



UNIVERSITY
OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione



KOS

delivering iTelos as a web application

Speaker: Simone Bocca

KnowDive Seminars
17th February 2021

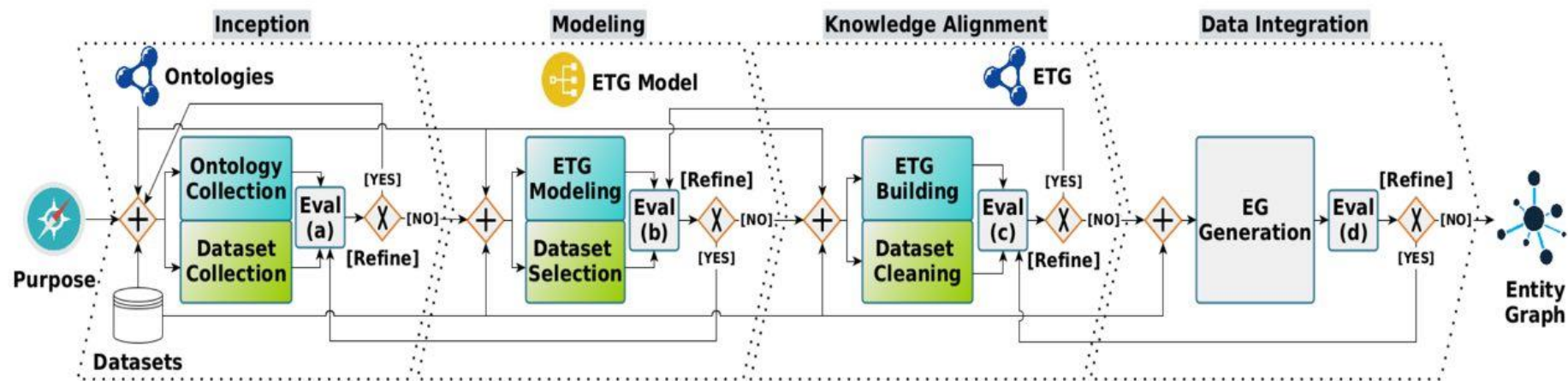


Index:

1. The iTelos methodology
2. The KOS web application
3. KOS project overview
4. Input and outputs resources
5. Appendix

The iTelos methodology

- Knowledge graph engineering process specs
- Focus on
 - re-use of existing resources
 - shareability of output
- Defines
 - roles
 - resources
 - pipeline

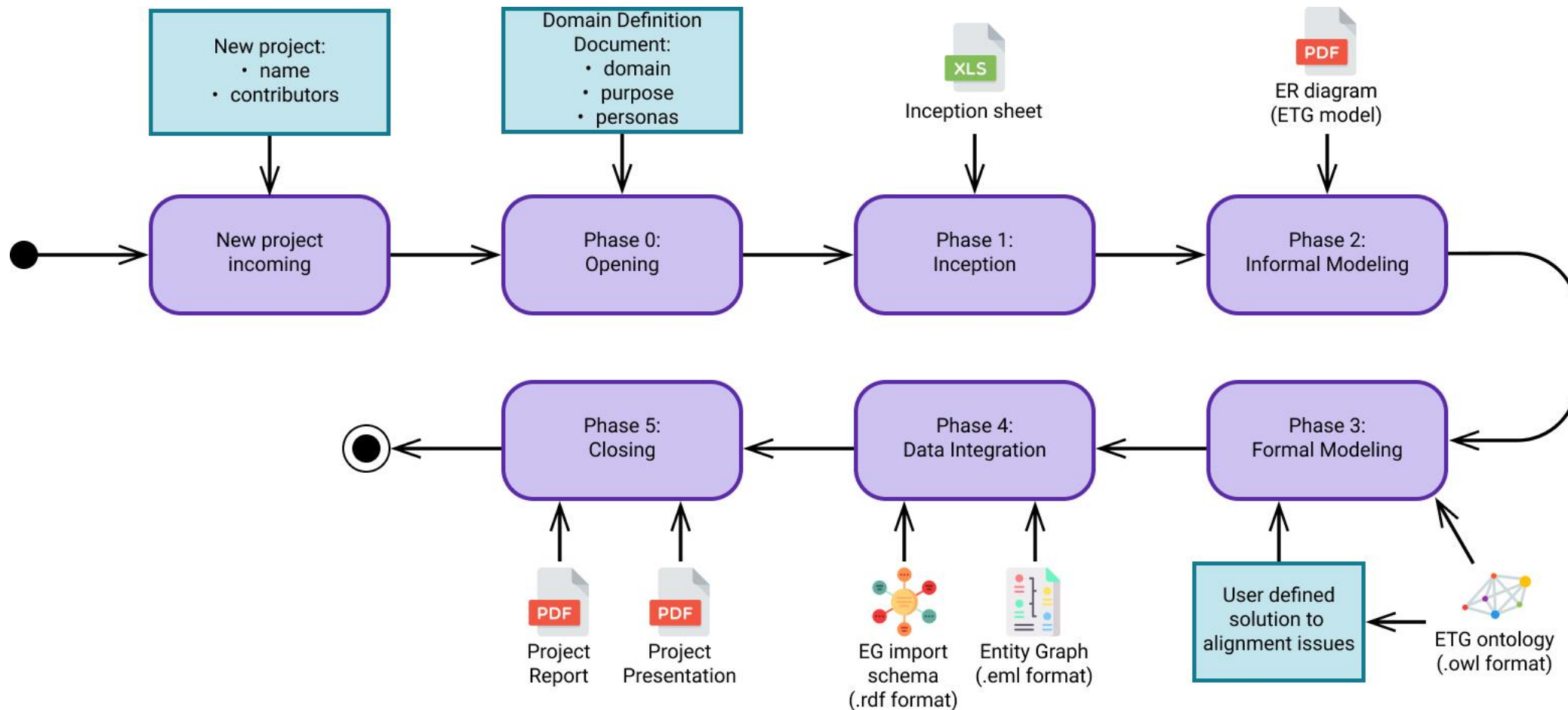


The KOS web application

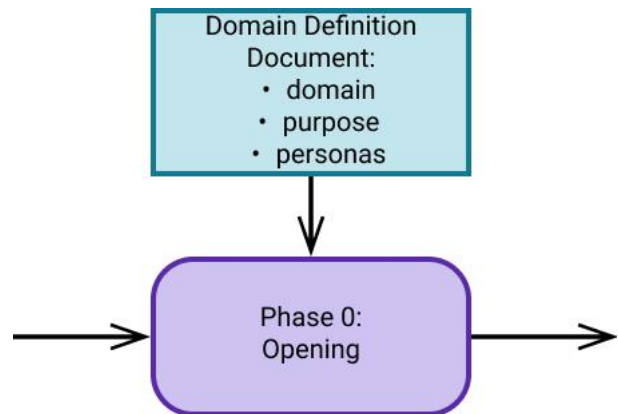
- GUI wrapper/hub for iTelos
- Allows user to define and execute *projects*
- A KOS project:
 - keeps track of state of process
 - helps user to collect required resources
 - provides alignment functionality
 - manages Github backbone repository



KOS project overview

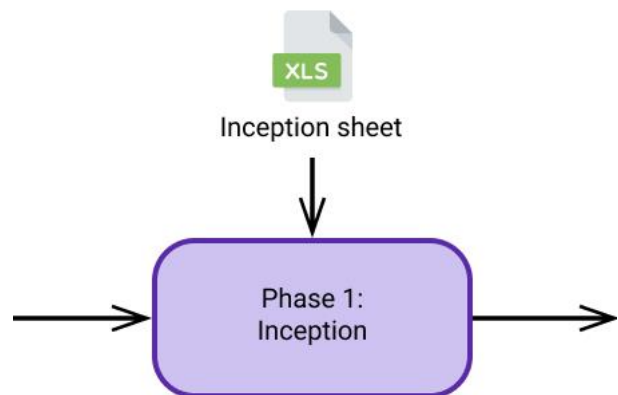


Phasewise input/outputs



0: Opening

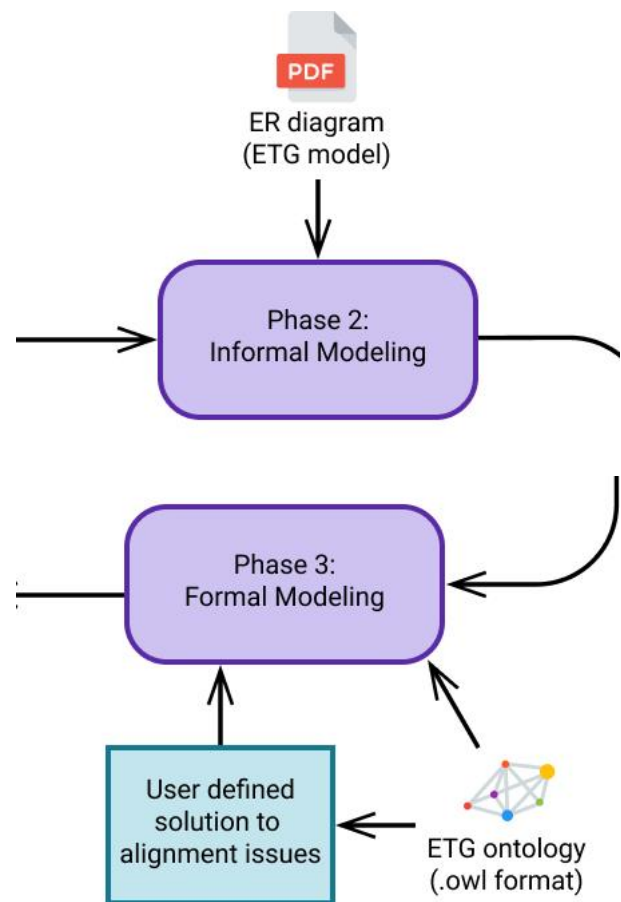
- domain definition document (.md)
 - domain, purpose, personas
- defined via GUI



1: Inception

- Inception sheet (.xlsx)
 - purpose formalization
- fill spreadsheet (MS Excel, Libreoffice Calc)

Phasewise input/outputs



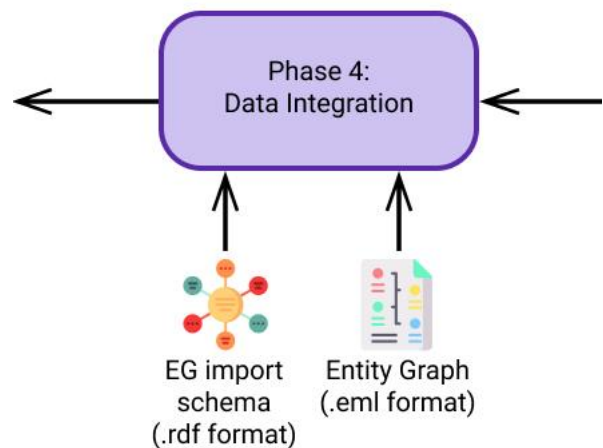
2: Informal modeling

- ETG model (.pdf/.jpg/.png)
 - first formalization of application ontology
- draw ER diagram (draw.io, Lucidchart, Figma)

3: Formal modeling

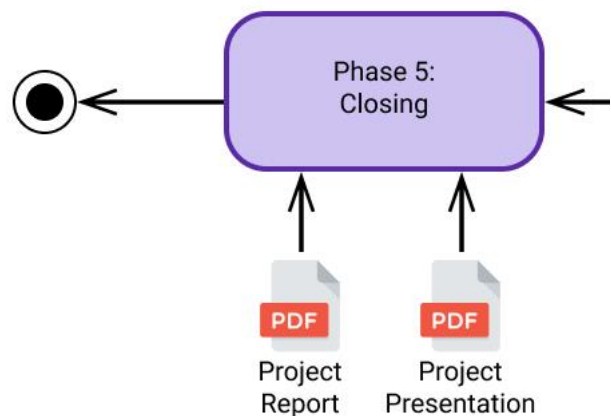
- final ontology (.owl)
 - some classes may require annotation
 - annotation is done via GUI tool
- Changelog (.xlsx)
- create ontology with Protégé

Phasewise input/outputs



4: Data integration

- final EG (.eml)
- RDF version of EG (.rdf)
- perform integration of ETG and dataset with Karmalinker



5: Closing

- project report (.pdf)
- project presentation (.pdf)
- any word processor/presentation program/LaTeX

Appendix: register form

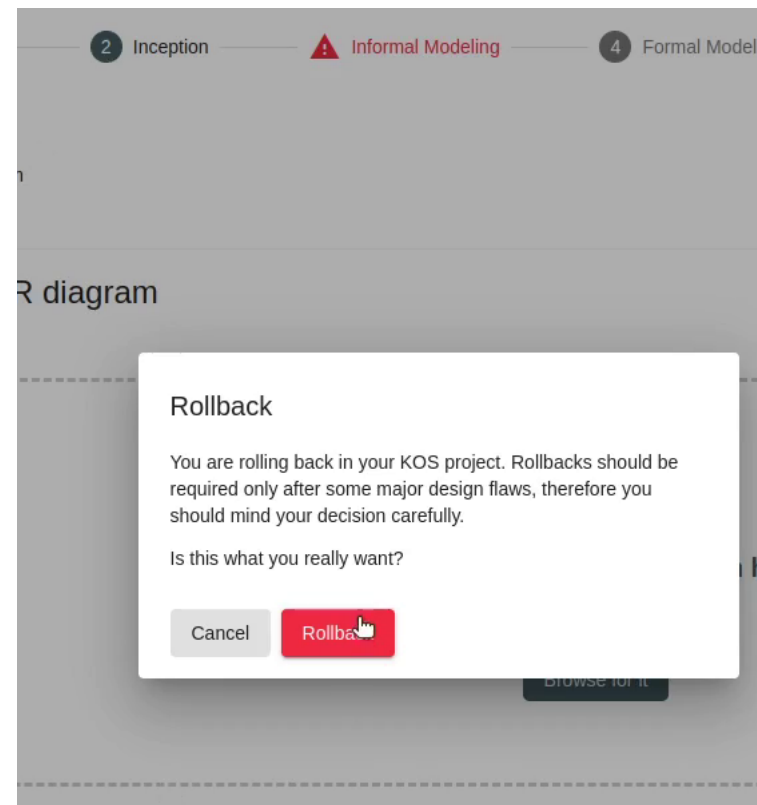
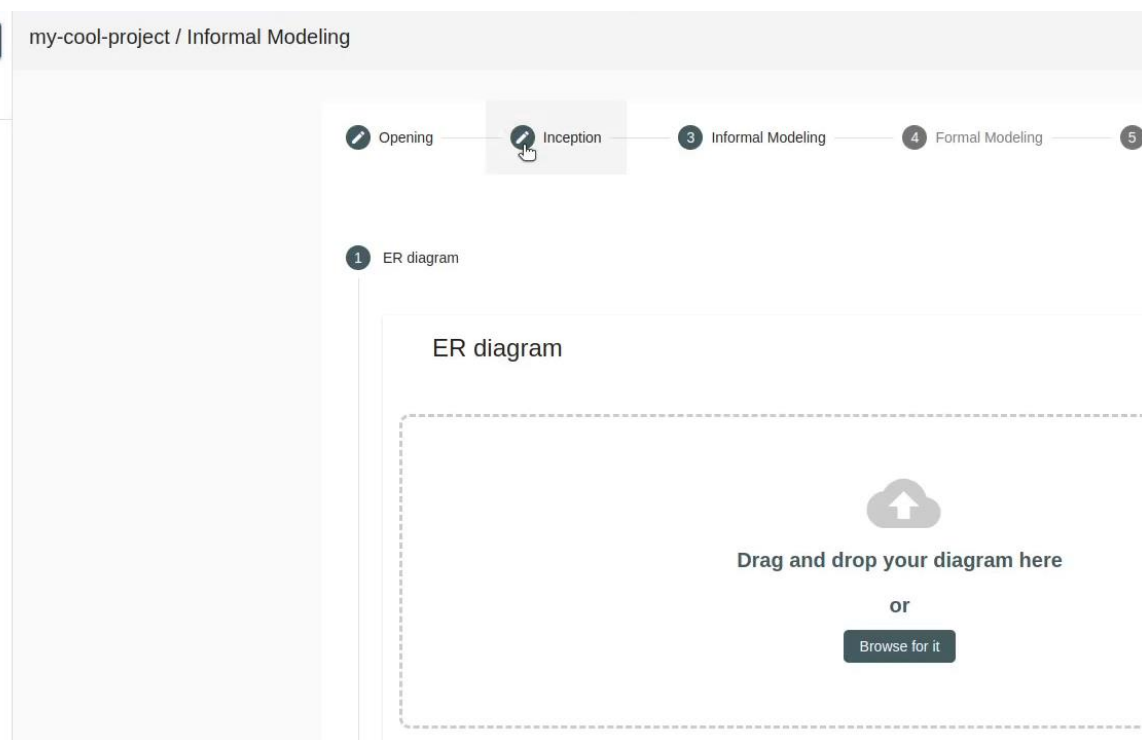
- username
- password
 - both used to identify user in the platform
- Github username
- Github email
 - username and email of a valid Github profile
- Github Personal Access Token
 - should be valid
 - select *repo* scope
 - see [here](#)

Register

What is a Github Personal Access Token?

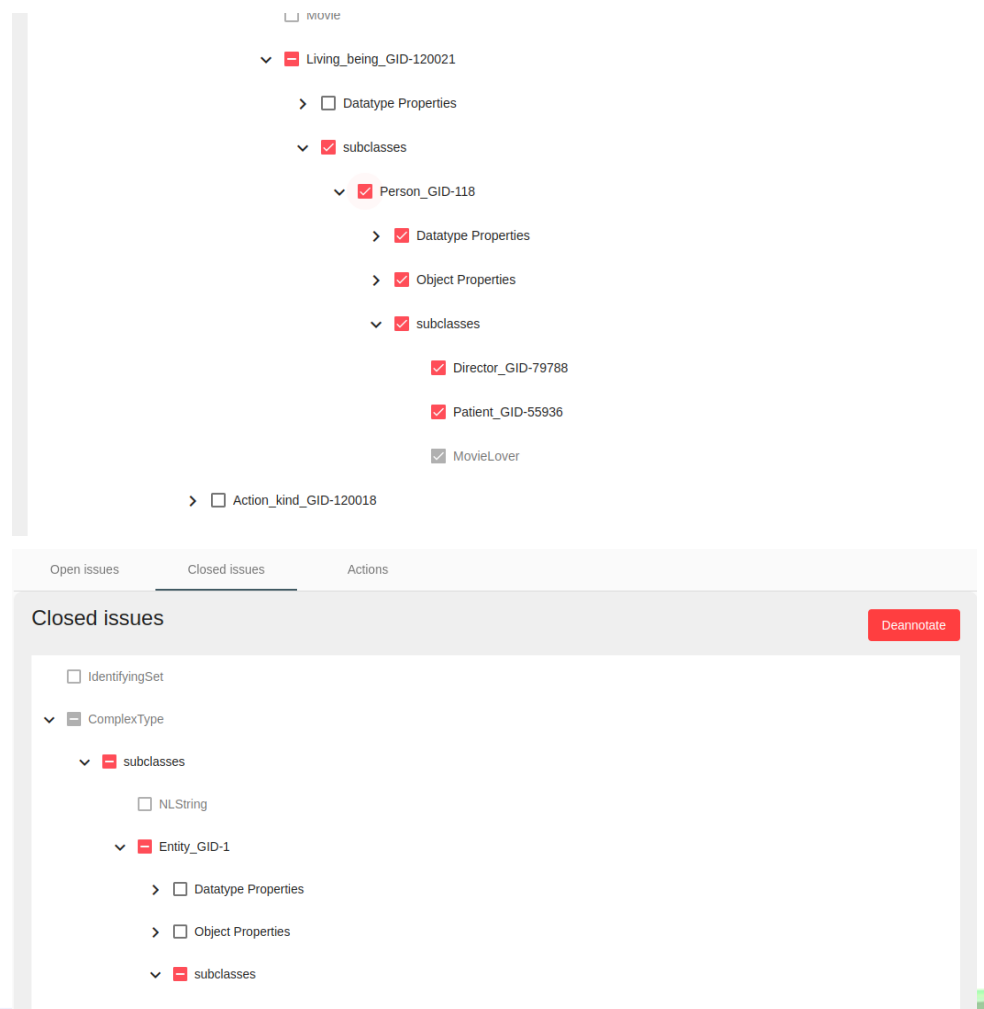
Appendix: rollback

- Click on the target phase in the main stepper to rollback



Appendix: De-annotation

- Go to the *closed issues* tab
- expand ontology tree
- look up for the desired concept
- mark checkbox and hit *deannotate*
 - this will de-annotate properties and subclasses recursively



The screenshot shows a web interface for de-annotation. At the top, there is a tree view of an ontology. The tree is partially expanded, showing a path: `MOVIE` > `Living_being_GID-120021` > `Datatype Properties` > `subclasses` > `Person_GID-118`. Under `Person_GID-118`, there are three checked checkboxes: `Datatype Properties`, `Object Properties`, and `subclasses`. Under `subclasses`, there are two checked checkboxes: `Director_GID-79788` and `Patient_GID-55936`, and one unchecked checkbox: `MovieLover`. Below the tree view, there are three tabs: `Open issues`, `Closed issues` (which is selected), and `Actions`. The `Closed issues` tab shows a list of issues. The first issue is `IdentifyingSet`. Below it, there is a `ComplexType` section, which is expanded to show `subclasses`. Under `subclasses`, there is a checkbox for `NLString` and a checked checkbox for `Entity_GID-1`. Under `Entity_GID-1`, there are three checked checkboxes: `Datatype Properties`, `Object Properties`, and `subclasses`. In the top right corner of the `Closed issues` tab, there is a red button labeled `Deannotate`.



UNIVERSITY
OF TRENTO - Italy

Dipartimento di Ingegneria e Scienza dell'Informazione

Know
dive



Thank you