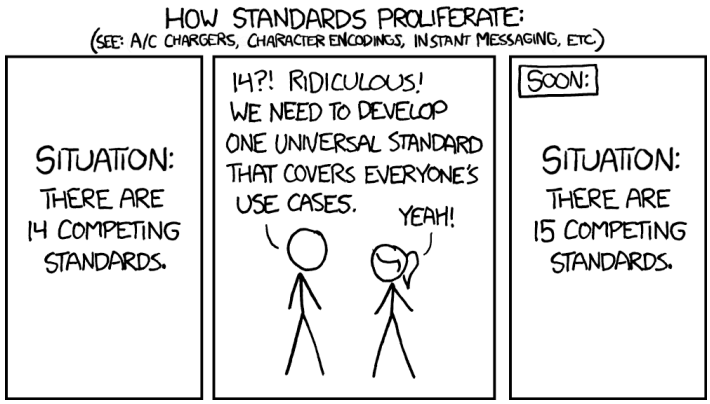
Building organization-level interop w/ a Focus on Research Software Engineering

GSOC Mentor Summit
6 Oct 2024
Jacob - Julia

- Core question : “how do we build organizational level interop with a focus on research software”
- Research software is software which helps answer/support research questions (e.g. simulations, pde solver, etc)
- Within Julia health, HADI and others, a lot of the different tools have a somewhat common data standard, but hoping to bridge between different languages/tools
- If we have R input, how do we get it working on Julia? Some languages have bindings and bridging, but is there some stronger organizational-level approach?
- Cooperation between different fields/languages/tools!
- Examples of problems/solutions
    - Random number generation and seeding inconsistent.
    - Floating point and numerical methods can be inconsistent, both software and hardware
- Reproducibility generally
    - R-python, and making sure package versions the same and other environmental things (used packrat to help)
    - Trying to avoid reimplementing things from scratch.
- Input/output and relationships for pipelines/processes
    - Documenting input and output standards. In practice, companies have mixed results on actually doing this.
    - “By design”
    - A waveform data data sharing project partially documented for consistency, but the documentation was incomplete. Some ad-hoc standards come up just for sharing.
    - How can we get ad-hoc standards to be more solid and better documented?
    - Metadata and discovery, “schema.org”
    - An aerospace project, MBDyn, talks about industry collaboration and data collection. Needed to find an existing standard across academic and industry, “shoehorn it into our use”. Finding an existing standard and making it work easier than making a new standard.
    - “If not in <X>, I’m not interested”
    - Some industries have open standards, some don’t. Some ISO standards, some a little less direct but still pretty formal.
    - Regarding metadata politics, OpenAustronomy mentions how NASA has a strong sway related to funding and similar. Sometimes this is top down, sometimes collecting representatives to make a standard.

    - Dicom! Well documented standards for data, metadata, and access mechanisms (dicom web)
- Containers and related environment standardization
    - Including web assembly, singularity, docker
    - Using containers (or packrat, etc) to make sure the computing environment is consistent.
    - Some possibility of getting different versions, and even within some standards (huggingface), sharing consistently.
    - Provenance too - how its run may have an impact too, especially for subtleties in inputs and outputs.
    - When dependencies change, sometimes the field may change under you
    - Tracking dependencies, git sha hashes. Sometimes people backport bugfixes to older releases.
- “Midpoint summary”
    - Containers
    - Metadata
    - Fully describing inputs and outputs
    - Sometimes a field may be developing/changing too rapidly to have a solid standard
    - “If it’s not in <X> I’m not interested”- Developing a workflow can cause optimal efficiency.
    - Implicit and explicit standards. Sometimes a tool ends up being the standard (e.g. matlab + simulink)
- Containers and io standards, continued
    - In house vs out of house work.
    - “Zero dependencies” - how do you distribute code via git sha? Or docker images.
    - NIX for bringing together dependencies and source code for reproducibility.
    - Generally hardware related standards are pretty solid, networking, etc.
    - APIs can standardize information transfer with an intermediate form. Enforcing things like radians not degrees.
- Human-friendly
    - If you’re working between standards. A lot of the difficulty is from data from humans without any idea there is a standard. Data dictionaries. OMOP.
    - Cleaning input data can be a lot of the work. Neuroscience/neuropsych data. Patient records, etc. Often no standard, only sometimes internally consistent.
    - Data is different from person to person within studies. Focus often on publishing, and data quality/formatting is not high on the concerns list.
    - Sometimes reimplement code to a more generalizable format.
    - Extending tools beyond use of standard, then we get things out of spec.
    - Data quality/standard failures seem to lead to retractions a significant amount of the time.
    - Collecting data as a design/interface/human problem, it should be useful to people otherwise they’ll bypass part of all of the system
    - Examples help people in addition to standards.

- Problems with new standards, and how they rarely replace other standards, just add to them.
  - 

  - (https://xkcd.com/927/)
- Wrap up **lightning round**
  - Transposing schemas to other schemas automatically
  - Data standardization and ease of use, combining too many things