modelling (NGCM).

The integration of Sun Grid Engine and Project Jupyter, done as part of OpenDreamKit (add deliverable XXX), has led to some educators considering using the notebook to introduce various aspects of High Performance Computing. The use of notebooks at Sheffield, Southampton and other universities has grown significantly. Based on the e-infrastructure work of OpenDreamKit and our dissemination activities, the notebook is emerging as the de-facto standard in the symbolic mathematics domain.

Our development and dissemination activities on nbval and nbdime (XXX add deliverable) within OpenDreamKit had resulted in researchers switching to using the notebook for all of their software documentation and tutorials. The unique selling point of nbval is to make it possible to ensure that notebook-based documentation will always work as development of code progresses and the documentation is in danger of becoming outdated: Using nbval allows the documentation to become part of the formal testing and continuous integration framework.

- Tania's course template
- Marcin's bookbook cookie-cutter

1.5. **Meta discussion**

1.5.1. *Evaluation and Adoption.* In this section, we build on all the gathered witnessing to reflect on how each tool we have promoted appeared adequate for the users needs and adopted by the community

The Jupyter notebook. Practice over hundreds of students in class or other learners at dissemination workshops have shown that it takes no more than an hour of guided instruction to become sufficiently acquainted with the basics of the Jupyter notebook to be able to autonomously explore properly structured collections of notebooks (e.g. an interactive text-book, or a collection of tutorials in a workshop). At this stage, the simple, linear narrative structure of a notebook is a precious guide to the reader. It does take practice however for the user to not get any more confused by the potential disprecancy between the visual order and execution order of cells. This can be mitigated by short notebooks, and also the use of notebook extensions that enforce the execution order of cells.

Properly authoring one's own notebooks also takes a lot of practice. The simplicity of incrementally building code by testing code snippets and aggregating them is a big strengh of the notebook. It's also a weakness, as it encourages ... leads to bloated messed up notebooks.

TODO: nthiery

Jupyter widgets.

3D visualization.

Computational systems.

Converting tools. nbconvert, sphinx, ...

Binder.

Version control nbdime.

Validation with nbval.

Installation with Conda.

VRE access / deployment. JupyterHub
at EOSC

Class management. There are various methods for managing classes using OpenDreamKit technologies and OpenDreamKit participants have tried most of them. None of them are perfect and many non-specialist lecturers require support in choosing and using the various options which include:

- Commercial Cloud-based systems such as Microsoft Azure and CoCalc. These are very easy to set up and use but remove a lot of control from educators. Issues include updates occuring in the middle of exam sessions that modified user-interface behaviour and even computational results in some cases. For one computationally intense course, the amount of CPU power available in the cloud environment was insufficient for students to complete some project work which led to complications.

- Bring your own laptop. Such sessions ensure that participants leave teaching sessions with a fully functioning computational environment but supporting diverse operating systems and hardware can be extremely challenging for those running the course.
- Servers ran by University IT. Can provide a very controlled and stable environment but at the cost of teachers not being able to update in a timely fashion unless the servers are supported by dedicated Research Software Engineerign staff.

When using this type of technology across many subjects in a University, it quickly becomes apparent that the only way to fully support lecturers properly is to have on-site specialists that can provide the requsite assistance. Such work forms the foundation of the Research Software Engineering groups that have started to form all over the world. OpenDreamKit participants formed the first demonstrations of such partnerships

nbgrader is a tool for managing assignments in a class: authoring, production and distribution of instructor and student version, collection, semi-automatic grading, feedback. It is modular and each piece can be used either from a graphical user interface (UI) within Jupyter or from the commandline. This gives quite some flexibility to integrate it into the local information system and workflow.

Experience shows that usage through the UI is straightforward, for students and instructors alike, even novices. However, as of now, setting it up requires some computer literacy, if not admin right on the local system. The tricky part is to setup the *exchange zone* through which assignments are exchanged back and forth between instructor and students. Support for multiple courses and multi-instructor courses is also very recent. We expect the entry barrier to decrease greatly with the ongoing development of an *exchange service*; once deployed locally at an institution and integrated with the local information system, instructors will be able to seamlessly setup new courses.

Interactive documents. TODO: reference to section 4 of D2.14

1.5.2. *EOSC.* All technologies developed in this work-package can be transferred to the European Open Science Cloud in fairly straightforward manner, as they are using open, standard, and modern web technologies.

???

1.5.3. *Reflection about "small contributions" making large impact.* was it efficient?

- feedback loop
- not easy to fund
- not easy to find people who want to do this
- the value of what we disseminated was for many participants dominated by learning about the basics of the tools rather than the more advanced extensions implemented through OpenDreamKit

1.6. Stuff that could be added

1.6.1. *Evaluation and contribution to marking systems.*

- @nthiery: could write 2-3 paragraphs about use of nbgrader for Info 111 + invitation to workshop + a few pull requests

1.6.2. *Somewhere: a brief story about Sheffield stoping and the project taking over the dissemination activities.*

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.