



KNOWDIVE



KDI ● **Knowledge and Data Integration**

Evaluation

iTelos Formal Modeling & Data integration

Fausto Giunchiglia

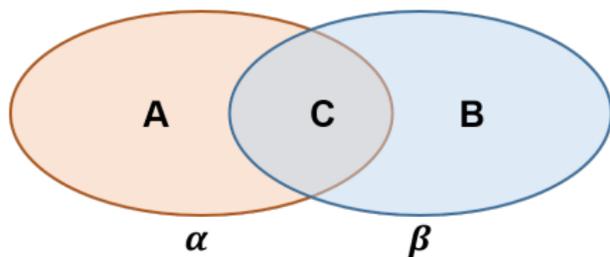
Contents

- 1 Metrics Definitions**
- 2 Evaluation on Formal Modeling phase**
- 3 Evaluation on Data Integration phase**

Contents

- 1 Metrics Definitions**
- 2 Evaluation on Formal Modeling phase
- 3 Evaluation on Data Integration phase

Metrics Definitions



- Coverage (*Cov*)
- Extensiveness (*Ext*)
- Sparsity (*Spr*)
- Cue Validity (*Cue*)

Metric definitions: Cue validity

Cue is a set of metrics to measure the quality of the Etype/ETG. By applying Cue, we focus on:

- Shareability and unity [1], we measure if the Etype/ETG is well-described by its features.
- Property richness [2], since we calculate the average number of properties that assigned to different Etypes.

[1] Giunchiglia, F. and Fumagalli, M., 2020, July. Entity type recognition—dealing with the diversity of knowledge. In Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (Vol. 17, No. 1, pp. 414-423).

[2] Tartir, S., Arpinar, I.B., Moore, M., Sheth, A.P. and Aleman-Meza, B., 2005. OntoQA: Metric-based ontology quality analysis.

Metric definitions: Cue validity

Cue is a set of metrics to measure the quality of the Etype/ETG.

Cue for Etype:
$$Cue_e(e) = \sum_{i=1}^{|\mathit{prop}(e)|} Cue_p(p_i, e) \in [0, |\mathit{prop}(e)|]$$

Cue for ETG:
$$Cue_k(K) = \sum_{i=1}^{|E_K|} Cue_e(e_i) \in [0, |\mathit{prop}(K)|]$$

Metric definitions: Cue validity

$$\text{Cue for Etype: } Cue_e(e) = \sum_{i=1}^{|\text{prop}(e)|} Cue_p(p_i, e) \in [0, |\text{prop}(e)|]$$

$$Cue_p(p, e) = \frac{PoE(p, e)}{|\text{dom}(p)|} \in [0, 1] \quad PoE(p, e) = \begin{cases} 1, & \text{if } e \in \text{dom}(p) \\ 0, & \text{if } e \notin \text{dom}(p) \end{cases}$$

- e represents an Etype. $Cue_e(e)$ represents the *cue* validity of the Etype e .
- $|\text{prop}(e)|$ is the number of properties associated with the specific entity type e .
- $Cue_p(p, e)$ returns 0 if p is not associated with e . Otherwise returns $1/n$, where n is the number of entity types in the domain of p . Cue_p takes the maximum value 1 if p has only one entity type.
- $|\text{dom}(p)|$ presents the cardinality of entity types that are the domain of the specific property p .
- $PoE(p, e)$ determines if the Etype e is in the domain of property p .

Metric definitions: Cue validity

Cue for ETG:
$$Cue_k(K) = \sum_{i=1}^{|E_K|} Cue_e(e_i) \in [0, |prop(K)|]$$

- The $Cue_k(K)$ is calculated as a summation of the cue validity $Cue_e(e)$ of all the entity types e_i in a given ETG K ,
- E_K presents the number of Etypes in a given ETG K .
- $|prop(K)|$ refers to the number of the properties in the ETG, as the maximization of $Cue_k(K)$.

Metric definitions: Cue validity

About $Cue_e(e)$ and $Cue_k(K)$:

- Values are always within the interval $[0,1]$.
- It captures the idea, that is Etypes are properly described by more specific properties.
- High values of Cue mean that there are enough number of properties for specifically describing the target Etype/ETG, which makes the target Etype more likely belongs to contextual category.
- Low values of Cue mean that the target Etype/ETG is describe by few general properties, which makes the target Etype more likely belongs to common category.

Contents

- 1 Metrics Definitions
- 2 Evaluation on Formal Modeling phase**
- 3 Evaluation on Data Integration phase

Evaluation purpose on Formal Modeling phase

During Formal modeling phase, we evaluate the on schema level. We have the formal ETG and several reference ontologies. We aim to measure:

- If the formal ETG and its Etypes are properly defined by their properties, using metric $Cue_k(ETG)$ and $Cue_e(Etype)$
- If the proposed ETG is different from the reference ontologies, using metric *Sparsity*.
- If the ETG is well-designed, and information in the ETG is correct. By sampling from ETG and then **checking manually**.

Examples: Formal ETG vs Reference Ontology

To calculate Cue validity, we should first generate a FCA lattice for the target ETG.

Here we select an example from DBpedia:

FCA Context		Properties						
		name	date	citizenship	settlement	academy award	gold medalist	race track
Etypes	Person	O	X	O	X	X	X	X
	Event	X	O	X	X	X	X	X
	Place	O	X	X	O	X	X	X
	Artist	O	X	O	X	O	X	X
	Athlete	O	X	O	X	X	O	X
	Sports Event	X	O	X	X	X	X	O

Examples: Formal ETG vs Reference Ontology

According to the FCA lattice, we further calculate $Cue(ETG)$, $Cue(Etypes)$ by the Cue metrics.

$$\text{Cue for Etype: } Cue_e(e) = \sum_{i=1}^{|\text{prop}(e)|} Cue_p(p_i, e) \in [0, |\text{prop}(e)|]$$

$$\text{Cue for ETG: } Cue_k(K) = \sum_{i=1}^{|E_K|} Cue_e(e_i) \in [0, |\text{prop}(K)|]$$

Cue_p Map		Properties							$Cue_e(E)$
		name	date	citizenship	settlement	academy award	gold medalist	race track	
Etypes	Person	0.25	0	0.33	0	0	0	0	0.29
	Event	0	0.5	0	0	0	0	0	0.5
	Place	0.25	0	0	1	0	0	0	0.625
	Artist	0.25	0	0.33	0	1	0	0	0.79
	Athlete	0.25	0	0.33	0	0	1	0	0.79
	Sports Event	0	0.5	0	0	0	0	1	0.75
$Cue_k(K) =$									3.745

Cue calculation service can be found at: http://liveschema.eu/service/cue_generator

Generating Cue by LiveSchema service

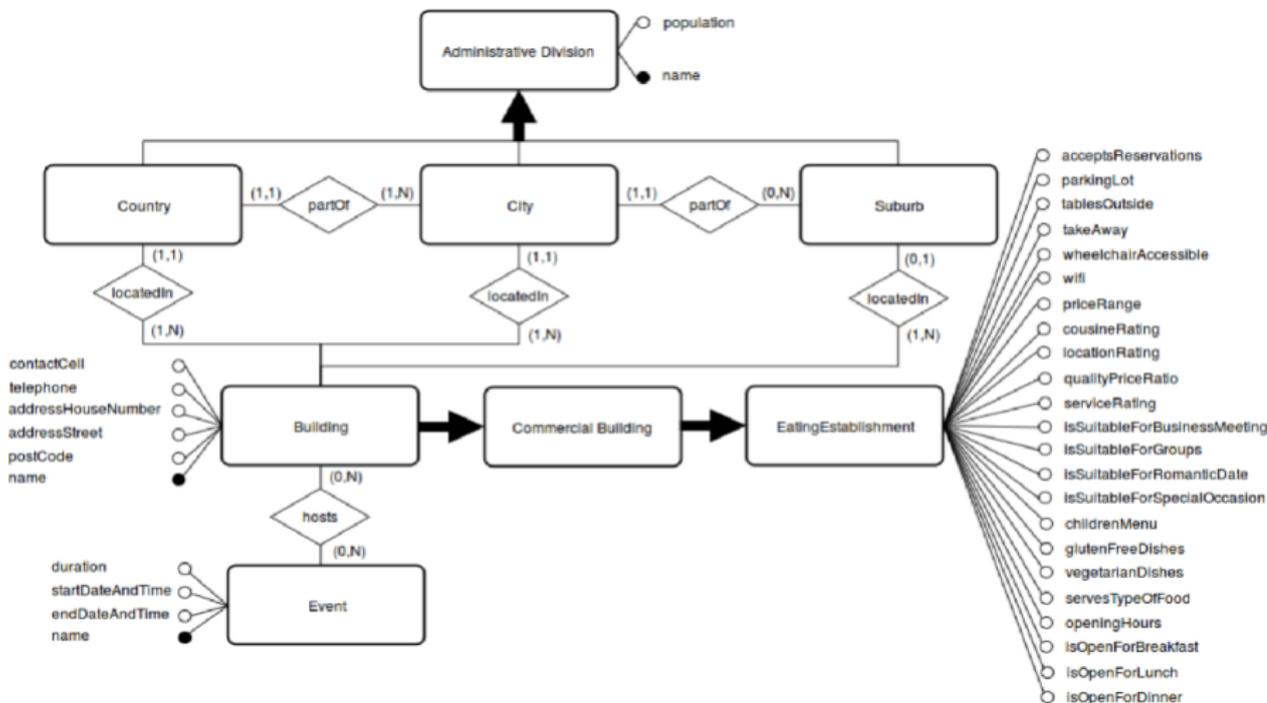
Cue calculation service can be found at LiveSchema:

http://liveschema.eu/service/cue_generator

#Type	Cue_e	Cue_er	Cue_ec	Cue_c	Cue_cr	Cue_cc
1.1	0.0	0.0	1.0	0.0	0.0	1.0
1.8	0.0	0.0	1.0	0.0	0.0	1.0
1.9	0.0	0.0	1.0	0.0	0.0	1.0
4.0	0.0	0.0	1.0	0.0	0.0	1.0
Class	1.25	0.3125	0.6875	1.3333...	0.2666...	0.7333...
DatatypeProperty	3.0	0.25	0.75	3.0	0.25	0.75
Graph	3.4166...	0.2628...	0.7371...	3.3333...	0.2564...	0.7435...
ObjectProperty	0.8333...	0.2777...	0.7222...	0.75	0.25	0.75
Ontology	0.8333...	0.2777...	0.7222...	0.75	0.25	0.75
Product	0.25	0.25	0.75	0.25	0.25	0.75
Property	0.3333...	0.3333...	0.6666...	0.5	0.25	0.75
System	0.25	0.25	0.75	0.25	0.25	0.75
Thing	0.5	0.25	0.75	0.5	0.25	0.75
ThingGraph	0.25	0.25	0.75	0.25	0.25	0.75
anyURI	0.5	0.25	0.75	0.5	0.25	0.75
bbc	0.0	0.0	1.0	0.0	0.0	1.0
boolean	0.5	0.25	0.75	0.5	0.25	0.75
dateTime	0.5	0.25	0.75	0.5	0.25	0.75
provenance	5.0833...	0.2675...	0.7324...	5.0833...	0.2541...	0.7458...
provenance.ttl	0.0	0.0	1.0	0.0	0.0	1.0
string	1.5	0.25	0.75	1.5	0.25	0.75
terms	0.0	0.0	1.0	0.0	0.0	1.0
KNOWLEDGE	19.0	0.2638...	0.7361...	19.0	0.2533...	0.7466...

*Notice, the Column in red refers to $Cue_e(E)$, and the value box in blue refers to $Cue_k(K)$

Examples: Formal ETG



Examples: Formal ETG

Classes/Etypes in ontology:

$C_c = \{\text{AdministrativeDivision, Country, City, Suburb, Building, CommercialBuilding, EatingEstablishment, Event}\}$

(Num class = 8)

Properties in ontology:

$C_p = \{\text{rating, suitableForGroup, childrenMenu, Vegetarians, contact, reservation, parkingLot, OpenForlunch}\}$

(Num property = 38)

Examples: Reference Ontologies

Reference ontology have similar structure with Formal ETG, which contains a set of Etypes and a set of properties.

Classes/Etypes in ontology:

$C_c = \{\text{Region, City, Suburb, Building, Customer, EatingEstablishment, \dots, Event, FestivalEvent}\}$

(Num class = 21, 5 of them aligned with formal ETG)

Properties in ontology:

$C_p = \{\text{rating, suitableForGroup, childrenMenu, Vegetarians, contact, reservation, parkingLot,duration, menu, \dots, parkingArea}\}$

(Num property = 50, 17 of them aligned with formal ETG)

Examples: Formal ETG vs Reference Ontology

Given the reference ontology (Ont), the sparsity (Spr) of the Formal ETG (ETG) is:

$$\text{Etype sparsity } Spr(ETG_c) = \frac{|ETG_c - Ont_c| + |Ont_c - ETG_c|}{|ETG_c \cup Ont_c|} = 1 - \frac{|ETG_c \cap Ont_c|}{|ETG_c \cup Ont_c|}$$

$Spr = 1$ Full Sparsity

$Spr \approx 0.5$ Ideal

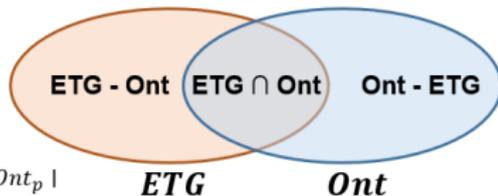
$Spr = 0$ No Sparsity

$$\text{Property sparsity } Spr(ETG_p) = \frac{|ETG_p - Ont_p| + |Ont_p - ETG_p|}{|ETG_p \cup Ont_p|} = 1 - \frac{|ETG_p \cap Ont_p|}{|ETG_p \cup Ont_p|}$$

$Spr = 1$ Full Sparsity

$Spr \approx 0.5$ Ideal

$Spr = 0$ No Sparsity



Examples: Formal ETG model vs Reference Ontology

Given the example ETG , and the example reference ontology, we have sparsity (Spr) as:

$$\text{Etype Sparsity } Spr(ETG_c) = 1 - \frac{|ETG_c \cap Ont_c| (5)}{|ETG_c \cup Ont_c| (24)} = 0.79$$

$$\text{Property Sparsity } Spr(ETG_p) = 1 - \frac{|ETG_p \cap Ont_p| (17)}{|ETG_p \cup Ont_p| (71)} = 0.76$$

**Notice that different (sparsity) information should be core or contextual information.*

Contents

- 1 Metrics Definitions
- 2 Evaluation on Formal Modeling phase
- 3 Evaluation on Data Integration phase**

Evaluation purpose on Data Integration phase

During Data Integration phase, we evaluate on data level. We have the proposed final EG. We aim to measure:

- If the CQs in inception phase can be answered by our constructed EG. We can do evaluation based on practical applications, like SQL.
- If our collected dataset is sufficiently used. By using **Sparsity** to check if the dataset schema is aligned to ETG properties. Otherwise, there will be a loss of dataset information.

Examples: EG vs CQs

One of the reasons for constructing a new EG is to answer the CQs we proposed. Thus, answering CQs is the key aspect for EG evaluation.

We apply our EG on applications to search/reason the results for CQs. In the current situation, we reorganize CQs into SQL commands to straightforwardly search the results from EG.

During the evaluation, we record the **accuracy** and **running time** to test if our constructed EG can effectively solve the CQs.

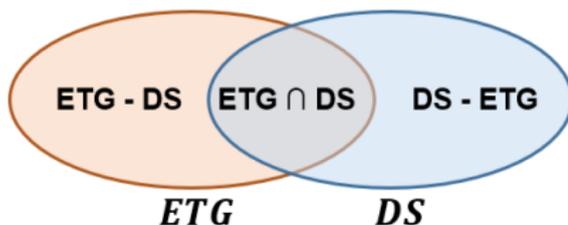
Examples: Formal ETG vs Dataset Schema

Given the Formal ETG (ETG), the sparsity (Spr) of the Dataset Schema (DS) is:

Property sparsity

$$Spr(DS_p) = \frac{|DS_p - Ont_p| + |Ont_p - DS_p|}{|DS_p \cup Ont_p|} = 1 - \frac{|DS_p \cap Ont_p|}{|DS_p \cup Ont_p|}$$

$Spr = 1$ Full Sparsity
 $Spr \approx 0$ Ideal
 $Spr = 0$ No Sparsity



*The calculation of the *Sparsity* in data integration phase is similar to the Formal ETG modelling phase.

Non-quantitative evaluation

Rather than the quantitative evaluation we introduced above, we also need to check in the schema-level and data-level of the EG:

Checklist

- Consistency Dimension
- Accuracy Dimension
- Completeness Dimension

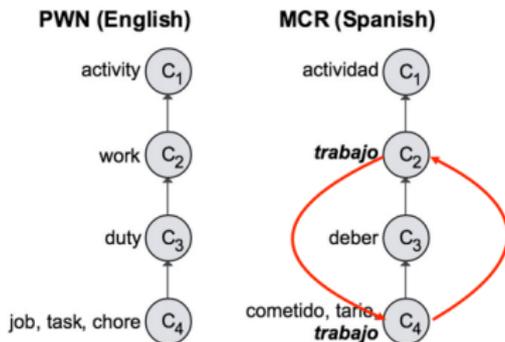
[1] Mc Gurk, S., Abela, C. and Debattista, J., 2017. Towards Ontology Quality Assessment. In MEPPaW/LDQ@ ESWC (pp. 94-106).

Consistency Dimension

Problem: Including Cycles in a class Hierarchy. Circulatory errors typically occurs, for example, when a class X1 is defined as a superclass of class X2, and X2 is defined as a superclass of X1 at the same time.

Solution: Do not use cycles in a class hierarchy.

Example:



Consistency Dimension

Problem: **Number of Polysemous Elements.** Number of properties, objects or datatypes that are referred by the same identifier. For example, 'contact' might refer to different but related concept, such as referring to 'contact information' or 'address'.

Solution: **Avoid polysemous term.**

Example:

The screenshot shows a window titled "Description: findingRelatedToBiologicalSex". It contains four sections, each with a plus icon for expansion:

- Equivalent To:** Lists two entries, both labeled "sex", each with a yellow background and control icons (question mark, @, X, O).
- SubProperty Of:** Lists one entry labeled "personalHistory" with a yellow background and control icons.
- Domains (intersection):** Lists one entry labeled "person" with a yellow background and control icons.
- Ranges:** Lists one entry labeled "xsd:string" with a red background and control icons.

Consistency Dimension

Problem: **Multiple Domains/Ranges.** Multiple domains and ranges are allowed, however, these should not be in conflict with each other (that is, no two domains or ranges should contradict each other).

Solution: Use only one domain and range for each property.

Example:



The screenshot shows a software interface for editing a property named "countryOfOrigin". The title bar reads "Description: countryOfOrigin". The main area contains several sections, each with a plus sign icon to its right:

- Equivalent To**: No items listed.
- SubProperty Of**: No items listed.
- Inverse Of**: No items listed.
- Domains (Intersection)**: One item, "person", is listed with a yellow dot to its left. To the right of this list are four circular icons: a question mark, an at-sign, an 'x', and a circle with a dot.
- Ranges (Intersection)**: One item, "country", is listed with a yellow dot to its left. To the right of this list are the same four circular icons as above.

Consistency Dimension

Problem: **Semantically Identical Classes.** This anomaly occurs when an ontology includes multiple classes with the same semantics. For example, creating a two classes such as 'airport' and 'airdrome' for representing an airfield those are equipped with control tower and hangars.

Solution: **Do not use different term to refer same element.**

Example:

The screenshot displays two panels from an ontology editor. The top panel is titled 'Annotations: surname' and shows the following annotations: 'rdfs:label [language: en] surname' and 'rdfs:comment [language: en] the name used to identify the members of a family'. The bottom panel is titled 'Annotations: familyname' and shows the following annotations: 'rdfs:label [language: en] familyname' and 'rdfs:comment [language: en] the name used to identify the members of a family'. Both panels have a blue header bar with a plus icon and a green bar above it with a search icon and other controls.

Accuracy Dimension

Problem: **Incorrect Relationship.** An incorrect relationship typically occurs with the vague use of 'is', instead of 'subClassOf', 'type' or 'sameAs'. For example, student isA person, uses isA as a relation (i.e. object property in Protégé).

Solution: **Avoid using relation name like isA or type.**

Example:

The screenshot displays the Protégé interface. On the left, a class hierarchy is shown under 'owl:Thing', with 'patient' selected at the bottom. The hierarchy includes categories like 'event', 'mentalObject', 'physicalObject', 'artifact', 'biologicalObject', 'geographicalLocation', 'specimen', 'substance', and 'role'. On the right, the 'Annotations: patient' panel is visible, showing 'isA some person' and 'personalRole' as annotations. The 'Description: patient' panel is also visible, showing 'Equivalent To' and 'SubClass Of' sections.

Accuracy Dimension

Problem: **Hierarchy Over specification.** Over specialisation occurs when a leaf class of a model does not have any instances associated with it. For example, having a class 'Mountain' in the model but did not have data for it.

Solution: Discard any leaf class for which there is no instances.

Example:

The screenshot displays a software interface with two main panels. The left panel, titled 'Class hierarchy', shows a tree structure of classes. The root is 'owl:Thing', which branches into 'event', 'mentalObject', 'physicalObject', 'role', 'socialObject', and 'stative'. Under 'role', there are 'personalRole' (with sub-classes 'baby', 'healthcareProfessional' which is a subclass of 'provider', 'parent', and 'patient') and 'socialObject' (with sub-classes 'organization' and 'healthBoard'). Under 'stative', there are 'process' and 'state' (with sub-class 'physicalCondition' which has a sub-class 'disease'). The class 'pharmaceuticalCompany' is highlighted in blue under 'healthBoard'. The right panel, titled 'Instances:', shows a list of instances for the selected class 'pharmaceuticalCompany'. The instances listed are: A-SMedicationSolutionsLLC, BayerAG, BoehringerIngelheimInternational, Bristol-MyersSquibb, IntrapharmLaboratoriesLtd, JanssenPharmaceuticals, MercuryPharmaceuticalsLtd, MerusLabsLuxcoIIS.à.R.L., PfizerEieg, RosemontPharmaceuticalsLtd, and SunPharmaceuticalIndustriesEuro.

Accuracy Dimension

Problem: Using a Miscellaneous Class. A class within the hierarchy of the ontology which is simply used to represent instances that do not belong to any of its siblings. For example, having the class 'Building' with subclasses 'Hospital, 'Hotel', 'Library' 'Commercial building' and Miscellaneous.

Solution: Do not use miscellaneous or other as a class name.

Example:

The screenshot displays an ontology editor interface. On the left, a class hierarchy tree is shown under 'owl:Thing'. The tree includes classes like 'event', 'mentalObject', 'physicalObject', 'artifact', 'device', 'hospitalFurniture', 'product', 'structure', 'bathingStation', 'building', 'medicalBuilding', 'burnsUnit', 'client'sRoom', 'healthEducationRoom', and 'hospital'. The 'hospital' class is highlighted in blue. On the right, the 'Annotations: hospital' panel is visible, showing 'rdfs:label [language: en] hospital' and 'rdfs:comment [language: en] a health facility where patients receive treatment'. Below this, the 'Description: hospital' panel shows 'Equivalent To' and 'Sub-Class Of' sections, with 'medicalBuilding' listed as a sub-class.

Completeness Dimension

Problem: **Number of Isolated Elements.** Elements, including classes, properties and datatypes are considered isolated if they do not have any relation to the rest of the ontology (declared but not used).

Solution: **Avoid to keep isolated elements.**

Example:

Annotations | Usage

Usage: address

Show: this disjoints

- address
 - DataProperty: address
 - address rdfs:comment "written directions for finding **some** location; written on letters **or** packages **that** are to be delivered to **that** local"
- B103H
 - B103H address "Station Road, Duns, TD11 3EL"^^xsd:string
- B104H
 - B104H address "Tweed Road, Galashiels, TD1 3EB"^^xsd:string
- B105H
 - B105H address "Victoria Road, Hawick, TD9 7AH"^^xsd:string

Completeness Dimension

Problem: **Missing Domain or Range in Properties.** Properties should be accompanied by their domain and range. Missing information about the properties may cause lack of completeness and may result in less accuracy and more inconsistencies.

Solution: **Define domain and range for all properties.**

Example:

The screenshot shows a property editor window titled "Description: ageAtDeath". It contains several sections:

- Equivalent To:** A plus sign icon.
- SubProperty Of:** A plus sign icon, followed by a list of properties: "age" (highlighted in green) and "age" (highlighted in yellow). Each entry has a question mark, at-sign, and X icon to its right.
- Domains (Intersection):** A plus sign icon, followed by a list of domains: "person" (highlighted in blue). It has a question mark, at-sign, and X icon to its right.
- Ranges:** A plus sign icon, followed by a list of ranges: "xsd:integer" (highlighted in red). It has a question mark, at-sign, and X icon to its right.



KDI : Knowledge and Data Integration



Fausto Giunchiglia



Evaluation

iTelos Formal Modeling & Data
integration