eosc | FAIR-IMPACT
Expanding FAIR solutions across EOSC

An overview of the Metadata landscape & Descriptive metadata curation

Morane Gruenpeter
Software Heritage
Inria, France

eosc | FAIRCORE4EOSC

Funded by the European Union

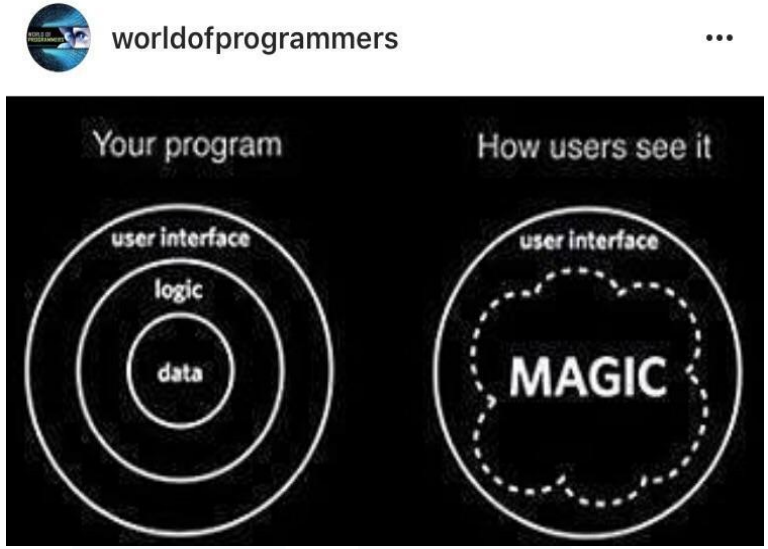
What is software? The metadata challenge

Software as a concept

- **project** or entity
- the **community** around the project
- the software **idea** / algorithms / solutions

Software artifact

- **source code** form
  - for each version and revision/commit
- **binaries/executables** produced (for different environments)



https://www.reddit.com/r/ProgrammerHumor/comments/70fuamp/programming_is_magic/

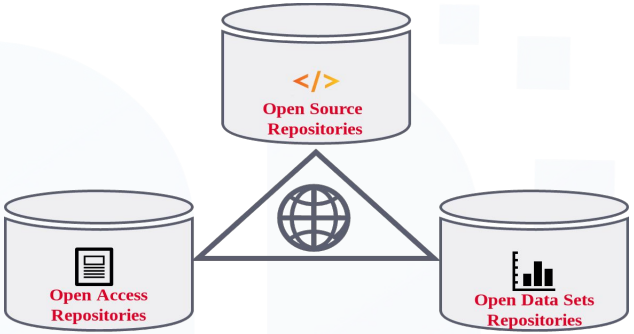
"Ontologies are **agreements**, made in a **social context**, to accomplish some objectives. It's important to understand those objectives, and be guided by them."

T. Gruber, The Pragmatics of Ontology, 2003

Defining Research Software

*Research Software includes **source code files**, **algorithms**, **scripts**, **computational workflows** and **executables** that were created **during the research process or for a research purpose**. Software components (e.g., operating systems, libraries, dependencies, packages, scripts, etc.) that are used for research but were not created during or specifically for research should be considered **software in research** and not Research Software. This differentiation may vary between disciplines. The minimal requirement for achieving computational reproducibility is that all the computational components (Research Software, software used in research, and hardware) used during the research are identified, described, and made accessible to the extent that is possible.*

FAIR4RS output: Gruenpeter et al. Defining Research Software: a controversial discussion (Version 1). Zenodo. https://doi.org/10.5281/zenodo.5504016



Three pillars of Open Science
Software Heritage CC-By 4.0 2019

Software has multiple facets:

- a **tool**
- a research **outcome** or result
- **the object** of research

Why are we here? A plurality of needs

Researchers

- **archive and reference** software used and created in articles
- **find** useful software
- **get credit** for developed software
- **verify/reproduce/improve** results

Laboratories/teams

- **track** software contributions
- **produce** reports
- **maintain** web page

Research Organization

know its **software assets** for:

- technology transfer,
- impact metrics,
- strategy

FAIR-IMPACT - D4.4:
Guidelines for recommended metadata standard for research software within EOSC

Where is the metadata available ?

Catalogs and registries

- ASCL
- swMath
- OpenAire
- libraries.io
- Research Software Directory - escience center
- …

Software development platforms (on platform page)

- GitHub
- Bitbucket
- SourceForge
- …

Scholarly repositories

- Zenodo (InvenioRDM)
- HAL
- …

Scholarly publishers

- IPOL
- eLife
- Dagstuhl
- Episciences
- …

Extrinsic metadata - Advantages and drawbacks

	Catalogs and Registries	Development platforms	Scholarly repositories	Publishers
Accuracy	- not created by author	+created by author	+ added by authors	+created by author
Completeness	+ very detailed (curators)	- not a priority	- depends on the author or the repository's requirements	+strict requirements and review process
Longevity	- depends on registry	- depends on platform (not archived)	+preservation strategy	~ depends on archival strategy

The case of intrinsic metadata

In the *software source code* itself

- README
- LICENSE
- AUTHORS
- **codemeta.json**
- package management
  - pom.xml
  - package.json
  - …
- CITATION.cff
- .About
- …

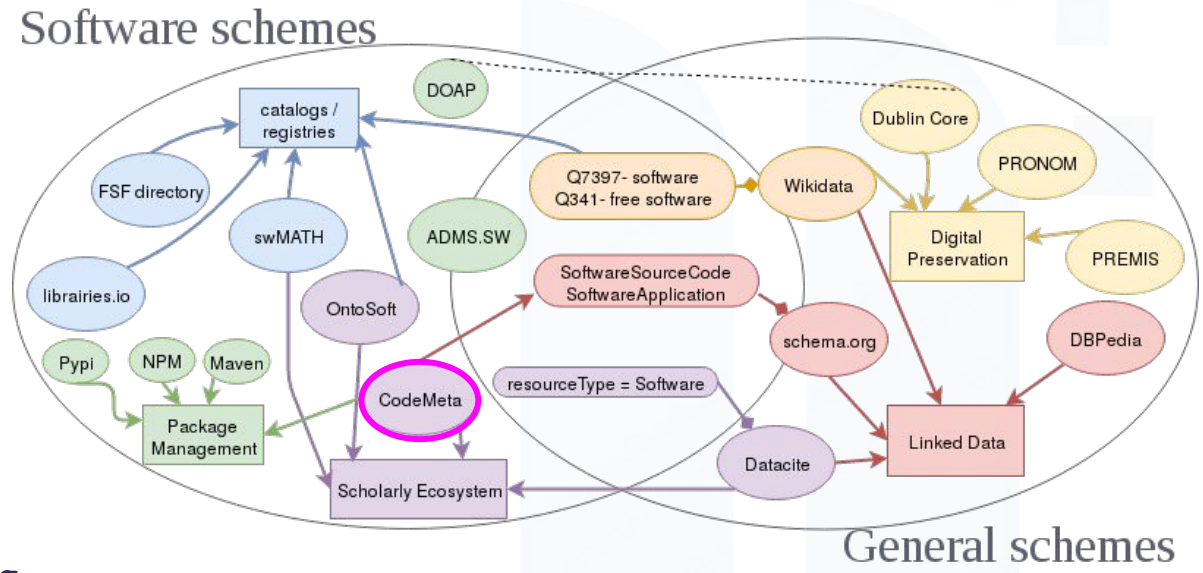
Advantages and drawbacks

	metadata file
Accuracy	+ created by author and evolves with code
Completeness	- depends on the authors knowledge of metadata
Longevity	+ not dependent on platform or infrastructure

Software vocabularies landscape

CodeMeta Initiative

- A subset of schema.org
- An academic community discussing software metadata
- A crosswalk table - mapping the metadata landscape



Gruenpeter M. and Thornton K. (2018) Pathways for Discovery of Free Software (slide deck from LibrePlanet 2018).
https://en.wikipedia.org/wiki/File:Pathways-discovery-free.pdf

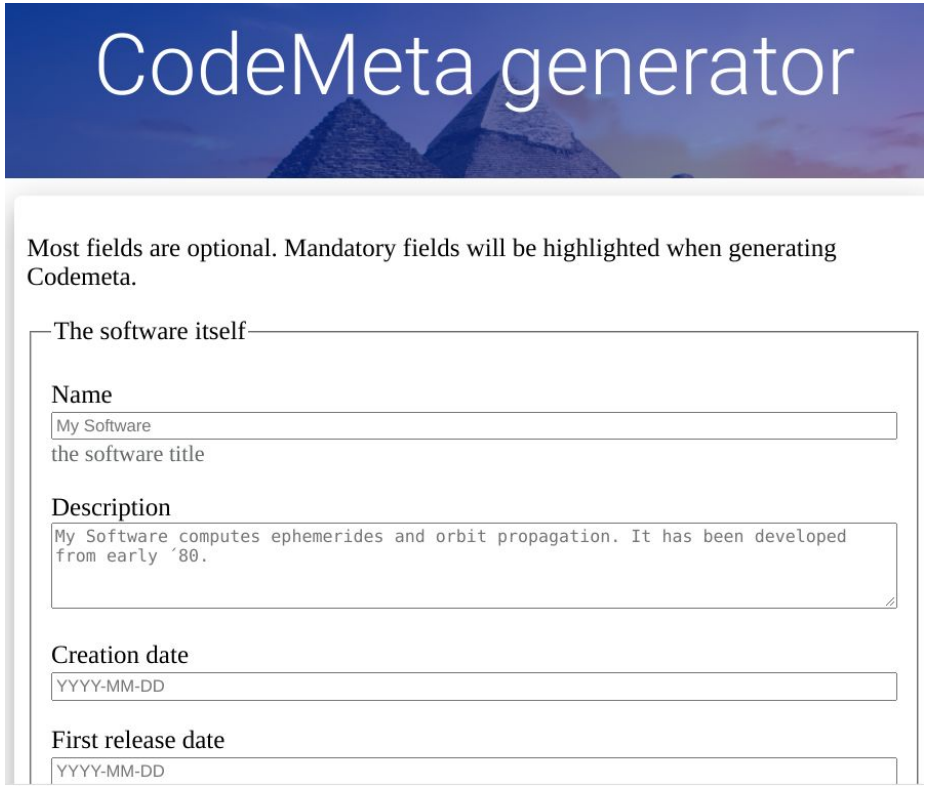
Metadata tools

An open source tool to create codemeta.json files

- Use it directly on the CodeMeta hosted version
- Contributions are welcome on the code repository

Contributed to the community by

Software Heritage
THE GREAT LIBRARY OF SOURCE CODE



Know your basics: CodeMeta properties

Identify

- identifier
- name
- author
- version, softwareVersion

Execute

- codeRepository
- operatingSystem
- softwareRequirements
- buildInstructions

Classify

- description
- releaseNotes
- keywords
- supportingData (in/out data)
- fileFormat
- programmingLanguage

Administrate

- maintainer
- copyrightHolder
- funder
- license
- editor
- publisher
- dateCreated
- dateModified
- datePublished
- developmentStatus

Know your basics: CodeMeta properties

Identify

- identifier
- name
- author
- version, softwareVersion

Execute

- codeRepository
- operatingSystem
- softwareRequirements
- buildInstructions
- **Not in CodeMeta:**
  a. Examples
  b. Compiler
  c. Executable link
  d. Other documentation

Classify

- description
- releaseNotes
- keywords
- supportingData (in/out data)
- fileFormat
- programmingLanguage
- **Not in CodeMeta:**
  a. references
  b. algorithms

Administrate

- maintainer
- copyrightHolder
- funder
- license
- editor
- publisher
- dateCreated
- dateModified
- datePublished
- developmentStatus

Session A - Exercise 1: codeRepository (type: URL)

What?	**Why?**	**Where?**	**Who?**	**How?**	**Bonus:** Metrics
(Description from CodeMeta)	*(Use case, purpose to use term)*	*(Intrinsic, extrinsic or both, which platform? Information source, provenance)*	*(Provider of metadata)*	*(availability of the information, metadata creation process - manual or automatic)*	*(Which **metric** can be used to verify this metadata and it's quality)*
Link to the repository where the un-compiled, human readable code and related code is located (SVN, GitHub, CodePlex, institutional GitLab instance, etc.). (from CodeMeta)	Access the development platform where the source code is developed and maintained As a software developer I want to open an issue or contribute to an existing tool to improve it	Extrinsic (but can appear in the metadata file as intrinsic information) Development platform (GitHub, GitLab, etc.)	Author	Manual process when creating a repository. URL is given by the platform and author can add the URL in the README or codemeta.json file	• URL exists in the record • URL is resolving • URL is also archived in SWH as an origin

Session A - Exercise 1: codeRepository (type: URL)

Examples of values: *(add 1-3 examples)* ● https://github.com/moranegg/deposit-template ● https://github.com/rdicosmo/parmap	**Challenges** - When code isn't developed on a collaborative platform (locally, privately on a non public repository, on a closed institution network) - releases can be then public and open source - When code is not Open Source	**Priority:** MUST / **SHOULD** / MAY **Bonus: Reference** to existing guidelines: *(provide link or quote from the guidelines collection)* - HAL - Create deposit guide: code repository isn't mandatory but suggested Archive | Reference | Describe | Cite *(highlight match from SIRS)* Find | Access | Interoperate | Reuse *(highlight match from FAIR4RS)*
How to improve metadata quality? ● Archive code repository in SWH with all dev history ● Insert codeRepository in intrinsic file to verify provenance	**Difficulty level** to complete quality metadata term: easy | medium | hard	

eosc | FAIR-IMPACT

Expanding FAIR solutions across EOSC

@fairimpact_eu /company/fair-impact-eu-project

Funded by
the European Union