Science & Technology Facilities Council

Persistent Identification and Citation of Software

Catherine Jones (STFC)

B. Matthews, I. Gent, T. Griffin and J. Tedds

June 2016

Introduction

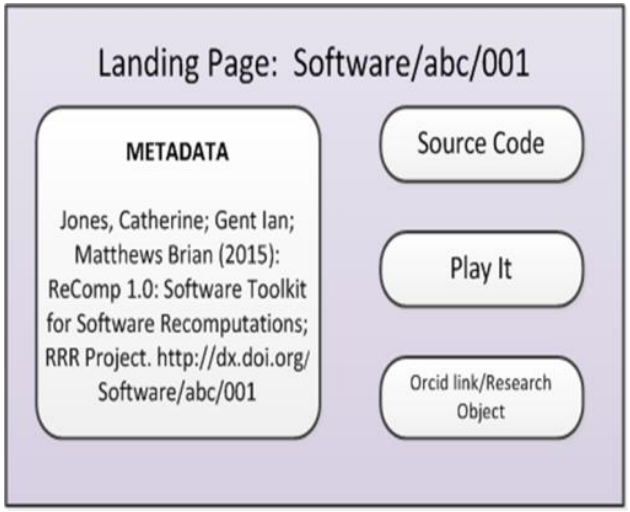
- Science and Technology Facilities Council
  - 1 of 7 UK funding bodies
  - Funds Universities in Particle & Nuclear Physics & Astronomy
  - Also provides large scale Facilities – Pulsed Neutron Source, Large lasers
- Scientific Computing Department
  - Supercomputing, Storage, LHC Tier 1, scientific code & data management
- Software Engineering Group
  - Software tools, publication & data repos

Science & Technology Facilities Council

Introduction II

- I lead the Software Engineering Group
- Background in Computer Science & also qualified Librarian
- Research Interest: data and software curation; linking different parts of the research life-cycle together

Aims



- Jisc funded **Software Re-use, Re-purposing and Reproducibility** project looked at how the DOI metadata schema should be applied to software and explored capturing software in a running state.

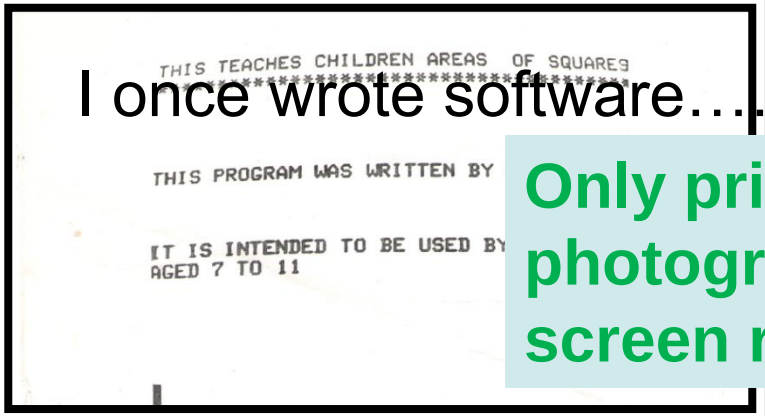
Starting point:

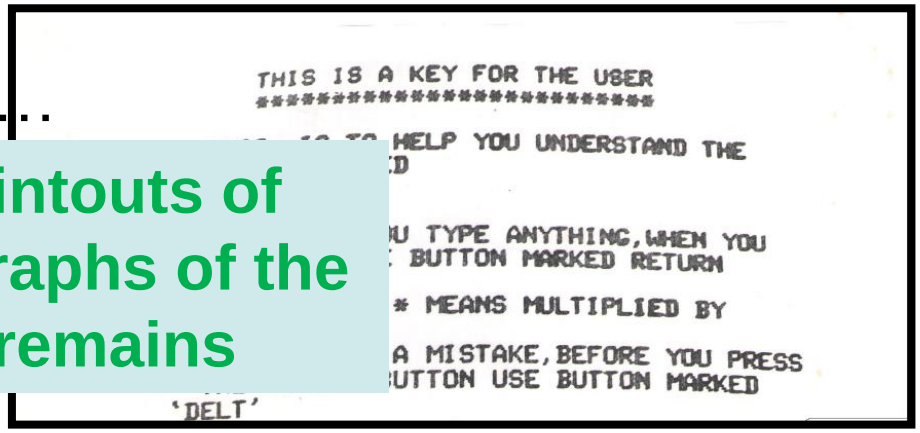
- Software underpins research in many disciplines
- Data can be meaningless without the software which created, analysed or displays them
- Software is complicated with many dependencies
- Writing software is an intellectual endeavour in its own right

Science & Technology Facilities Council

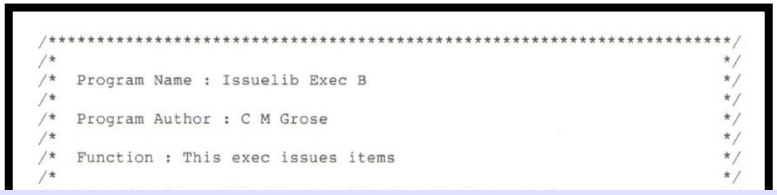
Why am I interested?

I once wrote software……





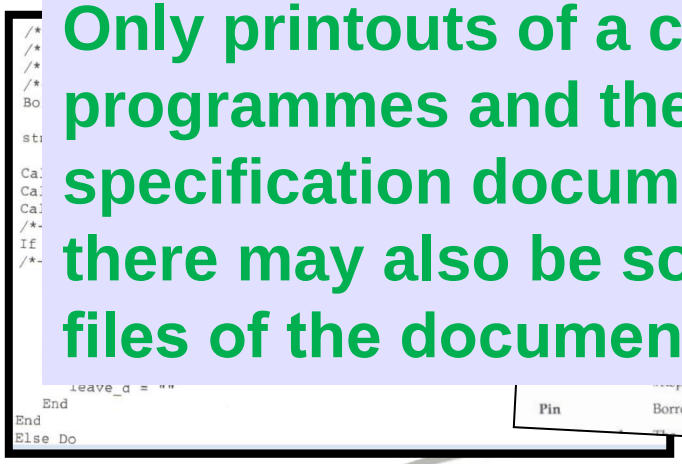
Only printouts of photographs of the screen remains

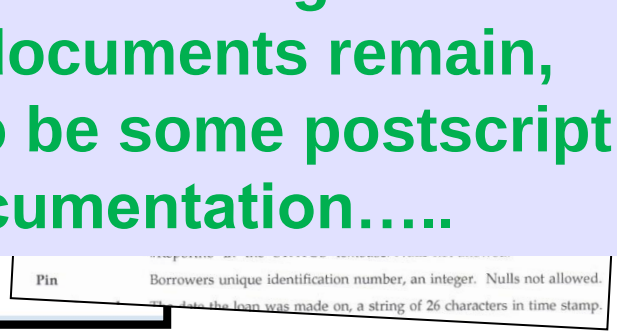


1983
Basi
.19
& Re

But I kept them because they were an important intellectual record for me….

Only printouts of a couple of programmes and the original specification documents remain, there may also be some postscript files of the documentation…..



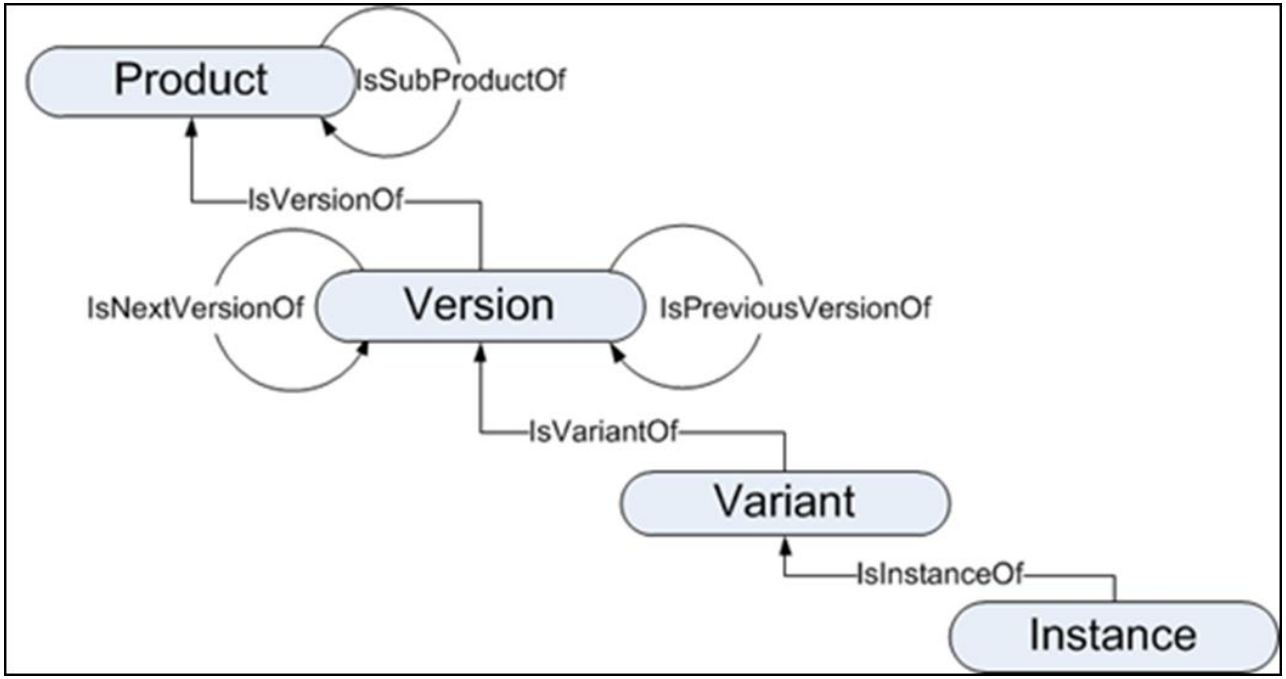


ogy

What do we mean by software?

- Software is a general term, scale can vary from:
  - One off script (post-it note?)
  - Script used on a regular basis (technical report?)
  - Complete programme providing a set of functionality (journal article?)
  - Suite of programmes providing a wider range of functionalities (series? Journal issue?)
- It can be written from scratch or a local modification of existing code
- Don’t forget the environment and dependencies…

What is being Identified?



Product is the concept. **Version** is a specific set of functionality. **Variant** is a version for a specific environment. **Instance** is a specific variant on a specific machine.

Any of these may be needed to be persistently identified depending on the situation

Science & Technology Facilities Council

Why persistently identify software?

- The identification of the software used in a specific circumstance
- Citation of the software so that appropriate credit can be given to the creators
- The ability to rerun the correct experiment/software to verify results recorded elsewhere
- A preservation repository needs to know what is in the collection.
- To distinguish between different versions

Science & Technology Facilities Council

Stakeholders & Motivation

- The further from the creation of the code, the greater the interest in preserving it is.
  - **Research software engineers**:
    - “Good software management practice is all that is needed”
    - We suspect those who need to reuse code may not agree …
  - **Computational scientists who write code**:
    - Haven’t thought about it but acknowledgement/credit and reproducibility are good in theory
  - **Digital Preservation experts**:
    - Very interested as they know they will have to do it
    - Recognised as a key trend in 2016 at iPres
- The idea of being able to prove reproducibility of results from software analysis is gaining traction
- There are overlaps with Continuous Integration tools or deployment processes

Science & Technology Facilities Council

Finding & Using software

- To be able to find & reuse software the following information may be needed:
  - **Purpose**: what was it designed to do
  - **Programming language:** what is it written in?
  - **Environment:** what tools and operating system will I need to be able to run or modify it
  - **Who wrote it:** Do I trust them & their organisation?
  - **Where does it live?** Do I trust the software repository, does the DOI point to the version being developed?
  - **What license is it issued under?**
- Some of this is relevant to many research outputs, some is specific.

Science & Technology Facilities Council

DataCite metadata

- Report: http://purl.org/net/epubs/work/24058274
- Gave guidance on how to apply DataCite to software
  - 1 recommendation will be in V4 of the DataCite Schema

DataCite Property	**<name of the property>**
DataCite Description	Explanatory text provided by DataCIte
Purpose	Discussion on how this applies to Software. This may include examples
Stakeholders	Who is involved in the decision and why they might add this metadata
Questions	Questions to help those who want to add DOIs to software to make the right decisions for their case

Science & Technology Facilities Council

DataCite Creator

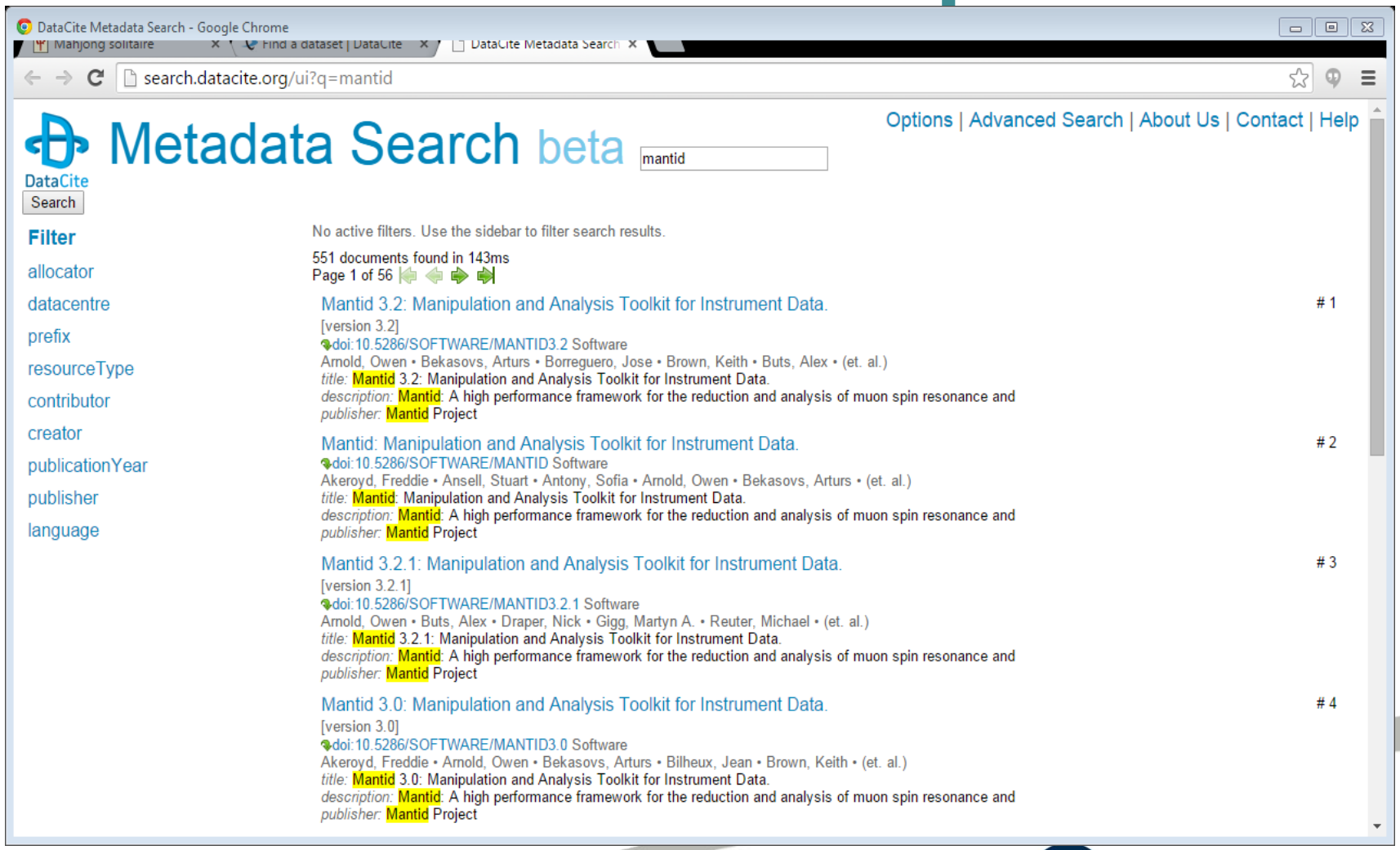
- To identify the people responsible for the software
- This field will be used to create the citation.
- The creator may not be a straightforward item to ascertain as software has a long life-span and may be worked on by many people.
- The point during the development cycle that the first DOI is given may also affect those identified as creators.
- The creators need to be listed in order of importance
  - How can this be identified?

Science & Technology Facilities Council

DataCite Creator Examples

- Student project/single developer
- Project team – DOI on first production release
  - The current team should be straightforward to identify.
- Project team - DOI after years of production releases
  - It may be hard to identify all those who contributed to creating the software. The current release's team will have built on the work of others.
- Project team – DOI for every major version
  - The creators for each DOI can reflect those who contributed to that specific version and the versions can be related through relationships.

Real life example



Science & Technology Facilities Council

DataCite Title

- This is used to form the citation
- Is the mandatory field containing the most information.
- Questions
  - If it a piece of software written by a single person for a specific project does it actually have a name?
  - Is the official name different from the common name?
  - What effect is versioning or branching of code going to have on the name?
  - Are there any naming conventions that need to be adhered to?
  - Will the name used be unique enough for it to be found and distinguished from other search results?

DataCite Relation type

- Relationships of particular relevance are:
  - RELEASES**: IsNewVersionOf** and **IsPreviousVersionOf**
  - MODULES **IsPartOf** and **HasPart**
  - FORK: **IsContinuedBy** and **Continues** or perhaps **IsSourceOf** and **IsDerivedFrom**
  - ENVIRONMENT: **IsVariantFormOf** and **IsOriginalFormOf** (Different operating systems)
  - DOCUMENTATION: **IsDocumentedBy** and **Documents**
- Make clear that IsCompiledBy and Compiles are not used in the computing sense
- Under consideration by the DataCite

Science & Technology Facilities Council

DataCite Description

- Enables extra information to be added
- **Abstract** and **Other** most commonly used
- Existing content:
  - More information about the purpose of the software or releases or live repo
- Some information needed to understand the object doesn't have an obvious field
- Report suggested new DescriptionType of **TechnicalInfo** (will be in the next version of the schema)

Science & Technology Facilities Council

Example: Mantid

- Mantid is an open source development for data analysis in the Neutron Scattering Community with a large software development team
- The software is used “as is” and there is no expectation that there will be local user modifications
- Approach
  - Product level DOI for the concept of the software
  - Each new version has its own DOI, crediting those who worked on that version.
    - Uses IsPartOf to link back to the Product and IsNextVersion/IsPreviousVersion to relate version levels
- Users of the software can cite the software version used for the analysis.

Science & Technology Facilities Council

Software Citation – Force 11 Principles

- **Importance**: Software should be considered a legitimate and citable product of research.
- **Credit and Attribution**: citations should facilitate giving scholarly credit.
- **Unique Identification**: citation should include unique identifier
- **Persistence**: Unique identifiers and metadata describing the software …should persist.
- **Accessibility**: citations should permit and facilitate access to the software.
- **Specificity**: citations should facilitate identification of, and access to, the specific version of software that was used.

https://www.force11.org/software-citation-principles

Science & Technology Facilities Council

Conclusions and Next Steps

- For persistent identification and citation to become commonplace, culture around credit needs to change
- Exploring persistent identification within computational science community (long-lived codes)
- Unresolved issues around the preservation and reuse of modified code
- Looking at overlaps with Jenkins/Deployment
- Catherine.jones@stfc.ac.uk