CW15 Discussion session 2 outcomes

Open Access Mandate for Software

Reporter: James Hetherington

* Case is easy, and already won for data and papers:
    * Public funded, public asset
    * Open science is verifiable science
* Collaborating on closed source projects is bad for RSEs as we can't point to what we've done.
    * Working closed source is bad for your career as a working scientist
* Data sharing now has a mandate. Data sharing is a better model than Open Access Publishing.
* Software is harder as there exist profit-making legacy codes with senior owners.
* BBSRC, H2020 already have this to a certain extent…
* Where are JISC in this? We don't know
* Where are Wellcome? We don't know
* Software management plans
* Some journals have it (Nature!)
* Can be incorporated as part of data sharing policies?
* Different for research that is software heavy?
* Grandfathering existing closed-source software?
* Need to do this with education of university enterprise people:

    * License advice
    * Business model advice
    * Contributor agreement advice
* Need to research international landscape.

* We recommend this as a new campaign to the SSI.

What are the best ways of communicating technical requirements from researchers to developers? Are there examples of good common vocabularies that have been created to aid interdisciplinary projects?

Name of Topic Owner: Natalie Stanford

Name of Topic Reporter: Mihaela Duta

What are the five most important things learnt during this discussion:
1. Lesson learnt from UCL Software Development Team: the pilot phase of a project may be the most effective aspect to implement in future projects.

What are the problems, and are there solutions?

1. Often software is not properly documenting, making code maintenance and further development almost impossible if the original programmer has left the project.
a. incentives for better documentation in code
b. better code and time management practices and teaching
2. Language barriers between scientists and software developers.
a. participatory code design
b. within large projects - project liaison officers that act as ambassadors for their groups, acting as an intermediate layer/buffer between software developers and research groups themselves
3. Initial requirements coming from researchers may not always be the best solution to the problem they have.
a. start with a pilot phase - proof of concept; this helps to reach a point where there is an agreed design that both sides understand and approve as the best solution.
What further work could be done, and who should do it (make a pledge)?

Are there any useful resources that people should know about?
1. Fossbox: http://www.fossbox.org.uk
2. IDEO: http://www.ideo.com/uk/
3. UCL Software Development Team: http://www.ucl.ac.uk/research-it-services/our-work/research-software-development

Effective management for successful interdisciplinary projects

Topic owner: Mikail
Topic reporter: Russell Garwood

Five important points
- Think carefully about what researchers and tools you will work with.
- Researchers are not often prepared for management
- Most principles are common strategies for good project management but interdisciplinary research is fragile. Communication and collaboration are critical to success, not only beneficial.
- Have a clear leadership structure
- Have a clear understanding of the problems in each domain

Problems
Some projects are a great success (e.g. CERN) but some interdisciplinary projects completely fail.

What management strategies can we use to ensure success?

Solutions
A lot of the solution is common sense for project management but it is important to pay close attention to the communication aspects of the team. Are people talking the same language? Do we

understand each other and our problems across the group?

Resources
https://github.com/maxogden/async-team

What are the best formats for sharing data and results across disciplines?

Name of Topic Owner: Bruno Vieira
Name of Topic Reporter: Boris Adryan

Martin, Michael, Kristina, Mark, Devasena, David, Alison, Richard

What are the five most important things learnt during this discussion:

1. Our group work with (a) binary data encoded, (b) ascii encoded, or (c) a mix. Rougly 1/3rd per category. However, encoding is not difficult. It's the actual FORMAT that can create issues.

2. Instrumentation and domain-specific software often spits out proprietary formats [independent of binary/ascii format]. However, large communities seem to agree on open formats for data exchange. At Diamond, this conversion happens at the time of generating the (raw) data. They also provide incentives for end-users to use open formats.

3. Meta information is CRUCIAL. A dataset without meta info is an incomplete dataset.

4. Data harmonisation takes a lot of time. Example from comparing simulation of tsunamis: Converting 15 different inputs for a comparative analysis becomes a job in itself.

5. Best practices in data storage: Annotate the data [meta info: what is x,y,z, which were the experimental parameters], keep this info close to actual measurement

What are the problems, and are there solutions?
- Proprietary data may not be useful if, e.g. your license expires to use relevant software. => requires additional software to convert to open formats

- Data exchange between large-scale facilities across the world. => "nexus" project tries to tackle that

- Variables, origins and directions are different in different fields. One man's x is another woman's y. => "Do we need meta data for meta data?"

- Writing good data formats. => "There should be unit testing for data." (see CSVLint)

What further work could be done, and who should do it (make a pledge)?

- Advocating the use of hdf5

Are there any useful resources that people should know about?
http://en.wikipedia.org/wiki/Hierarchical_Data_Format - HDF
http://csvlint.io - schemas in JSON
http://dat-data.com - like Github, for data

What generic methods can you use to couple/bridge/glue models from different disciplines together?

Name of Topic Owner: Derek Groen
Name of Topic Reporter: Graham Klyne

What are the five most important things learnt during this discussion:

1. The question is "faulty" - "generic" is a slippery term: a tool may technically work across disciplines, but trhis doesn't mean it is useful across all disciplines

2. No silver bullet

The "couple/bridge/glue" implies a single solvent - but there is no such single solution, as the problem needs to be addressed at several levels; e.g. syntactic, vocabulary/denotation, conceptual, operational, etc.

3. There are partial solutions that may work across disciplines, but are applicable to particular contexts; i.e. not all possible aspects of the glueing problem

Partial solutions include:
For the Web:
- RDF (syntactic) (discuss?)
- OWL (vocabulary)
- REST (operational)

We think that the conceptual level is not amenable to a technical solution because the "Frame problem" - no way to guarantee sufficient common context; so needs to be resolved socially.

For compute-intensive models:
- MML (conceptual, but partially)
- CellML,SBML etc. (syntactic, denotation)

- MUSCLE2 (operational)

Some solutions are quite comprehensive, but are not directly of use across disciplines.

4. For particular communities, there may be a base of strongly held common convention that is sufficient to address the conceptual issues, but only within that community.

Corollary: life gets harder when you go outside a community.

5. Another approach: simplify the problem (remove detail) to make the models more "glue-able", use more widely accepted conventions

Tension between breadth of community and depth of detail addressed

What are the problems, and are there solutions?

E.g. for simulation, combining models/codes which originate from different disciplines and platforms.

By "Model", we mean (not raw data), abstractions/simplifications of datasets / natural phenomena. The frame problem?

Linguistic/ontological concerns here? (cf. Quine "Ontological Relativity")

Pragmatic problem: how to combine information from *independent* sources.
Levels: Syntactical, Vocabulary/denotation, conceptual, operational/access/exchange

The challenge of crafting a common reference frame. There cannot be a technical fix to a conceptual issue.

Technical fixes are abundant for a wide area of applications/models, but many are specific to a domain.

The "general-purpose" fallacy. Just because something can be used by models from any discipline, doesn't mean that it actually provide required/useful functionalities.

What further work could be done, and who should do it (make a pledge)?

Are there any useful resources that people should know about?

(REST)
Linked data/RDF
Standard modelling languages / UML / SBML / MML / CellML
Universal access layer - translation layer - http://www.sciencedirect.com/science/article/pii/S0010465512003670
MUSCLE - http://www.qoscosgrid.org/trac/muscle

How does software support reproducibility of results without requiring expert domain knowledge?

Name of Topic Owner: James Baker

Name of Topic Reporter: Matthew Gamble

What are the five most important things learnt during this discussion:

1. There are different type of reproducibility - verification, repeatability, reuse

2. Verification requires domain knowledge.

3. Software can lower the burden of documenting experiments so that they might be reused and repeated

4. Linux containers are a specific technology that can be used in the goal of repeatability

5. There is a level of trust required in reproducibility.

What are the problems, and are there solutions?

What further work could be done, and who should do it (make a pledge)?

Are there any useful resources that people should know about?

The very fact that we are discussing this topic suggests that software can support reproducibility of results by others beyond the original domain.

climate research - have to take the model on faith - models are millions of lines of code.

Question of what level of reproducibility do we require? Are we validating, do we wish to to understand enough to reuse.

Linux container to capture software environment that was used so that you can “replay” the experiment helps with a lesser form of reproducibility.

Different levels of “domain expert” - might want to move a single level up to a different sub domain, or we might want to communicate.