From Open Data & Linked Data to Ontology

example: http://www.disit.dinfo.unifi.it/siimobility.html

From Open Data to Linked Data

To map Open Data into Linked Data:

1. **Map** the data **to RDF**: selecting/writing a **domain ontology** with standard terminology, convert data to RDF according to this ontology;
2. **Link** to **external** source: find **links** from the **metadata** to **other repositories** (Dbpedia, GeoNames, etc),
3. **Curate** the **Linked Data**: to ensure that the published information/link to other source are accurate.

First step useful tools (Map the data to RDF)

Extract, Transform and Load (ETL): Process used in database and data warehousing that involves:

- **Extracting** data from outside sources;
- **Transforming** it to fit operational needs, which can include quality levels;
- **Loading** it into the end target (database, operational data store or data warehouse).

Useful tools to prepare data to RDF translation

Pentaho Data Integration (Kettle)

- Free, **open source** (LGPL) ETL (Extraction, Transformation and Loading) tool;
- **Powerful** Extraction, Transformation and Loading (ETL) capabilities;
- It use an innovative, **metadata-driven** approach;
- Graphical, **drag and drop** design environment;
- **Scalable**, standards-based architecture;
- K.E.T.T.L.E, a recursive acronym for "Kettle Extraction, Transport, Transformation and Loading Environment".

Pentaho Data Integration (Kettle)

Designed to:

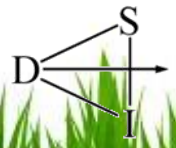
- Collect data from a **variety of sources** (extraction);
- Move and modify data (transport and transform) while cleansing, denormalizing, aggregating and enriching it in the process;
- Frequently (daily) store data (loading) in the final target destination, usually a **large**, dimensionally modelled **database** (or **data warehouse**).

Kettle's 4 main programs

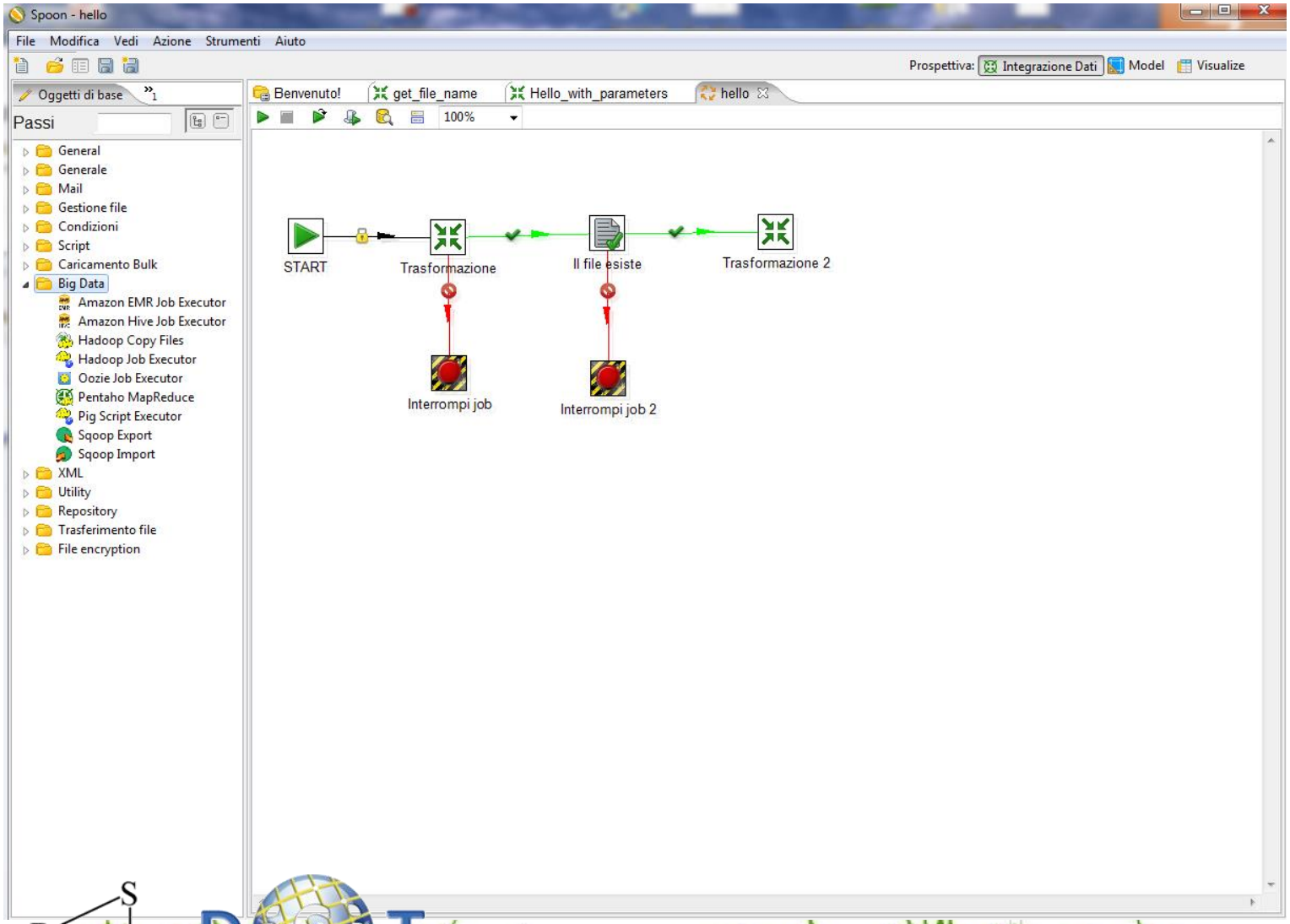
- **Spoon:** graphically oriented end-user tool to model the **flow of data** from input through transformation to output (**transformation**);
- **Pan** is a **command line tool** that executes transformations modelled with Spoon;
- **Chef**: a graphically oriented **end-user tool** used to model **jobs** (transformations, FTP downloads etc. placed in a flow of control);
- **Kitchen** is a **command line tool** to execute jobs created with Chef.

Kettle features

- Interesting feature: Kettle is **model-driven**;
- **Spoon** and **Chef** have a graphical user interface to define the ETL processes on a **high level**;
- **Pan** and **Kitchen** can read and interpret the models created by Spoon and Chef respectively;
- Models can be saved to a particular **XML format**, or they can be stored into a relational database (**repository**);
- Handling many models with repository: models are stored in a structured manner, arbitrary queries can be written against the repository.

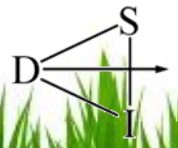



Spoon Interface



Spoon Concepts: steps and hoops

- One **step** denotes a particular kind of **action** that is performed **on data**.
- Steps are easily created by **dragging** the icon from the treeview **and dropping** them on the graphical model view.
- Kettle provides a lot of different step types, and can be **extended with plugin**.
- Three different kinds of steps: **input**, **transform**, **output**.

Type of steps in Spoon (1/2)

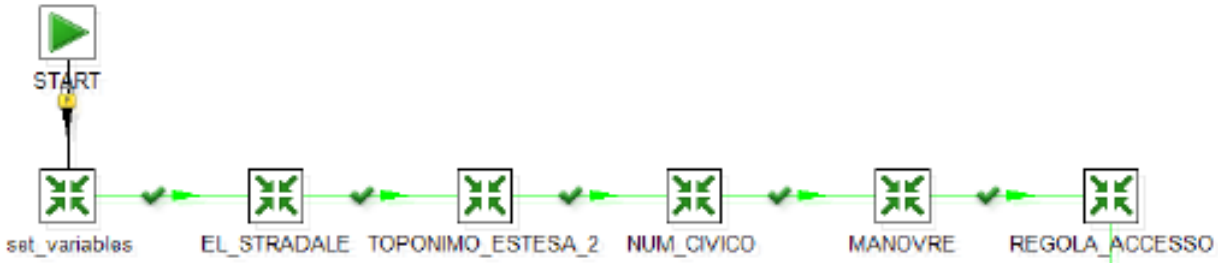
- **Input steps** process some kind of 'raw' resource (file, database query or system variables) and create an outputstream of records from it.
- **Transforming steps** process inputstreams and perform particular action on it (adding new fields/new records); This produce one or more outputstreams. Kettle offers many transformation steps out of the box, very simple tasks (renaming fields) and complex tasks (normalizing data, maintaining a slowly changing dimension in a data warehouse);

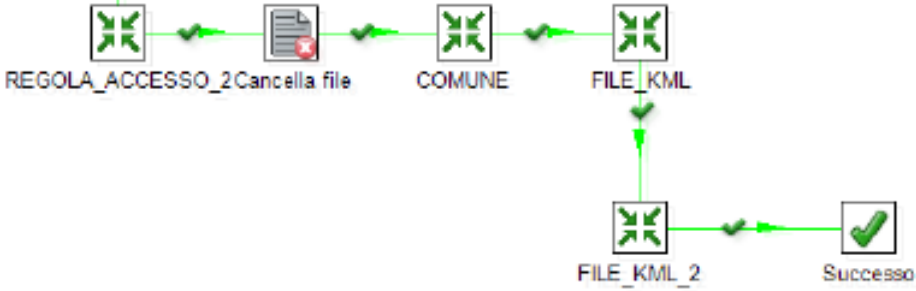
Type of steps in Spoon (2/2)

- **Output steps** (the reverse of input steps): accept records, and store them in some external resource (file, database table, etc).

- Connections between the steps are called **hops.**
- Hops between steps behave like **pipelines**: records may flow through them from one step to the other.
- **Main.kjb** is the primary job.

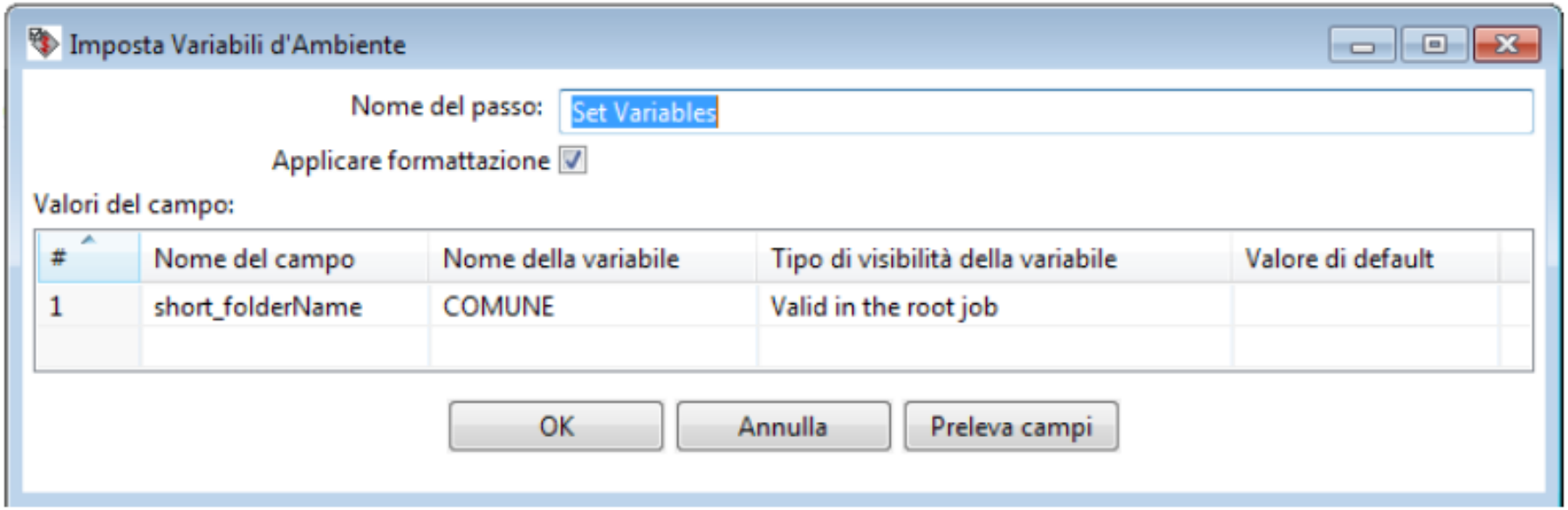
Kettle - SiiMobility Example (1/8)





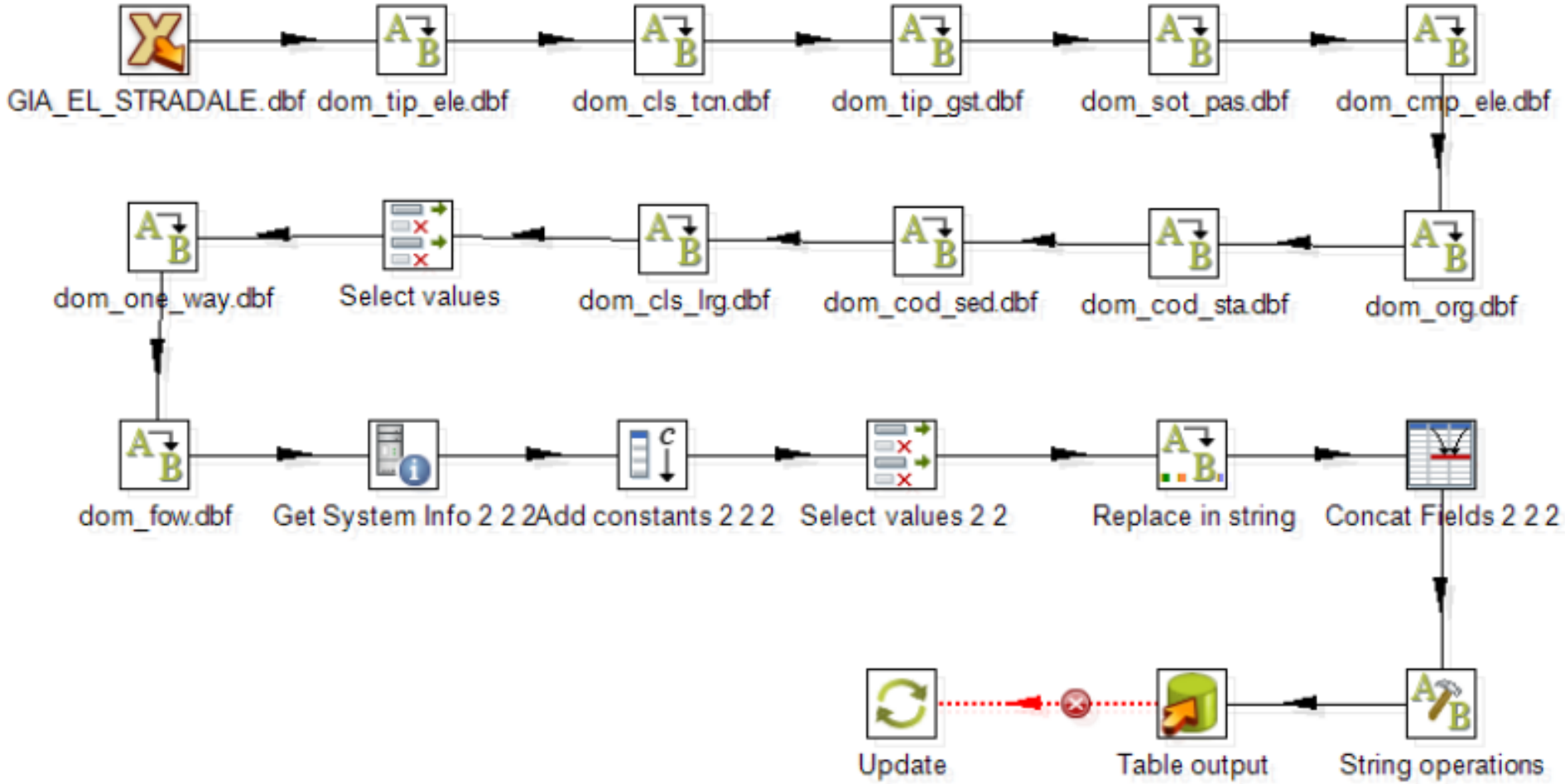
- 10 trasformazioni sequenziali per trattare i dati dell'osservatorio trasporti.
- Elementi stradali, Toponimi, Numeri Civici, Manovre, Regole di Accesso, Comuni, Giunzioni, Accessi, Cippi chilometrici.

Kettle - SiiMobility Example (2/8)



- **SET_VARIABLES**: copia in una variabile di ambiente il nome della sottocartella nella quale sono eseguite le operazioni. Nelle trasformazioni successive sarà possibile referenziarla tramite il comando ${COMUNE};

Kettle - SiiMobility Example (3/8)



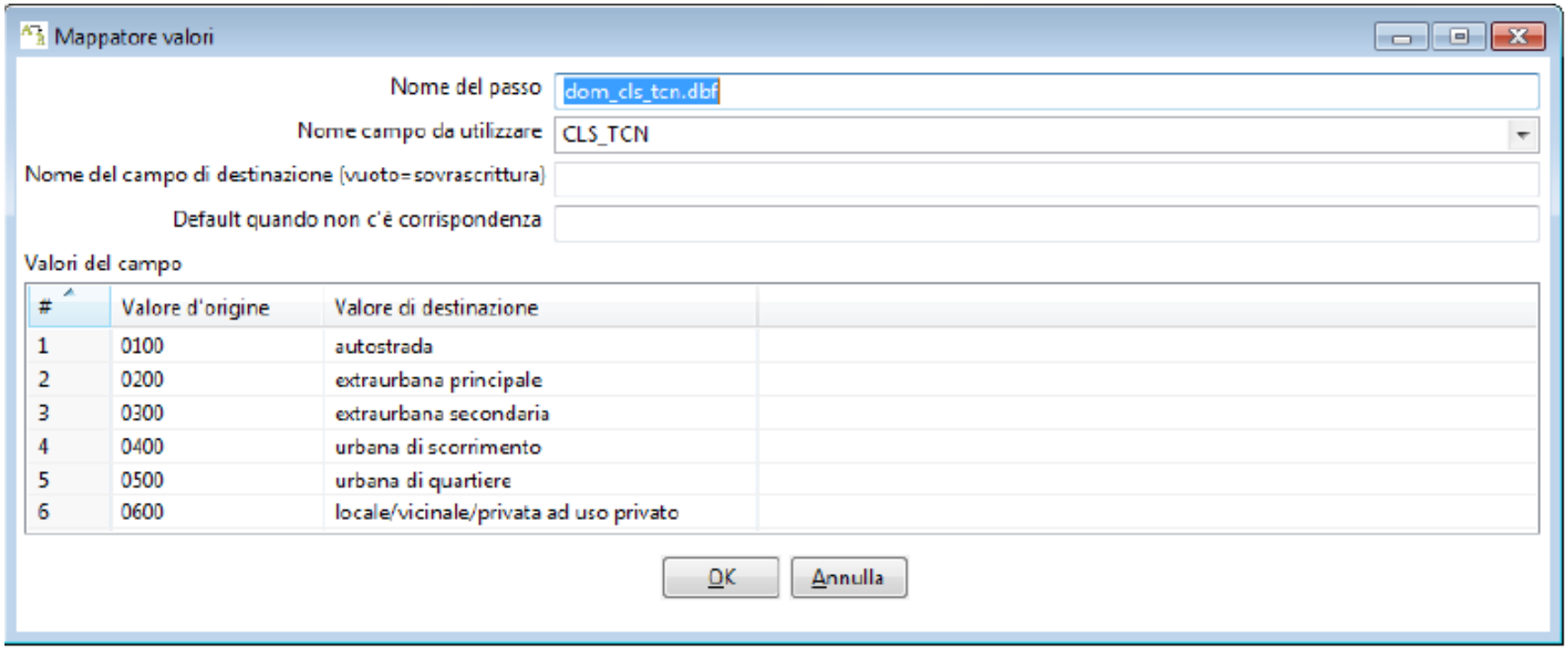
Kettle - SiiMobility Example (4/8)

- GIA_EL_STRADALE: Legge i dati da un file in formato Xbase (DBF);
- DOM_TIP_ELE: Permette di mappare i valori di una colonna attraverso una tabella;
- SELECT_VALUE: Seleziona o rimuove campi in una riga;
- GET_SYSTEM_INFO: Recupera informazioni dal sistema operativo come date, argomenti, ecc.
- ADD_CONSTANTS: Aggiunge una colonna ai dati e valorizza ogni riga con un valore costante;

Kettle - SiiMobility Example (5/8)

- REPLACE_IN_STRING: Funzione di sostituzione di sottostringhe;
- CONCAT_FIELDS: Concatena il valore di due diverse colonne e lo inserisce in una nuova colonna;
- STRING_OPERATIONS: Classiche operazioni sulle stringhe (trim, uppercase, rimozione caratteri);
- TABLE_OUTPUT: Salvataggio dei dati in una tabella di database;
- UPDATE: Aggiornamento di un record all'interno di una tabella di database.

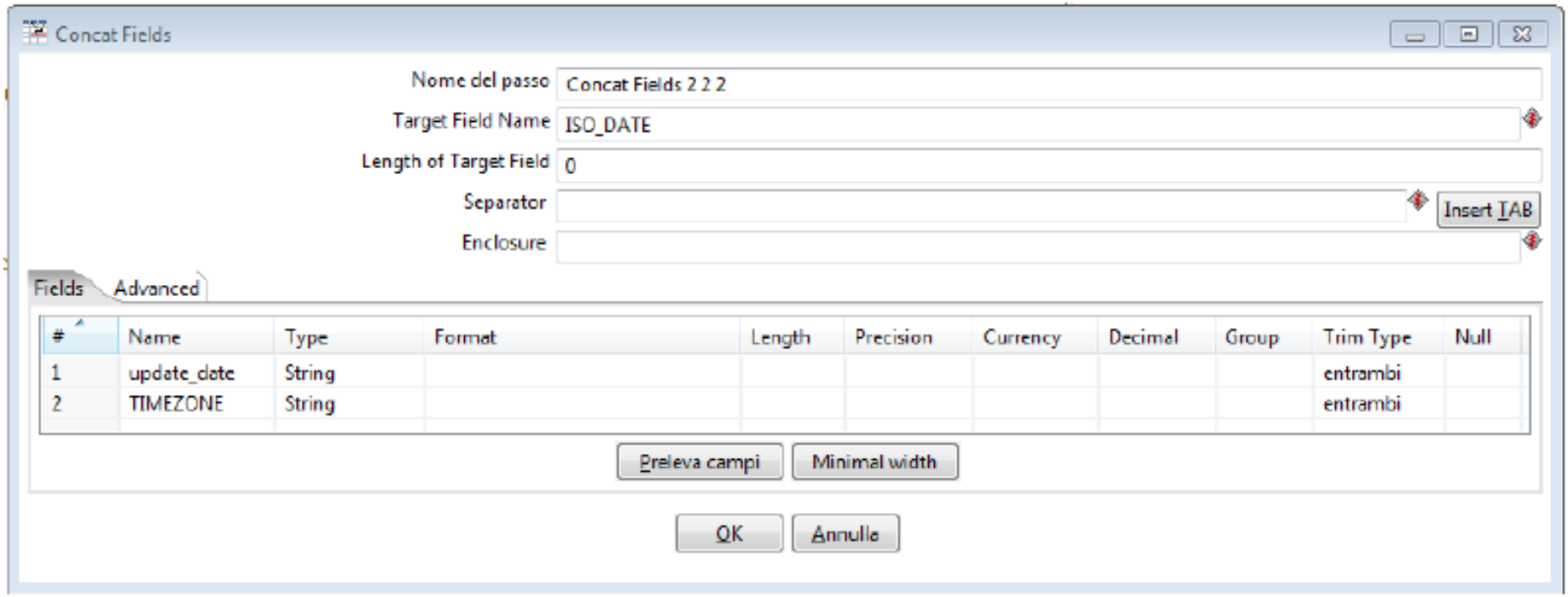
Kettle - SiiMobility Example (6/8)



Value Mapper:

- scegliere su quale campo andare ad eseguire la modifica
- definire la mappatura di tutti i possibili valori di input in altrettanti valori di output.

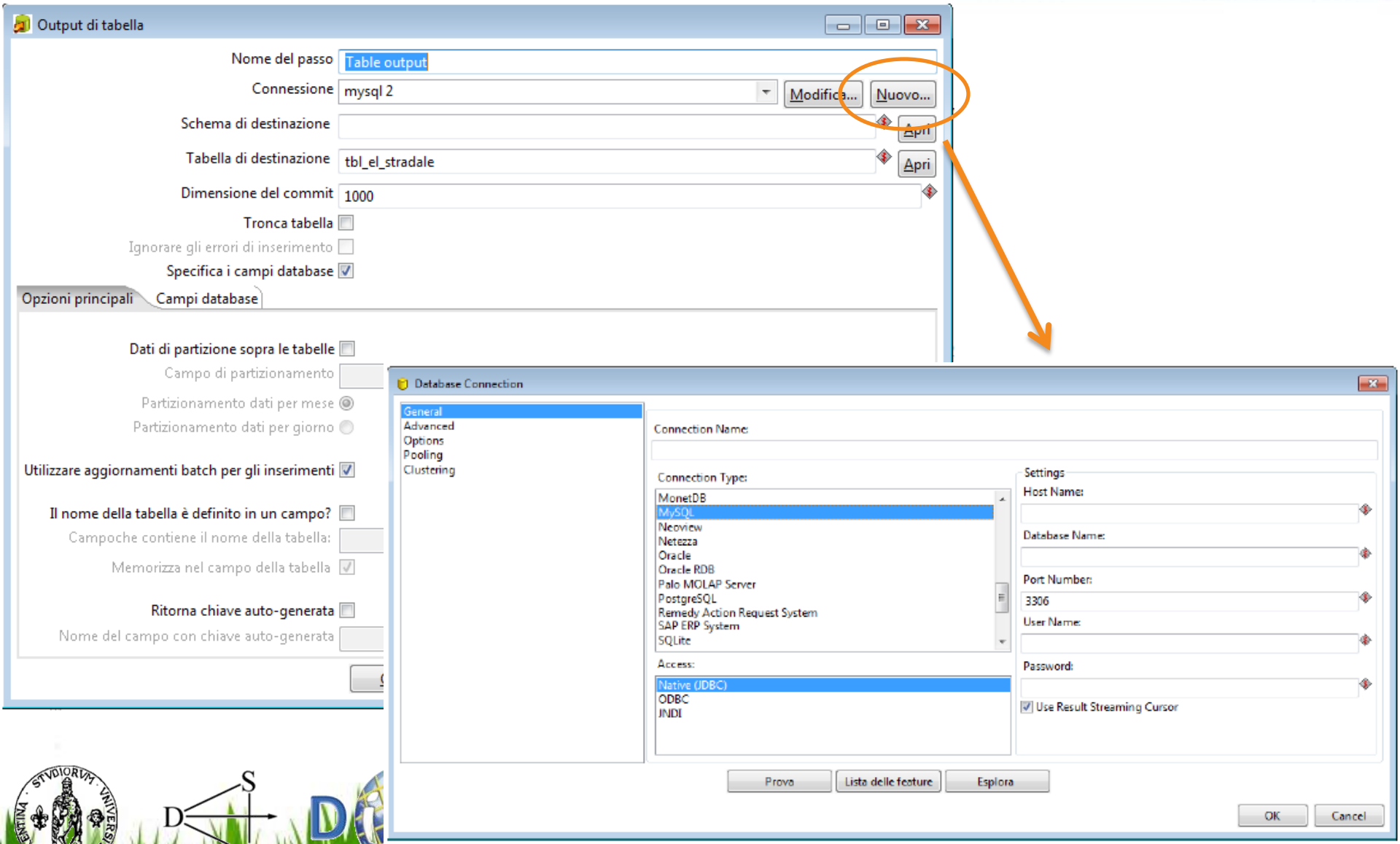
Kettle - SiiMobility Example (7/8)



Concat Field:

- Concatenazione dei campi **update_date** e **TIMEZONE** per formare il nuovo campo **ISO_DATE.**

Kettle - SiiMobility Example (8/8)



KARMA (1/2)

- Tool for **mapping** structured sources **to RDF according** to an **ontology;**
- Provides a **visual interface** to display the KARMA-proposed mapping (users can adjust them)
- Users can **work with** example **data** (not only schemas or ontologies);

http://www.isi.edu/integration/karma

KARMA (2/2)

- Assigment of **semantic type** to data **columns**, specification of **relationship beetween** semantic **types** (an OWL class, a DataProperty, etc).
- Use a **Conditional Random Field** (CRF) model to **lear** the **assigment** semantic type/column;
- Thank to CRF, Karma can **suggest** a **semantic type** for unassigned data columns.
- Create a graph that defines the space of all possible mapping between the data source and the ontology. Node = class.

KARMA – SiiMobility Example (1/5)

- Loading at least one **ontology** and a **dataset**;
- After that it is possible to start **mapping**;
- **Command History** displays the sequence of the last steps performed.

KARMA – SiiMobility Example (2/5)

Karma v1.110

Import Database Table | Import from Service | ✚ Import File... | Reset

Command History

Import CSV File: fi_firenze_loc.csv

Show Model: fi_firenze_loc.csv

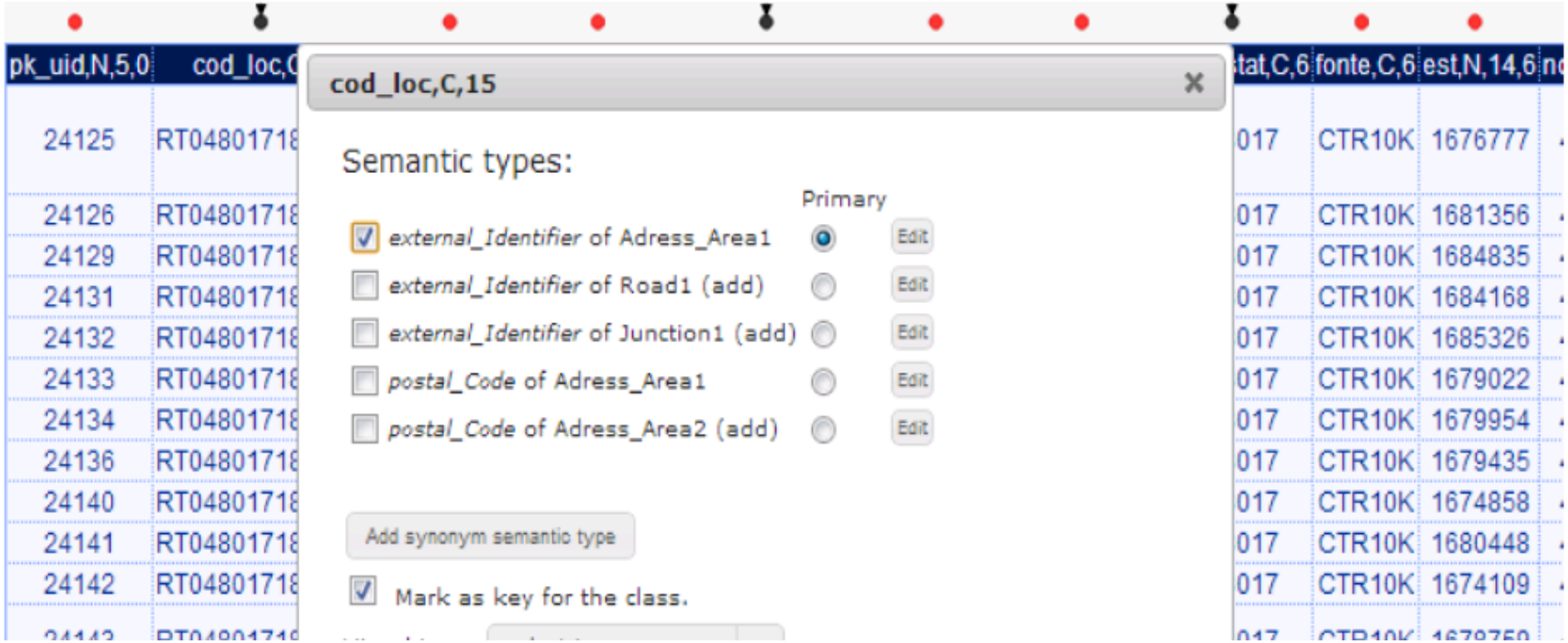
fi_firenze_loc.csv

Adress_Area1

external_Identifier* | official_Name | postal_Code

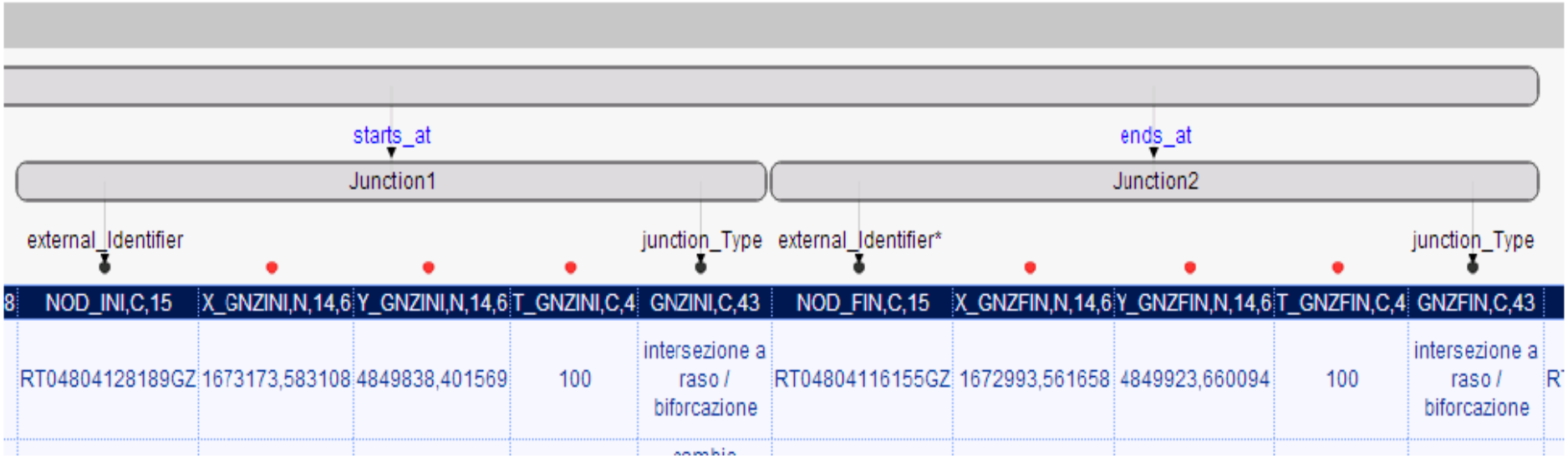
pk_uid,N,5,0	cod_loc,C,15	siglaprov,C,2	comune,C,7	toponimo,C,28	istatprov,C,3	istatcom,C,3	cod_istat,C,6	fonte,C,6	est,N,14,6	nord,N,14,6
24125	RT04801718893TL	FI	FIRENZE	AEROPORTO AMERIGO VESPUCCI	048	017	048017	CTR10K	1676777	4852932
24126	RT04801718894TL	FI	FIRENZE	ARCETRI	048	017	048017	CTR10K	1681356	4846580
24129	RT04801718899TL	FI	FIRENZE	BADIA A RIPOLI	048	017	048017	CTR10K	1684835	4847180
24131	RT04801718902TL	FI	FIRENZE	BARBACANI	048	017	048017	CTR10K	1684168	4845919
24132	RT04801718903TL	FI	FIRENZE	BARBADORO	048	017	048017	CTR10K	1685326	4846704
24133	RT04801718904TL	FI	FIRENZE	BARONTA	048	017	048017	CTR10K	1679022	4845970
24134	RT04801718905TL	FI	FIRENZE	BELLARIA	048	017	048017	CTR10K	1679954	4853957
24136	RT04801718909TL	FI	FIRENZE	BIGOZZI	048	017	048017	CTR10K	1679435	4847868
24140	RT04801718913TL	FI	FIRENZE	BORGO GALLI	048	017	048017	CTR10K	1674858	4849584
24141	RT04801718914TL	FI	FIRENZE	BOTTEGUZZA	048	017	048017	CTR10K	1680448	4844185
24142	RT04801718915TL	FI	FIRENZE	BROZZI	048	017	048017	CTR10K	1674109	4851453
24143	RT04801718917TL	FI	FIRENZE	BUCA DELLA CERTOSA	048	017	048017	CTR10K	1678759	4844246
24144	RT04801718918TL	FI	FIRENZE	C. AL BOSCO	048	017	048017	CTR10K	1678571	4844299
24145	RT04801718919TL	FI	FIRENZE	C. AL PINO	048	017	048017	CTR10K	1679951	4844690

KARMA – SiiMobility Example (3/5)



First step: to establish the **relationship** between the **columns** of the dataset and the **classes/properties** of the ontology

KARMA – SiiMobility Example (4/5)



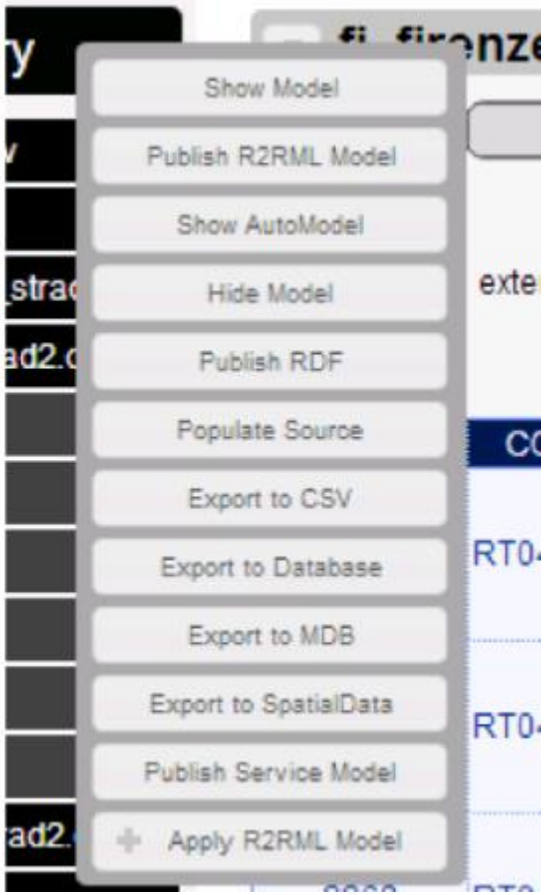
8	NOD_INI,C,15	X_GNZINI,N,14,6	Y_GNZINI,N,14,6	T_GNZINI,C,4	GNZINI,C,43	NOD_FIN,C,15	X_GNZFIN,N,14,6	Y_GNZFIN,N,14,6	T_GNZFIN,C,4	GNZFIN,C,43	
	RT04804128189GZ	1673173,583108	4849838,401569	100	intersezione a raso / biforcazione	RT04804116155GZ	1672993,561658	4849923,660094	100	intersezione a raso / biforcazione	R

- Is possible to define multiple instances of the same semantic class and bind each column to the correct instance;
- Through the check-box 'Mark as key for the class', specific URIs can be defined.

KARMA – SiiMobility Example (5/5)

After the data mapping, the resulting model can be exported in various formats:

- RDF Turtle
- R2RML model (usable to automatically map a relational data source in RDF)
- Notation 3 (N3) model
- CSV, Access, KML



Second step useful tools (Link to external source)

- **Dbpedia** uses the URI **http://dbpedia.org/resource/Berlin** to identify Berlin;
- **Geonames** uses the URI **http://sws.geonames.org/2950159** to identify Berlin;
- **URI aliases**: both URIs refer to the same non-information resource (common on the Web of Data);
- **Social function**: URI aliases allow different views and opinions to be expressed.
- **owl:sameAs** to link to URI aliases.

How to discovered URI Aliases

- **Manually**: **identify** particular **datasets** as suitable linking targets, manually **search** in these for the **URI** references you want to link to;
- **Automatically** : use some tools (**Silk Link Discovery Framework**, Mannheim University) for discovering **relationships** between data items within **different Linked Data** sources.
- Finally, set the built-in OWL property **owl:sameAs** for pairs of URIs identified.
- **PROV** Is a Suitable Technology for Curating the Links. In addition to supporting the user interface for human verfication of links.

Third step useful tools (Curate the Linked Data)

- Linked Data will remain usable twenty years from now **only if** URIs persist and remain resolvable to documentation of their meaning;
- **Changes** or additions in **interlinked** datasets can **invalidate** existing **links** or imply the need to generate new ones;

Problems:

- Most vocabularies reside on a **single Web server**, representing a **single** point of **failure**.
- Vocabularies used today are developed and **curated** by **private maintainers** (individuals or institutions).

SiiMobility Project