Part 4.7
Knowledge Graph Evaluation

1 A Methodology for Data Reuse

2 Phase 1 - Purpose Definition

3 Phase 2 - Information Gathering

4 Phase 3 - Language Definition

5 Phase 4 - Knowledge Definition

6 Phase 5 - Data Definition

7 Knowledge Graph Evaluation

The KG's evaluation

- **How to evaluate the quality of the final KG, as well as the entire KG construction porcess ?**
- The iTelos methodology's structure includes different **evaluation activities** to be executed at the end of each phase, to check if the intermediate outputs is suitable to be processed by the next phase or it needs to be revised. [44]

[44] In this course, for lack of time, the evaluation is done only at the end of the process.

The KG's evaluation

- iTelos provides different criteria to evaluate the primary and secondary objectives of a process execution.
- The criteria, described below, consider both the Knowledge and Data layer evaluation.
  - **Primary objective - Purpose satisfaction**: How much the final KG is able to satisfy the Competency Queries ?
    - **Knowledge layer**: Evaluation of CQs vs KG's Teleonotlogy
    - **Data layer**: Evaluation of KG connectivity
  - **Secondary objective - Reusability**: How much reusable is the final KG ?
    - **Knowledge layer**: Evaluation of Teleontology vs Reference Ontologies

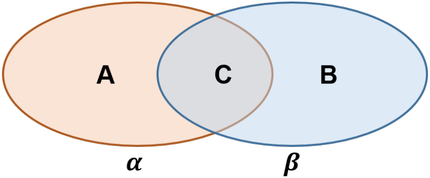
KG's Evaluation

- Knowledge Layer Evaluations
  - Primary goal - Purpose-based evaluation
  - Secondary goal - Reusability evaluation
- Data Layer Evaluations
  - Final KG evaluation
  - KG construction process evaluation

Evaluation metrics

- iTelos provides a set of metrics to be used for the above evaluations.
- Between them one of the most usefull is:
  - **Coverage**: How much a portion of knowledge (shaped as etypes and properties) is covered by a KG.
- TO evaluate the **Knowledge layer** for the primary and secondary objectives the coverage is used as follows:
  - **Primary objective** (Teleontology vs CQs): How much the Teleontology covers the Entities and properties extracted from the CQs.
  - **Secondary objective** (Teleontology vs Reference Ontologies): How much the Teleontology covers the etypes, and properties, extracted from the reference ontologies.

Metric definitions: Coverage



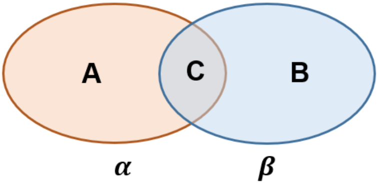
The Coverage is computed as the ratio between the intersection of α and β and the whole α sets:

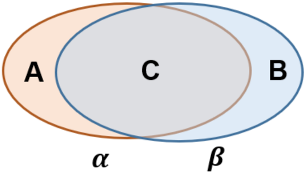
$$Cov = (\alpha \cap \beta)/\alpha = C/(A + C) \tag{1}$$

Where:

- α is a portion of knowledge to be verified.
- β is the KG's Knowledge layer.

Metric definitions: Coverage (extreme cases)





$$Cov = (\alpha \cap \beta)/\alpha = C/(A + C)$$

$Cov \simeq 0$

$Cov \simeq 1$

Metric definitions: Coverage

About the Coverage used to evaluate KGs: $\mathit{Cov} = (\alpha \cap \beta)/\alpha$

- Values are always within the interval [0,1].
- **High values of Coverage** mean that the KG's knowledge is appropriate for the domain.
- For **low values of Coverage**, we can have two possibilities.
    - The reference schema is not appropriate for the domain and maybe a further lookup should be performed.
    - The domain targeted by the knowledge graph is mostly unexplored.

Teleontology vs CQs - EType level

Given a set of (CQ), the **etype coverage** (Cov_E) of the Teleontology (T) is:

$$Cov_E(CQ_E) = \frac{|CQ_E \cap T_E|}{CQ_E} \tag{2}$$

Where:

- CQ_E is the number of etypes extracted from the CQs.
- T_E is the number of etypes of the Teleontology.

Teleontology vs CQs - Property level

Given a set of (CQ), the **property coverage** (Cov_p) of the Teleontology (T) is:

$$Cov_p(CQ_p) = \frac{|CQ_p \cap T_p|}{CQ_p} \tag{3}$$

Where:

- CQ_p is the number of properties extracted from the CQs.
- T_p is the number of properties of the Teleontology.

Teleontology vs Reference Ontologies (ROs) - EType level

Given a set of (RO), the **etype coverage** (Cov_E) of the Teleontology (T) is:

$$Cov_E(RO_E) = \frac{|RO_E \cap T_E|}{RO_E} \tag{4}$$

Where:

- RO_E is the number of etypes extracted from the ROs.
- T_E is the number of etypes of the Teleontology.

Teleontology vs Reference Ontologies (ROs) - Property level

Given a set of (RO), the **property coverage** (Cov_p) of the Teleontology (T) is:

$$Cov_p(RO_p) = \frac{|RO_p \cap T_p|}{RO_p} \tag{5}$$

Where:

- RO_p is the number of properties extracted from the ROs.
- T_p is the number of properties of the Teleontology.

KG's Evaluation

- Knowledge Layer Evaluations
  - Primary goal - Purpose-based evaluation
  - Secondary goal - Reusability evaluation
- Data Layer Evaluations
  - Final KG evaluation
  - KG construction process evaluation

The KG's evaluation - Data layer

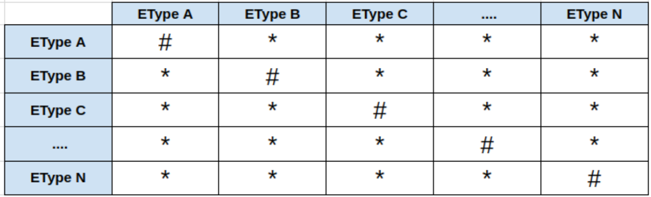
- Evaluating the KG's data layer, aims to understand how **"dense"** or **"connected"** is the KG, at the end of the iTelos process, and during the KG's construction.
- We can evaluate **the KG's connectivity** in two different moments:
  - **On the final KG**: this evaluation aims to understand how much connected is the KG at the end of the process.
  - **During the KG's construction**: this evaluation aims to understand how much each single dataset, handled during the process, improve the connectivity of the final KG.
- **Note**: the improvement of connectivity brought by a single dataset to the KG, can be different when the dataset is added to the partial KG (during construction), respect to the connectivity evaluated over the same dataset's values, over the final KG.
  - The difference is caused by the **entity matching conflicts** and their solutions.

The KG's evaluation - Data layer

- The **connectivity** of a KG can be evaluated over two dimensions:
  - **Entity connectivity**: How much the entities are connected to each other. It evaluates the grade of connection between the different entities in the KG.
  - **Property connectivity**: How much the entities are connected to their properties. It evaluates the grades of connection between each single KG's entity and its properties values.

The KG's evaluation - Data Layer - Final KG

- The Entity and Property Connectivity can be calculated by using the **connectivity matrix**.



	EType A	EType B	EType C	....	EType N
EType A	#	*	*	*	*
EType B	*	#	*	*	*
EType C	*	*	#	*	*
....	*	*	*	#	*
EType N	*	*	*	*	#

- Where the value of cell (X,Y) is:
  - # : (X = Y) the number of **non-null data properties** values, for the all the entities mapped on the EType X (or Y).
  - * : (X ≠Y) the number of **non-null object properties** values, for the object properties having the EType X as domain and the EType Y as range.

The KG's evaluation - Data Layer - Final KG

- The **Entity Connectivity** is calculated EType by EType, as the sum of the **"* values"** for each EType row in the matrix. The resulting row sum, relative to the EType X, is then divided by the number of **object properties** defined for the EType X.
    - The sum of the Entity Connectivity of all the KG's ETypes, defines the values of Entity Connectivity of the whole KG.
- The **Property Connectivity** is calculated EType by EType, dividing the **"# values"** for the cell (X,X) by the number of **data properties** defined for the EType X.
    - The sum of the Property Connectivity of all the KG's ETypes, defines the values of Property Connectivity of the whole KG.

The KG's evaluation - Data Layer - Final KG

- In formulae:
  - **EC(X) Entity Connectivity for the EType X**:

$$EC(X) = \frac{\sum_{Y=1}^{N}(X, Y)}{OP(X)} \tag{6}$$

  - **Where**: (X, Y) is a cell in the connectivity matrix, and OP(X) is the number of object properties of the ETypes X.
  - **EC(KG) Entity Connectivity for the whole KG**:

$$EC(KG) = \sum_{X=1}^{N} EC(X) \tag{7}$$

The KG's evaluation - Data Layer - Final KG

- In formulae:
  - **PC(X) Property Connectivity for the EType X**:

$$PC(X) = \frac{(X, X)}{DP(X)} \tag{8}$$

  - **Where**: (X, Y) is a cell in the connectivity matrix, and DP(X) is the number of object properties of the ETypes X.
  - **PC(KG) Property Connectivity for the whole KG**:

$$PC(KG) = \sum_{X=1}^{N} EC(X) \tag{9}$$

The KG's evaluation - Data Layer - Construction

- To evaluate the connectivity improvement brought by a new dataset that has to be integrated into the KG, we have to consider the following cases.

- It is possible to calculate the **entity and property connectivity** (see previous slides) to measure the impact of new datasets over the KG, in construction.

- **Assumption**: There are, a new dataset D_1 and the partially built graph KG. Moreover, D_1 has an etype E_1, with its property set A_1 and KG has an etype E_2, with its property set A_2.

The KG's evaluation - Data Layer - Construction

- **Case 1**: $[E_1 = E_2]$ The E_1 in D_1 is already present in KG.
- **Consequence**: By integrating D_1 into KG we are increasing the number of entities of E_1, thus **increasing the entity connectivity**.
    - **Case 1.1**: $[A_1 = A_2]$ The etypes share the same set of properties.
    - **Consequence**: Conflicts are possible between the value set of A_1 and A_2.
        - How many conflicts ?
        - How many new entities from D_1 are integrated into the KG ?
        - How many properties, in the property set A_1, with not null values remain after solving such conflicts ?

The KG's evaluation - Data Layer - Construction

- **Case 1**: $[E_1 = E_2]$ The E_1 in D_1 is already present in KG.
- **Consequence**: By integrating D_1 into KG we are increasing the number of entities of E_1, thus **increasing the entity connectivity**.
  - **Case 1.2**: $[A_1 \neq A_2]$ The etypes have different sets of properties.
  - **Consequence**: There are no conflicts between the value set of A_1 and A_2, and there is a greater increase of the integration over the etype E_1. Notice how in this case also **the property connectivity increases**.
    - How many new entities from D_1 are integrated into the KG ?
    - How many properties, in the property set $A_1 \cup A_2$, with not null values remain after the integration of D_1 ?

The KG's evaluation - Data Layer - Construction

- **Case 2**: $[E_1 \neq E_2]$ The E_1 in D_1 is not yet present in KG.
- **Consequence**: By integrating D_1 into KG we are increasing the number of etypes of KG.
  - **Case 2.1**: E_1 and E_2 are linked by at least one object property.
  - **Consequence**: The resulting KG, after the integration of D_1, is connected.
    - How many connections ?
    - How many entities of E_1 have not null values for the object properties linking E_1 with KG ?

The KG's evaluation - Data Layer - Construction

- **Case 2**: $[E_1 \neq E_2]$ The E_1 in D_1 is not yet present in KG.
- **Consequence**: By integrating D_1 into KG we are increasing the number of etypes of KG.
  - **Case 2.2**: There are no object properties linking E_1 and E_2.
  - **Consequence**: The resulting KG, after the integration of D_1, is not connected.
  - The integration of D_1 doesn't increase the connectivity, thus the information carried by D_1 cannot be reached by the KG.