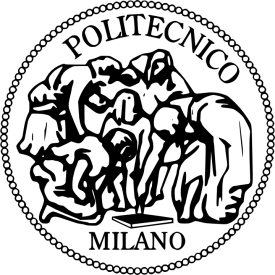
****

****

Advanced Software Engineering

SmartCity, IoT, Social

Network Data Modeling

Group 9

Amin Endah Sulistiawati | 854527

Daniel Rosato |

2016

**Scenario**

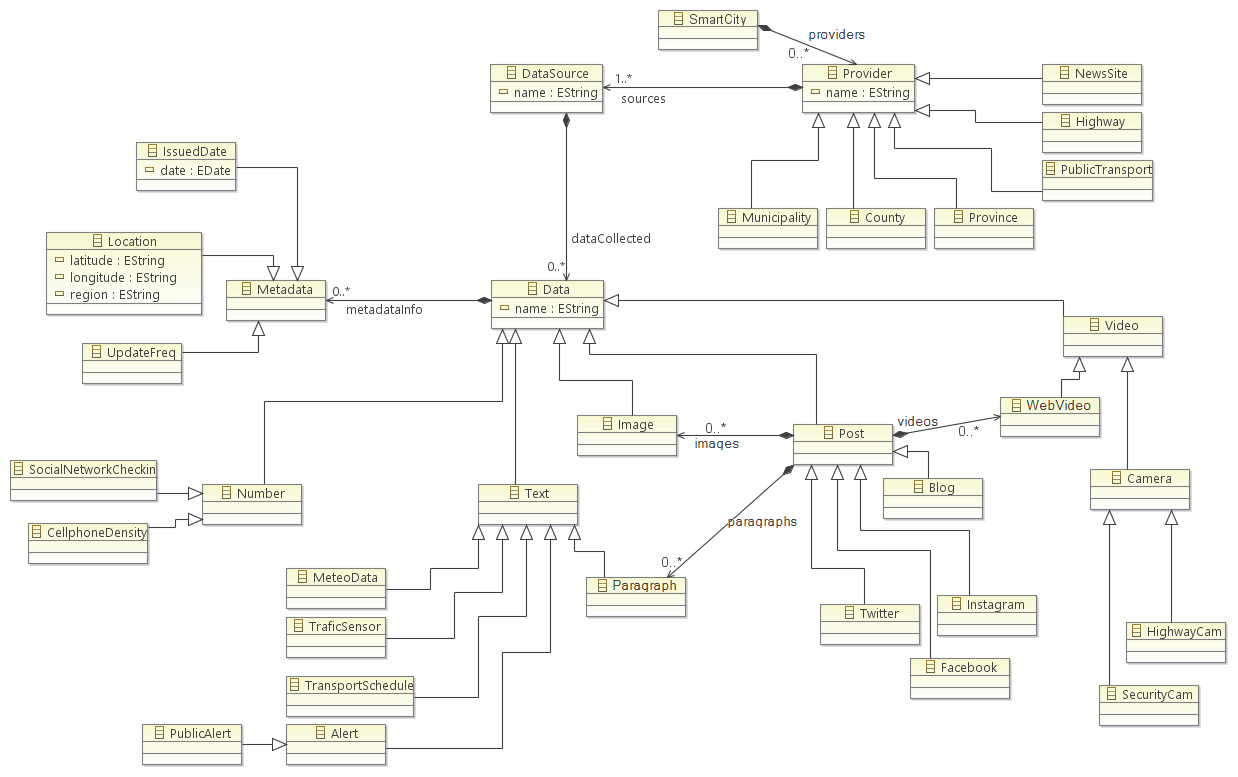
Design modeling language that supports **the specification of the information sources** related to a city, aiming at collecting information about the behavior of citizens. The metamodel should describe the various data providers: municipality, county, province, public transport authority, highway mgmt companies, news sites and feeds, social networks and so on. It must also describe any possible data source, and associate it to the provider that owns it (e.g., tweets, web cams on the highway, security cameras, traffic sensors, photos e.g. from Instagram or other sources, videos, meteo data, public transportation schedules, public alerts, density of presence of cell phones by area, checkins on social networks at locations, …). The metamodel must describe a taxonomy of data sources, their data types and metadata, such as the frequency of update and so on. (Particular care must go into the definition of the hierarchy of concepts here).

**Tasks**

1. The abstract syntax of the language with the appropriate OCL constraints wherever needed
2. A concrete textual syntax of the language
3. A modeling environment (tool)
4. At least two example models conforming to the language
5. A model­to­text transformation that generates in output a textual report with statistics from the data derived by the data sources, for instance aggregating such information by location and / or time.

**META MODEL**

The following diagram represents meta-model for Smart City. Smart City has various provider including news sites, highway, public transport, municipality, county and province. Smart City is rich of data. The data source associated with providers can be various. Data source groups the data based on the type of data that are number, text, image and video. Meta data is also described to store location, time and update frequency of the data. Uniquely, social media data are not only grouped by types of data, but also maintained in a post.

****

*Fig. 01: MetaModel for Smart City*

**OCL Constrain**

In order to build the model and the tools some assumptions have been made regarding to the general idea of the Smart City project, for each one an OCL constraint has been created, we discuss these assumption in detail below.

|  |  |
| --- | --- |
| **For Post** |  |
| * The properties of a post are videos, images, and paragraph. Post videos must be type of web video, post images must be type of image and post paragraphs must be type of paragraph. | |
|  | |
| **class** Post **extends** *Text*,*Image*,*Video*,*Data*  {  **property** videos : *WebVideo*[\*] { **ordered** **composes** };  **property** images : *Image*[\*] { **ordered** **composes** };  **property** paragraphs : *Paragraph*[\*] { **ordered** **composes** };  } | |

|  |  |
| --- | --- |
| **Facebook Post** |  |
| * A facebook post contains at least a paragraph or one or many images or one video. Image and web video cannot be in one post | |
|  | |
| **class** Facebook **extends** *Post*  {  **invariant**  validPost: (  (**self**.*images* -> *size*() > 0 **and** **self**.*videos* -> *size*() = 0) **or**  (**self**.*images* -> *size*() = 0 **and** **self**. *videos* -> *size*() =1) **or**  **self**. *paragraphs* -> *size*() =1  );  } | |

|  |  |
| --- | --- |
| **Twitter Post** |  |
| * A twitter post contains at least a paragraph or one or many images or one video. Image and web video cannot be in one post | |
|  | |
| **class** Twitter **extends** *Post*  {  **invariant**  validPost: (  (**self**.*images* -> *size*() > 0 **and** **self**.*videos* -> *size*() = 0) **or**  (**self**.*images* -> *size*() > 0 **and** **self**.*videos* -> *size*() = 1) **or**  **self**.*paragraphs* -> *size*() = 1  );  } | |

|  |  |
| --- | --- |
| **Instagram Post** |  |
| * An instagram post contains at least one or image or one video. Paragraph or text is optional | |
|  | |
| **class** Instagram **extends** *Post*  {  **invariant**  validPost: (  (**self**.*images* -> *size*() = 1 **and** **self**.*videos* -> *size*() = 0) **or**  (**self**.*images* -> *size*() = 0 **and** **self**. *videos* -> *size*() = 1) **and**  (**self**.*paragraphs* -> *size*() = 1 **or** **self**.*paragraphs* -> *size*() = 0)  );  } | |

|  |  |
| --- | --- |
| **Transport Schedule** |  |
| * A transport schedule must be provided by public transport. | |
|  | |
| **class** TransportSchedule **extends** *Text*  {  **Invariant**  validProvider: (**self**.*DataSource*.*Provider*.*oclIsTypeOf*(*PublicTransport*));  } | |

**TEXTUAL CONCRETE SYNTAX**

For the textual concrete syntax we’ve used .xtext in order to provide a textual tool environment.

**grammar** com.polimi.ase.concretesyntax.xtext.SmartCity **with** org.eclipse.xtext.common.Terminals

**generate** smartCity "http://www.polimi.com/ase/concretesyntax/xtext/SmartCity"

SmartCity:

(providers+=Provider)+

(dataSources+=DataSource)\*

;

Provider:

NewsSite | Highway | PublicTransport | Province | County | Municipality

;

NewsSite:

"NewsSite " name=ID " {"

"sources " sources+=([*DataSource*]) (","sources+=([*DataSource*]))\*

"}"

;

Highway:

"Highway " name=ID " {"

"sources " sources+=([*DataSource*]) (","sources+=([*DataSource*]))\*

"}"

;

PublicTransport:

"PublicTransport " name=ID " {"

"sources " sources+=([*DataSource*]) (","sources+=([*DataSource*]))\*

"}"

;

Province:

"Province " name=ID " {"

"sources " sources+=([*DataSource*]) (","sources+=([*DataSource*]))\*

"}"

;

County:

"County " name=ID " {"

"sources " sources+=([*DataSource*]) (","sources+=([*DataSource*]))\*

"}"

;

Municipality:

"Municipality " name=ID " {"

"sources " sources+=([*DataSource*]) (","sources+=([*DataSource*]))\*

"}"

;

DataSource:

"DataSource " name=ID " {"

(data+=Data)+

"}"

;

Metadata:

Location | UpdateFreq | IssuedDate

;

IssuedDate:

"IssuedDate {"

"date" date=STRING

"}"

;

Location:

"Location {"

"latitude" latitude=STRING

"longitude" longitude=STRING

"region" region=STRING

"}"

;

UpdateFreq:

"UpdateFreq {"

"time" time=INT

"}"

;

Data:

Number | Text | Image | Video | Post | "Data {"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

Number:

SocialNetworkCheckins | CellphoneDensity

;

SocialNetworkCheckins:

"SocialNetworkCheckins {"

"amount" amount=INT

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

CellphoneDensity:

"CellphoneDensity {"

"measure" measure=INT

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

Text:

MeteoData | TrafficSensor | TransportSchedule | Alert | Paragraph

;

Paragraph:

"Paragraph " data=STRING

;

MeteoData:

"MeteoData {"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

TrafficSensor:

"TrafficSensor {"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

TransportSchedule:

"TransportSchedule {"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

Alert:

PublicAlert

;

PublicAlert:

"PublicAlert {"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

Post:

Twitter | Facebook | Blog | Instagram

;

Twitter:

"Twitter {"

"texts {"

(paragraphs+=Paragraph)\*

"}"

"images {"

(images+=Image)\*

"}"

"videos {"

(videos+=WebVideo)\*

"}"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

Blog:

"Blog {"

"texts {"

(texts+=Paragraph)\*

"}"

"images {"

(images+=Image)\*

"}"

"videos {"

(videos+=WebVideo)\*

"}"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

Facebook:

"Facebook {"

"texts {"

(texts+=Paragraph)\*

"}"

"images {"

(images+=Image)\*

"}"

"videos {"

(videos+=WebVideo)\*

"}"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

Instagram:

"Instagram {"

"texts {"

(texts+=Paragraph)\*

"}"

"images {"

(images+=Image)\*

"}"

"videos {"

(videos+=WebVideo)\*

"}"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

Image:

"Image url " url=STRING

;

Video:

Camera | WebVideo

;

WebVideo:

"WebVideo url " url=STRING

;

Camera:

SecurityCam | HighwayCam

;

HighwayCam:

"HighwayCam {"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

SecurityCam:

"SecurityCam {"

"metadataInfo {" (metadata+=Metadata)+ "}"

"}"

;

**MODEL TO TEXT TRANSFORMATION**

For the Model-To-Text Transformation we use Acceleo to generate a report in text format. The output folder (where the .txt files are generated is named smartcityreport.txt)

We have numerous data, with various type and location described in metadata. We want to organize the data by giving statistic report. The idea is we have two approaches to make the statistic report. The firs approach is showing the statistic data based on location and the second one is based on provider and location.

In location based, we collect the data and group based on the region name in which the data obtained. Then, we count each type of data and sum up them.

In the second approach, we divide data firstly based on the provider. Then, we make statistic repost for each location in the same provider.

---------------------

Report Location Based

---------------------

Bergamo

Twitter = 1

TransportSchedule = 1

----------------------

Total = 2

Lecco

Twitter = 2

MeteoData = 7

----------------------

Total = 9

Como

Twitter = 3

TransportSchedule = 1

Facebook = 2

Instagram = 2

----------------------

Total = 8

Monza

TransportSchedule = 1

Instagram = 2

----------------------

Total = 3

Milan

Twitter = 6

PublicAlert = 2

TransportSchedule = 1

Facebook = 4

MeteoData = 4

Instagram = 5

----------------------

Total = 22

----------------------------------

Report Provider and Location Based

----------------------------------

Lombardy Post

Bergamo

-----------

Twitter = 1

----------------------

Total = 1

Lecco

-----------

Twitter = 2

----------------------

Total = 2

Como

-----------

Twitter = 3

Facebook = 2

Instagram = 2

----------------------

Total = 7

Monza

-----------

Instagram = 2

----------------------

Total = 4

Milan

-----------

Twitter = 6

Facebook = 4

Instagram = 5

----------------------

Total = 15

Trenord

Bergamo

-----------

TransportSchedule = 1

----------------------

Total = 1

Monza

-----------

TransportSchedule = 1

----------------------

Total = 1

Milan

-----------

TransportSchedule = 1

----------------------

Total = 1

Como

-----------

TransportSchedule = 1

----------------------

Total = 1

Lombardy

Lecco

-----------

MeteoData = 7

----------------------

Total = 7

Milan

-----------

PublicAlert = 2

MeteoData = 4

----------------------

Total = 6