URI Design and Mappings in StatSpace

Ba Lam Do

Contents

1. URIs patterns for metadata description

1.1. Patterns

Base URI: http://statspace.linkedwidgets.org

No	Patterns	Description
1	/metadata/{datasource-dataset}	URI of a metadata
2	/dataset/{datasource-dataset}	URI of a dataset
3	/codelist/cl_{name}	URI of a code list
4	/dimension/{name}	URI of a dimension

1.2. Example

No	Patterns	Description
1	http://statspace.linkedwidgets.org/metadata/ONS-Population-1851-2014	URI of a metadata
2	http://statspace.linkedwidgets.org/dataset/ONS-Population-1851-2014	URI of a dataset
3	http://statspace.linkedwidgets.org/codelist/cl_area	URI of a code list
4	http://statspace.linkedwidgets.org/dimension/economicActivity	URI of dimension of economic activity

2. List of components used to create metadata

2.1. Dimensions

No	URI	Label
1	http://purl.org/linked-data/sdmx/2009/dimension#refArea	Reference Area
2	http://purl.org/linked-data/sdmx/2009/dimension#refPeriod	Reference Period
3	http://purl.org/linked-data/sdmx/2009/dimension#age	Age
4	http://purl.org/linked-data/sdmx/2009/dimension#educationLev	Education Level
5	http://purl.org/linked-data/sdmx/2009/dimension#occupation	Occupation
6	http://purl.org/linked-data/sdmx/2009/dimension#currency	Currency
7	http://purl.org/linked-data/sdmx/2009/dimension#civilStatus	Civil Status
8	http://purl.org/linked-data/sdmx/2009/dimension#freq	Frequency
9	http://purl.org/linked-data/sdmx/2009/dimension#sex	Sex
10	http://statspace.linkedwidgets.org/dimension/economicActivity	Economic Activity
11	http://statspace.linkedwidgets.org/dimension/expenditure	Expenditure

2.2. Measure

No	URI	Label
1	http://purl.org/linked-data/sdmx/2009/measure#obsValue	Observation

2.3. Attribute

No	URI	Label
1	http://purl.org/linked-data/sdmx/2009/attribute#unitMeasure	Unit of Measure

3. URI patterns of code lists

Base URIs:

- http://reference.data.gov.uk for code list of reference period dimension
- http://statspace.linkedwidgets.org/codelist for the remaining code lists

3.1. Code list of reference area dimension (cl_area)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_area
- RDF Data: http://statspace.linkedwidgets.org/code/cl_area.ttl
- Pattern

URI Pattern	Description
/cl_area/{Country}/{Area in level 2}/…/{Area in level n}	URI of a geographical area

- Example

URI	Description
http://statspace.linkedwidgets.org/codelist/cl_area/Austria/Vienna	Vienna, Austria

3.2. Code list of reference period dimension (cl_period)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_period
- RDF Data: http://statspace.linkedwidgets.org/code/cl_period.ttl
- Patterns

URI Patterns	Description
/id/gregorian-year/{year}	URI for a year
/id/gregorian-half/{year}-{half}	URI for one-half year
/id/gregorian-quarter/{year}-{quarter}	URI for a quarter
/id/gregorian-month/{year}-{month}	URI for a month
/id/gregorian-day/{year}-{month}-{day}	URI for a day
/id/gregorian-hour/{year}-{month}-{day}T{hour}	URI for a hour
/id/gregorian-hour/{year}-{month}-{day}T{hour}:{min}	URI for a minute
/id/gregorian-hour/{year}-{month}-{day}T{hour}:{min}:{sec}	URI for a second
//id/gregorian-week/{year}-{week}	URI for a week
/id/gregorian-instant/{dateTime}	URI for an instant
/id/gregorian-interval/{dateTime}/{duration}	URI for a duration

- Example

URIs	Description
http://reference.data.gov.uk/id/gregorian-year/2016	URI for year 2016
http://reference.data.gov.uk/id/gregorian-day/2016-01-01	URI for day 01/01/2016

3.3. Code list of age dimension (cl_age)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_age
- RDF Data: http://statspace.linkedwidgets.org/code/cl_age.ttl
- Patterns

URI Patterns	Description
/cl_age/Y{n}, n=0, 1, 2,…,105	URI for an individual age
/cl_age/Y{n}T{n+4}, n=0, 5, 9,…,105	URI for an age group (5 years)
/cl_age/Y{n}T{n+9}, n=25, 35,…, 95	URI for an age group (10 years)
/cl_age/Y_GE_{n}, n=65, 70,…,90	URI for an age group (equal or above a specific age)
/cl_age/Y_LE_{n}, n=15, 20	URI for an age group (under a specific age)
/cl_age/TOTAL	URI for the top concept
/cl_age/UNK	URI for an unknown age

- Example

URIs	Description
http://statspace.linkedwidgets.org/codelist/cl_age/Y80	URI for age 80
http://statspace.linkedwidgets.org/codelist/cl_age/Y80T84	URI for an age group from 80 to 84

3.4. Code list of education level dimension (cl_educationLev)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_educationLev
- RDF Data: http://statspace.linkedwidgets.org/code/cl_educationLev.ttl
- URIs of values

URIs	Description
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L0	Pre-primary education
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L1	Primary education
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L2	Lower secondary
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L3	Upper secondary
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L4	Post-secondary non-tertiary education
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L5	Short-cycle tertiary education
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L6	Bachelor or equivalent
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L7	Master or equivalent
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L8	Doctoral or equivalent
http://statspace.linkedwidgets.org/codelist/cl_educationLev/L9	Not elsewhere classified

3.5. Code list of occupation dimension (cl_occupation)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_occupation
- RDF Data: http://statspace.linkedwidgets.org/code/cl_occupation.ttl
- Pattern

URI Pattern	Description
/cl_occupation/{code}	URI of an occupation

- Example

URIs	Description
http://linkedwidgets.org/resource/codelist/cl_occupation/OC1	URI for Managers
http://statspace.linkedwidgets.org/codelist/cl_occupation/OC11	URI for Chief executives, senior officials and legislators

3.6. Code list of currency dimension (cl_currency)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_currency
- RDF Data: http://statspace.linkedwidgets.org/code/cl_currency.ttl
- Pattern

URI Pattern	Description
/cl_currency/{ISO 4217 code}	URI for a currency

- Example

URI Pattern	Description
http://statspace.linkedwidgets.org/codelist/cl_currency/AED	URI for United Arab Emirates dirham
http://statspace.linkedwidgets.org/codelist/cl_currency/EUR	URI for Euro

3.7. Code list of civil status dimension (cl_civilStatus)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_civilStatus
- RDF Data: http://statspace.linkedwidgets.org/code/cl_civilStatus.ttl
- URIs of values

URIs	Description
http://statspace.linkedwidgets.org/codelist/cl_civilStatus/D	Divorced person
http://statspace.linkedwidgets.org/codelist/cl_civilStatus/E	Person whose registered partnership was legally dissolved
http://statspace.linkedwidgets.org/codelist/cl_civilStatus/L	Leggaly separated person
http://statspace.linkedwidgets.org/codelist/cl_civilStatus/M	Married person
http://statspace.linkedwidgets.org/codelist/cl_civilStatus/P	Person in Registerd partnership
http://statspace.linkedwidgets.org/codelist/cl_civilStatus/Q	Person whose registered partnership ended with the death of the partner
http://statspace.linkedwidgets.org/codelist/cl_civilStatus/S	Single person
http://statspace.linkedwidgets.org/codelist/cl_civilStatus/W	Widowed person

3.8. Code list of freq dimension (cl_freq)

- URI of the code list: http://purl.org/linked-data/sdmx/2009/code#freq
- RDF Data: http://statspace.linkedwidgets.org/code/cl_freq.ttl

- URIs of values

URIs	Description
http://purl.org/linked-data/sdmx/2009/code#freq-H	Hourly
http://purl.org/linked-data/sdmx/2009/code#freq-D	Daily
http://purl.org/linked-data/sdmx/2009/code#freq-N	Minutely
http://purl.org/linked-data/sdmx/2009/code#freq-S	Half yearly, semester
http://purl.org/linked-data/sdmx/2009/code#freq-A	Annual
http://purl.org/linked-data/sdmx/2009/code#freq-Q	Quarterly
http://purl.org/linked-data/sdmx/2009/code#freq-M	Monthly
http://purl.org/linked-data/sdmx/2009/code#freq-B	Daily-business week
http://purl.org/linked-data/sdmx/2009/code#freq-W	Weekly

3.9. Code list of sex dimension (cl_sex)

- URI of the code list: http://purl.org/linked-data/sdmx/2009/code#sex
- RDF Data: http://statspace.linkedwidgets.org/code/cl_sex.ttl
- URIs of values

URI Pattern	Description
http://purl.org/linked-data/sdmx/2009/code#sex-M	URI for Male gender
http://purl.org/linked-data/sdmx/2009/code#sex-F	URI for Female gender
http://purl.org/linked-data/sdmx/2009/code#sex-T	URI for Total
http://purl.org/linked-data/sdmx/2009/code#sex-U	URI for Unknown gender
http://purl.org/linked-data/sdmx/2009/code#sex-N	URI for not applicable gender

3.10. Code list of economic activity dimension (cl_economicActivity)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_economicActivity
- RDF Data: http://statspace.linkedwidgets.org/code/cl_economicActivity.ttl
- Pattern

URI Pattern	Description
/cl_economicActivity/{code}	URI of an economic activity

- Example

URI Pattern	Description
http://statspace.linkedwidgets.org/codelist/cl_economicActivity/A	URI for activity of Agriculture, forestry, and fishing

http://statspace.linkedwidgets.org/codelist//cl_economicActivity/A01	URI for activity of Crop and animal production, hunting and related service activities

3.11. Code list of expenditure dimension

- Contains 4 code lists
  - Classification of individual consumption by purpose (COICOP),
  - Classification of the functions of government (COFOG),
  - Classification of the purposes of non-profit institutions serving households (COPNI) and,
  - Classification of outlays of producers by purpose (COPP).

3. 11.1. Code list of COICOP (cl_coicop)

- URI of the code list COICOP: http://statspace.linkedwidgets.org/codelist/cl_coicop
- RDF Data: http://statspace.linkedwidgets.org/code/cl_coicop.ttl
- Pattern

URI Pattern	Description
/cl_coicop/{code}	URI for expenditure of an individual

- Example

URI	Description
http://statspace.linkedwidgets.org/codelist/cl_coicop/CP01	URI for expenditure of Food and non-alcoholic beverages

3.11.2. Code list of COFOG (cl_cofog)

- URI of the code list COFOG: http://statspace.linkedwidgets.org/codelist/cl_cofog
- RDF Data: http://statspace.linkedwidgets.org/code/cl_cofog.ttl
- Pattern

URI Pattern	Description
/cl_cofog/{code}	URI for expenditure of government

- Example

URI	Description
http://statspace.linkedwidgets.org/codelist/cl_cofog/GF01	URI for expenditure of government for General public services

3. 11.3. Code list of COPNI (cl_copni)

- URI of the code list COPNI: http://statspace.linkedwidgets.org/codelist/cl_copni
- RDF Data: http://statspace.linkedwidgets.org/code/cl_copni.ttl
- Pattern

URI Pattern	Description
/cl_copni/{code}	URI for expenditure of non-profit organization

- Example

URI	Description
http://statspace.linkedwidgets.org/codelist/cl_copni/PN1	URI for expenditure of non-profit organization for Housing

3. 11.4. Code list of COPP (cl_copp)

- URI of the code list COPP: http://statspace.linkedwidgets.org/codelist/cl_copp
- RDF Data: http://statspace.linkedwidgets.org/code/cl_copp.ttl
- Pattern

URI Pattern	Description
/cl_copp/{code}	URI for expenditure of producer

- Example

URI	Description
http://statspace.linkedwidgets.org/codelist/cl_copp/PP1	URI for expenditure of producer for Outlays on infrastructure

3.12. Code list of unit of measure (cl_unitMeasure)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_unitMeasure
- RDF Data: http://statspace.linkedwidgets.org/code/cl_unitMeasure.ttl
- Pattern

URI Pattern	Description
/cl_unitMeasure/{unit}	URI of a unit

- Example

URI Pattern	Description
http://linkedwidgets.org/resource/codelist/cl_unitMeasure/P1	URI for unit "People"
http://statspace.linkedwidgets.org/codelist/cl_unitMeasure/TU	URI for unit "Twenty-Foot Equivalent"

3.13. Code list of subject (cl_subject)

- URI of the code list: http://statspace.linkedwidgets.org/codelist/cl_subject
- RDF Data: http://statspace.linkedwidgets.org/code/cl_subject.ttl
- Pattern

URI Pattern	Description
/cl_subject/{code}	URI of a subject

- Example

URI Pattern	Description
http://statspace.linkedwidgets.org/codelist/subject/AG.SRF.TOTL.K2	URI for subject "Surface area (sq. km)"
http://statspace.linkedwidgets.org/codelist/subject//SP.POP.TOTL	URI for subject "Population, total"

4. Methods for identifying co-reference

4.1. Identifying co-reference URIs for dimensions

Input:

- a URI and its label
- e.g., ex:ref-Area, Reference area

Output:

- co-reference URI of this URI
- e.g., http://purl.org/linked-data/sdmx/2009/dimension#refArea

No	Dimension	Methods
1	Reference Area	- Define a set of possible names for representing this dimension.

		- Check the existence of one name in the input URI or label - {"ref-area", "refarea", "country", "refdistrict", "refstate", "place", "geocode", "region", “reference area”}
2	Reference Period	- {"ref-period", "ref-date","ref-year","refperiod", "timeperiod", "date", "year", "time-period", “time period”}
3	Age	- {“/age","_age", "#age","refage”}
4	Education Level	- {"educationlev", "edulev", “education level”};
5	Occupation	- {“occupation”}
6	Currency	- {“currency”}
7	Civil Status	- {“civil”, “status”}
8	Frequency	- {"freq”}
9	Sex	- {"sex", "gender"}
10	Economic Activity	- {"activity" ,”economy”}
11	Expenditure	- cofog = {"funcofgov", "function of government", "functions of government"}; - coicop = {"indvcons", "individual consumption"}; - copp = {"outlayofproducer", "outlay of producer", "outlays of producer"}; - copni= {"purposeofnpi", "purpose of non-profit institution", "purposes of non-profit institution" };

4.2. Identifying co-reference URIs for values of a dimension

Input:

- a URI and its label
- e.g., ex:AT, Austria

Output:

- co-reference URI of this URI
- e.g., http://statspace.linkedwidgets.org/codelist/cl_area/Austria

No	Value of dimension	Methods
1	Reference Area	- Detect hierarchical relation among areas in the input dataset - Algorithm: presented in section 4.3

2	Reference Period	- Use Patterns e.g.,
		Interval: [1-9][0-9]{3}-[1-9][0-9]{3}
		Year: [1-9][0-9]{3}

		Month:[1-9][0-9]{3}-[0-1][0-9] Quarter: [1-9][0-9]{3}-Q[1-4] Date: [1-9][0-9]{3}-[0-1][0-9]-[0-3][0-9]
3	Age	- Literal values e.g., {Value}^^http://www.w3.org/2001/XMLSchema#long => identify value, then builing corresponding URI - URIs e.g., ex:Y{Value1}-Y{Value2}, ex:{Value}%2B => identify age group, then building correpsonding URI
4	Education Level	- Compare label and code of the URI with values in the corresponding code list. - For example: if the URI ends with code "L0" or its label contains "Pre-primary education" => map to http://statspace.linkedwidgets.org/codelist/cl_educationLev/L0
5	Occupation	- Compare label and code of the URI with values in the corresponding code list. - For example: if the URI ends with code "OC11" or its label contains "Chief executives, senior officials and legislators => map to http://statspace.linkedwidgets.org/codelist/cl_occupation/OC11
6	Currency	- Compare label and code of the URI with values in the corresponding code list. - For example: if the URI ends with code "EUR" or its label contains "EURO" => map to http://statspace.linkedwidgets.org/codelist/cl_currency/EUR
7	Civil Status	- Compare label and code of the URI with values in the corresponding code list. - For example: if the URI ends with code "D" or its label contains "Divorced person" => map to http://statspace.linkedwidgets.org/codelist/cl_civilStatus/D
8	Frequency	- Compare label and code of the URI with values in the corresponding code list. - For example: if the URI ends with code "H" or its label contains "Hourly" => map to - http://purl.org/linked-data/sdmx/2009/code#freq-H
9	Sex	- Compare label and code of each URI with values in the corresponding code list.

		- For example: if the URI ends with code "M" or its label contains "Male" => map to http://purl.org/linked-data/sdmx/2009/code#sex-M
10	Economic Activity	- Compare label and code of each URI with values in the corresponding code list. - For example: if the URI ends with code "A" or its label contains "Agriculture, forestry, and fishing" => map to http://statspace.linkedwidgets.org/codelist/cl_activity/A
11	Expenditure	- Compare label and code of each URI with values in the corresponding code list. - For example: if the URI ends with code "GF01" or its label contains "General public services" => map to http://statspace.linkedwidgets.org/codelist/cl_cofog/GF01

4. 3. Identifying co-reference URIs for values of area reference dimension

Input:

- A set contains URIs and their lables
- $L = \{l_1, l_2, \ldots, l_n\}$, $l_i = \{uri\_l_i, label\_l_i\}$

Output:

- Mapping L to G
- $G = \{g_1, g_2, \ldots, g_n\}$, $g_i = \{uri\_g_i, label\_g_i, lat\_g_i, lng\_g_i, type\_g_i\}$

Methods

1. **Procedure sortInAscendingOrder(L)**

```
// sort areas in L in ascending order of uri
```

2. **Procedure isBroaderArea(uri_l$_i$, uri_l$_j$)**

```
//return true if uri_l$_i$ is a broader area of uri_l$_j$
if   (uri_l$_j$ startsWith(uri_l$_i$ + "/") &&
  length(uri_l$_j$) > length(uri_l$_i$) + 1 &&
  uri_l$_j$.substring(length(uri_l$_i$)+1).indexOf("/")==-1)
     return true;
if(uri_l$_j$ startsWith(uri_l$_i$)&&length(uri_l$_j$) == length(uri_l$_i$) + 1
     return true;
return false;
```

3. **Procedure indexOfBroaderArea(L, l$_i$)**

```
//remain index of the area which is a broader area of l$_i$ in list L
(use isBroaderArea procedure)
```

4. **Procedure filterByDistance(G, g_i)**

```
//remain only one result in gi, that is, the one which has the
minimal distance to adjacent areas $g_{i-2}$, $g_{i-1}$, $g_{i+1}$, $g_{i+2}$
```

5. **Procedure getLevel(uri)**

```
//split the input based on "/" character, and return the
length/size of the returned list
```

6. **Procedure isGoogleBroaderArea(uri_g_i, uri_g_j)**

```
//return true if uri_$g_i$ is a broader area of uri_$g_j$
if  (uri_$g_j$ startsWith(uri_$g_i$ + "/"){
  if(length(uri_$g_j$) > length(uri_$g_i$) + 1 &&
  uri_$g_j$.substring(length(uri_$g_i$)+1).indexOf("/")==-1)
     return true;

  if(uri_$g_j$.contains("/") && uri_$g_i$.contains("/")){
      String[] $area_1$ = uri_$g_j$.split("/"); //Austria/Vienna
      String[] $area_2$ = uri_$g_i$.split("/"); //Austria/Vienna/Vienna
      remove duplicate elements in $area_1$, $area_2$
      if(arrUri2.length==arrUri1.length+1)
          return true;
  }
}
return false;
```

7. **Procedure identifyMapping(L)**

```
// L ={$l_1$, $l_2$,…,$l_n$}, $l_i$ ={uri_$l_i$, label_$l_i$}
// G = {$g_1$, $g_2$,…,$g_n$}, $g_i$={uri_$g_i$, label_$g_i$, lat_$g_i$, lng_$g_i$, type_$g_i$}
// query Google's geocoding API

boolean bUseBroaderArea
string sLabel, sQuery
int level, googleLevel

sortInAscendingOrder(L)

//step 1. query labels with Google's geocoding API
for each area $l_j$ in L do
  //construct a query for this area
     i = indexOfBroaderArea(L, $l_j$)
     if(i!=-1) then
      if(label_$l_i$ != label_$l_j$) then
          sLabel = label_$l_j$ + " " + label_$l_i$
          bUseBroaderArea = true
      else
          k = indexOfBroaderArea(L, $l_i$)
          if(k!=-1) then
              if(label_$l_i$ != label_$l_j$) then
                  sLabel = label_$l_j$ + " " + label_$l_i$
                  bUseBroaderArea = true
              else
```

```
                  sLabel = label_$l_j$
                  bUseBroaderArea = false
              end if
          else
              sLabel = label_$l_j$
              bUseBroaderArea = false
          end if
      end if
  else
      sLabel = label_$l_j$
      bUseBroaderArea = false
  end if

  //query Google
  sQuery = “https://maps.googleapis.com/maps/api/geocode/xml
      ?address”= + sQuery

  responseCode = URL(sQuery) //query this URL
  if(responseCode==200) then
      gj <- results from the query
  else
      if(bUseBroaderArea==true) then
          sQuery = “https://maps.googleapis.com/maps/api/
                  geocode/xml?address=” + label_$l_j$
          responseCode = URL(sQuery) //query this URL
          if(responseCode==200) then
              gj <- results from the query
          end if
      end if
  end if
end for

//step 2. identify mappings
//step 2.1. identify mappings for areas which do not have broader
areas

for each area $l_j$ in L do
  //step 2.1.1. check if this is a country
  if(label_$l_j$ is name_of_a_country or
      uri_$l_j$.endswith(name_of_a_country) or
      uri_$l_j$.endswith(iso-alpha-2) or
      uri_$l_j$.endswith(iso-alpha-3)) then
      set uri_$g_j$ based on name_of_this_country
      set type_$g_j$ = administrative-area
      continue;
  end if

  if(indexOfBroaderArea(L, $l_j$)==-1) then
      if(size($g_j$)==0) then
          if(label_$g_j$!=””) then
```

```
              set uri_$g_j$ to “/undefined/”+label_$g_j$
          else
              set uri_$g_j$ to “/undefined/”+ending-part-of uri_$l_j$
          end if
          type_$g_j$ = non-administrative-area
      else
          if(size($g_j$)==1) then
              type_$g_j$ = administrative-area
          else
              //filter by level of area
              level = getLevel(uri_$l_j$)
              googleLevel = -1
              for each area $l_k$ in L do
                  if(getLevel(uri_$l_k$)==level && size(uri_$g_k$)==1))
                  then
                      googleLevel = getLevel(uri_$g_k$)
                  end if
              end for
              //assume $g_j$ = {$g_{j1}$,..,$g_{jm}$}, $g_{jk}$ = (uri_$g_{jk}$, label...)
              if(googleLevel!=-1) then
                  for each $g_{jk}$ in $g_j$
                      if(googleLevel!=getLevel(uri_$g_{jk}$)) then
                          remove $g_{jk}$
                      end if
                  end for
              end if
              if(size($g_j$)==0) then
                  type_$g_j$ = non-administrative-area
              else
                  if(size($g_j$)==1) then
                      type_$g_j$ = administrative-area
                  else
                      //filter by narrower areas, assume
                      $g_j$={$g_{j1}$,..,$g_{jm}$},$g_{jk}$=(uri_$g_{jk}$, label...)
                      in L, identify narrower areas of uri_$l_j$
                      => set of indexes{$j_1$,..,$j_t$}
                      in G, identify uri_$g_{jk}$ that has the largest
                      apperance in {$g_{j1}$, $g_{j2}$,.., $g_{jt}$)
                      keep $g_{jk}$ in $g_j$ and remove other results
                      type_$g_j$ = administrative-area
                  end if
              end if
          end if
      end if
  end if
end for

//step 2.2. identify mappings for areas which have broader areas
for each area $l_j$ in L do
```

```
  if(size($g_j$)==1 && type_$g_j$!=null) then
      continue;
  end if

  i = indexOfBroaderArea(L, $l_j$)
  if(size($g_j$)==1 && type_$g_j$==non-administrative-area) then
      i = indexOfBroaderArea(L, $l_i$)
  end if

  if(size($g_i$)==1 && type_$g_i$==administrative-area) then
      assume $g_j$ = {$g_{j1}$,..,$g_{jm}$}, $g_{jk}$ = (uri_$g_{jk}$, label...)

      //filter by boarder area
      for each result $g_{jk}$ in $g_j$ do
          if (! isBroaderAreaInGoogleGeo(uri_$g_i$, uri_$g_{jk}$)) then
               remove $g_{jk}$ from $g_j$
          end if
      end for

      //filter by distance
      if(size($g_j$)>1) then
          filterByDistance(G, $g_j$)
      end if

      //identify mapping
      if(size($g_j$)==1) then
          type_$g_j$ = administrative-area
      else
          if(label_$g_j$!="") then
              set uri_$g_j$ to "/undefined/label_$g_j$
          else
              set uri_$g_j$ to "/undefined/ending-part-of uri_$l_j$
          end if
          type_$g_j$ = non-administrative-area
      end if
  end if
end for

//step 2.3. identify mappings for areas which haven't defined yet
for each area $l_j$ in L do
  if(size($g_j$)==1 && type_$g_j$!=null) then
      continue;
  end if

  //filter by distance
  if(size($g_j$)>1) then
      filterByDistance(G, $g_j$)
  end if
```

```
          //identify mapping
          if(size(gj)==1) then
              type_gj = administrative-area
          else
              if(label_gj!="") then
                  set uri_gj to "/undefined/label_gj
              else
                  set uri_gj to "/undefined/ending-part-of uri_lj
              end if
              type_gj = non-administrative-area
          end if
      end for
      end procedure
```

4.4. Identifying co-reference URIs for unit of measure

Input:

- a URI and its label
- e.g., http://dd.eionet.europa.eu/vocabulary/eurostat/unit/1000PERS, 1000 persons

Output:

- co-reference URI of this URI
- e.g., http://statspace.linkedwidgets.org/codelist/cl_unitMeasure/P1.3

Method:

- Step 1. Identify unit (i.e., P1)
  Use step of possible names for a unit to identify the co-reference URI for the input URI. For example, to detect unit “People”, our set is as follows:
  {"people", “person", "worker", "population", "migration", "migrant", "labor", "births", "adults"}
- Step 2. Indentify scale/power of unit (i.e., 3)
  Detect value in the input label, then identifying scale of this unit. For example, if a label contains “1000” or "1,000" or "1.000" or "thousand", returns value 3.