



Ricerca di Sistema elettrico

Smart City Platform Specification Information Level 1.0

A. Brutti, A. Frascella, N. Gessa, C. Novelli

SMART CITY PLATFORM SPECIFICATION INFORMATION LEVEL 1.0

A. Brutti, A. Frascella, N. Gessa, C. Novelli (ENEA)

Settembre 2018

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA
Piano Annuale di Realizzazione 2017

Area: Efficienza energetica e risparmio di energia negli usi finali elettrici e interazione con altri vettori energetici

Progetto: Sviluppo di un modello integrato di smart district urbano

Obiettivo: Piattaforma ICT per la gestione della Smart District

Responsabile del Progetto: Claudia Meloni, ENEA

Indice

INTRODUZIONE.....	4
INFORMATION LEVEL.....	5
1 IL MODELLO DATI ASTRATTO.....	6
1.1 MODELLO DATI ASTRATTO - SEZIONE "SPECIFICATION"	7
1.2 MODELLO DATI ASTRATTO - SEZIONE "CONTEXT"	8
1.3 MODELLO DATI ASTRATTO - SEZIONE "VALUES "	9
1.4 MODELLO DATI ASTRATTO E ONTOLOGIA	10
2 SINTASSI DI RIFERIMENTO E VALIDAZIONE.....	12
3 DESCRIZIONE ED ESEMPIO DI UTILIZZO DI ALCUNI ELEMENTI DEL MODELLO DATI ASTRATTO.....	15
3.1 SPECIFICATION REFERENCES	15
3.2 PROPERTY DEFINITION	16
3.3 LINE.....	20
4 CREARE LE ISTANZE DEGLI URBANDATASET	24
5 ESEMPIO: URBANDATASET WHATEVER.....	28
6 RISORSE DISPONIBILI	33
7 RIFERIMENTI	35

Introduzione

Questa specifica tecnica è parte delle **Smart City Platform Specification (SCPS)**
<http://smartcityplatform.enea.it/specification/>

Questo documento è la specifica **Smart City Platform Specification (SCPS) Information Level**
<http://smartcityplatform.enea.it/specification/information/>

Information Level

Le Specifiche dell'Information Level definiscono il formato che i sistemi che aderiscono all'Architettura di Riferimento per la Smart City Platform devono adottare per scambiare UrbanDataset, ovvero informazioni significative su aspetti rilevanti della Smart City.

Il formato è definito da:

- un **Modello Dati Astratto** che, in modo indipendente dalla sintassi, rappresenta e definisce il contenuto che deve avere un documento utilizzato per scambiare UrbanDataset
- l'**implementazione sintattica** del Modello. Le implementazioni sintattiche possono essere molteplici. La specifica fornisce due sintassi di riferimento: XML (eXtensible Markup Language) e JSON (JavaScript Object Notation).

Il Modello Dati Astratto determina la semantica del formato (concetto diverso dalla Semantica degli UrbanDataset che è gestita dalle "Specifiche del Semantic Level"). Le implementazioni sintattiche definiscono le sintassi supportate dalla Smart City Platform.

Insieme, Modello Dati Astratto e implementazioni sintattiche di riferimento, garantiscono l'interoperabilità con la piattaforma a *Livello Information*.

Questa sottospecifica è organizzata nelle seguenti parti:

- descrizione del Modello Dati Astratto definito per rappresentare i documenti;
- artefatti per l'implementazione del Modello Dati Astratto nelle due sintassi (XML e JSON);
- procedura per la creazione delle istanze degli UrbanDataset;
- descrizione ed esempio di utilizzo di alcuni elementi del Modello Dati Astratto;
- un esempio di un UrbanDataset "Whatever";
- risorse a supporto dell'implementazione delle Specifiche dell'Information Level.

1 Il Modello Dati Astratto

Il Modello Dati Astratto ha l'obiettivo di definire, in modo indipendente dalla sintassi, il contenuto e il significato dei documenti utilizzati per scambiare set di dati rilevati sulle proprietà che compongono un UrbanDataset.

Il suo obiettivo è facilitare la mappatura tra le implementazioni sintattiche utilizzate dalle applicazioni esistenti e quelle supportate dalla Smart City Platform.

È stato progettato sulla base di una serie di requisiti e caratteristiche individuati analizzando sia le esigenze che questa sottospecifica deve soddisfare, sia la struttura degli UrbanDataset.

Il Modello Dati Astratto è descritto per mezzo di tabelle e si compone delle seguenti sezioni:

- **Specification:** contiene le informazioni che descrivono l'UrbanDataset utilizzato (per esempio il riferimento alla specifica a cui aderisce e le proprietà che lo compongono)
- **Context:** fornisce le informazioni che contestualizzano i valori trasmessi (per esempio il fuso orario dei timestamp)
- **Values:** contiene i dati rilevati sulle proprietà che compongono l'UrbanDataset, raggruppati in righe.

Come leggere le tabelle che rappresentano il Modello Dati Astratto

Le tabelle che rappresentano il Modello Dati Astratto sono composte dalle seguenti colonne:

- **Elemento:** etichetta scelta per un'informazione contenuta nel modello.
- **Descrizione:** descrizione testuale dell'informazione.
- **Occorrenze:** numero di ripetizioni ammesse (cardinalità) per l'informazione. È espresso come coppia di valori: il primo valore indica il numero minimo di ripetizioni (se valore minimo >0 l'informazione è obbligatoria), il secondo valore indica il numero massimo di ripetizioni ammesse.
- **Tipo:** tipo di dato ammesso per l'informazione (p.es. intero, stringa,...).
- **Esempio:** valore di esempio per l'informazione.

Il colore delle righe delle tabelle indica il livello di aggregazione delle informazioni:

- **riga con sfondo bianco:** indica una informazione elementare (es.: "data", "temperatura", ...) o, se il nome dell'elemento è preceduto dal simbolo @, una informazione che qualifica un'altra informazione elementare. Per queste due tipologie di informazione sono ammessi solo tipi primitivi (stringa, intero, booleano,...);
- **riga con sfondo grigio:** indica una informazione aggregata, composta da più informazioni elementari (p.es.: "periodo", "source", ...). Le informazioni aggregate sono sempre di tipo aggregato.

Si consideri, per esempio, il seguente frammento estratto da una riga del blocco *Values* del modello:

Elemento	Descrizione	Occorr.	Tipo
coordinates	Coordinate WGS84 del centro geometrico della rete applicativa su cui sono stati rilevati i valori trasmessi.	0..1	aggregato
@format	Formato WGS84 in cui sono espresse le coordinate. Opzioni possibili: – Gradi, Minuti, Secondi (WGS84-DMS) – Gradi Decimali (WGS84-DD)	0..1	string codelist

longitude	Longitudine.	1..1	double
latitude	Latitudine.	1..1	double
height	Altitudine.	0..1	double

la sua interpretazione è la seguente:

Il blocco "Values" può contenere un elemento "coordinates", di tipo aggregato, che contiene i dati che identificano il centro geometrico della rete applicativa su cui sono stati rilevati i valori trasmessi; questo è opzionale ("coordinates" ha cardinalità minima=0) e non può essercene più di uno (massima cardinalità=1) . I dati che occorre indicare sono:

- *opzionalmente, il formato WGS84 in cui sono espresse le coordinate; se questo elemento è presente, il suo valore deve essere una stringa e appartenere ad una lista di codici predefinita*
- *longitudine e latitudine (obbligatori poiché "longitude" e "latitude" hanno cardinalità minima=1) di tipo double*
- *opzionalmente, l'altitudine di tipo double.*

Il valore di alcune delle informazioni presenti nel modello dati astratto sarà vincolato dall'ontologia (vedere Specifiche Semantic Level).

L'uso degli identificatori, invece, è descritto nella specifiche "SCPS_Core_e_Terminology"¹.

1.1 Modello Dati Astratto - sezione "Specification"

La sezione **Specification** è composta dalle informazioni indicate nella seguente tabella:

Elemento	Descrizione	Occorr.	Tipo	Esempio
specificationReferences	Insieme di riferimenti che consentono l'identificazione della specifica dell'UrbanDataset utilizzato.	1..1	aggregato	
<i>@version</i>	Versione della specifica.	0..1	string	01
id	Codice identificativo della specifica.	1..1	string	UD0000000000
<i>@schemeID</i>	Identificativo dello schema di identificazione secondo il quale è stato definito l'identificatore.	0..1	string	SCPS
name	Nome associato all' UrbanDataset.	1..1	string	Smart Building Average Consumption
uri	URI che identifica la specifica (può essere un urn indicate il dominio della specifica, o un url).	1..1	string	http://ontologia.esempio.it/UD0000000000
propertyDefinition	Questo elemento consente di inserire, all'interno del messaggio stesso, la descrizione di una delle proprietà che compongono l' UrbanDataset. <i>Per la struttura del blocco, si veda i paragrafo propertyDefinition. L'elemento deve essere ripetuto per ciascuna proprietà che compone l'UrbanDataset.</i>	0..n	aggregato	

L'elemento aggregato **propertyDefinition**

¹ http://smartcityplatform.enea.it/specification/WP211-002-v7x6-SCPS_Core_e_Terminology.docx

Il Modello Dati prevede la possibilità di definire sia **proprietà elementari** (p.es.: *magnitudo di un evento sismico, latitudine, longitudine, consumo elettrico di un edificio, consumo termico di un edificio*), sia **proprietà aggregate**, ovvero proprietà composte da più proprietà elementari (p.es.: *epicentro di un evento sismico, che è composto da latitudine e longitudine, o consumo energetico totale di un edificio, che è composto da consumo elettrico e consumo termico*).

La definizione di una **proprietà elementare** deve essere data dalle seguenti informazioni:

propertyDefinition	Struttura dell'elemento in caso di proprietà elementare .			
propertyName	Nome della proprietà.	1..1	string	<i>ElectricConsumption</i>
propertyDescription	Descrizione della proprietà.	0..1	string	<i>Consumo energia elettrica, calcolato su uno specifico periodo.</i>
dataType	Tipo di dato con cui viene espresso il valore della proprietà.	1..1	string	<i>double</i>
codeList	Riferimento ad una lista di codici contenente il set di valori ammessi per questa proprietà e il relativo significato.	0..1	string	
unitOfMeasure	Unità di misura in cui è espresso il valore.	0..1	string	<i>kWh</i>

La definizione di una **proprietà aggregata** deve fornire le seguenti informazioni:

propertyDefinition	Struttura dell'elemento in caso di proprietà aggregata .	0..n	aggregato	
propertyName	Nome della proprietà.	1..1	string	<i>Consumption</i>
propertyDescription	Descrizione della proprietà.	0..1	string	<i>Consumo energetico (elettrico e termico) elettrico, calcolato su uno specifico periodo.</i>
subPropertyRef	Riferimento (nome) di una proprietà che compone la proprietà indicata nell'elemento <i>propertyName</i> . <i>Anche la proprietà riferita deve essere definita, seguendo la struttura propertyDefinition.</i>	2..n	string	<i>ElectricConsumption</i>
				<i>ThermalConsumption</i>

1.2 Modello Dati Astratto - sezione "Context"

La sezione **Context** è composta dalle informazioni indicate nella seguente tabella:

Elemento	Descrizione	Occorr.	Tipo	Esempio
producer	Dati del sistema (Piattaforma/Soluzione) che ha prodotto i dati rilevati.	1..1	aggregato	
id	Identificatore univoco del sistema.	1..1	string	<i>Smart Home Test</i>
@schemeID	Identificativo dello schema di identificazione secondo il quale è stato definito l'identificatore.	0..1	string	<i>SCPS</i>

timeZone	Fuso orario di qualsiasi marca temporale presente nel documento.	1..1	string	<i>UTC+1</i>
timestamp	Istante di generazione di questo UrbanDataset (ovvero il momento in cui i dati sono stati aggregati).	1..1	dateTime	<i>2017-01-30T10:52:50</i>
coordinates	Coordinate WGS84 del centro geometrico della rete applicativa su cui sono stati rilevati i valori trasmessi.	1..1	aggregato	
<i>@format</i>	Formato WGS84 in cui sono espresse le coordinate. Opzioni possibili: - Gradi, Minuti, Secondi (WGS84-DMS) - Gradi Decimali (WGS84-DD)	0..1	string	<i>WGS84-DD</i>
longitude	Longitudine.	1..1	double	<i>49.5215</i>
latitude	Latitudine.	1..1	double	<i>43</i>
height	Altitudine.	0..1	double	<i>3.0987</i>
language	Lingua in cui sono espressi i campi testo dell'istanza.	0..1	string	<i>IT</i>

1.3 Modello Dati Astratto - sezione "Values "

La sezione **Values** è composta di **una o più righe (1 riga = 1 gruppo di proprietà)**, ognuna contenente le informazioni indicate nella seguente tabella:

Elemento	Descrizione	Occorr.	Tipo	Esempio
id	Numero riga.	0..1	integer	<i>1</i>
description	Descrizione testuale dei valori rilevati.	0..1	string	<i>Consumo elettrico giornaliero dell'unità abitativa.</i>
timestamp	Data e ora (marca temporale) di generazione dei dati riportati in questa riga. <i>Se coincide con il timestamp di generazione dell'Urban Dataset, può essere omesso.</i>	0..1	dateTime	<i>2017-01-30T10:52:50</i>
coordinates	Coordinate WGS84 dell'oggetto a cui sono riferiti i dati rilevati indicati in questa riga. <i>Se questo elemento è presente, specifica ulteriormente la posizione del dato rispetto alle coordinates della sezione Context.</i>	0..1	aggregato	
<i>@format</i>	Formato WGS84 in cui sono espresse le coordinate. Opzioni possibili: - Gradi, Minuti, Secondi (WGS84-DMS) - Gradi Decimali (WGS84-DD)	0..1	string	<i>WGS84-DD</i>
longitude	Longitudine.	1..1	double	<i>49.52</i>

	latitude	Latitudine.	1..1	double	43
	height	Altitudine.	0..1	double	3.09
	period	Periodo durante il quale sono stati rilevati i dati. <i>Nel caso di grandezza istantanea, start_timestamp e end_timestamp coincidono.</i>	0..1	aggregato	
	start_timestamp	Marca temporale indicante l'inizio del periodo.	1..1	dateTime	2014-01-01T10:52:50
	end_timestamp	Marca temporale indicante la fine del periodo.	1..1	dateTime	2014-02-28T10:52:50
	property	Proprietà che compone l'UrbanDataset. <i>Per la struttura dell'elemento, si veda la sezione property. L'elemento deve essere ripetuto per ciascuna proprietà che compone l'UrbanDataset.</i>	1..n	aggregato	

L'elemento aggregato **property**

Il modello dati stabilisce come definire sia i valori delle **proprietà elementari**, sia i valori delle **proprietà aggregate**.

Una **proprietà elementare** deve essere espressa conformemente alla seguente struttura:

property	Struttura del blocco in caso di proprietà elementare .				
	name	Nome identificativo della grandezza rilevata.	1..1	string	<i>ElectricConsumption</i>
	value	Valore rilevato.	1..1	string	123

Una **proprietà aggregata** deve essere espressa conformemente alla seguente struttura:

property	Struttura del blocco in caso di proprietà aggregata .				
	name	Nome identificativo della proprietà rilevata.	1..1	string	<i>Consumption</i>
	property	Proprietà elementare che compone la proprietà aggregata.	2..n		

1.4 Modello Dati Astratto e Ontologia

Il Modello Dati Astratto è stato definito in modo tale da essere generico e, per questo motivo, non impone vincoli né fornisce indicazioni sulle proprietà da specificare in fase di creazione delle istanze degli UrbanDataset.

Questo livello di interoperabilità riguarda la **semantica degli UrbanDataset** ed è gestito attraverso l'utilizzo di un'ontologia (si veda la "Specificazione Semantica").

A livello semantico sono gestite anche le liste di codici (*Code List*) ammesse per gli elementi per i quali il Modello Dati Astratto le richiede, ovvero:

- TimezoneCode --> elemento *timeZone*
- LanguageCode --> elemento *language*

- FormatCode --> elemento *@format*

A livello semantico sono inoltre gestiti:

- i tipi di dato ammessi per i valori delle proprietà (elemento *dataType*)
- i valori da utilizzare per le unità di misura (elemento *unitOfMeasure*), attraverso l'importazione dell'ontologia "www.ontology-of-units-of-measure.org/resource/om-2/" sono gestiti.

Nella SCP Ontology sono infine gestite le liste di codici associate a specifiche proprietà, ovvero:

- BuildingEnergyAnomalyCode --> proprietà *BuildingEnergyAnomalyCode*
- DeviceAnomalyCode --> proprietà *DeviceAnomalyCode*
- TreatmentPlantAnomalyCode --> proprietà *TreatmentPlantAnomalyCode*

2 Sintassi di riferimento e validazione

La sottospecifica relativa al livello Information fornisce la definizione formale dell'implementazione del Modello Dati Astratto nelle due sintassi supportate dalla Smart City Platform: XML (1) e JSON (2).

Il formalismo utilizzato per la sintassi XML è **XML Schema 1.0** (3); XML Schema (XSD) è un linguaggio di descrizione del contenuto di un file XML; definisce:

- gli elementi e gli attributi che possono apparire in un documento XML
- la struttura del documento (quindi la sequenza e il numero di occorrenze di elementi e attributi)
- i tipi di dato di elementi e attributi
- eventuali valori di default e limiti sui valori ammessi per elementi e attributi.

In Figura 1 è riportato uno screenshot che mostra un sottoinsieme dell'XML Schema che implementa il Modello Dati Astratto.

Il formalismo utilizzato per la sintassi JSON è **JSON Schema**² (4); JSON Schema è un linguaggio di descrizione del contenuto di un file JSON; definisce:

- gli elementi che possono apparire in un documento JSON
- la struttura del documento (quindi la sequenza e il numero di occorrenze di elementi)
- i tipi di dato di elementi
- eventuali limiti sui valori ammessi per gli elementi.

In Figura 2 è riportato uno screenshot che mostra un sottoinsieme del JSON Schema che implementa il Modello Dati Astratto.

Gli Schema, XML e JSON, costituiscono il riferimento per la costruzione dei documenti utilizzati per scambiare UrbanDataset; consentono, inoltre, di **validare la sintassi e la struttura** dei documenti scambiati.

A tal fine possono essere utilizzati sia per realizzare propri strumenti di validazione, sia per alimentare tool di validazione realizzati da terze parti (per esempio validatori disponibili su web o editor XML).

Gli Schema sono disponibili al link: <http://smartcityplatform.enea.it/specification/information/1.0/>

Invece, per la **validazione della semantica degli UrbanDataset** (quindi, per esempio, per verificare che le proprietà specificate siano effettivamente quelle che compongono l'UrbanDataset indicato, *si veda paragrafo 1.4*) sono stati realizzati degli artefatti di validazione Schematron (5), specifici per tipo di UrbanDataset. I file Schematron sono generati automaticamente utilizzando le informazioni contenute nell'ontologia; possono essere applicati direttamente a UrbanDataset in formato XML, ma anche per validare UrbanDataset in formato JSON previa trasformazione dei file JSON in formato XML.

² JSON Schema è una specifica draft; l'Urban Dataset JSON Schema definito in questa sottospecifica basa sulla versione *draft 07*.

4

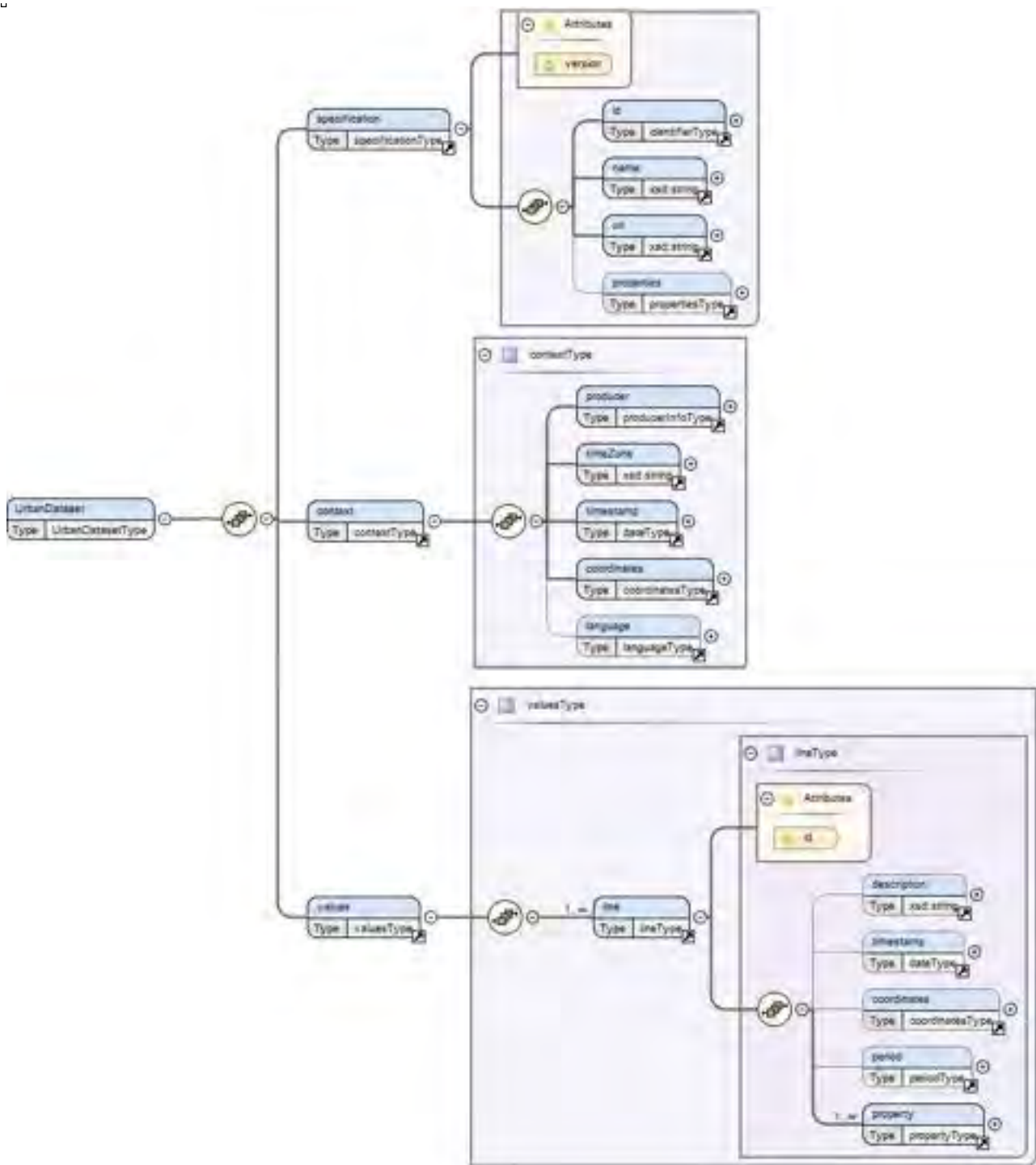


Figura 1 – Screenshot UrbanDataset XML Schema

```

    ▼ UrbanDataset {5}
      type : object
      ▼ properties {4}
        ▼ specification {4}
          type : object
          ▼ properties {5}
            ▶ version {1}
            ▶ id {3}
            ▶ name {1}
            ▶ uri {1}
            ▶ properties {4}
          ▶ required {3}
          ▶ additionalProperties {1}
        ▼ context {4}
          type : object
          ▼ properties {5}
            ▶ producer {4}
            ▶ timeZone {1}
            ▶ timestamp {1}
            ▶ coordinates {4}
            ▶ language {1}
          ▶ required {4}
          ▶ additionalProperties {3}
        ▼ values {4}
          type : object
          ▼ properties {1}
            ▼ line {3}
              type : array
              minItems : 1
              ▼ items {3}
                type : object
                ▼ properties {6}
                  ▶ id {1}
                  ▶ description {1}
                  ▶ timestamp {1}
                  ▶ coordinates {4}
                  ▶ period {4}
                  ▶ property {1}
                ▶ required {1}
              ▶ required {1}
              ▶ additionalProperties {1}
            ▶ additionalProperties {1}
          ▶ required {3}
  
```

Figura 2 – Screenshot UrbanDataset JSON Schema

3 Descrizione ed esempio di utilizzo di alcuni elementi del Modello Dati Astratto

3.1 *specificationReferences*

L'elemento *specificationReferences* è composto da un insieme di elementi che consentono di indicare la specifica a cui aderisce l'UrbanDataset.

L'elemento *specificationReferences* e gli elementi che lo compongono sono stati resi obbligatori poiché, al fine di favorire l'interoperabilità, si richiede che l'UrbanDataset abbia sempre una specifica di riferimento; a tale scopo si raccomanda l'uso dell'Ontologia ma, poiché l'Architettura di Riferimento può essere adottata a livelli, sono ammesse anche altre modalità (p.es. un documento di testo in formato pdf), che però rendono l'implementazione non conforme all'intera SCPS.

È invece opzionale l'indicazione della versione della specifica adottata (attributo *version*); se non indicata, si assume che sia la più recente versione disponibile.

È opzionale anche l'indicazione dello schema di identificazione (attributo *schemeID*) a cui appartiene il codice identificativo della specifica. Se la specifica dell'UrbanDataset è definita nell'Ontologia, se ne raccomanda l'indicazione e il valore da utilizzare è "SCPS" (che identifica le specifiche registrate nella Smart City Platform).

I casi di valorizzazione del blocco *specificationReferences* sono sostanzialmente due:

- caso 1: UrbanDataset registrato nell'Ontologia, in cui rientra anche il caso particolare dell'UrbanDataset Whatever
- caso 2: UrbanDataset **non** registrato nell'Ontologia.

Di seguito alcuni esempi di valorizzazione del blocco *specificationReferences* nelle sintassi XML e JSON.

Il primo esempio, Figura 3 e Figura 4, mostra il caso di un UrbanDataset registrato nell'Ontologia; tutti gli elementi, anche quelli opzionali, sono valorizzati.

<pre><specification version="1.0"> <id schemeID="SCPS">BuildingEnergyAnomalies-1.0</id> <name> Building Energy Anomalies</name> <uri> http://smartcityplatform.enea.it/...</uri> </specification></pre>	<pre>"specification": { "version": "1.0", "id": { "value": "BuildingEnergyAnomalies-1.0 ", "schemeID": "SCSP" }, "name": "Building Energy Anomalies ", "uri": " http://smartcityplatform.enea.it/..." }</pre>
---	---

Figura 3 – Esempio di implementazione XML del blocco *specificationReferences*

Figura 4 – Esempio di implementazione JSON del blocco *specificationReferences*

Il secondo esempio, Figura 5 e Figura 6, mostra il caso di un UrbanDataset **non** registrato nell'Ontologia la cui specifica potrebbe essere un file pdf contenente un testo come quello illustrato Figura 7; gli elementi opzionali non sono stati valorizzati.

```
<specification >
  <id>MyUD010</id>
  <name>Parking monitoring</name>
  <uri>http://myDomain.it/ MyUD010.pdf</uri>
</specification>
```

Figura 5 – Esempio di implementazione XML del blocco `specificationReferences`

```
"specification": {
  "id": {
    "value": "MyUD010"
  },
  "name": "Parking monitoring ",
  "uri": "http://myDomain.it/ MyUD010.pdf "
}
```

Figura 6 – Esempio di implementazione JSON del blocco `specificationReferences`

Specifica dell'Urban Dataset "Parking monitoring"

L'Urban Dataset "Parking monitoring" ha l'obiettivo di monitorare le auto presenti in un parcheggio.

Identificatore: MyUD010

Nome: Parking monitoring

Riferimento di questa specifica: [http://myDomain.it/ MyUD010.pdf](http://myDomain.it/MyUD010.pdf)

Proprietà che compongono l'Urban Dataset

- ParkID: identificatore parcheggio, tipo stringa
- CarN: numero auto presenti nel parcheggio in un certo istante, tipo intero

Figura 7 – Esempio specifica di un `UrbanDataset` non registrato

3.2 *propertyDefinition*

L'elemento *propertyDefinition* consente di dichiarare e descrivere nell'istanza dell'`UrbanDataset` le proprietà che lo compongono, rendendo così il messaggio autodescrittivo.

Tra le proprietà dichiarate possono esserci anche le proprietà elementari *DataDescription* e *timestamp*, e le proprietà aggregate *coordinates* e *period*, con le loro rispettive sottoproprietà. Queste proprietà sono esplicitamente dichiarate nel Modello Dati Astratto perché comuni a tutti gli `UrbanDataset`, ma sono opzionali. Attraverso l'Ontologia e coerentemente inserendole nelle definizioni del blocco *propertyDefinition* se ne rende obbligatorio l'uso per gli `UrbanDataset` che le richiedono (un esempio di loro dichiarazione è dato dall'esempio in Figura 10 e Figura 11).

L'elemento *propertyDefinition* è opzionale: se ne raccomanda l'utilizzo nei casi di `UrbanDataset` non formalizzati nell'ontologia, o di `UrbanDataset` "Whatever"; negli altri casi il suo utilizzo è a discrezione degli implementatori, ricordando che le proprietà dichiarate attraverso le diverse occorrenze dell'elemento *propertyDefinition* devono comunque essere coerenti con la specifica dell'`UrbanDataset`.

In

```

<specification >
  <id>MyUD010</id>
  ...
  <properties>
    <propertyDefinition>
      <propertyName>ParkID </propertyName>
      <propertyDescription>Identificatore parcheggio.</propertyDescription>
      <dataType>string</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>CarN</propertyName>
      <propertyDescription>Numero auto presenti nel parcheggio in un certo istante.</propertyDescription>
      <dataType>integer</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
  </properties>
</specification>

```

Figura 8 e Figura 9, per esempio, le implementazioni XML e JSON del blocco *propertyDefinition* di un UrbanDataset **non** presente nell'Ontologia; è stato ripreso l'esempio illustrato nella precedente Figura 7.

```

<specification >
  <id>MyUD010</id>
  ...
  <properties>
    <propertyDefinition>
      <propertyName>ParkID </propertyName>
      <propertyDescription>Identificatore parcheggio.</propertyDescription>
      <dataType>string</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>CarN</propertyName>
      <propertyDescription>Numero auto presenti nel parcheggio in un certo istante.</propertyDescription>
      <dataType>integer</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
  </properties>
</specification>

```

Figura 8 – Esempio di implementazione XML del blocco *propertyDefinition*

```

"specification": {
  "id": {
    "value": "MyUD010"
  },
  ...,
  "properties": {"propertyDefinition": [
    {
      "propertyName": "ParkID ",
      "propertyDescription": "Identificatore parcheggio.",
      "dataType": "string",
      "unitOfMeasure": "adimensionale"
    },
    {
      "propertyName": "CarN ",
      "propertyDescription": "Numero auto presenti nel parcheggio in un certo istante.",
      "dataType": "integer",
      "unitOfMeasure": "adimensionale"
    }
  ]}
}

```

Figura 9 – Esempio di implementazione JSON del blocco *propertyDefinition*

In Figura 10 e Figura 11, per esempio, un frammento delle implementazioni XML e JSON del blocco *propertyDefinition* per l'UrbanDataset "Building Energy Anomalies" definito nell'Ontologia.

Notare che, coerentemente con la specifica ontologica di questo UrbanDataset, nella lista delle proprietà dichiarate sono presenti anche la proprietà semplice *DataDescription* e la proprietà aggregata *period*.

```

<specification version="1.0">
  <id schemeID="SCPS">BuildingEnergyAnomalies-1.0</id>
  ...
  <properties>
    <propertyDefinition>
      <propertyName>BuildingName</propertyName>
      <propertyDescription>Etichetta associata al palazzo</propertyDescription>
      <dataType>string</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>BuildingEnergyAnomalyCode</propertyName>
      <propertyDescription>Codice identificativo dell'anomalia</propertyDescription>
      <dataType>string</dataType>
      <codeList>http://smartcityplatform.enea.it/specification/semantic/1.0/gc/BuildingEnergyAnomalyCode.gc</codeList>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>BuildingEnergyHighAnomalies</propertyName>
      <propertyDescription>Numero delle anomalie individuate con priorit  high</propertyDescription>
      <dataType>integer</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>BuildingEnergyLowAnomalies</propertyName>
      <propertyDescription>Numero delle anomalie individuate con priorit  low</propertyDescription>
      ...
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>BuildingEnergyMeanAnomalies</propertyName>
      <propertyDescription>Numero delle anomalie individuate con priorit  mean</propertyDescription>
      ...
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>BuildingID</propertyName>
      <propertyDescription>Identificatore del palazzo</propertyDescription>
      ...
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>DataDescription</propertyName>
      <propertyDescription>Descrizione testuale dei dati inviati</propertyDescription>
      <dataType>string</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>period</propertyName>
      <propertyDescription>Periodo durante il quale sono stati rilevati i dati riportati nella riga</propertyDescription>
      <subProperties>
        <propertyName>start_ts</propertyName>
        <propertyName>end_ts</propertyName>
      </subProperties>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>start_ts</propertyName>
      <propertyDescription>Marca temporale indicante l'inizio del periodo</propertyDescription>
      <dataType>dateTime</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>end_ts</propertyName>
      ...
  </properties>

```

Figura 10 – Esempio di implementazione XML dell'elemento *propertyDefinition*

```

"specification": {
  "version": "1.0",
  "id": {
    "value": " BuildingEnergyAnomalies-1.0 ",
    "schemeID": "SCPS"
  },
  ...
  "properties": {"propertyDefinition": [
    { "propertyName": "BuildingName",
      "propertyDescription": "Etichetta associata al palazzo",
      "dataType": "string",
      "unitOfMeasure": "adimensionale"
    },
    { "propertyName": "BuildingEnergyAnomalyCode",
      "propertyDescription": "Codice identificativo dell'anomalia",
      "dataType": "string",
      "codeList": "http://smartcityplatform.enea.it/specification/semantic/1.0/gc/BuildingEnergyAnomalyCode.gc",
      "unitOfMeasure": "adimensionale"
    },
    { "propertyName": "BuildingEnergyHighAnomalies",
      "propertyDescription": "Numero delle anomalie individuate con priorit  high",
      ...
    },
    { "propertyName": "BuildingEnergyLowAnomalies",
      "propertyDescription": "Numero delle anomalie individuate con priorit  low",
      ....
    },
    { "propertyName": "BuildingEnergyMeanAnomalies",
      "propertyDescription": "Numero delle anomalie individuate con priorit  mean",
      ...
    },
    { "propertyName": "BuildingID",
      "propertyDescription": "Identificatore del palazzo",
      ...
    },
    { "propertyName": "DataDescription",
      "propertyDescription": "Descrizione testuale dei dati inviati",
      "dataType": "string",
      "unitOfMeasure": "adimensionale"
    },
    { "propertyName": "period",
      "propertyDescription": "Periodo durante il quale sono stati rilevati i dati riportati nella riga",
      "subProperties": {
        "propertyName": [
          "start_ts",
          "end_ts"
        ]
      },
    },
    {
      "propertyName": "start_ts",
      "propertyDescription": "Marca temporale indicante l'inizio del periodo",
      "dataType": "dateTime",
      "unitOfMeasure": "adimensionale"
    },
    {
      "propertyName": "end_ts",
      ....
    }
  ]
}

```

Figura 11 – Esempio di implementazione JSON dell'elemento "specification"

3.3 line

Un'istanza di UrbanDataset deve avere almeno un elemento *line*, ovvero almeno un set di valori rilevati sulle proprietà che compongono l'UrbanDataset.

Ai valori deve essere associato l'istante di generazione; se esso coincide con l'istante di creazione dell'UrbanDataset, può essere omissso a livello di *line*, altrimenti deve essere indicato (elemento *timestamp* in *line*). Se non indicato a livello di *line*, si assume che il timestamp della riga sia uguale al timestamp dell'UrbanDataset (*timestamp* nel blocco *context*).

Opzionalmente, nell'elemento *line* è inoltre possibile indicare:

- il numero della riga (elemento *id*).

Se la definizione semantica dello specifico UrbanDataset lo richiede, nell'elemento *line* è obbligatorio indicare:

- una descrizione testuale del set di valori riportati nella riga (elemento *description*);
- le coordinate WGS84 dell'oggetto a cui sono riferiti i valori indicati nella riga (elemento *coordinates* in *line*); se non indicate a livello di *line*, si assume di poter associare all'oggetto le coordinate WGS84 del centro geometrico della rete applicativa su cui sono stati rilevati i valori trasmessi (*coordinates* nel blocco *context*);
- il periodo durante il quale sono stati rilevati i valori. Il periodo è espresso come coppia di marche temporali, una di inizio (elemento *start_timestamp*) e una di fine periodo (elemento *end_timestamp*) che devono essere indicate entrambe, anche in caso di misurazione istantanea (in questo caso coincidono). Se l'istante di misurazione coincide con l'istante di generazione dei valori (*timestamp* in *line*), il periodo può essere omissso.

È invece sempre obbligatorio indicare almeno una coppia proprietà-valore (elemento *property*) poiché si assume che un UrbanDataset abbia almeno una proprietà (elementare o aggregata).

In Figura 12 e Figura 13, per esempio, un frammento delle implementazioni XML e JSON del blocco *line* per l'UrbanDataset "Building Energy Anomalies" definito nell'Ontologia.

Per un esempio di riga con proprietà aggregata, si veda il *Capitolo 0*.

```

<values>
  <line id="1">
    <description>Anomalie negli ultimi tre anni.</description>
    <period>
      <start_ts>2014-08-02T00:00:00</start_ts>
      <end_ts>2017-08-02T00:00:00</end_ts>
    </period>
    <property name="BuildingName">
      <val>F40</val>
    </property>
    <property name="BuildingEnergyAnomalyCode">
      <val>1</val>
    </property>
    <property name="BuildingEnergyHighAnomalies">
      <val>301</val>
    </property>
    <property name="BuildingEnergyLowAnomalies">
      <val>115</val>
    </property>
    <property name="BuildingEnergyMeanAnomalies">
      <val>280</val>
    </property>
    <property name="BuildingID">
      <val>1</val>
    </property>
  </line>
  <line id="2">
    <description>Anomalie negli ultimi tre anni.</description>
    <timestamp>2017-08-02T11:22:15</timestamp>
    <period>
      <start_ts>2014-08-02T00:00:00</start_ts>
      <end_ts>2017-08-02T00:00:00</end_ts>
    </period>
    <property name="BuildingName">
      <val>F66</val>
    </property>
    <property name="BuildingEnergyAnomalyCode ">
      <val>1</val>
    </property>
    ...
  </line>
</values>

```

Figura 12 – Esempio di implementazione XML dell'elemento "line"

```

{"line": [
  {
    "id": 1,
    "description": "Anomalie negli ultimi tre anni.",
    "timestamp": "2017-08-02T11:22:15",
    "period": {
      "start_ts": "2014-08-02T00:00:00",
      "end_ts": "2017-08-02T00:00:00"
    },
    "property": [
      { "name": "BuildingName",
        "val": "F40" },
      { "name": "BuildingEnergyAnomalyCode",
        "val": "1" },
      { "name": "BuildingEnergyHighAnomalies",
        "val": "301" },
      { "name": "BuildingEnergyLowAnomalies",
        "val": "115" },
      { "name": "BuildingEnergyMeanAnomalies",
        "val": "280" },
      { "name": "BuildingID",
        "val": "1" }
    ]
  },
  {
    "id": 2,
    "description": "Anomalie negli ultimi tre anni.",
    "period": {
      "start_ts": "2014-08-02T00:00:00",
      "end_ts": "2017-08-02T00:00:00"
    },
    "property": [
      { "name": "BuildingName",
        "val": "F66" },
      { "name": "BuildingEnergyAnomalyCode",
        "val": "1" },
      ....
    ]
  }
]}

```

Figura 13 – Esempio di implementazione JSON dell'elemento "line"

4 Creare le istanze degli UrbanDataset

Il percorso che gli implementatori dovrebbero svolgere per realizzare un'applicazione capace di abilitare lo scambio di UrbanDataset tra un Contesto Applicativo e la Smart City Platform dovrebbe prevedere i seguenti step:

1. Identificazione o definizione della specifica dell'UrbanDataset che si vuole implementare

Il primo step consiste nell'analizzare gli UrbanDataset esistenti (nell'ontologia e nei casi studio resi disponibili) al fine di individuare un UrbanDataset capace di supportare lo scambio di informazioni che si vuole attivare. Se nessun UrbanDataset risulta adeguato, decidere se utilizzare la specifica dell'UrbanDataset "Whatever" o attivare la procedura per la richiesta di inserimento di un nuovo UrbanDataset nell'ontologia (si veda la "Specificazione Semantica Level"); in questo secondo caso, la specifica da utilizzare sarà quella rilasciata al termine della procedura (la definizione dell'UrbanDataset "Whatever" può essere anche il primo passo per iniziare a testare l'uso del formato oppure per scambiare i dati nell'attesa che l'UrbanDataset richiesto venga approvato. In generale non si dovrebbe considerare l'uso del "Whatever" come la soluzione definitiva).

2. Analisi del Modello Dati Astratto e della specifica individuata

Il contenuto che devono avere i messaggi utilizzati per scambiare UrbanDataset è definito dal Modello Dati Astratto; quindi, prima di passare all'implementazione vera e propria, occorre analizzarlo e confrontarlo con la specifica dell'UrbanDataset selezionato, al fine di stabilire quali proprietà specificare e quali dei suoi elementi opzionali utilizzare (in generale, l'uso degli elementi opzionali nei blocchi "Specification" e "Context" è a discrezione degli implementatori; l'uso degli elementi opzionali nel blocco "Values" è guidato dalla specifica dell'UrbanDataset).

Come descritto in precedenza, infatti, il Modello Dati Astratto è stato concepito per supportare qualsiasi tipo di UrbanDataset. Per questo motivo alcuni suoi elementi sono stati definiti come opzionali in modo da poter essere utilizzati o meno a seconda dei casi e, per la stessa ragione, non impone vincoli né fornisce indicazioni sulle proprietà da specificare in fase di creazione delle istanze degli UrbanDataset.

Entrambe questi aspetti, però, sono definiti sia negli UrbanDataset disponibili nei casi studio, sia nell'ontologia che, in questo step, devono essere analizzate e confrontate con il Modello.

Si consideri, per esempio, la specifica dell'UrbanDataset "Building Energy Anomalies" (Tabella 1):

Tabella 1 - Definizione UrbanDataset "Consumo elettrico medio dell'edificio"

UrbanDataset	Building Energy Anomalies		
Scopo	Contegge le anomalie (low, mean, high) associate a una particolare causa, a un determinato palazzo, calcolate su una particolare finestra temporale.		
Aggregazione spaziale	Facility		
Categoria / Sottocategoria	Built environment / Smart Building		
Aggregazione temporale	Valori totali		
SPECIFICAZIONE DELL'URBANDATASET			
UrbanDatasetId	BuildingEnergyAnomalies-1.0		
UrbanDatasetUri	http://smartcityplatform.enea.it/specification/semantic/1.0/ontology/scps-ontology-1.0.owl#BuildingEnergyAnomalies		
UrbanDatasetName	Building Energy Anomalies		
CONTESTUALIZZAZIONE DELL'URBANDATASET			
Timestamp	Unità di Misura:	Formato	Descrizione / Esempio
	Adimensionale	Date-Time aaaa-mm-ggThh:mm:ss	Tempo di generazione dell'UrbanDataset
Timezone	Adimensionale	String	Es. UTC, UTC+01, UTC-05
Coordinates	Adimensionale	WGS84 (World Geodetic System)	Coordinate del centro geometrico della rete di

Producer			Smart Building es. "42.04"; "12.30"
	Adimensionale	String	Identificatore del Sistema che ha prodotto i dati
PROPRIETÀ DELL'URBANDATASET			
Description	Unità di Misura:	Formato	Descrizione / Esempio
	Adimensionale	String	Descrizione dei dati inviati
BuildingID	Adimensionale	String	Identificatore del palazzo monitorato e generatore di anomalie (es. 1). Questo diventerà il codice di riferimento per tutti gli UrbanDataset associati a questo edificio
BuildingName	Adimensionale	String	Es. "F40"
StartPeriod	Adimensionale	Date - Time aaaa-mm-ggThh:mm:ss	Data di inizio del conteggio delle anomalie (es. 2017-09-26T00:00:00)
EndPeriod	Adimensionale	Date - Time aaaa-mm-ggThh:mm:ss	Data di fine della conteggio delle anomalie(es. 2017-09-26T23:59:59)
BuildingEnergyAnomalyCode	Adimensionale	String	Codice anomalia (definito da tabella delle anomalie). Es. 1
BuildingEnergyHighAnomalies	Adimensionale	integer	Numero di High Anomalies nell'intervallo di riferimento. Es. 4
BuildingEnergyMeanAnomalies	Adimensionale	integer	Numero di Mean Anomalies nell'intervallo di riferimento Es. 3
BuildingEnergyLowAnomalies	Adimensionale	integer	Numero di Low Anomalies nell'intervallo di riferimento Es. 1

La specifica indica che l'UrbanDataset:

- non ha proprietà aggregate
- non richiede l'indicazione delle coordinate a livello di riga
- non richiede l'indicazione del timestamp a livello di riga
- richiede l'indicazione del periodo a livello di riga
- richiede la descrizione a livello di riga
- è composto dalle proprietà *BuildingID*, *BuildingName*, *BuildingEnergyAnomalyCode*, *BuildingEnergyHighAnomalies*, *BuildingEnergyMeanAnomalies* e *BuildingEnergyLowAnomalies* di cui definisce anche tipo ed eventuale unità di misura.

Per implementare questo UrbanDataset, quindi, nel blocco *Values* è sufficiente supportare il sottoinsieme del Modello Astratto illustrato in Tabella 2.

Tabella 2 – Blocco "Values" dell'UrbanDataset "Building Energy Anomalies"

Elemento	Occorr.	Tipo	Valori ammessi (nel range ammesso dal Tipo indicato)
id	0..1	integer	<i>qualsiasi</i>
description	10..1	string	<i>Qualsiasi descrizione coerente con lo scopo dell'UrbanDataset</i>
timestamp	0..1	dateTime	
coordinates	0..1	aggregato	
@format	0..1	string	
longitude	1..1	double	
latitude	1..1	double	
height	0..1	double	
period	10..1	aggregato	
start_timestamp	1..1	dateTime	<i>qualsiasi</i>
end_timestamp	1..1	dateTime	<i>qualsiasi</i>
property	1..#6	aggregato	
name	1..1	string	<i>BuildingID, BuildingName, BuildingEnergyAnomalyCode, BuildingEnergyHighAnomalies, BuildingEnergyMeanAnomalies BuildingEnergyLowAnomalies</i>
value	1..1	string	<i>qualsiasi</i>

3. Mapping tra Modello Dati e sistema

Questo step ha l'obiettivo di definire come saranno popolati i messaggi creati a partire dal sottoinsieme del Modello Dati Astratto individuato dallo step precedente; a tal fine occorre identificare l'eventuale sorgente dati (per es. un database) che fornirà i dati per generare gli UrbanDataset e mappare ogni elemento del Modello in un elemento/campo della sorgente.

4. Implementazione

L'ultimo passaggio riguarda la scelta della sintassi per i messaggi da scambiare, l'analisi del relativo schema (che è l'unico riferimento normativo per la sintassi e la struttura dei messaggi) e la predisposizione del sistema per la generazione dei messaggi nel formato scelto.

Strumenti/documentazione di supporto per questo step:

- Guida all'implementazione del Modello Dati Astratto (che definisce la corrispondenza tra gli elementi dell'intero Modello gli elementi degli schemi XML e JSON)
- sottoinsieme del Modello Dati individuato allo step 2 e mappatura definita allo step 3 che, insieme, indicano quali elementi dello schema utilizzare e come valorizzarli.

In Figura 14 è illustrato un esempio di mappatura tra sorgente dati e sottoinsieme del Modello Dati e tra quest'ultimo e sintassi scelta per l'implementazione.

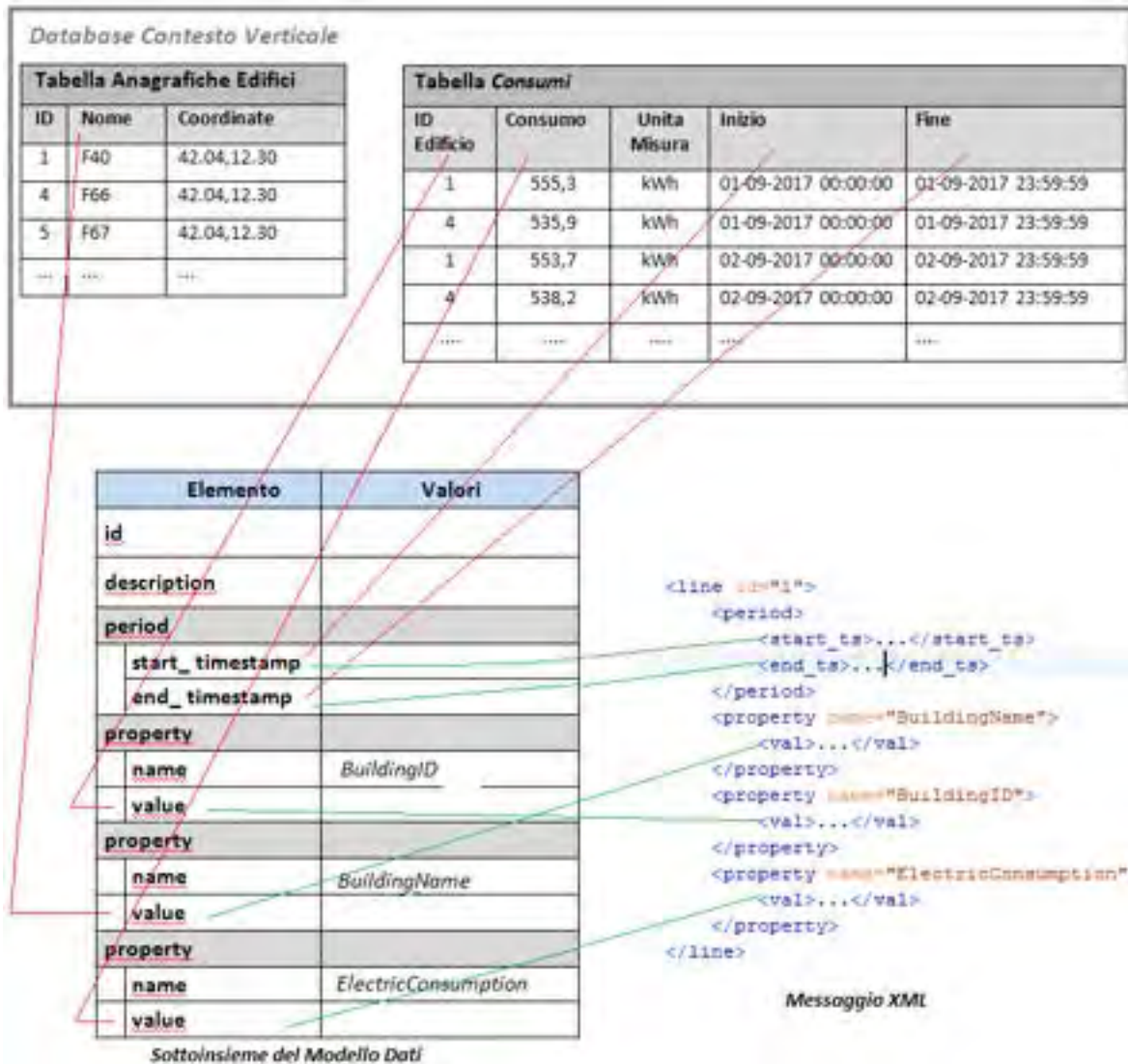


Figura 14 – Esempio di mappatura

5 Esempio: UrbanDataset Whatever

Figura 15 e Figura 16 mostrano un esempio di UrbanDataset "Whatever" in sintassi XML e JSON; ricordiamo che UrbanDataset "Whatever" può essere utilizzato per inviare set di dati relativi a proprietà non ancora definite nell'Ontologia.

Questo esempio mostra anche come definire e valorizzare una proprietà aggregata.

L'esempio riguarda l'invio di dati relativi a eventi sismici rilevati in Italia; per ogni evento vengono indicati l'istante temporale in cui si è verificato, l'epicentro e la magnitudo.

Si noti che:

- nel blocco *context* sono indicate le coordinate di riferimento utilizzate per l'Italia, visto che l'UrbanDataset fa riferimento a eventi che si verificano in questa zona geografica;
- a livello di riga non sono riportate le coordinate poiché il contesto geografico è uguale per tutte le righe ed è quello indicato a livello di UrbanDataset;
- per ogni riga è indicato il timestamp, poiché l'istante di generazione dei valori riportati varia di riga in riga ed, quindi, diverso dall'istante di generazione dell'UrbanDataset;
- a livello di riga non è, invece, indicato il periodo poiché si tratta di misurazioni istantanee che coincidono con l'istante di generazione dei valori indicati.

```

<UrbanDataset>
  <specification>
    <id schemeID="SCPS" > 1.0-Whatever </id>
    <name>Whatever</name>
    <uri> http://smartcityplatform.enea.it/specification/semantic/1.0/ontology/scps-ontology-1.0.owl#Whatever
  </uri>
  <properties>
    <propertyDefinition>
      <propertyName>DataDescription</propertyName>
      <propertyDescription>Descrizione testuale dei dati inviati</propertyDescription>
      <dataType>string</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>Epicentre</propertyName>
      <propertyDescription>Epicentro Evento Sismico</propertyDescription>
      <subProperties>
        <propertyName>Latitude</propertyName>
        <propertyName>Longitude</propertyName>
      </subProperties>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>Latitude</propertyName>
      <propertyDescription>Distanza angolare di un punto dall'Equatore in Gradi decimali.</propertyDescription>
      <dataType>double</dataType>
      <unitOfMeasure> WGS84-DD </unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>Longitude</propertyName>
      <propertyDescription>Distanza angolare di un punto dal Meridiano 0 in Gradi decimali.</propertyDescription>
      <dataType>double</dataType>
      <unitOfMeasure> WGS84-DD </unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>Magnitude</propertyName>
      <propertyDescription>Dimensione dell'evento sismico.</propertyDescription>
      <dataType>double</dataType>
      <unitOfMeasure>ML</unitOfMeasure>
    </propertyDefinition>
    <propertyDefinition>
      <propertyName>timestamp</propertyName>
      <propertyDescription>Tempo di generazione della riga .</propertyDescription>
      <dataType>dateTime</dataType>
      <unitOfMeasure>adimensionale</unitOfMeasure>
    </propertyDefinition>
  </properties>
</UrbanDataset>

```

```

<context>
  <producer>
    <id>INGV-ISIDE</id>
  </producer>
  <timeZone>UTC</timeZone>
  <timestamp>2017-11-19T13:50:00</timestamp>
  <coordinates>
    <longitude>12.56</longitude>
    <latitude>41.87</latitude>
  </coordinates>
  <language>IT</language>
</context>
<values>
  <line id="1">
    <description>Dati evento sismico in Italia.</description>
    <timestamp>2017-11-19T13:37:44</timestamp>
    <property name="Epicentre">
      <property name="Latitude">
        <val>46.06</val>
      </property>
      <property name="Longitude">
        <val>10.03</val>
      </property>
    </property>
    <property name="Magnitude">
      <val>4.4</val>
    </property>
  </line>
  <line id="2">
    <description>Dati di un evento sismico in Italia.</description>
    <timestamp>2017-11-19T13:42:17</timestamp>
    <property name="Epicentre">
      <property name="Latitude">
        <val>44.65</val>
      </property>
      <property name="Longitude">
        <val>10.07</val>
      </property>
    </property>
    <property name="Magnitude">
      <val>2.4</val>
    </property>
  </line>
  <line id="3">
    <description>Dati di un evento sismico in Italia.</description>
    <timestamp>2017-11-19T13:49:41</timestamp>
    <property name="Epicentre">
      <property name="Latitude">
        <val>44.68</val>
      </property>
      <property name="Longitude">
        <val>10.01</val>
      </property>
    </property>
    <property name="Magnitude">
      <val>2.1</val>
    </property>
  </line>
</values>
</UrbanDataset>

```

Figura 15 – Esempio di implementazione XML di un UrbanDataset Whatever

```

{"UrbanDataset": {"specification": {
  "id": {"value": " 1.0-Whatever "},
  "name": "Whatever",
  "uri": " http://smartcityplatform.enea.it/specification/semantic/1.0/ontology/scps-ontology-1.0.owl#Whatever",
  "properties": {"propertyDefinition": [
    {
      "propertyName": "DataDescription",
      "propertyDescription": "Descrizione testuale dei dati inviati",
      "dataType": "string",
      "unitOfMeasure": "adimensionale"
    },
    {
      "propertyName": "Epicentre",
      "propertyDescription": "Epicentro Evento Sismico",
      "subProperty": [
        {"propertyName": "Latitude"},
        {"propertyName": "Longitude"}
      ]
    },
    {
      "propertyName": "Latitude",
      "propertyDescription": "Distanza angolare di un punto dall'Equatore in Gradi decimali.",
      "dataType": "double",
      "unitOfMeasure": " WGS84-DD "
    },
    {
      "propertyName": "Longitude",
      "propertyDescription": "Distanza angolare di un punto dal Meridiano 0 in Gradi decimali.",
      "dataType": "double",
      "unitOfMeasure": " WGS84-DD "
    },
    {
      "propertyName": "Magnitude",
      "propertyDescription": "Dimensione dell'evento sismico.",
      "dataType": "double",
      "unitOfMeasure": "ML"
    },
    {
      "propertyName": "timestamp",
      "propertyDescription": "Tempo di generazione della riga.",
      "dataType": "dateTime",
      "unitOfMeasure": "adimensionale"
    }
  ]}]
},
"context": {
  "producer": {
    "id": "INGV-ISIDe"
  },
  "timeZone": "UTC",
  "timestamp": "2017-11-19T13:50:00",
  "coordinates": {
    "longitude": 12.56,
    "latitude": 41.87
  },
  "language": "IT"
},
"values": {"line": [
  {
    "id": 1,
    "description": "Dati evento sismico in Italia.",
    "timestamp": "2017-11-19T13:37:44",
  }
]}

```

```

"property": [
  {
    "name": "Epicentre",
    "property": [
      { "name": "Latitude",
        "val": "46.06"
      },
      { "name": "Longitude",
        "val": "10.03"
      }
    ]
  },
  { "name": "Magnitude",
    "val": "4.4"
  }
]
},
{
  "id": 2,
  "description": "Dati evento sismico in Italia.",
  "timestamp": "2017-11-19T13:42:17",
  "property": [
    {
      "name": "Epicentre",
      "property": [
        { "name": "Latitude",
          "val": "44.65"
        },
        { "name": "Longitude",
          "val": "10.07"
        }
      ]
    },
    { "name": "Magnitude",
      "val": "2.4"
    }
  ]
},
{
  "id": 3,
  "description": "Dati evento sismico in Italia.",
  "timestamp": "2017-11-19T13:49:41",
  "property": [
    { "name": "Epicentre",
      "property": [
        { "name": "Latitude",
          "val": "44.68"
        },
        { "name": "Longitude",
          "val": "10.01"
        }
      ]
    },
    { "name": "Magnitude",
      "val": "2.1"
    }
  ]
}
]
}
}
}

```

Figura 16 – Esempio di implementazione JSON di un UrbanDataset Whatever

6 Risorse disponibili

Le risorse disponibili per il Livello Information possono essere suddivise in tre macro-categorie:

- *Specification*
- *Case Studies Resources*
- *Tools*

La categoria **Specification** include le specifiche di riferimento del formato UrbanDataset, quindi:

- questa specifica
- gli schemi che implementano il Modello Dati Astratto nelle due sintassi di riferimento:
 - XML Schema (*scps-urbandataset-schema-1.0.xsd*)
 - JSON Schema (*scps-urbandataset-schema-1.0.json*)
- un **JSON** e un **XML Campione**: sono istanze con dati fittizi che hanno lo scopo di illustrare la struttura delle istanze UrbanDataset
- il **Modello Dati Astratto** (documento *NN224-015-vXX-ModelloDatiAstratto.pdf*) estratto da questa specifica.

La categoria **Case Studies Resources** include del materiale di supporto (esempi, template,...) realizzato per i casi studio analizzati volto a facilitare e illustrare l'adozione delle specifiche in questi scenari. Per esempio:

- istanze XML e JSON di esempio relative ai casi studio supportati
- materiale a supporto dell'implementazione dei casi studio supportati:
 - **template JSON e XML** (sono delle istanze in cui stati valorizzati solo gli elementi e attributi che hanno valore fisso per l'UrbanDataset del caso studio a cui si riferiscono, per esempio i "propertyName"; gli altri elementi e attributi, per esempio i "val", non sono valorizzati)
 - **Guide all'implementazione del caso studio**: descrive lo scenario di riferimento e gli UrbanDataset utilizzati nel caso studio
 - **Schede di documentazione degli UrbanDataset** coinvolti nel caso studio.

La categoria **Tools** include un set di applicazioni, basate su tecnologie Java e XSL, capaci di fornire servizi o di creare artefatti che possano supportare l'implementazione e l'adozione degli UrbanDataset definiti nell'Ontologia, nei formati previsti (XML e JSON). Nello specifico le applicazioni disponibili sono:

- un **generatore automatico di documentazione**: consente di generare la scheda di descrizione di ciascun UrbanDataset presente nell'ontologia;
- un **generatore automatico di Template** degli UrbanDataset in formato XML e JSON;
- un **generatore automatico di file Schematron** per la validazione semantica degli Urban Dataset: il generatore è in grado di produrre uno specifico file Schematron per ciascun UrbanDataset definito nell'Ontologia; ogni file Schematron implementa le regole semantiche per lo specifico Urban Dataset, ad esempio: set di proprietà che lo compongono, il loro tipo di dato, eventuali unità di misura o liste di codici da utilizzare. I file Schematron possono essere utilizzati in validatori XML esistenti o per la realizzazione di propri validatori (per esempio lato Solution);
- un **trasformatore di UrbanDataset in formato XML** (conforme all'XML Schema definito nel livello SCPS Information) in equivalenti UrbanDataset in formato JSON (conforme al JSON Schema definito nel livello Information delle SCPS);
- un **trasformatore di UrbanDataset in formato JSON** (conforme al JSON Schema definito nel livello Information delle SCPS) in equivalenti UrbanDataset in formato XML (conforme all'XML Schema definito nel livello Information delle SCPS);
- un **validatore di UrbanDataset in formato XML o JSON**: consente di validare la conformità sintattica e semantica degli UrbanDataset XML o JSON. La validazione sintattica verifica la

correttezza delle istanze rispetto al JSON Schema, o XML Schema a seconda del formato. La validazione semantica verifica la conformità alla specifica ontologica dell'UrbanDataset e viene effettuata tramite Schematron.

Le suddette applicazioni saranno fruibili attraverso l'interfaccia di navigazione dell'Ontologia³ (Figura 5).

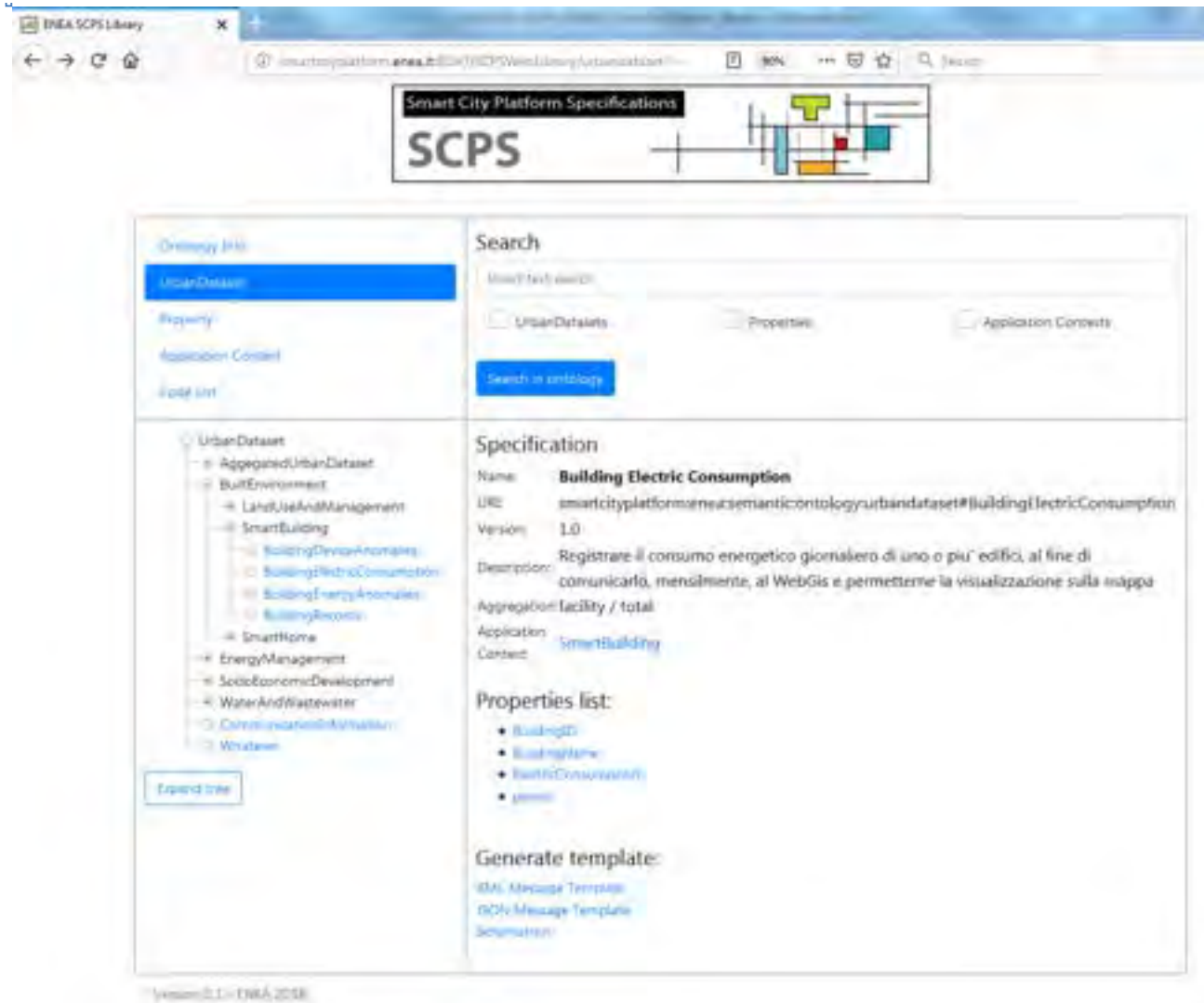


Figura 17 – Interfaccia di navigazione Ontologia

³ Interfaccia di navigazione Ontologia: <http://smartcityplatform.enea.it:8080/SCPSWebLibrary/ontologyinfo>

7 Riferimenti

1. **W3C**. Extensible Markup Language (XML). W3C. [Online] 11 10 2016. <https://www.w3.org/XML/>.
2. **IETF**. The JavaScript Object Notation (JSON) Data Interchange Format (RFC 7159). [Online] 14 10 2015. <https://datatracker.ietf.org/doc/rfc7159>.
3. **W3C**. XML Schema (XSD). [Online] W3C, 05 04 2012. <https://www.w3.org/standards/xml/schema>.
4. **JSON-Schema-org**. JSON Schema specification - draft 07. [Online] 19 11 2017. <http://json-schema.org/documentation.html>.
5. **schematron.com**. Schematron. [Online] 2017. <http://schematron.com/>.