2020-11-23 CTWG Meeting Notes

Meeting Date

- 23 November 2020

Attendees

- David Luchuk
- Drummond Reed
- Rieks Joosten
- Daniel Hardman
- Dan Gisolfi
- Scott Perry
- Steven Milstein

Main Goal of this Meeting:

Understand (and if possible decide on) Daniel Hardman's proposal for tooling and overall workflow.

Agenda

Time	Item	Lead	Notes
1 min	Welcome & Antitrust Policy Notice	Chairs	
2 mins	Introduction of new members	Chairs	
2 min	Agenda review & open Action Items	Chairs	
5 mins	Co-Chair volunteers	Chairs	
35 mins	Presentation and discussion on tooling and workflow	Daniel Hardman	
12 mins	Integration with Operations Team	David Luchuk	
2 mins	Review of Decisions and new Action Items	Chairs	
1 min	Next meeting	Chairs	

Recording

- **Link**

Presentation(s)

- *Daniel's slides*

Documents

- *File 1 - link*
- *File 2 - link*
- *File 3 - link*

Notes

1. Welcome and Linux Foundation antitrust policy
2. Introduction of new members
3. Agenda review & open Action Items
4. Daniel Hardman presented his slides and recommendations about terminology tooling
    a. His evaluation of existing tooling (open source and commercial) is that nothing that is still maintained really fits our needs well
    b. We need "just enough tooling"
    c. Daniel proposes the following data pipeline
        i. Capture—receive raw data from on-off ticket or batch submission PR or script
        ii. Scan—human sanity check to triage, catch basic issues
        iii. Merge—commit to repo, convert to internal data model, assign permalinks, becomes publishable
        iv. Mature—Run (semi-)automated QA. Generate tickets. Propose "Accepted" status for community = WG. Assign tickets for curators of other communities.
        v. Accept—Review and adjust ticket statuses in WG meeting.
    d. Daniel proposed a basic data model (see the slides)
        i. Rieks noted: "I'm concerned about the relation between concept and term having 1 - n multiplicities rather than n - m multiplicities. To be discussed."
    e. Daniel proposed a process by which every stakeholder community can review and decide on the status of a term without having to necessarily agree with other communities

f. Daniel proposed two major requirements for our tooling
    i. Major feature #1: Manage Curation
        1. Anybody can propose content
        2. Tickets are the way to change content status
        3. Anybody can raise a ticket
        4. Review tickets are tied to a community (scope)
        5. Each community has its own status
        6. Each community has its own review process and appoints one or more "**curators**" <== term proposed by Daniel
        7. Curators directly update status for their community (or admins update per instructions from a curator)
        8. Enforce some data integrity rules and workflows
        9. Track contributors, history
        10. Stats
    ii. Major Feature #2: Publish
        1. Emit content per community
        2. Timely updates (realtime desirable)
        3. Artifacts can be styled/customized per community
        4. Static, searchable, indexable HTML
            - One doc, or one doc per term / concept
            - Stable relative links
        5. Programmable data (CSV or JSON) and/or API
            a. An example is writing a script to analyze a glossary or a group of terms
        6. Full metadata available
            a. Contribution history
            b. Status change history
g. Data and permalinks
    i. Live data should be in the internal data model
        1. Browsable in internal data model
        2. https://github.com/<repo>/terms/agent-119 (term named by first EN label + concept num)
        3. https://github.com/<repo>/concepts/119-agent (concept named by num + first EN label)
        4. Hyperlinked to issues
        5. Links are stable across changes in terms, definition text <== permalinks are in place, so terms can be deprecated and still resolve
    ii. Published data
        1. Browsable in glossary data model format
        2. Published by communities on sites under their control (they put static HTML where they want)
        3. <glossary website>/agent.html (no concept links)
        4. Links are versioned (not guaranteed stable across releases) <== not permalinks
    iii. Dan asked if we could use GitHub Actions to publish "live data"
        1. Daniel said yes, that would result in the published data reflecting the live data
    iv. Drummond asked about how a version of a glossary can be "frozen" for a specific community, i.e., a spec
        1. Daniel said that the community could fork off a version of their glossary
        2. You can also point off to a specific version of the data at any point in time.
h. Specific tool proposals
    i. terminology database = github repo
        1. Ingest new data as Markdown documents in GitHub
        2. Then after processing into internal data model, still keep each "table" in as a Markdown document in GitHub
    ii. to do QA for WG review of submitted data: new python script (Daniel volunteering but inviting others)
    iii. to manage internal data model
        1. to convert from submit format to internal data model: new python script (Daniel volunteering but inviting others)
        2. to edit and browse internal data model: modified ESSIF / GRNet tool (the one Rieks has developed)
        3. to update status, add hyperlinks, propagate tags: new python script(s)
    iv. to emit static HTML: github action hooked up to #2 in preceding bullet
    v. to emit programmable data (CSV, ...)—TBD
i. Configuring a community
    i. Provide official name and #tag
    ii. Identify and train curators (github handles, contact info)
    iii. Configure artifacts
    iv. Configure data import
    v. Train community on curation and publication processes
j. Configuring an artifact
    i. Choose publication mechanism (output template, scripts, targets, collateral)
    ii. Setup schedule or triggers for publication
    iii. Provide selection criteria (tags, statuses)
    iv. Test run
k. Configuring data import
    i. One-time, ad-hoc, ongoing?
    ii. Write and/or tune script(s)
    iii. Dry runs with cleanup
    iv. First import
    v. Trigger for deltas
l. Working Group Duties
    i. Triage tickets
    ii. Train communities
    iii. Setup communities and artifacts
    iv. Liase with communities
    v. Approve "Accepted" status requests
    vi. Propose new data sources
    vii. Configure and maintain tool integrations
    viii. Develop output templates

        ix. Run tools for ingestion
        x. Review data quality
    m. Other proposals—for our actions
        i. Publish draft glossaries from our 3 datasets
            1. Data sanity check
            2. Convert to internal data model
            3. Configure artifacts and export
            4. Assign community curators to approve
            5. Forcing function for tools: first cut by mid Dec?
        ii. Divvy up WG work for #1 (tickets)
        iii. Figure out collaboration model outside WG meetings
        iv. Modify agenda so we spend a chunk of our time working tickets
5. Co-chairs
    a. At the conclusion of his presentation, Daniel Hardman volunteered to be a co-chair.
    b. Drummond Reed volunteered to join him in the interest of helping to coordinate vocabulary both at the Governance Stack WG (who will likely be the CTWG's biggest customer), across other WGs, and also across the wider ecosystem (he was one of the original co-chairs of the Decentralized Identity Foundation (DIF) Glossary WG).
    c. Daniel and Drummond were approved as the initial co-chairs of the CTWG by consensus.
6. We ran 15 minutes long, so there was no time for further agenda items.
7. The next meeting is at the regular time in two weeks.

Decisions

- [ ] Daniel Hardman and Drummond Reed were approved as initial co-chairs of the CTWG.
- [ ] The CTWG will publish draft glossaries from our initial three datasets following the process proposed by Daniel Hardman.

Action Items

- [ ] Daniel Hardman to develop a plan for implementing his proposed workflow, including how to collaborate outside bi-weekly meetings.