**Alert: This task was written up in early 2022 and refers to artifacts that are old. As such, the first part of this task is to update the description and references. Other will help with this.**

**(30 Mins) Deliverable 1**: Use the “second task owl Mapping file” below and reformat it into a csv mapping file. The goal Is to get the mapping below to have the same format as this mapping file: <https://github.com/I-AM-project/my-data-ontology/blob/master/dev/mapping-mydata_org-mdo.csv>

|  |  |  |
| --- | --- | --- |
| Data Model  (Customer Collaboratio) | Attributes | Example |
| class | Full name | cco:Person, cco:designated\_by, cco:PersonGivenName, cco:has\_value |
| class | Mail Box | cco:Person, cco:user, cco:EmailBox,  cco:StasisOfTelecommunicationEndpointAssignment |
| class | Street address | cco:Person, cco:agent\_in, cco:ActOfResiding, cco:ResidentialFacility, cco:designated\_by, cco:StreetAddress |
| class | Birthday | cco:Person, cco:is\_object\_of, cco:Birth, cco:occurs\_on |
| class | City location | cco:Person, cco:agent\_in, cco:ActOfResiding, cco:ResidentialFacility, cco:LocalAdministrativeRegion, cco:designated\_by, cco:DesignativeName, cco:has\_value |
| class | State location | cco:Person cco:agent\_in cco:ActOfResiding cco:ResidentialFacility, cco:LocalAdministrativeRegion cco:FirstOrderAdministrativeRegion cco:designated\_by cco:DesignativeName cco:has\_value |
| class | Full postal code | cco:Person cco:agent\_in cco:ActOfResiding cco:ResidentialFacility cco:PostalZone cco:designated\_by cco:PostalCode cco:has\_value |
| Class | Client Phone number | cco:Person cco:uses cco:MobileTelephone cco:StasisOfTelecommunicationEndpointAssignment cco:TelecommunicationEndpoint cco:designated\_by cco:TelephoneNumber cco:has\_value |
| class | As employer Names | cco:Person, cco:OccupationRole, cco:has\_organizational\_context, cco:Organization, cco:designated\_by, cco:DesignativeName, cco:has\_value |

**(30 mins) Deliverable 2**: Create fake source data for the “second task owl Mapping file.” The source data will use all the attributes of the mapping above. Create 4 Example people for the source data. The file should be in a csv format and look like this file: <https://github.com/I-AM-project/my-data-ontology/blob/master/dev/my-data-org-source-data.csv>

**(4 hours) Deliverable 3**: Use the dynamicTransformationScript.py https://github.com/I-AM-project/my-data-ontology/blob/master/dynamicTransformationScript.py to read the cvs file from deliverable 1 into a python dictionary where the keys are the attribute of the mapping, and the values are the MyDataOntoloy Expansions.

Then, the script will read in the csv file from deliverable 2 as the source data. It will match the attributes from the source data with the keys from the dictionary to retrieves the MyDataOntoloy Expansions.

Finally the script will concatenation the values from the source data with the MyDataOntoloy Expansions to create the conforming triples

**Examples for the deliverables**

1. Deliverable 1 first row of the csv

Attribute, rdf-type, relation, rdf-type, relation

Fullname , cco:Person, cco:designated\_by , cco:PersonGivenName, cco:has\_value

1. Deliverable 2

Full name = John Doe

1. Output of Deliverable 3 for the Fullname:

cco:Person\_{personGuid} cco:designated\_by cco:PersonGivenName\_{firstname\_uuid} .

cco:PersonGivenName\_{firstname\_uuid} a cco:PersonGivenName ;

obo:RO\_0010001 cco:InformationBearingEntity\_PersonGivenName\_{firstname\_uuid} .

cco:InformationBearingEntity\_PersonGivenName\_{firstname\_uuid} a cco:InformationBearingEntity ;

cco:has\_text\_value "John Doe”.