

<https://github.com/linkml/linkml/pull/2873>

LinkML Rolling Minutes (2024-)





Meeting time: Fridays, 4:00 PM ET, 1:00 PM PT (Except TISLab no-calls week)

Zoom link: <http://go.lbl.gov/cjmzoom>

Google Drive:  Meetings - LinkML

Meeting notes from 2022-2023:  2022-2023 LinkML Rolling Minutes

Recurring:

Description	Link
Topics for discussion	https://github.com/linkml/linkml/labels/meeting-topic
Feature voting	<ul style="list-style-type: none">• https://github.com/linkml/linkml/issues?q=is%3Aissue+is%3Aopen+sort%3Areactions-%2B1-desc• high priority tickets
Projects	https://github.com/orgs/linkml/projects?query=is%3Aopen
Open PRs	https://github.com/pulls?q=is%3Aopen+is%3Apr+user%3Alinkml+archived%3Afalse+
Open issues	https://github.com/issues?q=is%3Aopen+is%3Aissue+user%3Alinkml+archived%3Afalse+
Good first issues	https://github.com/issues?q=is%3Aopen+is%3Aissue+user%3Alinkml+archived%3Afalse+label%3A%22good+first+issue%22
Discussions	https://github.com/orgs/linkml/discussions/
Issue triage	https://github.com/linkml/linkml/issues
Milestones	https://github.com/linkml/linkml/milestone/1 2021-09-01
Stack overflow	https://stackoverflow.com/questions/tagged/linkml
Draft paper	 LinkML paper [2022]
Recordings	<ul style="list-style-type: none">• Schema-transformer• GA4GH VA in LinkML• LinkML w3ids (Harold)• LOINC in LinkML• run-linkml-examples (Mark)
Community meeting agenda and notes	 LinkML Community Meeting rolling agenda https://linkml.io/community
LinkML Technical Manager agenda	LinkML Technical Manager Agenda
LinkML SQL Meeting agenda	 LinkML SQL Meetings
LinkML-map WG agenda	 LinkML-Map Rolling Agenda and Notes

2025-09-05

Attendees:

Regrets:

Agenda:

- High level testing document to test Carlo's monorepo merge [PR](#):
 - LinkML Monorepo Migration Guide
 - linkml monorepo migration testing document – Sujay [Sujay]
 - confirm UV monorepo high-level understanding:
 - “packages” is now a “workspace” with two “members”: *linkml* and *linkml-runtime*. Each has their own pyproject.toml, but only one uv.lock file between both of them.
 - what is the consequence of this naming convention? (“packages” vs. “linkml” vs. something else).
 - “linkml” repo now looks for linkml-runtime modules locally, and not from PyPI.
 - how do we know what version of linkml-runtime is currently in use in linkml with this setup?
 - high level questions:
 - do we have a “release SOP” for folks making linkml mono-repo releases. (assume GH actions will distribute linkml-runtime and linkml packages separately “on release” - is this right?)
 - what happens if I only want to release linkml-runtime to pypi and not linkml? Do we still have separate release versions? - I assume the two pyproject.toml files make this doable for us, is this correct?
 - what is the story for folks that currently import linkml and linkml-runtime packages in their repos. what do they do going forward?
 - shall we consider lock-step release versions for both of these going forward? (does this mean always release linkml when a runtime change is released even with no linkml changes and vice versa?)
 - where are linkml-runtime tests? what does `make test` do at the top of the repo now? do we have an SOP for running tests in one member repo or another?
 - high level brainstorm on what can go wrong and how we mitigate it:
 - local virtual environments: install from linkml, then pull in from pypi with a specific version
 - install linkml-runtime vs different from linkml version
 - todo: pyproject.toml linkml-runtime dependency needs to change
 - todo: makefile changes need to move in to main Makefile
 - todo: make some test branches do more testing.

- todo: ontogpt, oak, biolink, nmnc-schema
- todo: merge in and make a release candidate.

○

notes from chat:

I would like linkml and linkml-runtime to stay synced with major version numbers but allowed to be different on minor version numbers.

tests/linkml_runtime

@Mark Miller The issue of datetimes in artifacts generated during test runs should have been solved a while ago, if that's what was underpinning your question

Yes, I was asking about that. I saw some issues about it when I was cleaning up email. I didn't realize how old those messages were

Ooh great catch!

uv pip install

"git+https://github.com/linkml/linkml.git@merge-linkml-runtime-into-linkml#subdirectory=packages/linkml_runtime"

Resolved 36 packages in 2.94s

Built linkml-runtime @

git+https://github.com/linkml/linkml.git@d7af431f2c9b61c94cfd7eadbd1344346e1815ce#subdirectory=packages

Prepared 2 packages in 1.24s

Installed 36 packages in 118ms

+ annotated-types==0.7.0

+ attrs==25.3.0

+ certifi==2025.8.3

+ charset-normalizer==3.4.3

+ click==8.2.1

+ curies==0.10.22

+ deprecated==1.2.18

+ hbreader==0.9.1

+ idna==3.10

+ iniconfig==2.1.0

+ json-flattener==0.1.9

+ jsonasobj2==1.0.4

+ jsonschema==4.25.1

+ jsonschema-specifications==2025.4.1

+ linkml-runtime==1.9.4.post1262.dev0+d7af431f (from git+https://github.com/linkml/linkml.git@d7af431f2c9b61c94cfd7eadbd1344346e1815ce#subdirectory=packages/linkml_runtime)

...Installing the top level of the repo gives me linkml-runtime from the branch, which makes sense

(runtime-test) → runtime-test uv pip install

"git+https://github.com/linkml/linkml.git@merge-linkml-runtime-into-linkml"

Resolved 83 packages in 465ms

Installed 83 packages in 358ms

+ alabaster==1.0.0

+ annotated-types==0.7.0

+ antlr4-python3-runtime==4.9.3

+ arrow==1.3.0

...

+ jsonschema-specifications==2025.4.1

+ linkml==1.9.4.post1262.dev0+d7af431f (from

git+https://github.com/linkml/linkml.git@d7af431f2c9b61c94cfd7eadbd1344346e1815ce)

```

+ linkml-runtime==1.9.4.post1262.dev0+d7af431f (from
git+https://github.com/linkml/linkml.git@d7af431f2c9b61c94cfd7eadbd1344346e1815ce#subdir
ectory=packages/linkml_runtime)
uv pip install "linkml_runtime==1.9.2"
Resolved 36 packages in 209ms
Prepared 1 package in 165ms
Uninstalled 1 package in 18ms
Installed 1 package in 6ms
- linkml-runtime==1.9.4.post1262.dev0+d7af431f (from
git+https://github.com/linkml/linkml.git@d7af431f2c9b61c94cfd7eadbd1344346e1815ce#subdir
ectory=packages/linkml_runtime)
+ linkml-runtime==1.9.2
(runtime-test) → runtime-test uv pip install
"git+https://github.com/linkml/linkml.git@merge-linkml-runtime-into-linkml"
Resolved 83 packages in 465ms
Installed 83 packages in 358ms
+ alabaster==1.0.0
+ annotated-types==0.7.0
+ antlr4-python3-runtime==4.9.3
+ arrow==1.3.0
...
+ jsonschema-specifications==2025.4.1
+ linkml==1.9.4.post1262.dev0+d7af431f (from
git+https://github.com/linkml/linkml.git@d7af431f2c9b61c94cfd7eadbd1344346e1815ce)
+ linkml-runtime==1.9.4.post1262.dev0+d7af431f (from
git+https://github.com/linkml/linkml.git@d7af431f2c9b61c94cfd7eadbd1344346e1815ce#subdir
ectory=packages/linkml_runtime)
uv pip install "linkml_runtime==1.9.2"
Resolved 36 packages in 209ms
Prepared 1 package in 165ms
Uninstalled 1 package in 18ms
Installed 1 package in 6ms
- linkml-runtime==1.9.4.post1262.dev0+d7af431f (from
git+https://github.com/linkml/linkml.git@d7af431f2c9b61c94cfd7eadbd1344346e1815ce#subdir
ectory=packages/linkml_runtime)
+ linkml-runtime==1.9.2

```

Within that venv with linkml from the branch, I downgraded to linkml_runtime 1.9.2 from regular pypi
And it may matter more to those people if there are frequent `_real_` updates rather than just version updates that are really just artificial (don't require testing).


2025-08-29

Attendees:

Regrets:

Agenda:

- validation plugin for Translator (data conformant to Biolink in JSONlines format – potentially also in some sort of flattened duckdb instance in memory).

- <https://linkml.io/linkml/data/validating-data.html>
- <https://linkml.io/linkml/code/validator.html#module-linkml.validator.plugins>
 - store TranslatorValidatePlugin in the Translator repo, but make it a descendent of the ValidatorPlugin (or whatever it is) so that its “interface” is fulfilled and validator.validate() works on this the same way.
- next week: discussion with Carlo w/re to monorepo - I need to dedicate some time to this next week, but also need one more reviewer for a change this large:
 - <https://github.com/linkml/linkml/pull/2873>
 -  LinkML Monorepo Migration Guide
 - Sujay to help review monorepo changes
 - Chris: futures - linkml-model is another issue here. Longterm we need to consider this.
 - we should consider the bigger vision - are we now lumpers? do we put everything in a giant package or keep separate (what are the splitting rules vs. lumping rules).
 - schemaview
 - consider closing PRs or making it clear the migration path for community members: not leave in an indeterminate space.
- linkml-map pydantic
 - <https://github.com/linkml/linkml-map/pull/87/files#diff-a347aa7339628cf59e88a338cff71de3c6de781fd1bdf0d4dd60163eb5397c9b>
- Are there automated docs for telling the world how we are using generated models?
 - linkml-store, linkml-solr?
 - how are folks storing data?
 - do they have a path to use it.
 - typescript generator exists, but beta
 - monarch build pipeline dev related to linkml-store but not in linkml-store yet, get indiv tools doing it the fast way, then we can migrate to a generic framework if need be.
 - linkml-solr schema generator!! (Kevin!!) - produced JSON that looks reasonable. change to schemaview, etc... vs. python dataclasses generation. made pydantic dataclasses, etc..
 - eventually: schema generation in linkml - talking to solr happens in linkml-store
 - linkml-store is not the most performant - it is a basic rewrite of some datastore specific methods.
 - to load neo4j fast use TSV, etc... for fast, you (currently) lose the “generic” benefits of linkml-store. Similar with Solr - POST a JSON, etc...
 - normalized schema -> normalized data -> linkml-map & translation spec -> denormalized schema -> denormalized data -> linkml-store (solr - not written, right now its linkml-solr)
 - can imagine a big SQL query that is the middle (linkml-map denormalizer)

- Chris: what would the patterns be for testing a codebase if we have a lot of optional dependencies
 - matrix GH action test perhaps on testing.
-

2025-08-22 - CANCELLED - TISlab no meeting week.

2025-08-15

Attendees:

Regrets: Kevin (ooo), Sarah

- linkml, linkml-runtime uv migration done. linkml-runtime v1.9.5 release today. - thanks Carlo!!
 - release on schedule
 - need a release process for linkml-map
 - https://docs.google.com/document/d/1A6R-KsMs_4cpyJH1SgabSRE0aEdfhbFf3QybfoxwrTc/edit?tab=t.0 - will try with release candidates, etc...
- copier/uv/just project template from David Linke:
 - <https://github.com/dalito/test-linkml-project-copier> - how best to make this our reference implementation? Make a parallel repo in linkml org? migrate David's to LinkML org?
 - can always have a Makefile if needed
 - make justfiles (makefiles) for linkml less complicated - move processing into python (like gen-project).
- From Nico - requirements for SSSOM:
 - A sophisticated system for propagation (propagating slot values from container to instance level, eg mapping set to mapping) and condensation (compressing certain instance-level fields to mapping set level)
 - Made issue (see above) <https://github.com/linkml/linkml/issues/2866>
 - From Chris: for simple expressions may be possible with <https://linkml.io/linkml/schemas/expression-language.html>
 - linkml-convert with infer flag.
 - possible path: sssom-py ← implement linkml-convert
 - first document functionality existing around path expressions
 - easy if in same object, in the general case we need more logic (what happens with path expression, inlining/not how does it resolve).
 - should not a blocker for SSSOM-py - document and have a utility class.
 - A system for slot versioning, that allows us to say: this slot exists since version v2, or this does not exist anymore, but we want our implementations to be "backwards compatible"

- Here is a 2 min video about this: <https://www.loom.com/share/b35581b6b07843ab912926df1fbee3f>
- Here the issue: <https://github.com/linkml/linkml/issues/2867>
- From Chris: yep, this is covered in a howto <https://linkml.io/linkml/howtos/deprecating-elements.html>. We still might like standardized ways to have “maturity indicators” but this shouldn’t be a blocker.
 - status/etc.. to say this covered.
- A complex system for extension slots, which allows users to define “extra slots” at container level, with some minimal typing as well
 - Docs: <https://mapping-commons.github.io/sssom/spec-model/#defined-extensions>
 - Loom: <https://www.loom.com/share/d7ad9f5530dc4efab0044f6512d6fb96>
 - From Chris: related issues: <https://github.com/linkml/linkml/issues/2241>, <https://github.com/linkml/linkml/issues/2238>, <https://github.com/linkml/linkml/issues/1595>, <https://github.com/linkml/linkml/issues/1404>
 - from meeting:
 - need a linkml-model change - simple boolean (check Chris’s comments in first ticket above)
 - see ticket above for each generator change needed (OWL unnecessary).
- We need to update linkml-model to uv, regenerate docs and push/update vendored artifacts into other linkml repos. (Sierra)
 - will do this separately from linkml/linkml-runtime condensing.
- PRs are building up
 - sierra reviews on the day before a release candidate is made and tries to merge as many as possible to the repo being released.
 - corey also helps here to make PRs go in more smoothly.
 - release calendar:
 - PR review guidelines:
 - rebase/merge
 - things we don’t like:
 - merging when you can’t tell who did what on the PR - and merge from main is a useless contribution
 - don’t like timestamp changes on own work.
 - when bringing a PR up to date with current main via GH interface, pick “rebase” – lets see how it goes. If annoying we will change.
 - should we set up something that closes PRs if no comments in a certain amount of time?
 - should we declare how many reviewers each PR needs before merge?

2025-08-08 - CANCELLED due to key absences

Attendees:

Regrets: Sierra (conflicting meeting), Kevin (ooo), Sarah (ooo),

2025-08-01

Attendees:

Regrets:

- If we have a quorum and folks from ISMB can join: give an overview of the LinkML collabfest at ISMB/BOSC: [summaries here](#) from Carlo, Moni, Daniel, Harry
 - cofest - Moni gave an intro
 - grad student - antimicrobial resistant resource for a long time, advisor to throw ontology in LinkML
 - *Question from a grad student: "My advisor told me to dump my ontology on LinkML." This is the resource <https://card.mcmaster.ca/> We explained to her - in general terms- how a modeling language like LinkML could help her improve her resource.*
 - *have a "data lake" - build an agent that gets all the data, adheres to a schema that has all this information structured.*
 - *generates datasheets for datasets.*
 - *Justin and Marcin has repos like this - ask him about it.*
 - two projects: text extraction and LLM extraction - ontogpt and aurelian approach. what they are doing is old school, explain how to use LinkML
 - showed ontogpt and how to use it
 - database needed to create a schema for it, reuse this for all the other things, also create the ontology out of it, validate everything out of it. had an existing JSON, use schema-automator, then do reverse engineering.
 - hard for her to pick up after Carlo and Harry have left.
 - metadata extraction proteomics publications
 - structured data out of unstructured data.
 - it would have been nice to have more specialized projects
 - next year - come and talk about your problems with us instead of just working on linkml itself.
 - project and thing you want to fix - with this issue - for these folks aggregated and worked on it together.
 - also can show up with a tool, then framework can be applied, clinic.
 - lots of folks did not know about LLM usage in ontogpt or curategpt or LinkML – folks are still at the level of "chatgpt" LLM use
 - don't know how to use a template to make results better
 - always helpful to use linkml

- People have a hard time understanding their problem much less implementing it.
 - tutorial on how to do it
 - Bridge2AI used for a lot of things - datasheets2datasets - also brought to ISMB - labor of love, people need to be acquainted with the tools, why LinkML.
 - examples would be helpful - small, contained
 - Community tab on the website should have the link to the community call.
- Update on LinkML-Map WG (if Corey is available)
 - low attendance, but going well! - possible moving linkml-map meeting time.
 - write out an alignment of two schemas in a spreadsheet -> script -> linkml-map transformation spec.
 - scripts may not be in linkml-map proper, but a demo would be good in linkml-map
- json-ld @type should be class name not 'dict'
<https://github.com/linkml/linkml/issues/2849>
- Review PRs from Vlad
 - Another simple one <https://github.com/linkml/linkml-runtime/pull/436>
 - <https://github.com/linkml/linkml-runtime/pull/428>
 - <https://github.com/linkml/linkml-runtime/pull/426>
 - <https://github.com/linkml/linkml-runtime/pull/435>
 - <https://github.com/linkml/linkml-runtime/pull/440>
 - <https://github.com/linkml/linkml-runtime/pull/439>
 - <https://github.com/linkml/linkml-runtime/pull/438>
 - <https://github.com/linkml/linkml-runtime/pull/437>
 - <https://github.com/linkml/linkml-runtime/pull/432>
 - <https://github.com/linkml/linkml-runtime/pull/430>
 - <https://github.com/linkml/linkml-runtime/pull/429>
 - <https://github.com/linkml/linkml-runtime/pull/442>
- W/re to other open PRs, not too many, a few, some very old
 - e.g., <https://github.com/linkml/linkml-runtime/pull/347> - we want to move to a system like pydanticai, using uv for everything, remove tox, add ruff as dev dependency, etc...
- UV migration strategy
 - linkml
 - do this one first
 - linkml-runtime
 - migration means PRs that currently exist may not get in before the migration.
 - some social engineering - let people know ahead of time.
 - set the repo to "frozen" - NO PRs, etc...
- linkml/linkml-runtime merge strategy
 - need to take to community
 - need to get Chris buy in

- e.g. how to preserve history - is this important, is it not

2025-07-25

Attendees:

Regrets: Chris, Nomi, Carlo, Daniel (all at ISMB/BOSC), Kevin (OOO), Trish

- If we have a quorum and folks from ISMB can join: give an overview of the LinkML collabfest at ISMB/BOSC
- Release schedule straw-person:
https://docs.google.com/document/d/1A6R-KsMs_4cpyJH1SgabSRE0aEdfHbFf3QybfoxwrTc/edit?tab=t.0
 - Sierra is going to try to stick to this for the rest of the year to see how hard it is and how much each release slips as a result of either broken codebase/merges/emergency changes/over-scheduling release managers, etc.
- Next week: let's try and get the linkml/linkml-runtime presentation going from Carlo (TODO: we need to make sure Chris is available too).

2025-07-18

Attendees: Mark, Madan, Patrick K, Sujay, Trish, Carlo, Kevin, Sarah

Regrets: Sierra, Chris

Agenda:

- Do we want to invite more people to this call that are really focused on developing right now?
 - For example: Vlad Korolev (ET), AJ Ireland (PT), Silvano Cirujano Cuesta (GMT +2), David Linke (GMT +2)
- [Trish] <https://github.com/linkml/dm-bip/tree/main/notebooks> - Trish to follow up with Corey on the needs for this.
- [Madan] Validate data in a CSV file against LinkML schema for include project. The slot is `multivalued: true`
 - Patrick: there's an issue in the linkml repo for linkml-validate not working with multivalued slots
 - Related issue: <https://github.com/linkml/linkml/issues/2581> (best bet to watch progress on this issue for now)
- [Sarah] <https://linkml.io/> upgrade coming — what do people think of the content on the current landing page?
 - Mark:
 - We definitely want to have the most actionable things on this splash page. when I first looked at this page, I did actually find it helpful as it is

- Could be nice to have this content on the splash: <https://linkml.io/linkml/faq/why-linkml.html>
 - Maybe we want a “playground” added to the site
 - Stealth semantics is a good topic, but the slide deck does not really apply to this topic in full
 - Schema view and intro page are the pages I go to the most often.
- Trish:
 - Might be good to have all the tools and functionality highlighted on the splash (they aren’t currently all there)
 - Maybe add who is involved
- Kevin
 - Maybe we do it persona driven? “For software engineer” “For scientists with data” “For semantic engineers”
 - Could have a grid with types of tools vs type of user
 - I could imagine a “why schemas are good” section <https://www.youtube.com/watch?v=fCkEo-1MG1A>
- Patrick
 - Look at what other people do and follow that. Example: <https://docs.pydantic.dev/latest/>
 - Turn the screenshots into something you can copy and paste
- [Trish] It would be nice to harmonize the tool documentation as well. They are all different (originating from each tool repo)
 - examples:
 - <https://linkml.io/schema-automator/>
 - <https://linkml.io/schemasheets/>
 - <https://linkml.io/dm-bip/>

2025-07-11 CANCELLED due to illness

2025-06-27

Attendees: Mark, Sierra, Carlo, Sarah, Sujay, Kevin, Corey

- Review failures in automated PR docs displayed in Carlo and Sarah’s PR to add the community meeting schedule to the website.
 - <https://github.com/linkml/linkml.github.io/pull/8>
 - will check into this preview action: https://github.com/microbiomedata/nmdc-schema/blob/main/.github/workflows/test_pages_build.yaml

- Touch base on progress with linkml-model repo updates.
 - refresh linkml-model doc pages (linkml-runtime 1.9.2 is currently not compatible with linkml-model). need to find the most up to date version that is compatible then go from there.
 - Review linkml-registry regeneration and migration to uv.
 - <https://github.com/linkml/linkml-registry> and <https://linkml.io/linkml-registry/registry/>
 - <https://github.com/linkml/linkml/network/dependents>
 - https://github.com/linkml/linkml-registry/blob/main/src/scripts/discover_linkml_repo.py
 - Decide on next steps for feedback on linkml, linkml-runtime monorepo with Carlo. - Jul 11, 2025
 - How large is too large for schema-automator-ing a JSON data file? Mark Miller
 - creating a microbe ontology
 - creating a microbe KG builder
 - one ETL script takes a BacDive JSON data file as input - schema-automator is slow. - using 35 G
-

2025-06-20

Attendees:

Agenda:

- Great Community Call! - Thanks Kevin & Sarah! - do this:
 - [LinkML Community meeting schedule](#)
- Next steps for LinkML CDE repos
 - Shall we move some of this code out the private repo
 - <https://github.com/monarch-initiative/cde-harmonization/tree/cde2vec/cde2linkml> (remember the repo is only private as we are unsure about whether we can share uber-ingests of CDEs).
 - generic question/answer schema? (CDE + legal processes)
 - answer link back to question, have a value, etc. questions as data.
 - BDC has these too (Corey)
 - Quantity class (value decimal, value string, value boolean, etc...) - dislike the 3 slots for this – rather have 3 classes with type designators.
 - BloodPressureMeasurement
 - has_quantity (range: Quantity):
 - value_string (or something): lod (limit of detection - this could be another slot, etc..) or 125, or even could be a max/min value! or passes a threshold!
 - value_unit: bp/min (or something)
 - value_decimal: null

- value_boolean: null
 - QuantityValue class (NMDC)
 - Next step is a submission in Oct for more work here (UNC, LBL).
 - linkml/linkml-runtime monorepo with uv
 - Carlo is working on a fork to see how this works, find pitfalls.
 - PyPI can still be distributed independently.
 - look at submodules.
 - linkml continues to have an external dependency on linkml-runtime
 - local development will be easier because you point to a sibling directory, not via pypi import.
 - We need to get to a point where we don't have open PRs on linkml-runtime.
 - Priority: get runtime PRs accepted, resolved, closed.
-

2025-06-13

Attendees: Sujay, Mark M, Kevin, Trish, Patrick, AJ, Carlo, Chris, Corey, Nomi

Regrets: Sierra (OoO)

Agenda:

- Next Friday is Community Meeting - is Madan aware that he is hosting? Or, is someone else hosting in his place?
 - Madan is OOO today
 - Kevin will send a message to Madan and check
- LinkML-map WG: summary of purpose, schedule, and deliverables of WG ~ 30 min meeting
 - Had a lot of interest – yayyy!! 8 people turned up for the first meeting this week (Harshad, Deepak among others)
 - Spent some time introducing people and what they would like to use linkml-map for (use cases)
 - Corey:
 - Has a project that's currently using linkml-map and below are the major blockers for that
 - Topic #1: Hindered by the kitchen sink model being used in our test suite currently (in linkml repo)
 - Important topic b/c it'll make it easy for community members to convert their bug/feature requests into unit tests
 - Topic #2: Value mappings
 - Might need to change the way the tests around this are written
 - Will ask in community chat as to who would like to be tagged a reviewers on this
 - Topic #3: Transforms on class objects that are nested and inlined

- AJ: should we keep linkml-map restricted to isomorphisms transformations? Should we have a map extension?
- Chris: might be a hard and fast restriction. Collect use cases from AJ. <Add link to slide deck discussing motivation for linkml-map>. Be wary of “recreating” tools like pandas, etc.
- AJ: looking at linkml-map as an alternative to some data engg tools. Would be nice to stick to tools in the ecosystem. Might need to do some SQL transforms, etc.
 - One set of s/w that’s written in GO and another set in Python. Unified transform language b/w the two spaces
-
- Top upvoted issues:
 - <https://github.com/linkml/linkml/issues?q=is%3Aissue+is%3Aopen+sort%3Areactions-%2B1-desc>
 - Chris favs:
 - <https://github.com/linkml/linkml/issues/2333>
 - Kevin/Carlo/TISLab/CZI:
 - <https://github.com/linkml/linkml/issues/2615>
- Open PRs: <https://github.com/linkml/linkml/pulls>

2025-06-06

Attendees: Mark, Carlo, Sujay, Patrick, AJ, Corey, Trish, Kevin
 Regrets: Sierra (OoO)

Agenda:

- [apply_to](#)
 - Appears to be supported by pythongen and schemaloader, doesn’t appear to be supported by schemaview
 - Implement in schemaview?
 - What would that look like
 - How much work
 - Consider deprecating meta-model elements that we aren’t supporting?
 - Support/code for this almost exclusively lives in linkml-model
 - Status of this concept in the metamodel: `status: testing`
 - But
 - `status:`
 - `range: uriorcurie`
 - Render the fact that the concept is marked as testing more visible in the web documentation pages [TODO: Sujay to make issue in linkml/linkml and work on it]
 - Example use case for `apply_to`? How does it work?
 -
 - Issue: <https://github.com/linkml/linkml/issues/2747>

- What we should do:
 - Let's see if this can be done with linkml-map, recommend if it works well
 - Then think about deprecating apply_to
- Misc.
 - Turn off codespell on certain folders like test folders altogether
 - Pin-it and try the pain of pinned instead of the pain of unpinned
- linkml-map is a new slack channel - please join for focused discussions on that repo
 - Kick off meeting? - cadence?
 - Expectation: be familiar with the repo/codebase (at least a little bit), read through the documentation prior to joining the calls (build the docs incrementally)
 - Not planning to spend too much time on background, overview, etc.
 - Hit the ground running with pair programming tasks, issues, and what not
 - What are folks' use cases for linkml-map?
 - Mark Miller : i would like to see an [implementation of the exclude copy directive](#)
- AOP-wiki data model -> LinkML - has this already been tried? [Chris?]
 - AOP – Adverse Outcome Pathway (eg <https://aopwiki.org/aops/433>)
- Test organisation: storing/categorising tests by github issue # vs by python package name
 - current docs: <https://linkml.io/linkml/contributing/contributing.html#linkml-testing-framework>
 - Tests should go in the appropriate test_*.py file, organised by python package; supporting files (schemas, etc.) can be named according to github issue and kept in separate folders for ease of access/categorization.
- Pycon talk about schemas as the key to data happiness, no linkml, lots of jsonschema
 - <https://www.youtube.com/watch?v=fCkEo-1MG1A>
- GitHub Action error on linkml-model PR: <https://github.com/linkml/linkml-model/pull/226>
 - Does anyone know what this error is about?
 - Latest versions of linkml/linkml-runtime are causing an Action to error out so we need to narrow down on what the most recent versions of linkml/linkml-runtime that are compatible with linkml-model

2025-05-30

Attendees: Sierra, Carlo, Corey, Kevin, Mark, Patrick, Sujay, Trish, Nomi, Nomi's kitten

Regrets:

Agenda:

- Main pipeline failing: <https://github.com/linkml/linkml/issues/2734>
 - Silvano: "we have a broken main pipeline for 11 days now. Fixing that should IMO be getting prio 1 in the project now. Why? Because it is blocking lots of open PRs (some of [mine](#) among them), what eventually increases the merge conflicts once we start merging them. ...I have developed a [PR](#) that should fix the issues. As

of now only @David Linke had a look at it, but other inputs/opinions are helpful for a prompt fix of this issue.”

- Silvano noted: main has currently also [issue 2728](#), which is fixed by the [first commit](#) of [PR 2727](#).
- Sierra said, “the first step I would take would be to see why the linkml-runtime (main) tests are failing and fix them (we should have enough shims in place to fix both runtime and linkml in branches to “handle” the circular dependency issues we’ve gotten ourselves into).”
 - Sierra privately noted that Silvano did not start at the root of the problem (I think its linkml-runtime, but I haven’t looked)
 - Sierra said, “I pulled both linkml-runtime and linkml main branches. All tests pass in both repos locally for me. So next step is to dive in and see why the PRs in question are failing.”
- David Linke said (in Slack), “<https://github.com/linkml/linkml/pull/2727> passes without errors. So a fix is already suggested. The PR could be merged (even by me). One question would be if you agree to the suggested reporting and its planned refinement in later PRs.
- Corey & Kevin:
 - working on a schema-automator release with some fixes
 - working on a linkml-map PR with Kevin
 - how integrated should we make linkml-runtime with linkml-map - right now it is very direct (with yaml parsing, etc...)
 - some complications:
 - refactor to use more linkml methods?
 - transform into linkml before dumpers?
 - performance considerations - python the fastest?
-

2025-05-23

Attendees: Madan, Carlo, Chris, Mark, Kevin, Sujay, Nomi, Trish

Regrets: Corey, Patrick

Agenda:

- Efficient validators [Chris]
 - best support is for JSONSchema serialization validation against data objects (could stream)
 - instead of bringing in postgres, sqlite, mongo dependencies into linkml - try linkml-store.
 - two different validation layers
 - individual KGX bundles - use JSONSchema validation - it probably will be fine

- use the nodes,edges schema (sits over top of biolink, with Node being parent of Entity and Edge being parent of Association) -> then
<https://linkml.io/linkml/data/validating-data.html>
- merged KG validation - something more performant on the backend
- action: swap out Nxgraph with duckdb in KGX (or at least make a separate path there to use a more performant/less(?) memory intensive backend).
- do we need enumerations of infores ids, enumerations of predicates
 - do we just call a method when we want to validate (bmt strategy)
 - dynamic enums - is there code for this? not really, except the method above – call a method at run time.
- id_prefix validation - recommended, not enforced.
 - need model work to figure out if id_prefixes (addition) is open or closed.
 - action: check that jsonschema validation handles (or not) id_prefixes:
 - action: add test_compliance for id_prefixes here:
https://github.com/linkml/linkml/tree/main/tests/test_compliance
 - does not appear to be a supported validation in jsonschema validation yet:
https://github.com/linkml/linkml/blob/3d4495b2eb9bf2d9ac6d3f29865d634f1aa69d05/linkml/validator/_init_.py#L72
 - <https://linkml.io/linkml-store/how-to/Use-Neo4j.html>
 - <https://bioportal.bioontology.org/ontologies/BIOLINK?p=classes>
 - Translator action: we need BMT/something to find the most descendent class for a multi-valued category.
 - Plater requires category closure
- linkml-convert on infores registry - got an error with an infores id
`infores:text-mining provider` ← curie not valid.
 - https://github.com/biolink/information-resource-registry/blob/165e5ddda2a4b3b523c3ab75ba97102d09901576/infores_catalog.yaml#L1669
 - action: need to test that the curie collection is validated via JSONSchema generator
 - action: need to add a GH action to information resource registry (plus examples and tests).
 - curies vs. “strict_curie” (w3id compliant)

- see: https://github.com/linkml/linkml/blob/main/tests/test_compliance/test_core_compliance.py for compliance test suite - we need this for all our generators.
- Request for PR review (async): <https://github.com/linkml/linkml/pull/2640> [Sujay]
 - AJ is helping with PR review - she has qualms about the way the tests have been organized? It's consistent with the pattern we've been following so far, but <https://linkml.io/linkml/contributing/contributing.html#linkml-testing-framework> might need some revision
- GitHub Actions workflows update in linkml-model repo: <https://github.com/linkml/linkml-model/pull/225> [Sujay]
 - Necessary before documentation refresh/upgrades can be applied
- What does it really mean to have a multivalued type designator [Kevin]

2025-05-16

Attendees: Sierra, Chris, Sujay, Mark, Patrick, Corey

Regrets: Kevin

Agenda:

- Community meeting update [Sujay]
 - DH 2.0 presentation from Damion Dooley
 - LinkML feature deprecation best practices presentation from Corey Cox
 - how do we want to use DH links between schema elements
 - application useful for using as a bundle
 - out of the box usage is 👍 - don't plan on customizing it, not quite as easy to incorporate the components separately.
- What release should the YUML deprecation target? [Corey]
 - <https://linkml.io/linkml/code/deprecation.html>
- Efficient validators [Chris]
 - wait for Kevin - next week.
- Request for PR review (async): <https://github.com/linkml/linkml/pull/2640> [Sujay]
- LinkML metamodel documentation pages ... not using latest bells & whistles eg column sorting Mark Miller ,
 - we've gotten better (thanks Sujay) at docs ergonomics, we want the linkml-model docs to take advantage of these.
 - sorting by slot/class metadata
 - rule display for metamodel elements
 - need to update linkml-model to use default templates available in linkml proper - need to take care with the existing URLs (e.g. don't break w3ids).
 - Sierra Moxon 's request for updating description in metamodel Mark Miller - we need a way to deploy this.
 - PRs would be wonderful
 - Might be useful to have the Action that allows for docs preview

- Top 3 priorities for LinkML (roadmap):
 - Migration from **poetry** → **uv**
 - Sujay Write SOP - use Claude/LLM
 - Corey test SOP on schemasheets
 - LinkML SQL
 - LinkML map
- When would we have a 1.x.x-->2.x.x-->3.x.x version upgrade?
 - metamodel change
 - any massive change to the repo - see semvar docs
 - Monorepo refers to GH, still distribute runtime, linkml separately

2025-05-09

Attendees: Sierra, Nomi, Sujay, AJ, Kevin, Madan, Trish, Corey

Regrets: Patrick, Chris

Agenda:

- linkml-runtime 1.9.2 - released, linkml updated to use it
 - hope to get pandera generator merged, then release candidate, but either way rc early next week.
 - agree on a release version for linkml - 1.9.2?
 - Deprecation
 - <https://linkml.io/linkml/developers/deprecation.html>
 - Need a presentation on this - Corey will present! :) ty
 - Running tests locally.
 - Sujay will investigate making ergonomics easier for standard test running and report back if he finds a clean solution.
- community meeting next week
 - HOST: Sujay - ty!!
 - presenters:
 - Damion Dooley - DH updates
 - Corey will present Deprecation Protocol - ty!!
 - ACTION: sierra cancel Juneteenth linkml community call.
- CZI (UNC) updates
 - Corey will make an effort to update on Slack
 - schema-automator updates and fixes to CI/CD bugz
 - Corey will lead linkml-map working group
 - Kevin's linkml-sql group is progressing: clean up and rollup transformer both merged
 - gen-linkml + rollup transformer -> schema that has classes rolled up.


- already added to cli option (parallel clis - not showing in either on atm, TBD)
 - gen-sql-table(...genddl equiv)
 - add to linkml-map as an operation (chain tooling so we can change names of rollup classes from the source class)
 - Kevin working on pydantic 1 deprecations - might need to bundle these two deprecations
 - Madan - deprecation protocol for yUML generator - needs PR review. Looking into another one; just a stub.
 - PR tags, triage, and stale PRs (AJ)
 - Can Nomi do a bit of tagging/flagging/triage for PRs?
 - Corey also helps push through PRs.
 - Let's continue closing PRs that are stale, having failing tests, etc.. and keep PR branches open
 - Tags would help Corey sort through ones he's already seen/updated/etc.
 - Setting reviewers is an important piece.
 - GH action for stale PRs?
 - see relation ontology GH actions (closed after 2 years of no effort)
 - also GH action called "stale PRs" – give it a time period and close PR if its older than 60 days, etc...

2025-05-02

Attendees: AJ, Sierra, Kevin, Mark, Sujay, David L,
 Regrets: Chris

Agenda:

- linkml-runtime 1.9.2-rc2 ready for testing
 - upgrade poetry to 2.x [David L]
 - path resolution bug fixes [AJ]
 - swap linkml-runtime to "trusted publisher" pypi mechanism (also coming is a swap of this for linkml)
- Slack Q/A - please contribute!
 - <https://github.com/linkml/linkml/issues/2655>
 - <https://github.com/linkml/linkml/issues/2654>
 - <https://github.com/linkml/linkml/issues/2653>
 - <https://github.com/linkml/linkml/issues/2652>
 - <https://github.com/linkml/linkml/issues/2651>
 - <https://github.com/linkml/linkml/issues/2650>
- UV migration progress updates?
 - schemasheets - Corey
 - linkml-runtime - Sujay

- Will use the PR on linkml-map as reference to migrate to **uv** in linkml-runtime
 - Turnaround ~ 1-2 weeks
- linkml-map - AJ (complete)
- linkml-store - Sierra (in progress, first update to poetry 2.x, streamline GH actions to match progress in linkml-runtime, create SOP with tips/tricks)
- GH action sharing - should we have a repo? - AJ, David
 - also includes an efficiency upgrade to our actions :)
 - where does tox fit in - do we actually start using it as a python-version test runner? instead of actions?
 - copier templates are good :thumbsup:
 - follow linkml-map
- Collabfest topics:
 - GH action refactor
 - cookie-cutter copier refactor
- Async PR review request (ready for review) from **Sierra Moxon** **AJ Ireland** / anyone else whenever you have some time:
 - Mockup for Rules section on class documentation pages: <https://github.com/linkml/linkml/pull/2640>
 - Added tests 

2025-04-25

Attendees: Sierra, Mark, Sujay, AJ, Corey, Patrick K, Kevin, Nomi

Regrets: Chris

Agenda:

- Best practices for making an artifact from a GH action available at a predictable web location [Mark]
 - <https://github.com/GenomicsStandardsConsortium/mixs/issues/923>
 - what do we do with artifacts? especially if we want external projects to pick this up regularly in a consistent spot.
 - check into main? gh-pages? dist/pypi? (last two are suggested best practices from GH)
 - e.g. JSONSchema serialization, OWL serialization generated by default with **gen-project**
 - should artifacts be stored manually by a human in the repo (main) vs. robots (GH actions, in some branch)
 - req'ts are to *only* see the "latest" (don't need previous versions)
 - w3ids -> gh-pages URL works for biolink, some similar process for AJ
 - <https://docs.github.com/en/pages/getting-started-with-github-pages/github-pages-limits#usage-limits> (1GB storage, 100GB bandwidth/month)

- OWL file generation takes 30 mins
- PR demonstrating deprecation of a legacy generator (`gen-markdown`) [Sujay]
 - <https://github.com/linkml/linkml/pull/2637>
 - Any volunteers who want to use the above as an example and deprecate `gen-yuml`? shouldn't take longer than ~15 mins
 - Sujay will create a **ticket** and assign to Madan
 - Ticket assigned to `madan@tislalab.org` : <https://github.com/linkml/linkml/issues/2644>
 - Think about ways to automate deprecation notification to LinkML developers
 - **ticket**: perhaps add owner metadata to each deprecation - GH handles as user keys
 - guideline: dev that does the deprecation should assume they will follow through on removal (OR notify/ask someone else to do it) when setting a removal release.
 - **ticket**: add a test that will notify if deprecations are still around, unremoved, and report the owner of that deprecation for release manager to bug.
 - try: add a release ticket with deprecations and assign to release manager.
- Simplify documentation pages deployment for linkml-model
 - Make sure the linkml-model docs are using the latest jinja templates
- Modify class documentation pages to show a section for *Rules* [Sujay]
 - Prototype: <https://github.com/linkml/linkml/pull/2640>
 - We like it! 🎉
 - CRITICAL feature of linkml is gen-doc and we love your additions here.
 - linkml-model doc is old and out of date.
 - linkml-model needs definitions
- Default view for index documentation pages [Sierra, AJ, Sujay]
 - Corey thinking about starting a doc WG
 - include Sujay!
 - Defaults should be decided by popularity (Sujay)
- Is linkml-runtime stuck in dependency conflict hell? [AJ]
 - Do you want dependabot updates in linkml, too? We added it to linkml-runtime in the poetry 2.x update but the PR is not yet merged.
 - upstream repo is still set to sneakers/linkml and python-3.13 in PR comment in this PR: <https://github.com/linkml/linkml-runtime/pull/378> change this and the dataops dependency will go away (gone from linkml:main)

2025-04-18

Attendees: Corey, Sierra, Nomi, Chris M, Patrick, Mark, Sujay, AJ
 Regrets: Kevin, Madan, Matt, Trish ← UNC holiday

Agenda:

- UV and ruff for LinkML-map summary (AJ? Patrick?)

- release?
 - need to include linkml updates as well.
 - what are best practices for back-support of linkml-runtime versions?
 - python = "^3.9" vs. python>=3.9, <=3.13
 - notes:
 - take care of library dependencies not to be overrestrictive, but python version is ok
 - need reminders to go on to the next version (October calendar reminder at the very least)
 - for linkml, linkml-runtime - support new first, deprecate next.
 - linting fixes, update to linkml 1.9.1 - then release.
- prioritize next repo to undergo treatment
 - linkml-store -
 - schemasheets - Corey
- poetry 2 upgrade across all repos? (David L.)
 - notes:
 - first do the pyproject.toml standards – poetry 2 helps here?
 - then do the uv migration
 - do not support both at once.
 - do it if it means that our job is easier in moving to UV
- Are there any shared workflows that we could move to the .github repo for easier maintenance/updates? (AJ will take a look)
 - notes:
 - we are not currently consistent here
 - we could be more consistent here
 - centralize ruff config
- notes:
 - extras vs. dev dependencies (duckdb and graphviz)
 - action: ticket for resolving this. we should handle this consistently in our linkml org
- SQL WG updates
 - did not meet today b/c UNC holiday.
 - actions:
 - enum handling
 - make sure documentation is up to date with how keys/constraints/indexes are generated in the SQL serialization.
 - who are the devs working on this? (Sierra will reach out to Kevin for assignments, Sierra and Kevin will rework the CZI board with these new priorities).
- Community meeting recap (sierra)
 - What is our ideal, longer-term vision for modeler-friendly modeling tooling with LinkML? Do we want to invest in Qleverize (.elmo? linkDSL?) DSL/DBML/roll-our-own-ui?

- notes:
 - we like a pluralistic approach, both are still in development, not core
 - web approach fits in with the roll-our-own-ui tool
 - great way for folks to get started
 - need to be feature-full implementation
 - we can imagine it stopping folks from using unsupported/in-development modeling languages
 - positive feelings about teaching here.
 - we have <https://github.com/linkml/semantic-dsl>
 - we are not experts, but we could become more expert
 - see: <https://linkml.io/semantic-dsl/overview.html#usage>
- May meeting
 - Damion Dooley wants to present curation improvements to DH
 - notes:
 - Mark has heard this presentation, its good! - similar to DSL in its helping of folks learn
 - Work has been directed towards DH, and its use cases, might not be generalizable to LinkML
 - separation of concerns between UI and data model/schema?
 - Community is somewhat interested in priorities discussion - Sierra thinks we should do a test of a two-month release cycle with these priorities so we can get a handle on what we can accomplish in that time frame before taking a priority discussion to the wider LinkML community - what do others think?
 - linkml-sql (Kevin)
 - linkml-jsonld (Silvano) - needs a dev
 - Sujay: I have some experience with JSON-LD from past work ex, so happy to volunteer?
 - uv migration (AJ)
 - linkml-rust (Jonny)
 - linkml-map (Matt)
 - deprecate validators package (and other things we've committed to for 1.10.0)
 - maintenance ← critical to good vibes from our community.
 - NMDC noticed this week that prefixes from imports two levels deep do not work (e.g. nmdd.yaml imports core.yaml which imports basic-slots.yaml to output nmdd-full-model. gen-linkml, gen-python both miss prefixes only specified in basic-slots.yaml).
 - host: Sierra ask Patrick G
 - notes:

- do some documenting internally, try something, then bring to community.
 - tag community PRs - need a release tag potentially
 - approved PRs - why don't they get merged
 - we don't want to appear abandoned
-
- Release update (sierra)
 - 1.9.1-rc3 is out - small change to docker image push that Patrick helped rectify.
 - 1.9.1-rc2 works with biolink, oak, nmcd - anyone else want testing of their components specifically done?
 - 1.9.1-rc1 did not release a pypi package correctly, so rc2 corrected that, rc3 corrects the docker image tagging.
 - need to update schemasheets, rest of sub components for python 3.13 – we are already seeing failures in schemasheets.
 - Strategy:
 - 2-month linkml release cycle with release candidates ahead of time.
 - release manager for linkml-runtime needs to test with biolink, nmcd, oak, ontogpt, probably mixs, probably gocam-py (Sierra volunteers for now but will be writing up SOP for hand off).
- Google drive permissions
 - strategy is:
 - if not in google group, no permissions.
 - if in google group, view permissions (on everything? somethings? e.g. paper vs. community meeting recordings?)
 - who should have write/edit permissions and to what - suggestion: monarch internal, linkml-dev group gets write/edit permissions to everything.
- Stats collection - what do we need and where can we get it.
 - dependency graph for linkml proper
 - how do we include sub modules
 - how do we handle redundancy (e.g. sierra's fork of linkml should not count toward dependency)
 - we curated a registry - its now out of date as expected. what do we do.
 - Nico's GH search for usage - should we have some special keyword to look for? (not every linkml project uses makefiles, nor the cookiecutter - do we put some linkml-specific text in somewhere to make this work).

2025-04-11

Attendees: Kevin, Patrick, Nomi, Mark, Sujay, Corey

Regrets: Chris M, Sierra

Agenda:

- LinkML SQL Meeting report out (Kevin)


- AJ, Scott P, James, ?, Kevin, Sierra
- Dynamic enumeration was a big focus - users want to be able to specify a foreign key to a data table
- James will do his own ROBOT thing for this
- All dynamic enumerations in same table vs. each enumeration in its own table
- Nested JSON
- Open PRs:
 - <https://github.com/pulls?q=is%3Aopen+is%3Apr+user%3Alinkml+archived%3Afalse+>
- LinkML, LinkML-runtime out of (unofficial) freeze?
 - 1.9.1-rc1 yesterday replaced 1.9.0-rc1
 - Continue hold until we have a non-rc release

2025-04-04

Attendees: AJ, David L, Chris M, Patrick K, Kevin, Trish, Corey, Nomi, Mark, Sierra, Sujay

Regrets:

Agenda:

- should we move our build/package management system to uv (AJ)
 - use linkml-map as a test bed here (AJ - PR: <https://github.com/linkml/linkml-map/pull/52/files>)
 - in general:
 - convert pyproject.toml files (there is a tool, there is also cborg code :D)
 - give people time
 - move to UV
 - dependencies don't matter (mostly) - if you use a package moved to uv, you can continue to import from pypi
- move to ruff from black and isort?
 - 
- Do we want the cookiecutter?
 - yep
 - do we want cruft?
 - what about copier?
 - how do we maintain folks who use cruft if we move to copier - maybe new repos with new cookiecutter wouldn't even know about cruft, old, well, deprecate
 - Corey has interest in improving cookiecutter
 - The copier-based template experiment is here <https://github.com/dalito/linkml-project-copier>
- What is the best way to deprecate a generator? Two cases –
 - Generator written using SchemaLoader
 - Generator written using SchemaView

- See deprecation guidelines here:
<https://linkml.io/linkml/developers/deprecation.html#deprecations>
- SQL stuff - AJ
 - BER Data (BERTON)
 - **action:** Data Lake House ← we should get a FAQ into LinkML soon to describe how LinkML may be used to support a Data Lake House
 - create, generate DDL from LinkML would be great
 - dbml - but doesn't do the dDI
 - dbml doesn't do the documentation like we can with LinkML.
 - Need ER Diagram ++.
 - Need joining support ++
 - Need FK support for CV/enum tables.
 - Need validators specific to SQL
 - general support for transformation of LinkML+ syntax -> RDBMS
 - **action:** refine the FAQ for SQL + LinkML Modeling frameworks
 - keep SQL focus in LinkML too.
 - have basic types
 - columns are atomic - not reusable
 - advanced features of SQL are sometimes really nice, but leads to modeling confusion (e.g. JSON in a RDBMS)
 - spark - glorified dataframes
 - can put JSON in them
 - SQLAlchemy is ORM for Python
 - need spark, etc...
 - “delta lake” is all table based – needs to be normalized - the data itself isn't boxy though.
 - translator is also recreating a data lake house with medallions.

2025-03-28

Attendees: Sierra, Corey, Chris, Sujay, Patrick G, Patrick K, Mark, Trish, Madan, Kevin, David Linke, Nomi

Regrets:

Recording:


New Items:


- Old PRs to discuss (Corey):
- linkml-runtime PRs that have failing tests but notification from community that they would be great to get into the next release (let's diagnose):
 - <https://github.com/linkml/linkml-runtime/pull/378> - “#378 fixes a real bug and the PRs in linkml depend on the fix being added to linkml-runtime first. So it is the same kind of mutual dependency problem (linkml <-> linkml-runtime), which would be perfect to fix now.” - dalito

- <https://github.com/linkml/linkml-runtime/pull/380> - dalito (trivial bug fix)
 - <https://github.com/linkml/linkml-runtime/pull/371> (metamodel update - should we regenerate a metamodel update and close this one, or fix the conflicts?)
 - update: David L is ok with us merging linkml first (below) then we can do another linkml-runtime release in the next few days. The test fixing is getting a bit circular.
- now the big linkml PR for 3.13:
 - <https://github.com/linkml/linkml/pull/2358> - just a lint fix, created a branch off this PR here that we can interact with if need be: <https://github.com/linkml/linkml/pull/2589>
 - Most upvoted issues - are any of these low-hanging fruit? <https://github.com/linkml/linkml/issues?q=is%3Aissue+is%3Aopen+sort%3Areactions-%2B1-desc>
 - <https://github.com/linkml/linkml/issues/2010> – might just be a matter of better documenting our documentation/visualization generators [Sujoy to take point]
 - <https://github.com/linkml/linkml/issues/2313> – closed by <https://github.com/linkml/linkml/pull/2584> [Sujoy]
 - <https://github.com/linkml/linkml/issues/2029> – [Sujoy] follow up with Ryan (NWB). We added a feature to the cookiecutter to be allow for docs preview on PR, but never added it to the linkml repo which is what the feature was requested on
 - Nico's multi-valued CSV loading request and CSV booleans
 - <https://github.com/linkml/linkml/issues/2580>
 - <https://github.com/linkml/linkml/issues/2581>
 - part of a bigger discussion: <https://github.com/orgs/linkml/discussions/1996>
 - need to assemble all the stuff around this and decide if we want to refactor CSV parsing etc. (E.g. use pandas, stay with csv parser?)
 - do we want to try and fix all the different communities conventions?
 - is this a case for “configuration” ?
 - want more introspection of the domain schema, need less configuration
 - character delimiter isn't a schema element, but an annotation to the schema could be helpful.
 - another idea (Chris and Nico): super-csv (SSSOM + non-schema header which would be a good place for the delimitation specification).
 - we hope folks go towards more structure modeling frameworks.
 - old linkml-validate did pass things thru jsonflattener -> pretty permissive, allows JSON inside CSV (but has been stripped out - no extras, thought would be something else would be layered back in) <https://github.com/cmungall/json-flattener>
 - Kevin's doing multivalued simpledict values for pydantic & typescript because he wants it for monarch: <https://github.com/linkml/linkml/pull/2586>
 - Gregor - Madan
 - <https://github.com/linkml/linkml-model/pull/221>

2025-03-21

Attendees: Sierra, Corey, Chris, Sujay, Patrick G, Patrick K, Mark, Trish, Madan, Kevin
Regrets:

recording:  LinkML Community Meeting 2025-03-19 Pandera Generator: Tim Fliss, ...

- Shex library upgrade or removal [Corey]
 - two options:
 - release pyshexc - 0.5.0 - not maintained, but compliant with the current spec until a new spec comes up
 - remove dependency on pyshexc
 - remove tests for shex generator
 - yes, this will make us susceptible to errors in shex generation, but no one is maintaining shex upstream
 - Corey will make an issue to: move this dependency to an extra or remove it
 - Chris will tag Patrick G to an email thread; hopefully they will make a release, if they don't then we need to remove the dependency.
 - We do NOT want to be maintainers of this library.
 - longer term: make it an add-on, Corey will make an issue.
 - Arrays, iterators, and lists [Patrick G, Corey, Jonny?]
 - pydantics generation of jsonschema doesn't natively support any_shape_array conversion, so we need to sort of fake it as a list of lists?
 -  see PR discussion for more information.
 - Madan: adding FAQs
 - Kevin: roll up transformer - still in progress.
- Community meeting recap [Sierra]
 - more performant
 - Can community members merge PRs with one core-developer review? [David Linke]
 -
- Paper updates [Sierra]

2025-03-14

Attendees: Chris, Nomi, Sierra, Corey, Kevin, Madan, Mark, Patrick, Sujay

Regrets:

- [Community meeting](#) is next week:
 - Host: Sierra Moxon
 - Tim Fliss - pandera generator
 - SIERRA: make sure to remind Nico of this, and get feedback on time/questions/etc.
 - Mark Miller - MixS implementation of LinkML
- <https://github.com/linkml/linkml/issues/2523> (Mark Miller `exclude_type` linter config)
 - PR: <https://github.com/linkml/linkml/pull/2524>
 - Introduces `exclude_type` some rules (`standard_naming`, `recommended`, `no_empty_title`) in the linter config – “skip all items of a certain type”
 - Corey: if there’s duplicative code in the PR, is it generalizable enough that they can be abstracted out into their own functions?
 - Currently has all feedback needed.
 - next step: tests, in progress.
 - This is, in particular, useful for MixS
 - be able to ignore classes for linting, etc.
 - Sujay: Motivation:
<https://github.com/GenomicsStandardsConsortium/mixs/pull/902#issuecomment-2701981170> — lengthy linter reports
- CZI updates
 - Kevin: Kuzu
 - try claude code
 - does not work for Biolink
 - every entity is its own table, every association is its own table
 - instead some sort of refactor for “nodes” table and “edges” table (Matrix/Nico is doing this)
 - check “graph projection” in linkml-store for mapping “predicate” to edge, etc.... perhaps
 - TODO:
 - figure out how to not fork Biolink
 - use “instantiates” or “implements” (metamodel, range is open, but we should have a standard curie/URI for edge to map it to `rdf:Statement`) in Biolink
Associations/predicates/something...
 - “rollup” to “Nodes” class and “Edges” class more specifically with tooling (like KGX?). see:
<https://github.com/linkml/linkml/tree/main/linkml/transformers>
 - need a composition transformer?
 - need to say in mapping or in Biolink that predicate/association maps to an edge not a Node.
 - kuzu needs a roll->up (roll up to node/association)

- generally, this is useful (e.g. in the rel-model-transformer in linkml-map – do we really need a class for every thing, etc...)
 - action: Kevin will document in an issue
- Corey: PR Czar, python upgrade reconciliation, linkml-map TSV
 - outstanding PRs are creating a cognitive tax
 - close PRs at will, document in our Contributing.md
 - python upgrade - we need to take it on.
 - divide into chunks, small PRs
- Madan & Trish: working on schema-automator
 - EML - <https://github.com/linkml/linkml/issues/2168>
 - mapping a domain specific format and a modeling framework
 - <https://eml.ecoinformatics.org/schema/>
 - conditional enumerations
- Patrick G: spun up Data Harmonizer for Zebrafish tox/pheno atlas project, working on a “ontology term picker” widget-thing - can we incorporate that back into DH? ,
- GA4GH survey of participation (Matt) - please contribute your experiences: https://docs.google.com/document/d/1c0jL09jk_SPJOIswHh3wT7t95VGGSrYDQvQadmOXgUY/edit?tab=t.0
 - How can we extend the LinkML community to include GA4GH - is there anything that we do particularly well that we can share with GA4GH more formally?
 - Can we learn anything about supporting standards development from GA4GH?
- Schema Automator debugging (Trish, Madan, Sierra) - others welcome to stay on
 - Two strategies for SA:
 - deterministic:
 - heuristic ones:
 - e.g., generalize from x,y,z, annotate using NLP - claude code - try it.
 - when you can't use an LLM, sure, use SA
 - some level of curation is likely needed - this is a bootstrapper application.

2025-03-07

Attendees: Mark, Sujay, Trish, Madan, Corey

Regrets: Chris, Matt, Patrick K, Sierra, Kevin

** Note: Sierra is in person with Nomi, Melissa, Sabrina: we may not be able to attend. It would be very helpful if someone takes notes and walks through the agenda in our place.

- PR review
 - Removing support for Python 3.8 and adding support for Python 3.13 – at a point where it can be merged (after checking with Jonny)

- <https://github.com/linkml/linkml/pull/2358>
 - Recommendation from PK: make one PR to deprecate 3.8 and another one to add support for 3.13
- <https://github.com/linkml/linkml/issues/2523> (Mark Miller `exclude_type` linter config)
 - PR: <https://github.com/linkml/linkml/pull/2524>
 - Introduces `exclude_type` some rules (`standard_naming`, `recommended`, `no_empty_title`) in the linter config – “skip all items of a certain type”
 - Corey: if there’s duplicative code in the PR, is it generalizable enough that they can be abstracted out into their own functions?
- CZI updates
 - CZI priorities for LinkML board:
 - <https://github.com/orgs/linkml/projects/35/views/1>
 - LinkML Map for subsetting SEPIO - (Matt)
 - [Kuzudb generator](#), SSSOM in LinkML-Map - (Kevin)
 - Kevin & Corey working to add tsv functionality to LinkML-Map
 - Any PRs that need group discussion? - (Corey)
 - <https://github.com/linkml/linkml-runtime/pull/345>
 - <https://github.com/linkml/linkml/pull/2358>
 - These are the remove 3.8 support and add 3.13 support. I’m going to try to break these up into removing support for 3.8 from linkml first then remove from linkml-runtime and finally try to add support. I’ll try to get all of the changes from Johnny as well...
- March Community Meeting @ March 20th
 - Tim Fliss (pandera generator) - need dev group ready to answer questions, so review the Draft PR in prep:
 - <https://github.com/linkml/linkml/pull/2543>
 - Eric Cavanna (refscan possibly, but would prefer to talk at this call instead)
 - Mark, have we talked about LinkML + MlxS at a community meeting yet?
 - would you be interested in presenting this month (reuse an older GSC presentation)?
 - Action item: Mark Miller talk to Sierra Moxon
- Open issues sorted by popularity:
 - <https://github.com/linkml/linkml/issues?q=is%3Aissue+is%3Aopen+sort%3Areactions-%2B1-desc> - pls talk about taking on one or two issues if you currently don’t have LinkML work to do.
- `gen-linkml` followed by `gen-prefix-maps` of [nmdc-schema](#) fails in linkml 1.8.6 and 1.8.7 Mark Miller
 - <https://github.com/linkml/linkml/issues/2560>
 - `ValueError: Class "NucleotideSequencing" - multiple keys/identifiers not allowed (nucleotideSequencing_id, NucleotideSequencing_id)`

- biolink-model needs to do an linkml upgrade to get new pydantic features - are we good to try this or do we have a released issue? (Sierra, Patrick G)

2025-02-28

Attendees: Sierra, Kevin, Corey, Mark, Patrick, Sujay, Trish, Nomi

Regrets: Chris, Madan, Matt

Agenda:

- LinkML Community Meeting Recap and notes:
 - ☰ LinkML Community Meeting rolling agenda - (Sujay)
 - 3 sections of the meeting:
 - tutorial/demo of schemalink - Marcio and Emanuel
 - graphical editing interface that looks like cypher-query-interface and it creates a LinkML schema for you.
 - LLMs incorporated (predict the schema/relationships/hierarchies/node properties to make it comprehensive)
 - highlights of the releases 1.8->beyond
 - good feedback, questions - unified LinkML CLI tool (issue created)
 - tutorial on various viz tools on LinkML (Sujay)
 - nice to draw some correlations between diagrams between schemalink tool and our UML generators!
 - ~30 people
 - Next meeting: TBD - former workplace (Sujay)
 - March (Sierra)
 - CZI updates
 - [Modeling for property graphs](#) - (Matt)
 - [Kuzudb generator](#) - (Kevin)
 -
 - Any PRs that need group discussion? - (Corey)
 -
 - LinkML-Map
 - GA4GH: ☰ Toward a Profile Authoring Methodology and Tooling Support
 - NCATS Translator: 🟡 DogPark Strawman - (Sierra)
 - Next week we're going to do a linkml-map hacka-collab-athon - want to be invited? (Sierra)

- Corey, Kevin, Trish to be included
- LinkML collaborative project at ISMB CollaborationFest 2025 - (Deepak, Chris, Sierra) - Liverpool England (ISMB/BOSC - July 21-22 - 22 is a combined session with BioOntologies and Knowledge Representation - Chris M is the keynote speaker. July 23-24 is CollaborationFest & LinkML meetup).
 - TISLab folks: you might want to use CZI funds to send someone. (community development and meeting was funded as “travel” in that grant).
 - April 17 - ISMB talk submission deadline (posters are also an option, prob easier, but submit for a talk and abstract will be automatically considered for a poster if it doesn’t get a talk spot). Note: we already did LinkML in 2023, so it might be hard to get a talk spot.
 - BDC/INCLUDE ingest might be a good poster topic.
 - ACTION: Corey - talk to Matt about CollaborationFest.
 - (Sierra) ACTION: start a google doc with submission ideas.
 - (Sierra) ACTION: coordinate with Deepak on “things to do” at collaborationfest.
- SciPy (Tacoma, Washington - July 2025) [Abstract\(s?\)](#) - Mar 5 is the deadline (Sierra)
- Open issues sorted by popularity:

<https://github.com/linkml/linkml/issues?q=is%3Aissue+is%3Aopen+sort%3Areactions-%2B1-desc>
- `nmdc-schema` won’t build under `linkml` later than 1.8.5 Mark Miller
- From linkml community slack:
 - Linda Xiang: Hi team, does anyone have experience using `linkml generate json-schem` ? The above command will generate JSON Schema representation of a given linkml model using JSON schema version <https://json-schema.org/draft/2019-09/schema>, is it possible we can specify the JSON schema version instead of using the default version? Mark Miller

2025-02-21

Attendees: Sujay, Patrick, Trish, Mark, Kevin, Madan, Corey
 Regrets: Chris, Sierra

Agenda:

- Review schemalink
 - Was able to load `nmdc-schema` but layout was an unhelpful circle as previously reported

- Does it know how to process imports?
- RAG based “schema suggerster” feature built into tool
- We have schemasheets as a spreadsheet interface to creating LinkML schemas, SchemaLink is a graph based interface serving the same purpose
- Corey’s linkml, linkml-runtime PR czar activities
 - Will check failed PRs and poke owners, suggesting rebase
 - Address old drafts
- Kevin’s current broken env (due to coverage package?)
- Corey recommends `poetry install --with-dev`
- Ended at :35 after

2025-02-14

Attendees: Sujay, Patrick, Trish, Mark, Nomi, Chris, Sierra, Kevin, Madan

Regrets:

Agenda:

- KBase is using LinkML 🎉
- PR review
 - Very old, should it be abandoned or assigned to someone else?
 - <https://github.com/linkml/linkml-project-cookiecutter/pull/16>
 - linkml-store ([3 PRs](#) that I would appreciate a review of) (Sierra)
- CZI updates?(Matt/Kevin/Corey/Madan) - [project board](#)
 - Kevin - pydantic improvements
 - kuzu exploration - sqllite for graphdb (no automatic schema generation).
 - but with linkml we could write a kuzu generator!
 - put in a ticket
 - TODO: create ticket<https://docs.kuzudb.com/import/paraquet>
 - pandera generator PR (with Tim Fliss)
 - lets redo KGX? simple database composition. do like Kevin does.
 - schema for flattening, bring up properties from normalized source needs specification.
 - pyspark types - give generator a mapping table/typedict
 - Kbase also using data lake house, minio, medallions, pyspark. Essentially a big environmental genomics data warehouse. semsql/linkml/etc.

- Madan - gen-json-schema should materialize patterns by default - for 1.9.0
- Corey - PR Czaritude is going great!
- Plans for next week's [community meeting](#)
- Draft of PG docs
 - https://hackmd.io/ijEwkQnMRFyPtRKqMWV_MQ
 - property graph -> rdf -> other serialization discussions
 - <https://linkml.io/linkml-store/how-to/Use-Neo4j.html>
 - <https://hackmd.io/@cmungall/rJKeSe6Fkx>
 - <https://schemalink.anacleto.di.unimi.it/#/local/id=70-h2xqRERVTHOR8fONe>
 - cypher-ish in postgres
https://duckdb.org/community_extensions/extensions/duckpgq.html
 - bson queries in postgres (from Microsoft)

2025-02-07

Attendees: Chris, Sierra, Corey, Kevin, Madan, Mark, Patrick, Trish, Nomi (late)
regrets: Sujay

Agenda:

- Phil Stoemert Q form 2024-01-31 (Mark)
<https://obo-communitygroup.slack.com/archives/C04EU7JL1NF/p1738310415706729>
 - memory hole of slack is super frustrating
 - add to discussions on GH
 - add to Stack Overflow and answer your own question (repeated from slack) yourself.
 - put these into FAQs or discussions by default
 - for this call:
 - review any unanswered question on slack regularly
 - linkml-validate is stricter than linkml-convert or unit tests
 - some parts of our stack use dataclasses (auto-coercion), but others do not (linkml-validate) auto-coerce, so the schema doesn't validate, but the python dataclasses compile.
 - postelion: be liberal in what you accept, strict in what you coerce to.
 - [Different validation tools may complain or tolerate different patterns in data #2532](#)
 - sometimes json-schema validator error codes are obtuse - sometimes also consider how you design a schema (this also affects the error results)

- [linkml-validate error messages are not always clear or actionable #2533](#)
- typescript Questions from slack
 - Currently fits needs for TISlab.
 - Need more use cases for this
 - Working group maybe?
 - features should be accordant to a “playbook” for feature parity with the source language.
 - compliance suite helps you test for the “feature parity” if you lay it out ahead of coding.
 - get ahead of the “wait, this isn’t supported” convos - instead we have a compliance suite that says “this is what we support”
- JSONSchema “additional property” support
 - held up a bit by scope creep: <https://github.com/linkml/linkml/issues/2241>
 - we need the simpler thing implemented, the meta-model change has been started with Kevin and Chris
 - <https://github.com/linkml/linkml-model/pull/214> in progress here I think?
 - command line option for the generator that only affects the top level thing (tree root). not each individual element.
- CZI updates?(Matt/Kevin/Corey/Madan)
 - how goes the PR Czaritude?
 - What to do with all the drafts?
 - How should we manage PR’s that fail tests?
- deptry: shall we incorporate in all linkml repos?
 - <https://github.com/linkml/schema-automator/pull/151>
- Update LinkML [community meeting agenda](#) for Feb [Sujoy OOO]
 - Sujoy working on writing `mermaidclassdiagramgen.py` in advance of this month’s community meeting – see issue [here](#) 🎉
 - PR: <https://github.com/linkml/linkml/pull/2529>
 - Can we also present CMDR modeling work? Or other recent developments? (Sierra)
 - Community Meeting invite, new list? - core core: smoxon@lbl.gov, pkalita@lbl.gov, cjmungall@lbl.gov, spatil@lbl.gov, mam@lbl.gov, corey@tislab.org, kevin@tislab.org, madan@tislab.org, matt@tislab.org Plus (currently - this may change as folks come in and out of the project voluntarily) community core (Jonny, Silvano, David Linke, Vincent Keheller): j@nny.fyi, silvano.cirujano-cuesta@siemens.com, david.linke@catalysis.de, vincent.kelleher@gaia-x.eu, sigfried@jhu.edu + linkml-community google group.

- Quick PR review: <https://github.com/linkml/linkml/pull/2514> [Sujay OOO] – time permitting
- Shahim: I would like to discuss: <https://github.com/linkml/linkml-owl/issues/39> (I will be able to join the call around 1:35 PT)
 - Any workarounds or alternatives?
 - might've been considered an edge case during original design.
 - funowl has been converted to another, faster, rust-implementation (pyhornedowl) - PR in progress on linkml-owl to swap from funowl->pyhornedowl (lots of good code refactoring here). It's easier to deal with this one after that one.
 - Shahim will take a look at owl-dumper.py (dumpers are in linkml-runtime repo) and/or branch for supporting pyhornedowl.
 - <https://github.com/ontology-tools/py-horned-owl>
 - does the end user need to be aware of the rust implementation?
 - no.
 - e.g., oak installs semsimian but the python wrapper of rust, etc. (e.g. pre-build the wheel files, failures here are very messy but oak is well tested, etc.)

2025-01-31

Attendees: Corey, Chris, Trish, Patrick K, Patrick G, Sierra, Mark
 regrets: Madan, Sujay

Agenda:

- CZI updates - (Matt/Kevin/Corey/Madan)
 - CZI board: <https://github.com/orgs/linkml/projects/35/views/1>
 - Please keep the board up to date
- PR czar (Sierra/Nomi) - Corey is on the hook for February. Thank you!
 - Make sure all PRs in all linkml repos are assigned a reviewer if they are not in “draft” state, and have passing checks.
 - Do not assign Chris.
 - Add agenda items here for PRs that need discussion with the larger group and/or Chris.
 - Its ok/good to assign open-source developers as reviewers.
 - Follow up with reviewers you assign on slack - make sure reviews are happening (everyone is busy; its good to be a squeaky wheel).
 - For reviewers:
 - you don't have to be an expert in the code to do a review.

- you should try to test it locally if you don't have experience with the code you're reviewing, but lean on the automated tests to help if you don't have time.
 - if you absolutely can't review the code, take responsibility to assign someone else and follow up with them.
- Python 3.13 upgrade status - what do we need to do now? (Patrick G)
- <https://github.com/linkml/linkml/issues/2479> Class & slot names that aren't valid python, iterate through class/slot aliases to find one that's valid python and just complain? Try to generate a valid name (what we've done with enums, it's a little weird)
 - <https://github.com/linkml/linkml/issues/896>
 - short term - warn via linters
 - short term - make sure we have "alias" support in all generators (more computer friendly term)
 - NOT **aliases**
 - alias vs. aliases - take care here, one is for internal use, like this use case, the other is just used in a schema itself to help with search/display/etc.
 - the example of slot_slot is a good one to see how "alias" and "aliases" work.
 - compliance test for aliases also would be a good example. (all feature/framework combos are here for reference in making sure all support this convention). - actually we think this works across all generators (might just need a better doc/FAQ for this – e.g. an example) - pydanticgen should fail here for now.
 - generators should be able to handle language-specific reserved words (but maybe the linter could do this as a whole - using a word somewhere we should report it.)
<https://github.com/AnanthaRajuC/Reserved-Key-Words-list-of-various-programming-languages>
 - python has a "keyword" module too (helpful for python)
 - long term - can't reserve every word in every language we support.
- Linter - how do I edit (Mark)
 - how do I say "exclude elements from the schema from checking" in a config option
 - (Patrick) - start with the configuration model (there is a makefile to regen the model just like all other linkml models)

- regen the classes
 - then go splunking and make some new methods.
- Schema-automator (Trish)
 - TODO: add Trish to LinkML CZI group
 - using “annotator” module of the SA package
 - <https://linkml.io/schema-automator/packages/annotators.html#annotators>
 - Chris: every LinkML schema can be turned into OWL/RDF
 - once ontologized, use lexmlatch, etc. other tools to do the annotation manually. (gen-owl, gen-rdf)
 - <https://linkml.io/linkml/generators/owl.html>
 - then push the mappings back in manual
 - OR, merge SSSOM (literal->id mappings may not be specified in SSSOM, yet.) support into schema-automator annotator framework
- schema automator is for schemas/schema elements
 - schema automator would be this use case: e.g. “hole in heart” is a phenotype of a participant, I want to construct an enum of all the conditions of all the participants in all the studies with ontology ids instead of strings.
 - but if we’re just mapping labels to terms, something like lexmlatch (not schema-automator) would be better.
- schema-automator is a bootstrapping tool.
- possibly do experimentation in a notebook/colab so we can help debug/question/etc and then feed back into LinkML FAQ or schema-automator FAQ
- Chris: long term, we should probably fold schema-automator back into the core library (just a “take this serialization, and generate this other one”)
 - we might want to incorporate “curategpt”-like tooling here (though hte constant warning is that it likes to hallucinate ids).
- e.g. in INCLUDE - we have a list of conditions from the participant (disease/phenotype/MAXOs) in string format
 - assign ontology terms to these

2025-01-24

Attendees: Anne, Sierra, Mark, Patrick K, Sujay, Madan, Sigfried

Regrets: Corey (sick)

Agenda:

- Anne - agenda item for review - BDC
 - take 9 datasets with one spreadsheet/table with “patient”, and other “harmonized” variables in the rest of the table.
 - model is here: <https://rtiinternational.github.io/NHLBI-BDC-DMC-HM/>
 - still work to do, of priority is “observation”
 - in particular, interested in some sort of “rule” to say that if x is filled out, they need to fill in y
 - diastolic BP should be a value quantity, and fill out a value decimal
 - if 0 throw an error
 - <https://github.com/linkml/linkml/blob/main/examples/rules-example.yaml>
 - due in July
 - Can I ask a question about this same model? (Sigfried)
 - ER diagram generation: there might be issues here, see tickets
- Nextgen Linter (Mark, Chris)
 - statistical analysis of schema to warn about “missing” metamodel components
 - should folks have a tool for specifying their own, more restrictive, version of the metamodel for fodder to lint?
 - <https://github.com/linkml/linkml/issues/2469>
 - for MlxS something simpler is ok.
 - linter does talk to the metamodel
 - not really a way at the moment to feed in another meta model
 - statistical anomalies bit:
 - prototype it as a standalone tool? make a new linter rule
 - users should be able to provide their own rules as plugins
 - add to the standard linter framework
 - use case1:
 - MlxS convention is: “description” required, but not in metamodel, so we need a linting rule specific to MlxS
 - enumerate the kinds of linkml element that needs a description would be immediately helpful.
 - use case2:
 - no descriptions on enumerations in MlxS
 - linter defaults always result in a warning about each Enum having missing description
 - overwhelming - but in some cases we want them - we should look into configuration
 - recommended rule has include/exclude options.
<https://linkml.io/linkml/schemas/linter.html#recommended>
 - is there a rule index?
 - linkml-lint –help is limited, but go to teh doc above.

-
- CZI direction recap - Matt/Kevin/Madan
 -
- Pandera generator - <https://github.com/linkml/linkml/issues/671#issuecomment-2597580433>
- Feedback for Tim Fliss on pytest conversion questions: <https://github.com/linkml/linkml/issues/1722#issuecomment-2611220504> - Patrick will take a look.
- Cookiecutter engagement continues to engage: <https://github.com/orgs/linkml/discussions/2203#discussioncomment-11890565> (we probably want to wait for `corey@tislabs.org` to discuss)
 - Mark is interested in just as an experiment.
-

2025-01-10

Attendees: Chris, Sierra, Mark, Corey, Kevin, Madan, Patrick, Sujay, Nomi (last few minutes)
 Regrets:

Agenda:

- CZI priorities announcement (Matt, Sierra)
- Community Meeting (Sujay) – *speaker might not be ready to present in January, but could be ready in ~~February~~ March. Need to check with him, but it looks like he's still on vacation. Update – in Taiwan until March*
 - host: Sujay
 - speaker: Thomas Yu (Sage Bionetworks) – slated for March
 - tutorial needed?
 - Should we do a recap/tutorial on the various visualization generators in LinkML as part of the next community meeting? (Sujay)
 - Happy to do some work on splitting out (from docgen.py) mermaid class diagram generation into its own standalone generator (mermaidclassgen.py) – it's been a request from community members on the slack channel
 - cancel this month - be available for hackathon
- DDD recap (Sierra, Patrick)
 - [December Developer Days - 2024 Summary](#)
 - We need to make a new LinkML release.
 - volunteer: KEVIN voluntold
- GREGoR (ANVIL) meetup (Sierra)
- Should Harold's repos like hbreader and jsonasobj be moved under LinkML umbrella so that they don't cause (aka: we take on support for) unfixable degradation of LinkML? (dalito, tikapix, sneakers-the-rat)
 - <https://github.com/linkml/linkml/issues/1966>

- Vending proposal
 - Make a copy of jsonasobj2 in linkml_runtime
- Nextgen Linter (Mark, Chris)
 - statistical analysis of schema to warn about “missing” metamodel components
 - should folks have a tool for specifying their own, more restrictive, version of the metamodel for fodder to lint?
 - <https://github.com/linkml/linkml/issues/2469>
 - for MlXS something simpler is ok.
- just vs. make - having both in the cookiecutter (dalito, Corey)
 - <https://github.com/linkml/linkml-project-cookiecutter/pull/127>
 - we like the idea of making a just file (just) for users that need it vs. for all users all the time in the cookiecutter.
 - separate package that cookiecutter could import. (written by dalito)
 - concerns about dual build systems getting out of sync
 - concerns about just as a “don’t know of it” new tool
 - some enthusiasm to try new tools
 - from Chris: longer term, we’ll have more logic in python vs. make anyway.
- poetry 2.0 upgrades
 - need volunteer to do the pyproject.toml work in all LinkML repos
 - should we commit our cookiecutter to poetry 2.0?
 - is it time to look at uv more seriously?
 - volunteer owner of this mini project:
- DBML generator and viz tools (Sierra)
- Cookiecutter PRs with common “blocker” (decide on sub-folder for ad-hoc scripts)
 - <https://github.com/linkml/linkml-project-cookiecutter/pull/105>
 - <https://github.com/linkml/linkml-project-cookiecutter/pull/104>

2024-11-22

Attendees: Sierra, Patrick K, Mark, Matt B (briefly), Kevin, Nomi, Sujay, Madan, Corey
 Regrets: Chris M

Agenda:

- Community call went well
 - Several new people want to be assigned tickets!
- December Developer Days
- PR to update linkml-model docs to use latest Jinja templates:
 - <https://github.com/linkml/linkml-model/pull/208>
 - Unrelated error from some metamodel changes in extra_slots
- Feedback from Alicia (NMDC) regarding documentation generation:

- Currently, the deployed documentation pages are built from the contents of the `main` branch of the schema repo as opposed to using a pinned release, which can be confusing. She would like to enable Mike-style documentation pages which have a dropdown of the various versions (included `latest` which pulls from `main`). This is something that we need to prioritize as an issue and add into the cookiecutter

2024-11-15

(focus on LinkML Rust Generator)

Attendees: Jonny, Daniel, Corey, Chris, Sierra, Patrick K, Mark

Regrets: Sujay (working from India), Kevin, Madan, Aaron

Agenda:

- Developing a LinkML Rust Generator
 - scoping ticket: <https://github.com/linkml/linkml/issues/2360>
 - `#linkml-rust-generator` channel in obofoundry slack: <https://obo-communitygroup.slack.com/archives/C080Y28GC3T>
 - regular meeting time?
 - ticket creation and assignment?
 - teaching/review with non-rust developers - when should we do this?
- Minutes:
 - We're all on the same page: learning rust through something practical that we need to do. e.g. a generator!
 - In rust you have a lot of control, but can get into trouble. can implement your own logic for really basic types (e.g. what is the logic for boolean! you can define it).
 - JS: has a basic generator already started (mirrors the standard generator paradigms) - what should be the focus of this call?
 - design of the generator/what is the mapping to rust - what constructs are we going to use:
 - baseline mappings are:
 - `structs` as classes, `type aliases` have worked out for slots but unsure. not sure how to add metadata to a `type alias` or make a custom struct?
 - CC: what additional metadata will programmers expect to interact with when we've written the generator?
 - e.g. the uri attached to a thing – want to know both the literal value and also infer its relationship to the schema it comes from (is it representing another

LinkML class? what is that class and how is it used in the schema?).

- JS: probably need most of the info present...hard to predict use cases. could also allow the metamodel to serve the role instead.
 - CC: That is mostly how our other generators work... at least how the typescript generator works. We don't expect a developer to need the deeper parts of the schema when they are just using the schema to hold data.
- CJM: most programmers don't need to do a lot of introspection, but "we" will want RDF generation from the rust objects...then introspection is important. but we could focus less on introspection if it causes blocking.
 - what are things you can express in LinkML but not in Rust? (e.g. JSONSchema unrolling hierarchies).
 - is any_of not expressible in Rust? what do we do here – JS: this is largest most common language feature that Rust does not have.
 - DD: Type union sounds similar to a Rust enum to me, there is also a "union" type, but accessing the value of this is "unsafe."
 - CM: any_of can be translated, just need to figure out how.
 - DD: in memory represented as the largest enum variant, just one PV available at a time.
 - JS: need more logic to unwrap the value of that - DD: yes, a match statement or something similar - first figure out which enum variant (PV) dealing with, then go to the corresponding match expression arm.
 - CC: Daniel, is what you were talking about earlier, putting your own extensions on a type, a way to hold the extra metadata that Jonny was talking about?
<https://doc.rust-lang.org/reference/items/enumerations.html>

- JS: need some specific macros for implementing LinkML functionality
 - in the same way we have naming issues, there wouldn't be a name for "any_of" – this will be a temporary object.
 - JS: another issue: lots of derived statements that might be desirable
 - interested in making a python compatible mapping as well, py03 doesn't have a tidy way of expressing a ... might need to help py03? make dataclass macros work better and use those in the rust generator. For example: autogenerating the class constructor - handle the type construction in the struts
 - don't want the models to be *bound* exclusively to python too strongly b/c there is an overhead in performance there
 - might want multiple versions of the schem ain a crate:
 - py03, purely rust (use these internally, expose to python models, not sure of the performance here)
 - DD: so we need to decide if we generate py03 part - controlled by the user somehow? Not a pure dataclass?
 - simplest is to create a strut with public fields - no getters and setters, easy to access the values. but for py03, we need something more complicated needed to interacting with python. [FromPyObject](#) and [ToPyObject](#) can help (idiomatic rust - "things just work")
 - seems like a "feature" the user can turn on/off (JS: yes this is how it currently works).
 - DD: maybe some folks won't want [serde](#) - feature flag – but they should want serde.
 - CC: lots of stuff in the generator? too much?
 - what is the target for the generator to generate?
 - JS: target is to generate the metamodel and use SchemaView on that. Care has been taken to keep metamodel small and easy to understand.
- CJM: best java generator is written in Java - but ...
 - best generators are in Rust...? but maybe not. maybe the Jinja stack might be ok.
 - do the minimum amount to get the metamodel and then go from there?

- DD: generating code will be additive - have the base, then add extras, independent of each other. Rust features require this. crate has a base version, features should be additive not removing things. use same design pattern, starting from the meta model.
- CC: can we generate the metamodel without py03 and serde? Are these addons?
 - JS: we can generate a version of the metamodel without py03 (any ext. dependencies), take liberties with basic data types like "date" in particular – DD: look at protobuf, they use python package, include primitives like "date".

CURRENT METAMODEL IN RUST:

<https://github.com/p2p-lid/linkml-core/blob/main/src/metamodel/src/lib.rs>

- JS: small scope is good 👍 - almost there
 - next step... linkml rules? harder
 - using python generator as a reference?
 - in the middle: validating string patterns, value constraints, cardinality constraints – good for every day users
 - how do we implement this? (e.g. like pydantic on every field we validate – this could be version 2)
 - DD: when and who does validation?
 - JS: ideally just like pydantic, at runtime, at initialization
 - DD: to me this sounds like a type system - can make a custom type that is rule – JS: eg a type that has a regex pattern. DD: has to be a new type, then write a wrapper to do this validation in rust.

https://doc.rust-lang.org/rust-by-example/generics/new_types.html

 - DD: feet and miles have the same type under the hood, but different pattern validation.
 - JS: is this a macro that lives in a separate crate? DD: int is an int or a float or a float is another validation we have to do to.
 - JS: but is that validation code in the linkml crate, or is it in another crate? reuse here? crate could be very duplicative if in the same

crate? but all code in one place is simpler.

- CC: still confused if we are building a generator to make the metamodel?
 - making the generator to generate the metamodel, THEN using schemaview with the generated metamodel, we can maybe swap this in to the generator:
<https://linkml.io/linkml/howtos/port-linkml.html#step-2-generate-data-access-classes-for-the-metamodel>
- DD: what is the plan for testing this?
 - JS: all ears. currently planning on plugging into the existing LinkML compliance tests that test the output is compliant.
 - replace the existing LinkML schemaview, then put in the rust metamodel and if all the tests pass then we are on the right track.
 - need of course more specific unit testing for generator and generated metamodel
 - start with lives in python -> generates rust
 - compile the rust, use the models in rust, test they are working
 - could be slow
 - pydantic uses pytest to test their rust methods
 - CJM: tests designed to make intermediate files that could be copied over into the rust repo to make a compliance suite. Don't want to add overhead burden to the linkml library.
 - JS: would be great to separate the testing here. 👍
- Tasks? Next steps? (JS: will coordinate more meetings as needed, for now work offline)
 - JS: happy for contributors. right now just in a single feature branch. probably don't want half-ready code to LinkML repo. Steady rust dev branch in LinkML?
 - CJM: leave it to you. ok with leaving a branch as clearly experimental, ok to leave it in main and not add it into the docs, etc.
 - JS:
 - put the first PR (from fork) into main (experimental)
 - single large ticket, need to split a bit more (JS will do this)
 - branch from there.
 - what parts are there?
 - JS: the generator part - ✅(we've talked about this above)
 - JS: the second repo - the schemaview part - in a new repo (CJM: merge into the main LinkML org when ready)
 - similar to linkml-arrays repo
- CJM: GFF validators are an ideal use case for this!

2024-11-08

Attendees: Sierra, Chris, Mark, Corey, Kevin, Madan, Patrick, Nomi, Mark
Regrets: Sujay (working from India)

Agenda:

- [Sierra] - unique_keys and a linkml-mongo-index-generator
 - <https://github.com/linkml/linkml/blob/6789c8db23631f0d00d02a7b571ce22081f76780/linkml/generators/sqltablegen.py#L217> - unique key generates an index in SQL automatically. What about a simple generator that just converts some linkml language component into an index on an attribute in a mongo collection (second iteration would be to make combined indexes).
 - unique as a parameter on the keys metamodel
 - discussion topic for linkml - should functional indexing be part of the metamodel.
- [Sierra] - November Community Meeting
 - Silvano said yes to presenting last month, but haven't confirmed yet
 - feelers out to Noel (linkml cookiecutter friend), Cyril (he has a magic-the-gathering schema that would be fun to hear about as an end-of-the-year change of pace). Deepak is a colleague and is reaching out to him. I also tried to contact on LinkedIn.
 - We could do a discussion about top priorities for DDD - I could assemble some options and talk through them on the call, getting community feedback about priority.
 - Yes - do this.
 - Slight chance we could recruit AJ to talk about object-oriented→ relational database challenges with linkml. But I would rather wait until the new year to coordinate with the Alliance.
 - hesitate to make these two too close together (warehouse vs. normalized)
 - ask AJ if she could do a presentation/conversation at Nov.
 - Kbase data warehouse
 - Does anyone want to be the host?
 - Minutes:
 - Chris is away
- [Sierra] - Prioritizing community feature development by the community
 - right now its done through "core devs have to review and merge PRs"
 - do we feel like time spent reviewing tickets/PRs is manageable?

- is it ok and we take that the package is getting better “for free” (just our time to come up to speed with changes)?
 - some other ideas:
 - One suggestion was that we plan quarterly targets for development and prioritize those getting done before new features are reviewed. We go back to project boards. We have one for “help wanted” tickets.
 - yes, lets try getting ahead - but not suppress the good work
 - prioritize is different than PR review
 - extend an invite if we need to
 -
- [Patrick] `rdflib` upgrade (<https://github.com/linkml/linkml/pull/2375>)
 - Not out of the woods yet?
 - With regenerated snapshot files, the tests pass but they take ~3x longer now
 - could be another library? - maybe. we need to do some benchmarking or isolate a smaller bit for a PR, just to try to figure out what is slow.
 - if it is rdflib, we need more help.
 - Patrick G has rdf experience so maybe could help.
 - Corey could help.
- Issues/Slack/New development of the week
 - Rust generator - do we need more involvement here - does anyone want to step up and pair program with Jonny. <https://github.com/linkml/linkml/pull/2364>
 - positive feedback from TISlab about rust development teaching
 - Justin also has some rust experience
 - Daniel Danis has good rust experience - but could support the library if we had to support after Jonny goes on.
 - rust for phenopackets will be great.
 - might be a good way to bring Daniel - use phenopackets as a test case
 - KGX replacement made out of rust
 - Also obographs
 - Can we use in <https://github.com/ielis/ontolius/tree/master/src/ontology>
 - Corey +1 to learning through a project
 - **action:** reach out to Corey, Kevin, Jonny, Patrick G, Daniel, Aaron to suggest phenopackets as a “project” use case for rust linkml development. ask jonny to help coordinate the pair-programming/presentations/teaching – we could do this in a dev call!
 -
 - (Silvano) - type designator is wrong <https://github.com/linkml/linkml/issues/2399>
 - (karanthpallavi) - support “annotations” in gen-shacl <https://github.com/linkml/linkml/issues/2402> - good issue for SHACL community.

- (Vincent K) - attributes defined more than once in more than one class should generate errors in all generators? - <https://github.com/linkml/linkml/issues/2403>
- PRs to review
 -

2024-11-01

Attendees: Sierra, Patrick, Mark, Chris, Kevin, Corey,

Regrets: Sujay (working from India)

Agenda:

- [Patrick] Any RDF experts interested in taking a look at <https://github.com/linkml/linkml/pull/2375?>
- [Patrick] [Dependabot for version updates](#)
 - Could be useful to keep projects intended to be *libraries* tested with the latest version of dependencies
 - Patrick has been testing it out here: <https://github.com/geneontology/gocam-py/blob/main/.github/dependabot.yml>
 - For projects that are *applications* and you only want to update dependencies that have known security issues see the older Dependabot Alerts functionality: <https://docs.github.com/en/code-security/dependabot/dependabot-alerts/about-dependabot-alerts>
- Updates to `gen-linkml`: <https://github.com/linkml/linkml/pull/2341> [Sujay]
 - Request for review
 - Adding a *master* option called `--materialize/--no-materialize` that works in the way described in the PR. There are tests associated with this change as well. Do let me know if we want the functionality to behave differently
- this week's issues
 - [Corey] - document what kinds of things (curies/uris/class names/etc.) can be used in ranges <https://github.com/linkml/linkml/issues/2391>
 - [Jonny] - is it time to clean up our utils directories and make more independent packages out of linkml and linkml_runtime? <https://github.com/linkml/linkml/issues/2371>
 - [Silvano/Jonny] - PV serialization - <https://github.com/linkml/linkml/issues/2382>
 - [Jonny] - what do we do with validator results in linkml.converter.utils <https://github.com/linkml/linkml/pull/2385>

- PRs to review
 - [Jonny] - test speed up snags with rdflib (possibly) <https://github.com/linkml/linkml/pull/2388>
 - more test speed ups: <https://github.com/linkml/linkml/pull/2389>
 - [Patrick G] - some pydantic-generated changes to help with dev environments from MS: <https://github.com/linkml/linkml/pull/2390>
 - needs movement of one bit from the python to the jinja template
 - [Jonny, candleindark] - global config? <https://github.com/linkml/linkml/pull/2335>
 - TODO: <https://github.com/linkml/linkml/pull/2335> > devs should try this out and see if it works via checking out and testing it. review next week.
 - Sierra: invite Jonny to a dev call to review splitting repo into modules/etc. (also ask about global config once reviewed)
- Random extra
 - [Kevin] SchemaView like shared util for getting flattened types? I want to re-use the logic from pydanticgen (or equivalent) to know what raw type a field a slot will produce. Does that already exist? Does it belong in OOCCodeGen? Is it above that? Should schemaview support something like induced_type?
 - is it a class, enum, or type (recursive)
 - result = type(sv.slot.range)
 - then second_result = type(result)
 - then raw python type (second_result)
 - data model that is more like python data model
 - any_of [optional: foo]
 - not as nested in LinkML
 - in python terms, give me back the range, not in linkml terms (possibly the same use case for Eric's range checking at NMDC)
 - use case is whether MATRIX needs full generator, or just some methods

2024-10-25

Attendees:

Regrets: Sierra, Chris, Mark, Patrick K

Agenda:

2024-10-18

Attendees: Mark, Madan, Chris, Sierra

Regrets: Corey, Patrick K, Sujay, Kevin, Aaron, Patrick G

Agenda:

- [Mark] - Community Meeting Recap and Follow Up
 - Manuel - has funding, wants developers
 - webapp: “share my cloning”
 - metadata collection around cloning experiments
 - classes:
 - source, fasta sequence - annotate with what kind of analysis you’ve done, add cell lines and chromosomes, and organisms.
 - making good use of LinkML
 - looking for collaboration
 - see community meeting notes for more detail
 - TODO: CMDR should follow up with Sid’s work and details about his schema - see (this is Terra)
<https://github.com/DataBiosphere/biocore-data-model/blob/main/content/linkml/BioCoreFields.yaml>
 - Justin/Harry might also be interested here w/re to phenopackets.
 - We could have a page of “general” high level bigger picture of how things are deployed/developed/scoped etc. for community meeting leaders
- [Sierra/Matt] CZI funding 🎉🎉
 - These are the highest priority:
 - B.1 (developer days) - ✅
 - B.4 (basic maintenance)
 - B.7 (unittest to pytest)
 - B.8 (get rid of snapshot tests)
 - B.5 (finish moving generators to SchemaView)
 - A.1 (Typescript Generator)
 - A.2 (R generator / Rust Generator)
 - C.1 (simple web-based linkml data validation)
- PRs to review:
 - <https://github.com/linkml/linkml/pull/2281> - Vincent approved, why is it being weird?
 - <https://github.com/linkml/linkml/pull/2369> - Jonny adds lifecycle methods (e.g. “before class do this”) to JSON schema generation - come back to this next week
 - <https://github.com/linkml/linkml/pull/2329> - Silvanoc fixes if_absent processing

- <https://github.com/linkml/linkml/pull/2347> - Silvanoc parallelizes tests for local development! (not CI yet)
- Slack this week:
 - Request for work on enum support in Typescript generator
 - Also Kevin notes that TS generator from 1.8.4 release does not produce output. (1.8.3 works) - Sierra make an issue
- New issues this week:
 - Requests for core developer to maintain conda-forge version of prefixmaps: <https://github.com/linkml/prefixmaps/issues/77>

2024-10-11

Attendees: Sierra, Chris, Patrick, Corey, Sujay

Agenda:

- [Mark] - Next week is Community Meeting - good to go?
 - Follow up with hosts
 - @Manuel Lera Ramírez
 - @Sid
 - No December LinkML Community call
 - December Developers Days (how is that related to Triage Tickets Together?)
 - [LinkML - December Developer Days](#)
 - Regrets
 - CJM
 - Sierra
 - Patrick
 - Corey
 - Post to Monarch slack ??? 5 minutes before ([Sierra Moxon](#) will set up in advance)
- [Sierra] The story of pydantic and dataclasses for Biolink this week - how do I make this better?
 - Biolink distributes both pydantic and python dataclasses serializations in pypi
 - `__init__.py` generated (by poetry?) includes `from .model import *` but not `from pydanticmodel_v2 import *`
 - `from biolink.datamodel.pydanticmodel_v2 import Gene` - weird, but works.
 - but by default, python first processes all the dataclasses at runtime 😭
 - notably spent a bit of time lost in the weeds of –meta additions to gen-pydantic
 - if_absent: <https://github.com/linkml/linkml/pull/2329>
 - any good ideas on how to escalate breaking changes that the community finds?

- some sort of special label that we can use to elevate a ticket to an automatic slack message? – trick is how do we prevent everything from being special.
 - fail fast:
 - tests in Biolink - generate the python, try to use it
 - need a generic test with a broken serialization (maybe in cookiecutter)
 - tests in LinkML - these exist to eval the python
 - might be a hole in our test suite
 -
- this week' new developments:
 - [Sujay] - default materialize options - <https://github.com/linkml/linkml/pull/2341>
 - [Sierra] - schemaview tests to pytest - <https://github.com/linkml/linkml-runtime/pull/343>
 - [Silvanoc] - <https://github.com/linkml/linkml/pull/2365>
 - who merges approved PRs - can the submitter, just the PR reviewer, just us core folks, or just Chris?
 - action: put that in the Contributing guidelines
 - [request/community] - decentralize imports - <https://github.com/linkml/linkml/issues/2367>
 - [Jonny] - update the minimal python support to 3.9 vs. 3.8, increase support to 3.13 as well - <https://github.com/linkml/linkml/pull/2358#issuecomment-2398909404>
 - every october!
 - [bug?] - linkml valid <https://github.com/linkml/linkml/pull/2358#issuecomment-2398909404> changes the source schema: <https://github.com/linkml/linkml/issues/2359>
 - [bug?] - plantuml and its kroki server continues to irritate PR submitters: <https://github.com/linkml/linkml/issues/2342#issuecomment-2396742398>
 - [bug?] - makeflags in Makefile of cookiecutter: <https://github.com/linkml/linkml-project-cookiecutter/issues/124#issuecomment-2395160549>
 - Example of how to do tests in a cookie cutter repo <https://github.com/monarch-initiative/dragon-ai-cookiecutter/blob/main/examples/Makefile>
 - [Alicia/NMDC] - range: Enum + no **required** annotation - can the slot be None? <https://github.com/linkml/linkml/issues/2361>
 - follow up with Alicia to add “required” to data_category
 - [Jonny] - rust generator: <https://github.com/linkml/linkml/issues/2360>
 - rusty people at TisLab :D
 - maybe Madan?
 - Corey is interested in it
 - Justin can maybe take a look at this - compelling for phenopackets.
- older interesting ones:

- should data files that conform to a linkml schema have metadata built in to reflect that? <https://github.com/linkml/linkml/issues/1833>
- closable?
 - <https://github.com/linkml/linkml/issues/1821> - we now validate and materialize patterns right?
 - something about boorgats - <https://github.com/linkml/linkml/issues/1885>

2024-10-04

Attendees: Mark, Chris, Kevin, Madan, Corey, Sujay, Patrick, Sierra

Agenda:

- CZI funding has started
 - Prioritize issues with preference given to projects that we are actively supporting (NMDC, MlxS, DataHarmonizer, etc.)
- December Developer Days
 - tag issues with “developer days” label - before December.
 - get through as many as possible (they should be small, take between 1-8 hours to complete)
 - be available for some amount of time on zoom between 8-2 pm PT / 11-5 pm ET on Fridays in the month of December (holidays notwithstanding) to be “helpdesk” support for any user that wants to join.
 - Mark Miller will make an announcement about the December developer days at this month’s community call
 - Presentation on Pandera at a future community call?
 - Backup: Corey to present on SEPIO
 - Suggestion from Kevin:
 - Typescript generator on the frontend receiving data from FastAPI that passes through Pydanticgen
 - End-to-end toy example demonstrating a full stack web app using above components
 - LinkML recommendation: technologies will need to be evaluated on a per project basis
- this weeks new issues:
 - “additional properties” (Jonny) metamodel support - <https://github.com/linkml/linkml-model/pull/205> seems like we should approve “option 2”? - upvote option 2.
 - gen-owl defaults: <https://github.com/linkml/linkml/issues/2353>
 - Creates slightly fake “hold your nose” classes. Can visualize nicely using Protegé
 - Might want to consider changing the defaults for MlxS

- Add uber option called “materialize” which *combines* “materialize-X” (patterns and attributes) instead of removing the options entirely as is the case in the linked PR at the moment: <https://github.com/linkml/linkml/pull/2341>
- Document Container class with examples = <https://github.com/linkml/linkml/issues/1385#issuecomment-2392970664>
 - Chris will expand on the idea here
- mermaid/plantuml many<->many relationships - <https://github.com/linkml/linkml/issues/2350>
 - Is it a problem in the generators (Mermaid ER and plantUML) or is it a problem in the SchemaView methods?
 - Sujay will comment on issue and make PR to address it
- GraphQL generator: <https://github.com/linkml/linkml/issues/2346#issuecomment-2388103891>
- Pandera generator [Kevin via Matrix] - <https://pandera.readthedocs.io/en/stable/>
 - <https://github.com/linkml/linkml/issues/671>
- Issues updates
 - using boolean constraints on required state <https://github.com/linkml/linkml/issues/2282>
- recent discussion bumps:
 - linkml-map with DH use cases: <https://github.com/orgs/linkml/discussions/2337#discussioncomment-10823289>
 - cookiecutter feedback: <https://github.com/orgs/linkml/discussions/2203#discussioncomment-10842625>
 - Corey helped make it “just work” now and documented it - Makefile vs. bash etc...! 🎉
 - Chris: some common threads in feedback:
 - initializer, uber linkml command - e.g. we automate/website-itize this instead of making a cookiecutter noob friendly
 - Nico’s ODK community support is also helpful to feed into this conversation
 - but... who is the target?
 - python dev?
 - or is it at either end - expert but want it to be in their framework, and those who don’t want to be python experts
 - !!! don’t focus on an entire refactor without a community discussion about who are users should primarily be.
 - **ACTION:**
 - we need tests! (see dragon-ai cookiecutter for help)
 - get the cookiecutter to produce *runnable* tests (currently they fail)
 - Corey could take a crack and simplifying the config suite - as a stretch
 - possibly we need more than one cookiecutter to separate concerns and users. (e.g. those with schemasheets, those

with pypi pushing, etc... or some sane combination of features).

2024-09-27

Attendees: Chris, Mark, Patrick, Sujay, Kevin, Sierra

Agenda:

- Structured patterns and materialization
 - <https://github.com/linkml/linkml/issues/1557>
 - linkml-convert uses old validation module, still calls gen-json-schema
 - but... it may not pass the same parameters as linkml-validate in terms of materializing/inducing
- Jonny's issue about metamodel change for "additionalProperties"
 - <https://github.com/linkml/linkml-model/pull/205#issuecomment-2377823345>
 - action:
 - reach out to Jonny to come to dev call
 - reach out more broadly to the community about the use of this as a "flag" vs. a metamodel component, or a simple metamodel component vs. restrictions around the content of new metamodel component.
 - can we satisfy the parody issue now, but leave the hooks in for next time
 - more examples maybe?
 - trying to propose the minimal version of this to match additionalProperties in JSONSchema - yes, no or a schema.
 - prefer not to have any_of in the metamodel

2024-09-20

Attendees: Chris, Corey, Sujay, Sierra, Mark, Madan, Kevin, Patrick

Agenda:

- Community call
 - Recap - TNO
 - Thank you to Patrick and Corey for being flexible and providing content!!
 - Useful to see, see if he engages further
 - October:
 - host: Mark
 - topics:

- Sid Cox - confirmed - AnVIL (<https://anvilproject.org/>) data model and ontology development with LinkML
 - Anvil - NIH data compute system
 - NIH data with right logins, etc, run analysis
 - SBOL
- Corey - SEPIO and schemasheets
- Other ideas?
 - Sierra also reached out to Silvano Cuesta for a presentation - awaiting response.
 - Deprecation discussion?
 - Required vs. required with caveats discussion?
- Appetite/time for December Developer Days?: Sierra
- November: Patrick
- January: Sierra
 - Semantic web & LinkML - (perhaps Patrick G could do a walk through based on his learning over the next couple of months?)
 - chris introduces, then we have some “experience with linkml” short talks
- Check if my understanding of the **alias** metamodel slot is correct: <https://github.com/linkml/linkml/pull/2322> [Sujay]
 - The **alias** name needs to be used in the instance file and not in the schema file
- Deprecation
 - Goal: produce Discussion item in LinkML GH and invite stakeholders to discuss offline, and then in person as needed
 - stub discussion: <https://github.com/orgs/linkml/discussions/2324>
 - LinkML (so far):
 - metamodel slots:
 - slot: deprecated (string)
 - slot: deprecated_element_has_exact_replacement (uri/curie)
 - *slot: last_updated_on*
 - *slot: modified_by*
 - tool support
 - deprecated elements (not PVs) go into “deprecation.yaml”
 - generators optionally include elements in “deprecation.yaml” in imports (e.g. documentation brings in deprecation.yaml, json-schema does not).
 - user issues:
 - users see a difference between “deprecated: true” and “deprecated: this is the reason for deprecation”
 - users want a way to deprecate PVs but not the referencing enumeration
 - OboAcademy (<https://oboacademy.github.io/obook/howto/obsolete-term/>):

- reason for deprecation (rdfs:comment) -
 - advice is to add GH issue as a reason
<https://github.com/obophenotype/cell-ontology/issues/####>
)
 - also: Annotations (rdfs:comment)
 - also: has_obsolence_reason - not sure the type of thing this is?
 - also: term tracker item (type xsd:anyURI)
- replacement entity ('item replaced by' axiom)
 - or 'consider' tag
- 'OBSOLETE.' to the term definition
- Discussion:
 - “Deprecation” metamodel class to hold all the possible metadata about a deprecation?
 - change range of “deprecated” from string to “Deprecation” class.
 - Add “deprecation_reason”, “tracker_item” as slots, change range of “deprecated” to boolean.
 - “Derivations” in linkml-map
 - can we find parallels between “deprecation” and “derivation” framework/model?
 - minimally, should derivation tooling support deprecation syntax?
 - This should fit into an audit history/trail of metadata (DeprecationEvent as an Event)
 - can imagine a UI that helps us make these
 - KGCL for LinkML...could be big
- make a discussion, we don't deprecate that often - use what we have, then see how the discussion goes in the GH discussion item
- Required slots that have mixed range types
 - Goal: produce Discussion item in LinkML GH and invite stakeholders to discuss offline, and then in person as needed:
 - stub discussion: <https://github.com/orgs/linkml/discussions/2325>
 - LinkML (so far):
 - “required” is boolean
 - slot range is not directly modeled with its required status (e.g. boolean, string, date-time slots can all be required/optional)
 - User desired behavior:
 - parent element is not-required, descendent element is required (or vice versa). - this one is ok - just don't get less constrained as you go deeper.
 - subclassing something to make it less restrictive is an anti-pattern
 - linkml-mapping pattern is to make it less constrained

- standards need to be restrictive, but if too restrictive no one uses them.
 - permissive but enough structured metadata to generate reports after the fact.
 - e.g. slot is required or it is a value from the `can-not-provide-because` enumeration
- TODO: fix this in the `any_of` induced_slot method
 - support `any_of` (range):
 - date
 - string
 - never infer upwards
 - explicitly declare the “highest” - “any”.
 - incomplete check should not give a false positive
 - think about overriding default range when there is a `slot_usage`
 - doing both atm - need to formalize. (find issues)
- Koza and TSV/JSONlines [Kevin]
 - schema + classes -> becomes -> SSSOM TSV
 - column order? (rank)
 - needs to be added to TSV writer
-

2024-09-13

Attendees: Chris, Sierra, Mark, Patrick, Madan, Kevin, Sujay, Corey

Agenda:

- LinkML Community Meeting agenda
 - Sierra Moxon: LinkML Registry [spreadsheet](#)
 - TNO - 🙌
 - high level single cli interface [Patrick] - just an announcement and a short discussion
 - sepio-linkml and schemasheets [Corey]
- LinkML NMDC
 - structured_aliases:
 - `alias_predicate` - but fails
 - `predicate` works
 - metamodel -
 - docgen issue: <https://github.com/linkml/linkml/issues/2316>
 - in linkml any element can have an alias (not the same as “aliases”)
 - `alias_predicate` == `predicate` in the “alias” of the metamodel element- this takes precedence over the name of the element (docgen needs to show you the alias in the medamodel if it exists).

- See *alias* in LinkML YAML for [alias_predicate](https://linkml.io/linkml-model/latest/docs/alias_predicate/#linkml-source):
https://linkml.io/linkml-model/latest/docs/alias_predicate/#linkml-source
- if you are specifying the slots element:
 - - (actual name of “slots” is “slot_definitions” – :D – b/c the name “slots” is already used in the metamodel)
- induce_slots anyOf range element - populate range_expression, not range, range should still be “union”-- b/c it is single valued this needs to be the most generic of the possible ranges (e.g. NamedThing)
 - Sierra: get the PR up to date, fix conflicts, get merged.
 - NMDC: should query the range_expression field not expect it from range.
- Are there any exemplars in linkml talking about policies on how to use LinkML [subsets](#)?
 - not really, no - we don't need them for NMDC

2024-09-06

Attendees: Madan, Sierra, Mark, Chris, Corey, Kevin, Shahim

Agenda:

- schemasheets any_of with range constraints – union_of is worthwhile
 - Corey C will take this on at least in terms of making an issue, etc. Chris can hack along with him.
 - any_of ranges are really helpful for NMDC - but we are trying to get rid of them
- concern with linkml-convert CSV – Madan will try TSV
- how to specify a prefix for instances of a subclass that are different than the default prefix - Shahim
 - owl-dumper creates a document, overrides ontology iri
 - use a curie instead for slot_uri and it should work - default behavior
 - in general, user is in charge of prefixing when generating instances
- discussion of “in subset” as a metamodel component vs. implementation of logic in code
 - with focus on FHIR, Snomed – with the intention of understanding if the concept of “in subset” is a paradigm employed with a different syntax/architecture than “in_subset” in biological ontologies/OWL.

2024-08-30 - Canceled

2024-08-23

Attendees: Chris M, Sujay, Patrick, Anne, Mark
regrets: Sierra (vacation)


Agenda:


- Ask Matt if he wants to bring the GA4GH CPO model to one of these dev meetings sometime
- CDE discussion
 - CaDSR - already done
 - NLM - needs to be retrieved
 - PhenX - maybe - see if someone is willing to explore
 - Anne will try to find someone with FTE to do this

2024-08-16

Attendees: Sierra, Chris M, Chris R, Sujay, Patrick, Mark, Anne, Kevin, Corey, Trish

Agenda:



- Best way to [get meta.yaml if runtime has been imported, with no network traffic](#)
Mark Miller
 - https://github.com/linkml/linkml-runtime/blob/3db7d7c87c7d38d103389fe6f8baec8c50b8409/linkml_runtime/utils/introspection.py#L30-L38
 - ```
from linkml_runtime.utils.introspection import package_schemaview
```
  - ```
psv = package_schemaview('linkml_runtime.linkml_model.meta')
```
 - ```
print(psv.schema.name)
```
  -
- **linkml-map** for subsetting schema? Mark Miller
  -  LinkML tutorial (community meeting July 2024) - linkml-map
  - <https://github.com/sierra-moxon/linkml-map-example/tree/biolink-subset-example> ← branch with SchemaView + SchemaBuilder + linkml-map to subset biolink
- linkml-community call recap
  - schema registry - follow up with Damion Dooley
    - talked with him about linkml-store for DH
  - tutorial on linkml-store
    - backend management for linkml
  - Daniel Ji - CZI doing some work with CryoET modeling
    - CryoET is an imaging platform (electron microscopy to make tomographs)
    - <https://github.com/linkml/linkml-runtime/pull/330#event-13904442859> - Jessica Gradling is another schema developer “platformics”:  
[https://github.com/chanzuckerberg/platformics/blob/main/platformics/codegen/lib/linkml\\_wrappers.py](https://github.com/chanzuckerberg/platformics/blob/main/platformics/codegen/lib/linkml_wrappers.py)

- CDE harmonization paper
  -  CDE harmonization paper
  - <https://github.com/monarch-initiative/cde-harmonization>
  - Mark Miller to Everyone (Aug 16, 2024, 1:25 PM): <https://www.bodc.ac.uk/>
  - Trish Whetzel to Everyone (Aug 16, 2024, 1:25 PM): <https://cde.nlm.nih.gov/home>

2024-08-09

Attendees: Chris, Sierra, Patrick, Madan, Sujay, Trish


Agenda:

- Linkml-registry updates:  LinkML Schema and User Registry
  - Repos using linkml: <https://github.com/linkml/linkml/network/dependents>
  - Goal: first pass curation done by two weeks from today
- Merged in <https://github.com/linkml/linkml/pull/2262> and <https://github.com/linkml/linkml-runtime/pull/335> [Sujay]
  - Thanks to Jonny and Chris for the reviews!
- CMDR updates [Sierra]
  -  INCLUDE-BDC Merge to CMDR

2024-08-02

Attendees: Sierra, Chris, Mark, Sujay, Kevin, Corey, Nomi

Agenda:

- Announce: neo4j wrapper for linkml-store
  - <https://linkml.io/linkml-store/how-to/Use-Neo4j.html>
- Discussions
  - <https://github.com/orgs/linkml/discussions>
  - <https://stackoverflow.com/questions/tagged/linkml>
- CMDR/LinkML-Map updates
- Review open PRs: <https://github.com/linkml/linkml/pulls>
  - <https://github.com/pulls?q=is%3Aopen+is%3Apr+user%3Alinkml+archived%3Afalse+>
- PR review: <https://github.com/linkml/linkml-runtime/pull/335> [Sujay]
  - Approved by Jonny and Chris
- TODO: Make an issue on the GitHub tracker to capture a list of the current problems in [gen-plantuml](#) [Sujay]
- This week, or on the 16th:
  - <https://github.com/linkml/linkml/issues/2213> CURies/AnyURI  Mark Miller



- Chris: we should have two types in the metamodel – current metamodel CURIE (relaxed CURIE) and the W3C CURIE (strict CURIE)

2024-07-26

Attendees: Sierra, Sujay, Mark, Patrick, Corey, Kevin, Nomi, Chris

Agenda:

- <https://github.com/linkml/linkml/issues/2224> [Mark]
  - Gist: there's a problem in the `induced_class()` method in SchemaView which needs fixing (see issue for details)
  - Jonny/Mark/Sujay (everyone else welcome) to work on this
- CLI refactor from Johnny: <https://github.com/linkml/linkml/pull/2227> [Sierra]
- (depending on Chris's availability) `range_expression` vs. `range` in SlotDefinition as returned by SchemaView. Similarly, `attribute` vs. `slot` in ClassDefinition. [Sierra]
  - <https://gist.github.com/sierra-moxon/7179854c4d8000ab4942a8cc7b9a7a94>
  - should range be multivalued?
- vendor -
  - move python `linkml_files.py` and the like to runtime
  - move tests that exercise `linkml_files.py` to runtime
  - move path to include "latest"
- Might need some [more permissible values](#) for `pv_formula` for <https://github.com/INCATools/ontology-access-kit/pull/785> [Mark]
  - Or fstring template
- Why aren't Sujay Patil's awesome documentation Jinja2s being used in LinkML (linkml-model) documentation pages? <https://github.com/linkml/linkml/issues/2222> [Mark]
  - update linkml in dev site
  - 1.8.1 release with no content changes and a trigger a rebuild of the doc. [Sujay]
- Add a page to the linkml docs where we capture links to all our LinkML recordings
- Need to add a link to our youtube presentation at ISCB Academy on our homepage

2024-07-18

Attendees: Kevin, Sierra, Corey, Patrick, Mark, Sujay, Madan, Trish, Chris R,  
 Regrets: Chris M

Agenda:

- LinkML Community Meeting Recap [Sujay]
  - Nice continuation of last month's meeting - GaiaX - more clear what this actually is now. We can go to another hackathon if we want to.
  - Focusing on Shacl generator - since we have users here, maybe we need to focus on this generator.
  - Talked about ISMB

- results of the cookie cutter survey:
    - we don't advertise it well
    - slim-down is needed
      - remove schemasheets prompt
      - get rid of cruft or exchange it for ...
    - folks aren't really importing their schema via PyPI, raw YAML files
  - linkml-map
    - Demo & tutorial:
  - next month:
    - host: Corey!! :D
    - agenda:
      - Chris to talk about semantic web and LinkML - probably 30 mins
      - TNO presenter -
- PRs ready for review - volunteers?: [Jonny, Patrick]
  - <https://github.com/linkml/linkml/pull/2208> , <https://github.com/linkml/linkml/pull/2178> , <https://github.com/linkml/linkml/pull/2019> (approved but waiting on another)
  - one merged - thanks Patrick!
  - two others - dependent on the one that got merged, but failing tests/Drafts – now Jonny back to changing the PRs. - Patrick will be tagged and review.
- Some discussion of how to notify team that PRs are urgent
- handling `any_of` and `exactly_one_of` [Sierra] - code review
  - Use case: use SchemaView to enforce referential integrity. Requirement came out of NMDC schema which extensively uses `any_of`
  - <https://gist.github.com/sierra-moxon/7179854c4d8000ab4942a8cc7b9a7a94>
  - [Mark] Whatever decisions we make here regarding a concluded range (union or intersection of `range`, `any_of.range` etc), code and tests have to support instances whose type is any subclass of the concluded range
  - [Madan] & Mark universal quantification “all values from”, “some values from” - OWL (Sierra: check these)
    - used in an OWL “restriction”
      - owl:someValuesFrom
      - owl:allValuesFrom
    - qualified properties that can't be expressed as a binary
    - schema level and instance level
  - [Patrick] & Sierra
    - Not confident of the reason or usage of `range_expression`, nor why someone would query this
  - [Mark, Patrick] - This should all boil down to whether we agree on the logic of the tests - we will do a bunch more tests and ask Chris too :)
  - [Corey] - where do we expect to fail - make sure assert not's are there.

2024-07-12

Attendees: Madan, Sierra, Patrick, Kevin, Trish, Sujay, Mark

Regrets: Chris

- ISMB Workshop recap (see notes below)
- Cookiecutter revamp prototyping discussion
  - <https://github.com/orgs/linkml/discussions/2203>
  - Community meeting - shall we solicit feedback about the cookiecutter from our community members?
    - survey? live Q/A on the call? Other?
      - Have you used the cookie cutter?
      - Which generators do you use?
        - Mark: I recommend including most of them in the build process by default, so users can see when they start using language elements that are incompatible with one of the “serializations”, or really “data world views”
      - Do you write your own Python tests?
- Modularizing generators?
- What do we do with generator output (as a best practice)
  - what about using *mike* for viewing several versions of documentation ... to view different versions of our artifacts. (cp artifacts to gh-pages branches only - w3id to JSON schema artifact via /docs dir ... slightly weird but... )
    - Sujay: Documentation is fine. But the linkml project artifacts that we are creating are mostly developer artifacts right, so it's okay to expect people to fish that out from GitHub, or build it locally themselves? There is the possibility of making the web/public-facing documentation very developer focused and intimidate users who just want to browse the standard (classes, slots/terms, etc.). Also like Patrick said, the use case for using different versions of the various artifacts simply hasn't come up yet (for NMDC)
  - generate artifacts at the time of the tag – but only at the time of the tag?
  - can we do this without checking into the main branch of the repo - this might be preferred!
  - set of best practices -decision tree for various cases - maybe there are different kinds of linkml projects:
    - are you centralized?
    - do you use more than one serialization?
    - do you support outside project users?

Notes from ISMB workshop

- Sierra:

- Explaining the difference and the way to modify gen-project, the generator Makefile targets, and configuration for gen-project, and customizing these appropriately is tricky - we need a story here, or a streamlining of responsibility between these kinds of build methods
- [https://linkml.io/linkml-model/latest/docs/designates\\_type/](https://linkml.io/linkml-model/latest/docs/designates_type/) is a bit unintuitive (e.g. when you are subclassing in order to be more specific about a particular domain, it is easy to forget to add this metadata)
- lots of SMEs would like to be modelers, but they tend to still get overwhelmed by all the different ways to say the same thing - maybe we expand the linter (or something like it) to suggest model design patterns?
- a big selling point in LinkML is that we can import other LinkML models -- but finding out where those are and what folks are working on is difficult (luckily we plan to revitalize our registry, but more non-manual processes here would be wonderful)
- our cookiecutter is great, but explaining the need for cruft is also difficult (we may be too young of a project for this to pay off the way it will later in our evolution?)
- small things: why don't we include jinja documentation templates in our cookiecutter? why do we distribute a Person schema with failing tests? how do we explain prefixes and their need to novices? can we make non-technical development even easier (e.g. "I don't know what pypi is, why do I care about python serializations")
- do we always see linkml as an intermediate form (e.g. will we continue to use JSONSchema for a validation framework vs. writing our own custom validator, etc.)
- we have 16 standards, none of them work, so let's make our own ( :slightly\_smiling\_face: we clearly understand this
- Sometimes our expressivity in the language outpaces our support for our generators - there is new-user confusion about this. e.g. "subclassing is a nice feature, but its not supported in JSONSchema, so...." what is the LinkML "advice" to this kind of user - do we teach them how to a common serialization of the LinkML model that does support subclassing (E.g. pydantic) or try to educate/document/etc... ?
- Patrick
  - I think we need to cast a UX eye on the cookiecutter, beyond what Sierra already mentioned about the tooling (cruft vs. cookiecutter) and things that don't work out of the box (failing tests). I think we've tried to stuff so much into the cookiecutter that it can be overwhelming to a new user. Someone asked the question (paraphrasing) "Why did it make so many files and directories for one schema file? What is all this stuff?" It makes me wonder if/how we can [progressive disclosure](#)-ize the cookiecutter.
  - On a meta level, running a workshop virtually (even setting aside any questionable conferencing software) is *haaarrrd*. Maybe other people have it figured out more than me, but I found it so hard to tell if people were

following along (or if they even cared to follow along). There were times that if we were in person I might have just walked around and talked to people, but virtually you sort of have no choice but to just plow ahead. We should think about that when we choose what workshops we run and how we tailor the content.

2024-06-28

Attendees: Mark, Sujay, Patrick, Sierra, Chris

Regrets: Corey, Kevin

- Brief discussion around LinkML-CZI funding hassle/issues/kinks
- Mark Miller Attempts to automate the cookiecutter-based, TSV-centric [MIFC](#) project
  - Kai wants the primary data format (for examples) to be TSV, but `linkml-run-examples` requires data examples in YAML format, so modifying the build process through Makefile targets to convert TSV to YAML using `linkml-convert`
  - Also build process in general
  - Patrick: cookiecutter has some weirdness (like `__init__.py` file getting deleted) which needs to be fixed. Noticed this while developing tutorial for ISMB. Simply hasn't been tested enough
  - Mark: what is the best way to bring the "latest and greatest" features from the cookiecutter into downstream repos? Haven't seen enough of cruft in the wild to be confident about recommending it to other users
  - Documentation process in general
- Mark Miller Tools for managing LinkML YAML module files
  - Break monolithic schema into topically sensible modules that compile independently
- <https://github.com/orgs/linkml/discussions/2177> [Sujay]
  - Biolink has a Github Action that checks the YAML schema files against a `.yamllint` config:  
<https://github.com/biolink/biolink-model/blob/master/.yamllint-config>
  - Linkml-model does too:  
<https://github.com/linkml/linkml-model/blob/main/.yamllint-config>
  - We want to add this for MlxS and NMDC schemas too
  - YAML auto formatters like `yamlfmt` are less reliable so use with caution
    - Recommended by Sam Willison: <https://til.simonwillison.net/yaml/yamlfmt>
- 1.8.0

2024-06-21

Attendees: Sujay, Patrick, Chris, Sierra, Mark, Madan, Corey, Nomi (late)

Regrets: Kevin

- Debrief of LinkML community call [Corey]
  - Lots of excitement from users!! ❤️
  - Surprised to see Bart not having pushed back/up contributions into linkml org/repo
    - What's the best way (politically) to get Bart (and other users/potential contributors) to contribute back into the repo
    - How to contribute back to the org? - create a howto! [Corey]
- Next LinkML community call
  - Topics
    - TNO
    - LinkML and the semantic web [Chris] – put down for August
    - Nico on multilingual LinkML
  - Moderator - Sujay
- LinkML users meeting is not a bad idea [Nomi]
  - Host at LBNL? - ISB in Kansas in 2025 - workshop! in addition to user meeting
- <https://github.com/linkml/linkml/issues/2103> - any\_of and one\_of in “range” schemaview retrieval method
  - induced\_slot method already returns most specific, but does not include any\_of and one\_of yet – we should add that
  - if a user wants more, its on them
  - symbolic transformer - well defined transforms from first order logic - if there is a deep hierarchy or one\_of + any\_of mixes in the hierarchy, then it can be implemented.
  - also think about decreasing usage of any\_of in NMDC, but still support this!
- referential integrity checker as part of the core LinkML service (maybe as linkml-store)
  - Eric C from NMDC has a prototype
  - sometimes people point linkml-validate at a CSV file (uniqueness and referencing something else, you can assume the collection of CSVs are self-contained, you could do some validation). [Patrick]
  - or, I just want to validate a little bit of data against a larger store of data (e.g. on ingest?) [Patrick] – harder, how do you get access to the stuff already there.
- INCLUDE doc has been updated! [Madan, Matt, Trish]
  - <https://include-dcc.github.io/include-linkml/about/>
  - also interested in versions (e.g. in LinkML model - using “mike” to do versioning) - wants users to be able to pick the version of INCLUDE model
    - <https://github.com/biolink/biolink-model-toolkit/blob/ba442022c9e1a8b707ad17be0349031f5ffa169e/bmt/toolkit.py#L39> ← instatiates a Schemaview object based on a full URL to the version of Biolink we want to bring up (any tagged release, etc) (same kind of thing could be done to initialize a schema object to pass to validation).
  - Let's make some mappings to OBI and CMDR [Chris M]

- Let's include more explicit units - check the howto guide on modeling quantities and measurements - simple explicit scalar pattern - helpful documentation and it can be used by linkml-map to do data conversion
    - Some static enums to constrain values would be good too.
  - ParticipantExternalId is really a reference to the Participant class - but is a string - should be range of Participant [Chris]
  - Next time: more samples :) [Madan]
- DBMI/Gregor(?) [Madan/Chris]
  - using a drawing format as a schema
  - provide schema then convert to JSONSchema then covert to DBMI for visualization
  - if DBMI is just JSON, that's a great addition to schema automator
    - import gregor and make a linkmlification of gregor so we can use tools to compare it, etc. [https://madanucd.github.io/gregor\\_linkml/](https://madanucd.github.io/gregor_linkml/)
  - gen-dbmi -> kind of stale, it seems like a good schema-automator fit
- EML and incorporating Data Dictionaries [Chris M]
  - bundling a linkml schema next to a dataset (e.g. the data dictionary/schema that models the headers of an associated CSV)

2024-06-14

Attendees: Chris, Mark, Patrick, Sierra, Sujay, Nomi

Regrets: Kevin

- LinkML runtime release causing issues [Patrick]
  - Fails in 320 are false positives
  - linkml test pass with latest linkml-runtime (main) - time to make a new release
- Demo NMDC work that helped improve linkml-cookie-cutter [Sujay]
  - Generate documentation preview on PR
  - Ontology build repos have a "diff" function - we could introduce that here too, or this might be helpful for ontology repos as well.
- Review Jim S's gen-plantuml PRs [Sujay]
  - Some back and forth between open source devs on the standard for PlantUML
  - Likely the quotes introduced is a result of spaces in names of linkml elements
    - we should take a look at the PlantUML spec
    - we want to escape the spaces, but not see quotes in the diagram. quotes are likely not in the spec.
- Community meeting agenda:
  - Bart K on electric utility modeling with LinkML
  - Anja S on SHACL-LinkML hackathon (Gaia-X)
  - Corey C. is the host.
- CZI open science report out
  - The world loves LinkML ❤️
  - LinkML users meeting is not a bad idea

## 2024-06-07

Attendees: Chris, Mark, Patrick, Sierra, Sujay, Nomi

Regrets:

- Finish up PRs from folks failing on linting/tests
  - we can checkout their branches and do so: <https://github.com/linkml/linkml/pull/2125>
- Prioritize [#2146](#) - from NMDC (validation around a slot)
- [ISMB tutorial slides](#), [CZI flash talk slides](#)
  - Message to send to tutorial participants a week before: [draft](#)
- LinkML Community Meeting
  - June 20
    - possible: Bart K or <https://github.com/linkml/linkml/pull/2125> folks (who won 3rd prize in the [the 7th Gaia-X Hackathon](#) at [Tech-X 2024](#)) or Nico (SSSOM schema)
    - Chris - semantic web and LinkML

## 2024-05-30

Attendees:

Regrets:

- <https://github.com/linkml/linkml/issues/2142>
- Default license for distributing LinkML Schemas?
- 

## 2024-05-24

Attendees: Sujay, Sierra, Madan, Kevin, Patrick

Regrets: Chris, Mark

- Question from Madan: what do the various ingest (ingest\_jsonld, ingest\_biportal, ingest\_bioregistry, etc.) ingest pipelines do in schema-automator?
  - Seeing some usage in prefixmaps
- Question from Kevin: still running into problems loading/building his schema using SchemaLoader in linkml-solr. Schema in #linkml-dev
  - Working fine on v1.6 but running into issues on v1.7

## 2024-05-17

Attendees: Mark, Sierra, Patrick, Kevin, Corey, Sujay

Regrets:

- [Mark] - Root class of "Database" all its slots are multivalued



- Modeling pattern being adopted going forward – polymorphic slots
  - Ex. PlannedProcess and its subclasses
- validation pattern on an abstract class - caused some odd behavior
- <https://github.com/linkml/linkml-model/blob/main/examples/unique-key-example.yaml>
- Community meeting debrief
  - Montana on NMDC
  - Jonny on LinkML devops
  - linkml-store - Chris
- [Mark] - GitHub issue about class typing tests

2024-05-10

Attendees: Sujay, Mark, Corey, Kevin, Patrick, Madan, Sierra, Chris (late)

Regrets:

- [Madan] -
- [Sierra] Community Meeting next week: who is presenting for NMDC? (Montana might be out)
  - host: Mark
- [Sierra] order of precedence/SOT for metamodel annotations; e.g. “range” or “required” or “multivalued” slot attributes can be determined by any of the following:
  - slot definition
  - attribute definition (currently allowed to conflict with a slot with the same name)
  - slot\_usage in a class
  - any\_of/exactly\_one\_of (which can be overwritten by slot\_usage)
  - expression syntax
    - proposal (last/highest number is SOT):
      1. slot definition
        - a. range definition
        - b. then any\_of/exactly\_one\_of
        - c. then expression syntax
      2. slot\_usage in a class
        - a. range definition in slot\_usage,
        - b. then any\_of/exactly\_one\_of,
        - c. then expression syntax
      - .....
      3. if attribute redefines an existing slot - raise error.
      4. if slot\_usage has range and any\_of/exactly\_one\_of has range - raise error?
        - a. minutes:
          - instead of errors, raise warnings? (but people ignore us)
          - is this a documentation, linting, or model issue?

- can we write tests that verify this, but not enforce it as an error/warning
  - need consider external linkml tools (e.g. DataHarmonizer) - can they follow a loose (“documentation” - level spec)
  - can we have a precedence annotation in the model?
  - Eric can present on RI at the community meeting
  - Need doc on monotonicity - should slots be constrained lower in the order of precedence or higher - is it an error if a slot\_usage makes a slot range less constrictive than a slot definition’s range itself? (in schemaloader monotonicity is enforced but maybe not in schemaview)
- [Sierra] - referential integrity with LinkML (NMDC) - besides supporting structured patterns (we think structured patterns might be bad practice vs. specifying an object with an identifier as a range constraint), what more can we do on the LinkML side to validate object references in JSONSchema generator/validator?  
<https://github.com/linkml/linkml-runtime/pull/322>
  - <https://github.com/microbiomedata/nmdc-schema/issues/1846>
  - are we handling “\$ref” correctly in the jsonschema generator? - should all range constraints that reference other classes (not types or enums) lead to inlined objects in JSONSchema?
  - minutes:
    - the issue here is that the OWL generator needs to ignore patterns/structured\_patterns on slots - instead we need this to be the Class we referenced in the schema.
    - we need to be consistent in our advisory/schema authoring with this convention (at least in NMDC)

2024-05-03

- UML
  - Discussion on “aggregation” relationships and if we should even add them:  
<https://github.com/linkml/linkml/pull/2094> [PR closed]
  - Updated cardinality docs: <https://github.com/linkml/linkml/pull/2100> [Sujay/Chris]

2024-04-26 - cancelled

2024-04-19

- If bioregistry goes missing from your poetry environment, just run `poetry update bioregistry`

- Terrific Community Meeting - thanks Madan!
  - May is covered by Mark & NMDC
  - Should we do a “summer developer days” in June? (e.g. helpdesk hours for community)
- linkml poetry update ready for review - worked!
- please review the linkml paper: [LinkML paper - fresh 2024 start](#)

## 2024-04-12

- [Madan] Next week is the community meeting, all set with presenters?
  - [LinkML Community Meeting rolling agenda](#)
- [Patrick] GitHub Apps and tokens
  - GITHUB\_TOKEN – implicit in workflow, very limited in what you’re allowed to do with it
  - PAT (Personal Access Token) – useful but too powerful
  - Fine-grained personal access tokens – better scoping, but expires after 1 year at most
  - GitHub App – can generate short-lived tokens in actions that have fine grained controls and are not tied to a specific person. This is GitHub’s recommended way of generating tokens for automated tasks – <https://docs.github.com/en/apps>, <https://github.com/actions/create-github-app-token>
- <https://github.com/linkml/linkml-map>
- <https://github.com/linkml/linkml-store>
- Mark Miller QC/sniff-test items from last week

## 2024-04-05

- Need a screenshot of Sierra, Chris, Nomi, Ryan, Matt, Kevin - done
- Slack: should we move our linkml-dev channel from Monarch to OBO so it’s easier to add people?
  - Can make the Monarch channel more Monarch-specific. Rename it linkml-monarch-dev
  - Done. New channel is #linkml-dev in [OBO Community workspace](#).
- Mark Miller can discuss any time
  - I have been doing a lot of bad-smell reporting and refactoring in nmmdc-schema. Could move some of that functionality into linkml core
  - Testing for and squashing non-monotonic `slot_usages`
  - Documentation and diagrams that show `any_of.ranges`
  - I would like to normalize my schema modules and my examples dta files without losing comments and without emitting the most verbose dictionary form
  - Goal: be in a good position to run `linkml-sqlldb` dump on at least one of the core nmmdc-schema modules
- General refactoring discussion - overview

- Minutes:
  - swap the backend “main” serialization framework to “pydantic” vs. python data classes.
    - its probably going to take some time
    - not sure if python data classes generator should be completely refactored or generate python data classes from the “new” pydantic core - for discussion
    - piggyback off pypi?
      - generates nervousness from Chris
      - we want folks to use LinkML even in non-python environments
  - compliance test framework
    - basically it should be a feature x generator/framework/serialization matrix
    - sometimes our current framework does a lot of coercion - we should provide more flexibility and more transparency about what parameters to use to choose.
  - reviewed: <https://github.com/linkml/linkml/pull/1957>
    - will bring in with some extra docs
  - pretty print models
    - <https://github.com/linkml/linkml/pull/314>
    - 👍 - careful with tests though, they might need to be re-written
    - Recommended for anyone touching GH workflow files: <https://github.com/rhysd/actionlint>
  - base class imports
    - <https://github.com/linkml/linkml/pull/2036>
    - lots of discussion to be had here, this one can wait a bit
    - it seems like there are a lot of ways to “wayfind” a schema here, it might be good to establish a best practice here though.

2024-03-29

- Some good news! CZI funding
- [Patrick] Automatic/periodic poetry.lock updates
  - Came up again in <https://github.com/linkml/linkml/pull/2002>
  - I thought we had talked about this and roughly decided to keep the lock file checked in and keep using it for CI installs, but to update it more frequently.
  - I pitched the idea of having a scheduled action which does a **poetry update**, commits the changes to a branch, and opens a PR for a developer to review and possibly resolve test failures. Jonny may contribute something for that.
  - Minutes:
    - this automated PR won’t run the tests (to avoid infinite loops) - this must be fixed before we make this a process.
      - we need to make a generic user to create the PR to fix this.

- can't be the same person that opens this PR to be the one that reviews and merges.
  - only one will exist at any one time
  - need an SOP for reviewing the PR
    - check if the tests pass, then merge in
    - this might be automatable, PR reviewer comes into the loop when/if the test fail and then they will either fix or track down someone to fix the dependency that is breaking the tests.
  - need default environment to be pydantic 2, add a special process to test pydantic 1 - outstanding.
  - is this best practice for downstream projects?
    - if you are making an application (e.g. fastapi) - want the lockfile, needs to be used in deployed application so you get the same environment.
    - if you are making a library - might not, because people downstream are then not getting the same env as you use for development
- [Patrick] Publishing these notes somewhere publicly accessible? (maybe in more digested format of just decisions and action items)
  - Also came up in <https://github.com/linkml/linkml/pull/2002>
  - Minutes:
    - Nope.
    - Invite Jonny to these calls to be involved and see notes.
- [Nico] `linkml-transformer` is being renamed to `linkml-mapper`
  - <https://github.com/linkml/linkml-transformer/issues/22>
  - Minutes: linkml-mapper wins dev call
- Noticed while working on the paper that we have a lot of different doc styles going on across our repos:
  - <https://linkml.io/linkml-transformer/> windmill style mkdocs
  - TODO [Sierra] <https://linkml.io/schema-automator/> sphinx with alabaster (change to furo with a different color)
  - <https://linkml.io/linkml/> sphinx (for RST or MD), nice for blending in API documentation
  - <https://linkml.io/linkml-model/latest/> mkdocs material with palettes
  - TODO: [Sierra] <https://linkml.io/schemasheets/> mkdocs default (todo: use material)
  - various repos with README.md files only (TODO: make these have doc sites too **linkml-project-template**, **linkml-solr**, **linkml-dataops**, **linkml-runtime**)
  - TODO: [Sierra] - sphinx autodoc for linkml-runtime
    - keep generators as - pull out into sections and link perhaps
  - Minutes:
    - [Patrick] Some plugins belong to specific flavors of mkdoc themes (e.g. the versioner uses a specific plugin that works with mkdocs material)
    - [Sujay] Mkdocs-material has “admonitions”, similar to some Sphinx styling

- [Sujoy] <https://squidfunk.github.io/mkdocs-material/reference/admonitions/>
  - [Mark] it would be good for downstream projects to have a standard here, but we should do some research and pick a good one intentionally so we don't flip flop – there are downstream consequences to a choice here.
  - [Chris] - windmill makes the jupyter notebook content look nice
  - **[DECISION]** - would like mkdocs material for mkdocs, use sphinx furo by default.
- [Sujoy] UML class diagram documentation updates discussion (if time permits)
- Mark Miller LinkML representation ? the gradient of mobile phases in a liquid chromatography setup ?
- Look and feel/branding
  - <https://monarchinitiative.slack.com/archives/C01LDTVURH9/p1711667801169749>
- 

2024-03-22

- Madan Q on expressions
  - <https://linkml.io/linkml/developers/inference.html#expressions>
  - Action item: Chris Mungall and Mark Miller update expression language documentation
- Community meeting debrief
  - NWB recap
  - Presentation from Ben D on Django application with LinkML schema
  - hoping to have some uptake in development in the Django prototype space
    - Django can create data migration scripts based on changing python (model) - this would be nice in projects with relational models.
  - Who is chairing the next one (April 18)? Need volunteer.
- LinkML Tutorial for BOSC/ISMB July 8 or 9 - we get 3 free virtual registrations.
  - Sierra (free virtual registration), Patrick (free virtual registration), Kevin (free virtual registration)
  - Chris
  - Mark - deal with questions asynchronously (in a gdoc), windows issues
  - Sujoy?
  - anyone else want to participate? (would have to register)
  - Let's make a gdoc (and a shortened URL for it) where workshop participants can add questions and tutorial presenters AND people who are not registered can answer them; we can later mine that doc for new FAQs.
- Should we tag open source devs in bugs/tickets that result from their changes to the system?
  - options:
    - we fix, we tag them in the fix, we ask for their review of our changes.
    - we tag them in the ticket and ask them if they have bandwidth to fix, give them a day to respond (e.g. a breaking change), then fix ourselves.

- we assign them “bugs” that directly result from changes they’ve made.

2024-03-15

- Community meeting
  - do we want to chime in on namespace discussion: <https://github.com/orgs/linkml/discussions/1739>
    - should this come up on our community call?
  - Minutes (Sujay):
    - Ben Dichter - present on Django - putting together a UI for any LinkML schema
    - Mark QT - sierra will reach out with Sujay cc'd to confirm
      - Maybe replace with Namespace discussion.
    - Sujay will introduce with like 3 slides to summarize the Namespace issue and reach out to Frank, Jonny, Jsnyder to make sure they can come
- Increase cache size for schemaview methods: <https://github.com/linkml/linkml-runtime/pull/309>
  - Minutes:
    - discussed in meeting, we can't imagine a scenario where even having an infinite cache would break anyone – and if we do, then we can revisit.
    - Sierra will review/get it in with Tim.

2024-03-08

- Biolink categories
- Will make an issue demonstrating unsatisfiable combos (e.g. patterns on type designators that conflict)
- How not to lose Q&A to the Slack memory hole
- /faqbot
  - spend 15 mins/week converting (be sure to check stack overflow)
  - maybe we can dump messages: <https://slack.com/help/articles/201658943-Export-your-workspace-data>
- CMDR updates
- [https://app.diagrams.net/?libs=general:uml#G1AvBY0B08\\_l8Cj4z28hlnpt8\\_09f6l7Vg#%7B%22pageId%22%3A%224D5R9XwDL6HKGmEVIVDG%22%7D](https://app.diagrams.net/?libs=general:uml#G1AvBY0B08_l8Cj4z28hlnpt8_09f6l7Vg#%7B%22pageId%22%3A%224D5R9XwDL6HKGmEVIVDG%22%7D) (see bottom of the image, lots of tabs)
  - we're starting with person/participation/cohort/...and specimen/sample/biospecimen
  - C-PATH needs to involve fewer people until the end of their grant year (in September)
  - strategic direction would be helpful: clinical research data
  - comet: should help the larger use case (linkml-common)
    - CMDR could be a profile of comet

- harmonize: INCLUDE, C-PATH, BDC
    - INCLUDE will be happy for us to spend time on Specimen/Biosample/etc.
- CDE harmonization updates
- <https://github.com/linkml/schema-automator/releases/tag/v0.5.0>
- <https://github.com/monarch-initiative/cde-harmonization>
  - this is a central repository of all that we can find (currently private) - the “E” in ETL (just extract)
  - manually populated
  - e.g. phenx redcap forms, kind of like a DH/linkml CSV combo
  - just a large repo of questions! (aka: slots)
    - with enumerated response values (permissible values)
  - NCIT does the best job - caDSR isn’t downloadable but they do have a sane API that can download JSON - 75K - these can be extremely large – lots of metadata on the “slot” or CDE – some of that metadata should be somewhere else...but we need to analyse first.
    - could put this up on monarch ols
  - the analysis would be helpful to do
    - how many are reused, how many instances of the “same” thing are there, etc. (wait for funding)
- Open PRs:
- <https://github.com/pulls?q=is%3Aopen+is%3Apr+user%3Alinkml+archived%3Afalse+>

2024-03-01

- Go over PRs (most look ok, not too many besides Johnny’s that need review)
  - SQL representations of the schema
    - sql table gen - newly merged removal of deprecated generator (need a release, but use sqltablegen instead now)
    - some discussion of where we are and where we want to be
      - e.g. we know we want to handle enumerations differently
      - some discussion about the RelationalTransformer and whether there is an intermediate product that would help use cases like:
        - I get too many null attributes in a “flat schema” representation of my data (all the inherited properties)
        - I want a hierarchical object model, but a flatter SQL model, but not completely flat! (e.g I want to be able to specify where in my object model hierarchy to start my table generation – I want a “Gene table” but not a “NamedThing” table – or I want a “FeatureTable” with a transformed “type” column instead of specific classes that inherit from “Feature” becoming independent classes). This is likely a job for linkml-transformer but not clear if



RelationalTransformer intermediate class in sqltablegen is the start of this or not.

- Mark will try it with NMDC (biolink works)
- Pydantic schema gen
  - Proper data model for generating pydantic code (e.g. pydantic model of pydantic) - should eventually be a linkml model? it has functions on the classes....how do we introduce this downstream vs. in the schema - python wrapper around gen-pydantic.
    - e.g. “imports”, “initializer”
    - individual render functions
      - vs. render the entire schema
      - tests can now be segmented
  - we should use this as a template perhaps – make a model for your destination instead of printing strings as you go.
  - we think Johnny is “done” and the intermediates could be ignored - now is time for code review.
    - in particular, check the “module” methodology ( jinja templates that take arguments) vs. python that generates jinja with the arguments passed to the render function in python.
- <https://github.com/linkml/linkml/pull/1935>
- TTT - ticket triage - review PRs now, meet back at 2 to triage.

2024-02-23

- CDEs and LinkML (Anne)
  - CDEs - what are they? (Common Data Elements: some synonyms: enum name, controlled vocabulary name, survey question with standardized answers to blood pressure measurement, metadata elements, CF conventions, )
    - use the same Q on a survey with the same set of answers
    - use the same blood pressure measurement with the same units
      - currently there are 100s of these all a little different
      - metadata around CDE is not what you think they should be
    - MixS times 1000 with a registry of all of them (thousands of ids for PI name)
    - EXAMPLE: CDE - uses a value set (CDE is racial demographics, uses value set of all the races included)
  - what can LinkML do to help here? (Chris's slides)
    - schema-automator to transform CDEs to LinkML enums?
    - want to include environmental sciences in same bucket/lump of problem solving, BODC, MixS,
    - CDE bundles aren't clearly organized by biology (instead by projects like CRDC, basically a list of linkml slots)


- LOINC is lumped here as well
  - Darwin-core ++
  - ISO-11179 - is a Metadata Standard in this space
    - Data Element - slot
    - Value Domain - range
    - Concepts - things in a ontology - slot\_uri/class\_uri
  - COB issue 195
  - CDM
  - Drinking History is a LinkML Enum
    - min value 1, max value 4 turns into LinkML PVs: 1,2,3,4,
    - (vs. drinking history as a class)
- <https://github.com/monarch-initiative/cde-harmonization> - REMAIN PRIVATE
  - ingest all the CDEs we can find
  - can use ontogpt to lump the CDEs together using HP or OBA (might be better)
- Questions?
  - goals of the presentation: non-technical audience, stop-the-madness, promote LinkML
- Summary of migration to python 3.12 ? - anything left to do here? what is our official support matrix for python versions.
  - Silvano, Johnny working on it
  - linkml-runtime (new 2 day old PR that was merged)
    - tests happen on 3.12 (and 3.8, 3.9, 3.10, 3.11)
    - PR for tests failing on linkml on 3.11 and 3.12 - <https://github.com/linkml/linkml/pull/1704#>
  - Are we ready for not supporting pydantic 1.0?
- LinkML paper: [linkml serializations](#) - what other features (columns in the figure) need to be compared and highlighted as LinkML strengths.
- [Discuss Pkl](#), Apple's new [open source](#) schema driven configuration language / code generator
  - <https://github.com/apple/pkl/discussions/7> <-- someone should really respond here
  - Chris Mungall to Everyone (Feb 23, 2024, 2:00 PM)
    - <https://github.com/orgs/linkml/discussions/1944>

2024-02-16


- Triaging Tickets Together (TTT)

2024-02-09

- NWB debrief & native support for multi-dimensional arrays in linkml-model
  - ...
  - <https://github.com/linkml/linkml/releases/tag/v1.7.0>

- comet and cmdr
  - Should we import comet in cmdr straight away?
  - Is comet a replacement for cmdr?
- Announcements:
  - Next week is linkml-community call
    - AGR had a conflict - Sujay taking over their spot possibly?
    - Kenneth is doing Data Harmonizer
    - It has been talked about before but maybe N-D arrays now that 1.7 is out?
- Minutes:
  - [NWB \(neurodata without borders\) hackathon](#) + Allen Institute + DANDI:
    - NIH brain initiative with neurological data repository
    - BiCAN KB presentation -> schemasheets -> model (lots of feedback to schemasheets)
    - ANSRS schema (anatomy, resource, schema, etc) work was done
      - cookiecutter worked WELL!!
      - Developed by Lydia at the Allen Institute: <https://github.com/brain-bican/ansrs-schema>
    - Ben Dichter is part of this group: linkml->schema->Django->webform (DH with relationships) –
      - **[SUJAY]** - please link his repo in this ticket: <https://github.com/linkml/linkml/issues/1882>
        -  LinkML -> Django Admin
      - **[SUJAY]** - can you do March community meeting and invite Ben to present?
        - Invited him on 2/8 to present at the Feb community call. This was his response: *Hey, thanks Sujay! I would love to present the work but it's going to be tough that day. When is the next community call?*
        - Host for the community meeting in March – Sujay
    - Another person pydantic -> linkml
    - LinkML adding support for n-dimensional arrays 1.7.0
      - see doc
  - NWB is like a GIANT community of people that are 100% on board with LinkML (and more broadly, the idea of making ontologies easier to incorporate into models, etc.)
  - CMDR - the “C” should be clinical
    - Comet is higher level - an experiment, but can imagine a higher level here.
      - comet is modular - could focus on a single YAML file in comet for example.
    - CMDR should implement a subset of this?
      - maybe - lets see where this goes.

## 2024-02-02

- Reminder: TTT (Ticket Triage Together) today, after this meeting
- Cookie cutter changes from Noel M. (review .env direction) [Patrick]
- feedback on which direction I should go? <https://github.com/linkml/linkml/issues/1850> [Sierra]
- PR review: <https://github.com/linkml/linkml/pull/1856> [Sujay]
- INCLUDE/LinkML discussion on Thursday, 2/9 [Madan/Sujay]
  - Best practices to validate instance data
  - Customize documentation
- Review of the current state of automatic ER & UML diagram generation tools in LinkML next week [Sujay]
  - Slides:  Review of LinkML schema diagram generation tools
- Announcements
  - Next week: LinkML-NWB Hackashop
  - <https://github.com/linkml/linkml/releases/tag/v1.7.0rc2>
  - <https://github.com/linkml/schema-automator/releases/tag/v0.4.0>
  - <https://linkml.io/linkml-common/>

## 2024-01-26

- Mark presents NMDC local triple store for developers setup (with an RDF serialization as a middle person), graphdb and <https://jena.apache.org/documentation/fuseki2/>
  - using a combo because graphdb is limited - we use neo4j in other monarch
- CMDR/INCLUDE
  - generate google forms from LinkML classes (e.g. "Study metadata") - RedCap?
    - datasets - Harry and Marcin
      - sample sample processing objects - NMDC
    - Madan - help coordinate a presentation for March community meeting with RedCap folks
    - <https://linkml.io/linkml/developers/schemaview.html#schemaview>
    - <https://linkml.io/linkml/howtos/port-linkml.html#data-class-generators>
    - Bootstrapped via excel generator, probably better served by DataHarmonizer
  - give a user a spreadsheet with "Down Syndrome Subtype" and what it translates to is "CommonMeasureSourceText" (reminded me of a mixs use case)
    - lots of discussion about pain points in changing the model - difference between query model and input model too.
  - Presentation on the CMDR data model at the Monarch HTS? [Sujay]
    - Sierra and Anne to present together
- Presentation reviewing the current state of automatic ER & UML diagram generation tools in LinkML next week [Sujay]
- Boost! <https://genomic.social/@linkml@fosstodon.org/111823722495962822>
- Announcements

- Upcoming linkml neuro hackathon
- get someone to present at a community meeting (Sujay can help figure it out)

2024-01-19

- Mark presents NMDC local triple store for developers setup (with an RDF serialization as a middle person), graphdb and <https://jena.apache.org/documentation/fuseki2/>
  - using a combo because graphdb is limited - we use neo4j in other monarch
- Community meeting
  - Feb 15 - Patrick
    - Data Harmonizer - Kenneth B.
    - Alliance of Genome Resources Schema - Mark Quinton-Tulloch
  - Mar 21 - **NEED HOST and presenters/topics**
    - CMDR efforts? - (sierra, potentially)
    - Java/JVM LinkML implementation
      - Minutes:
        - there is a python based Java generator (jinja template to allow you to define the annotations/etc. you need for your flavor of java class)
        - Jules Jacobson has done another iteration with ...
- Schema Diffing
  - difference calculated via names of slots/classes or via slot\_uris
    - or more generically, let the user specify which metamodel component to do the diff on (e.g. user specifies if the diff should be on the name, slot\_uri, or name and description, or name and mappings, or difference in the hierarchy) etc.
    - which repo would be best for a prototype? (schema\_automator or linkml\_transform, or a new repo?)
    - minutes:
      - maybe spin up a prototype and see how it goes
      - Corey:
        - where do tests of a generated schema serialization live?
        - testing the typescript artifact itself
          - group: keep the tests maybe as close to the generator as possible.
      - Patrick:
        - NMDC does a lot of testing of the schema in the schema repo itself via linkml-run-examples
- How do I write a new generator - what are the best practices?
  - use existing generators as reference
  - use SchemaView
  - use Jinja when helpful
- Look at all these old data instances that don't say what their type is
  - schema view over induced slots to figure out whats supposed to be in the range, etc

- any slot that is marked as “inlined” or “inlined as list” would have an object in its range.
  - but induced\_slots isn’t great

## 2024-01-12

- Discussions: <https://github.com/orgs/linkml/discussions/>
  - Do any of these have actionable tasks in them? If so, these should be added as new Issues.
  - <https://github.com/orgs/linkml/discussions/1748> - Chris said “this should be fixed”, but it’s unclear what “this” refers to.
  - Lifecycle hooks for generators? [#1582](#)
- Deprecation best practices proposal:
  - deprecate element (leave references in tact) for at least 1 release
    - include replacement elements when possible
  - when doing a subsequent release, remove deprecated items and references
  - what do we do with w3ids?
  - does `deprecated == obsolete`?
    - maybe obsolete gets put in an obsolete.yaml
    - reusing is tricky - can’t use the same id for a new component.
      - avoid this, but it can’t stop us.
      - keep deprecated == obsolete but separate them into another, non-imported yaml file.
      - action: write up best practices doc :)
  - what’s best practice for generators and validators w/re to “deprecated” (what about “obsolete”?)
    - We should start seeing some warnings (validation warnings, etc.)
- Best practices on element name length
  - don’t make essays
  - visual comprehension are limited to a couple of words
  - sometimes long ones mess up doc or code lines
- CMDR updates (Corey)
  - two introductory meetings with INCLUDE, BDC, C-PATH, Kids First, CCDH, Bride2AI (maybe)
  - additional meetings on Thursdays at 2pm PT for introductory view of their current models (some young, some more mature/in prod)
  - We think CMDR should be focused on metadata to find and reuse cohorts/sample sets/study identification
    - e.g. GFF files vs. “Study uses sequences” metadata
    - but... does phenotype (because its in a phenopacket) necessarily make it “analysis” vs. cohort data? (Chris)
  - We also talked about core vs. extended models. We also talked about documented transforms. (e.g. linkml\_transform, etc.)

- Where is the linkml-dataops repo? is it still in development - can it be used in linkml-transformer?
  - see somewhat related linkml discussion: #1818 – maybe some overlap in here
  - find all the objects that have a name or id across all serializations – common abstractions like CRUD operations – could have a plugin architecture here that specifies how to do “retrieval” in different serializations.
    - curate-gpt has some starts here for “lookup” operation, etc.
    - linkml-solr abstracts stuff like this too (based on data-ops repo)

## 2024-01-05

- FAIR data standards - how can LinkML automatically generate metrics for these? <https://github.com/orgs/linkml/discussions/1811>
- [CMDR doodle poll](#): please fill it out by end of day today!
  - Document use cases as best we can before the meeting:
    - ➦ [CMDR Landscape Analysis - 2024-01-03](#)
  - Minutes:
    - hackathon potentially Feb 22 (details TBD)
    - CMDR is a schema for samples across clinical research + environmental spaces (inspired by CCDH, some FHIR, some Phenopackets)
      - classes are very high level at the moment
        - how to extend or extract is still under discussion - e.g. linkml-transformer, etc.
        - currently independent but C-Path, BDC are looking for collaboration here (we want one model). INCLUDE is very similar, we could merge the ideas here.
      - <https://github.com/linkml/cmdr/>
- LinkedIn is another place to look for questions from our users; please join the group if you haven't already. :) <https://www.linkedin.com/groups/14303246/>
  - [Einar Clementz](#) reached out to me directly via LinkedIn and has clearly spent a lot of time looking at ROBOT, OBOFoundry, OntoGPT, LinkML .... basically all of our tools.
  - Some desire for a SQL DDL -> LinkML roundtripping
    - schema-automator
    - transform would be between normalized -> denormalized
    - roundtripping might need hints to go back from linking table -> multivalued
  - Will present at the next Community meeting
- LinkML Community Meeting **Jan 18** agenda:
  - Chris - [linkml-transformer](#)
  - Einar C - representing FIBO in LinkML and using ontogpt...
- Community meeting host sign up:
  - Jan 18 - Sierra
  - Feb 15 - Patrick

- Data Harmonizer - Kenneth B.
  - Mark Quinton-Tulloch (Alliance of Genome Resources) - presentation
- Mar 21 -
- April 18 -
- May 16 -
- June 20 -
- What should our “theme” be for Q1, Q2 LinkML Community meetings and how do we want to document any training/tutorials we give in these meetings? (e.g. OBOAcademy style?)
  - back to basics
  - add recording links to our FAQ or tutorial.
- Shahim: how to write a custom dumper (transformer?)?
  - And how to hold the mapping information needed for the dumper?
  - Minutes:
    - loaders and dumpers going from python representation of data to serialization of that data into a different format, e.g JSON data to OWL
      - CSV + (INCLUDE) LinkML for that CSV -> python instance -> JSONSchema
        - conceptual mapping lives somewhere else
        - then row is represented in a JSON Object according to a FHIR Schema
      - linkml-transform could be used for this use case
        - linkml-transform -source\_schema xyz -mapped\_transform schema -source\_dataset
          - will derive the output schema from the mapped\_transform (later development will allow target schema to be submitted and we can validate the input to see if the mappings make sense, e.g. single value -> multivalued).
          - currently limited in the shape of transform - better at one level nesting (flat -> normalized is more difficult, we want to go there, but for now we would suggest doing this in python and using that python to dictate the more generic linkml-transform code)
          - can we generate the transform via mappings URIs? - not yet but good use case, and we can imagine it being doable.
          - does the transformer reference functions?
            - yes, any arbitrary python, but encouraged to use the KGCLanguage
              - protects against malformed transforms
              - could give it a try, but is in development



- <https://github.com/linkml/linkml-transformer>
- generators: turning a FHIR schema -> JSONSchema
  - constrain a FHIR schema (e.g. profiling)
  - vulcan IG ? is that FHIR or OMOP?
  - FHIR.Person.address (can have any string value), but in US, FHIR.Person.IGaddress must have only 5 characters. (IG - profiling)
  - can declare the profile in the same language, but not extending the schema
    - this is a linkml-transformer use case: from FHIR to FHIR IG (profile), restricting the FHIR schema. (this is really a next step, not this step)

Older minutes from 2022-2023:

 2022-2023 LinkML Rolling Minutes