HAL
open science

# Connecting Semantic 3D Models to the LOD Cloud for Mobile Applications: Discovering the Surroundings with ARCAMA-3D

Betül Aydin

## To cite this version:

Betül Aydin. Connecting Semantic 3D Models to the LOD Cloud for Mobile Applications: Discovering the Surroundings with ARCAMA-3D. Ubiquitous Computing. Université Grenoble Alpes, 2015. English. NNT : . tel-01273036

**HAL Id: tel-01273036**

**https://hal.science/tel-01273036v1**

Submitted on 11 Feb 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Communauté UNIVERSITÉ Grenoble Alpes

# THÈSE

Pour obtenir le grade de

## DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

**Betül AYDIN**

Thèse dirigée par **Mr Jérôme Gensel**
et codirigée par **Mme Sylvie Calabretto**

préparée au sein du **Laboratoire d'Informatique de Grenoble (LIG)**
et de l'**École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

# Connecting Semantic 3D Models to the LOD Cloud for Mobile Applications:

## Discovering the Surroundings with ARCAMA-3D

Thèse soutenue publiquement le **25 septembre 2015**,
devant le jury composé de :

**Mr Max Chevalier**
Maître de conférences, HDR, Université Paul Sabatier, Rapporteur
**Mr Thierry Joliveau**
Professeur, Université Jean Monnet, Rapporteur
**Mr Bernard Espinasse**
Professeur, Aix-Marseille Université, Examinateur
**Mr Gilles Falquet**
Professeur associé, Université de Genève, Examinateur
**Mme Claudia Roncancio**
Professeur, Grenoble-INP, Présidente
**Mme Sylvie Calabretto**
Professeur, INSA de Lyon, Co-Directrice de thèse
**Mr Jérôme Gensel**
Professeur, Université Pierre-Mendès-France, Directeur de thèse
**Mr Bruno Tellez**
Maître de conférences, Université Claude Bernard Lyon 1, Co-Encadrant de thèse

# Acknowledgments

Many have contributed to the completion of this thesis. This thesis is dedicated to all of you who with your guidance, support, smile and love gave me the strength to complete this endeavor.

I want to, first of all, thank the members of the jury: Mr Max Chevalier, Mr Thierry Joliveau, who have accepted to evaluate my dissertation, as well as Mr Bernard Espinasse, Mr Gilles Falquet, Mme Claudia Roncancio, who have accepted to assist to my defense.

I would like to express my deepest gratitude to my advisors, Mr Jérôme Gensel and Ms Sylvie Calabretto for giving me the opportunity to work with them. Thank you Jérôme, for your guidance, effort and support during these years, also for reading my drafts countless times and constantly challenging my ideas. Thank you Sylvie, for your constant support and kindness. I thank both of you for your advice, time and encouragement.

My sincere thanks to Mr Bruno Tellez, for giving me the chance to work again together. Our constant discussions in front of the white board or just near the coffee machine have definitely broadened my perspective. Thank you again for listening me, giving me suggestions and supporting me during the difficult moments of this thesis.

I would also like to thank to Mr Philippe Genoud, for his great help in the software development work that contributed to this thesis. Thank you for dedicating your time and interest to explain to me subjects that I was not familiar with.

I am particularly grateful for the friendly working environment in the STEAMER research group that has shaped my understanding of how research is carried on. Thank you Marlène, Paule-Annick, Danielle, Sylvain, Hervé, Benoit, Mahfoud, for your kindness, generosity, and especially for encouraging me to finish this thesis. Without the presence of my dear colleagues, this thesis wouldn't have been complete. Thank you Tales, Reinaldo, Carina, Mouna, Anthony, Camille et Cécile.

I want to extend my gratitude to the people of IUT Lyon 1, who have welcomed me and helped me during my year as a teaching assistant.

I would also like to thank the students with whom I had the chance to work with. Thank you Fanny, Matthieu, Erik and Julien for your patience as my ideas and research evolved during your internship.

My warm acknowledgment to the staff of the University of Grenoble, thanks to Elsa, Pascale, Ghislane, Samantha, Ms Felicette Boero and Mr Djamel Hadji for

making my stay easier.

Huge thanks to my friends for reminding me that there are more to life than science: Hortense, Ramon, Aysegül, Defne, Ozan, Dani, Flo, Sophie, Pierre, Ronan, Rada, Josy, Tales, Aastha, Prakhar, Nicolas, Paul-Antoine, Yeter, Lionel, Camille and Benjamin. This adventure became worthwhile thanks to your friendship.

My sincere thanks to Anne, Patrick, Luc, Marion, Annik and Monique, for being like my family when I was away from my own. I could not have done all of this without you.

Words cannot express how grateful I am to my family: my mother, my father, my brother and So-Hyun. Thank you for supporting me during this journey and my life. To my father: you are my best friend. Thank you for teaching me that with hard work, there is nothing one cannot achieve. To my mother, thank you for giving your best in understanding me and being my best supporter. To my brother, Bilal, even though we stayed apart during this time, thank you for your jokes that made me laugh and feel like we were always together. To So-Hyun, thank you for cheering me up and being always kind to me during my hard times.

There is one person, without whom I would have never been able to pursue and conclude this thesis. Thank you for your unconditional support, love and care. Thank you for walking by my side. Because this thesis is more yours than anyone else, it is dedicated to you.

To Jun-Young.

## Abstract

Discovering the surroundings while walking using a mobile device is now possible by coupling new technologies such as Augmented Reality (AR) and location-based services. However, the amount of geo-referenced information increases every day, so it is suitable to adapt a generic approach that aims at presenting information in a structured manner and adapted to different application domains and user profiles. Our work focuses on the definition of a data model for 3D objects to be integrated into mobile AR applications. The data model we propose prioritizes access to available spatio-temporal and thematic information associated to objects or points-of-interests of the real world. For this purpose, initially, we publish on the Web 3D objects corresponding to real-world objects using the data model, which allows interlinking them with available data sources on the LOD cloud. Then, building on this conceptualized knowledge base, which is constantly evolving, we propose an architecture, called ARCAMA-3D, whose modules allow application designers and developers to create their own mobile applications and multi-thematic to explore the surrounding area, exploiting both Augmented Reality and the LOD cloud.

**Keywords** Semantic Web, Augmented Reality, Mobile Devices, Spatio-temporal Modelling, Surrounding Discovery, Storytelling

## Résumé

Découvrir les environs tout en marchant à l'aide d'un appareil mobile est désormais possible en associant les nouvelles technologies telles que la Réalité Augmentée (RA) et les services basés sur la localisation. Toutefois, la quantité d'informations géo-référencées disponibles augmentant chaque jour, il convient d'adopter une approche générique qui vise à présenter l'information de manière structurée et adaptée à la fois aux différents domaines de découverte et aux profils divers des utilisateurs. Notre travail se concentre sur la définition d'un modèle de données pour les objets 3D destiné à être intégré dans des applications de RA mobiles. Notamment, le modèle de données que nous proposons donne la priorité à l'accès à l'information de nature spatiale, temporelle et thématique associée aux objets ou points d'intérêt du monde réel. À cette fin, dans un premier temps, nous publions sur le Web, en utilisant le modèle de données, les objets 3D correspondant aux objets du monde réel, ce qui permet de les lier avec des sources de données disponibles sur le nuage du Web de Données ou LOD Cloud. Puis, en nous appuyant sur cette base de connaissances conceptualisée et en constante évolution, nous proposons une architecture, appelée ARCAMA-3D, dont les modules permettent aux concepteurs et développeurs de créer leurs propres applications mobiles et multithématiques de découverte des environs, en exploitant à la fois la Réalité Augmentée et le Web des Données.

**Mots-clés** Web Sémantique, Réalité Augmentée, Appareils Mobiles, Modelisation spatio-temporelle, Découverte des environs

# Contents


**I General Introduction** **1**

**Introduction** **3**

Problem Definition and Thesis Contribution . . . . . 5

Thesis Outline . . . . . 8

**1 Surroundings Discovery Applications** **11**

1.1 Introduction . . . . . 11

1.2 Information processing: from sensing and querying to visualization . 14

1.2.1 Sensing the World with Mobile Devices . . . . . 16

1.2.2 Location-based Content Retrieval . . . . . 18

1.2.3 Visualizing Location-based Content during Surroundings Discovery . . . . . 22

1.3 Augmented Reality . . . . . 33

1.3.1 Mobile AR systems . . . . . 35

1.3.2 Tracking and Registration . . . . . 37

1.4 Conclusion . . . . . 40

**2 Semantic Information in Mobile Applications** **43**

2.1 Introduction . . . . . 43

2.2 Semantic Information . . . . . 44

2.2.1 Web x.0 Evolution . . . . . 45

2.2.2 Ontologies . . . . . 50

2.2.3 Ontology Description Languages . . . . . 53

2.2.4 Linked Open Data . . . . . 62

2.2.5 DBpedia . . . . . 68

2.2.6 Publishing Data on the LOD Cloud . . . . . 74

2.3 Different Ways to Use Semantic Information . . . . . 76

2.3.1 Extracting Information from the LOD Cloud . . . . . 76

2.3.2 Examples of Applications based on the LOD Cloud . . . . . 78

2.3.3 Semantic Information on Mobile Devices . . . . . 83

2.4 Conclusion . . . . . 86

**Synthesis** **89**

**II Proposition** **93**

**3 Overview: A New Augmented Reality Approach** **95**
3.1 Introduction . . . 95
3.2 Overview . . . 98
3.3 Conclusion . . . 103

**4 3D2T: A Data Model Mixing 3D, Thematic and Temporal Information** **105**
4.1 Introduction . . . 105
4.2 3D, Thematic and Temporal Data Modeling . . . 108
4.2.1 Thematic Layer . . . 109
4.2.2 3D Information Layer . . . 122
4.2.3 Temporal Layer . . . 126
4.3 3D Data Model . . . 128
Arcama-owl Ontology . . . 129
4.4 Publishing 3D Models on the LOD . . . 134
4.5 Conclusion . . . 138

**5 From LOD to Augmented Reality based Interaction** **141**
5.1 Introduction . . . 141
5.2 System Overview of the AR Application . . . 143
5.2.1 Application Modules . . . 144
5.2.2 The Application Flow based on the User's Interaction . . . 146
5.3 AR Objects as Points of Interests . . . 147
5.3.1 A Warehouse for 3D Objects . . . 149
5.3.2 Tracking and Sensor Module . . . 149
5.4 Querying the LOD Cloud . . . 153
5.4.1 Thematic and Temporal Visits of the Surroundings . . . 153
5.4.2 Creating Narratives using Multimedia Documents (Site Specific Storytelling) . . . 157
5.5 Conclusion . . . 161

**6 ARCAMA-3D** **165**
6.1 Introduction . . . 165
6.2 General Architecture and its Components . . . 166
6.2.1 Triple store . . . 166
6.2.2 Providing Further Information from the LOD Cloud Depending on Different Application Scenarios . . . 169
6.3 Use Case Scenario: Royal Monastery of Brou . . . 172
Stories Related to the Monastery . . . 173
6.4 Conclusion . . . 180

**Conclusion** **183**
Contributions . . . 185

Perspectives . . . . . 187

**Bibliography** **189**

**A Appendix** **215**

# List of Figures


1.1 A representation of the growing class of sensor-enabled phones. . . . 17
1.2 POI recommendations according to context in [Baltrunas *et al.* 2012]. 21
1.3 The tasks and sub-tasks in [Liagouris *et al.* 2011]. . . . . 22
1.4 2D maps on mobile devices for surroundings discovery. . . . . 24
1.5 POI visualization using Google Maps and Apple Maps. . . . . 26
1.6 3D maps on mobile devices for surroundings discovery. . . . . 27
1.7 3D city model of Munich generated by the OSM-3D project [Goetz & Zipf. 2012]. . . . . 29
1.8 Levels-of-detail introduced by CityGML [Kolbe *et al.* 2005]. . . . . 30
1.9 Complex object represented by CityGML [Stadler & Kolbe 2007]. . . 30
1.10 Head-mounted displays in late 90s and Google Glass. . . . . 34
1.11 The RV continuum introduced by [Milgram & Kishino 1994]. . . . . 35
1.12 Layer-based augmented reality [Lee *et al.* 2009a], Layar and Wikitude. 37
1.13 Vision-based tracking with markers. . . . . 38

2.1 Semantic Web Layers (Layer Cake) [Berners-Lee 2000]. . . . . 48
2.2 Components of an ontology [Keet 2004]. . . . . 51
2.3 Two dimension classification of ontologies [van Heijst *et al.* 1997]. . . 52
2.4 Geographical data expressed as a graph [Grimm *et al.* 2007]. . . . . 53
2.5 URL, URN and URI. . . . . 55
2.6 Examples of commonly used namespaces. . . . . 57
2.7 A simple RDF graph showing foaf relations. . . . . 57
2.8 Domain and range of RDFS properties [Gandon *et al.* 2012]. . . . . 60
2.9 The evolution of Linked Open Data. . . . . 64
2.10 The distributions of triples and domains in LOD. . . . . 65
2.11 Linked Open Data project as of April 2014. . . . . 67
2.12 Infobox of a Wikipedia article and Religious building infobox template. 69
2.13 SKOS vocabulary representing relations between Wikipedia categories. 71
2.14 RDF graph represented as a 3D tensor in [Mirizzi *et al.* 2012]. . . . . 79
2.15 Data Integration in LODAC project [Kamura *et al.* 2011]. . . . . 81
2.16 Yokohama Art Spot integrates different data sources [Matsumura *et al.* 2012]. . . . . 81
2.17 Semantically heterogeneous and mutually linked cultural heritage [Hyvönen *et al.* 2009a]. . . . . 82
2.18 The map view of DBpedia Mobile [Becker & Bizer 2008]. . . . . 83

2.19 The PRISSMA vocabulary [Costabello 2011]. . . . . 84
2.20 Semantic information on mobile devices. . . . . 85

3.1 An illustration of our approach. . . . . 97
3.2 Overview of the general approach. . . . . 100
3.3 The key elements in our approach. . . . . 101

4.1 Conceptualized 3D data model with geometry, thematic and temporal components . . . . . 106
4.2 An excerpt of the FTT-v01 ontology. . . . . 113
4.3 MediOnto classes are mapped with DBpedia ontology classes and category hierarchy. . . . . 114
4.4 A mapping example between MediOnto and DBpedia class ontologies. 115
4.5 An example of similar concepts in DBpedia and MediOnto. . . . . 117
4.6 A mapping between MediOnto classes and DBpedia SKOS concepts. 118
4.7 A representation of mapping MediOnto classes to DBpedia categories. 119
4.8 Mapping of a concept to DBpedia ontology and categories. . . . . 122
4.9 The 3D extrusion of a library building using OSM and Matlab . . . 124
4.10 A 3D object evolution in time. . . . . 125
4.11 The thematic and geometric evolution of the real world entity *Hagia Sophia*. . . . . 128
4.12 UML class diagram of the arcama-owl ontology. . . . . 130
4.13 Representations of *Hagia Sophia* according to its thematic and geometric evolutions. . . . . 132
4.14 The web-based application for interlinking 3D models with LOD cloud. 134

5.1 Four modules of the mobile AR application. . . . . 144
5.2 The flow diagram during the execution of the AR application. . . . . 145
5.3 The transitions based on the choices made by the user. . . . . 147
5.4 Semi-transparent 3D models are displayed in different colors. . . . . 149
5.5 The interface for adding 3D objects to the warehouse. . . . . 150
5.6 One of the 3D objects of the warehouse with related information. . . 151
5.7 Sensor data is fusioned in real-time for the superimposition. . . . . 151
5.8 The AR view with misalignment. . . . . 152
5.9 Screen capture of the interface to correct the misalignment. . . . . 152
5.10 The representation of a thematic list displayed on the screen based on the types of entities. . . . . 155
5.11 Temporal visit of an edifice: *Hagia Sophia, Istanbul, Turkey*. . . . . 156
5.12 Parts of the 3D object are rendered in different colors. . . . . 158
5.13 The relationships between types of structures, multimedia documents, site specific stories, 3D objects and object parts. . . . . 159
5.14 The user interacts with the highlighted object parts in order to access information. . . . . 160

6.1 Overview of the general architecture. . . . . 167

6.2 UML diagram of information query-retrieval process between mobile device and triple store. . . . . 168
6.3 Various applications can be developed using ARCAMA-3D. . . . . 170
6.4 The representation of the queried information. . . . . 174
6.5 The 3D temporal evolution of the monastery's bell tower. . . . . 176
6.6 The stories are attached to temporal representations. . . . . 177
6.7 The construction story of the monastery is told using multimedia files. 178

A.1 MediOnto.owl ontology. . . . . 215

# Part I

# General Introduction

# Introduction

Nowadays, combined with embedded sensors (Global Positioning System (GPS), accelerometer, gyroscope, compass, digital camera, *etc.*), the location, orientation (direction in which the device is held according to the local earth frame), and velocity (direction in which the user is heading to and speed of movement) of most of mobile devices can be instantly calculated. With the ability to derive location information at any time and anywhere and the increasing number of usages, mobile devices can provide the user with messages about her location. The *Context*, which can be defined differently according to the field of application (for instance, location, identities of nearby objects or people, time, environmental states *etc.*), is used by the systems to provide relevant information to the user, as her location changes over time. Using her mobile device, the user can gain access to services related to her current context or location [Suomela 2006].

On the other hand, Augmented Reality (AR) has opened up new horizons of interaction for discovering the surroundings by overlaying interactive digital information on the captured scene. However, there is no widely agreed design approach in accessing information using AR and most of the propositions remain domain specific (tourism, navigation, real estate, *etc.*). AR applications should provide selective location-based data that may be interesting to the user, instead of distracting her by providing *all* the available data. In this regard, intelligent approaches, such as conceptualizing the information belonging to the surrounding objects and delivering only the necessary information to the user (by taking into account user preferences and profile) should be considered. AR applications may also allow the user to demand information by pointing her mobile device towards an object around and receive information related to it by tapping on that object on the mobile device's screen. These are some of the interesting solutions to create user friendly interactions and to make the perception an easy task.

Virtual 3D models are widely used in surroundings discovery mobile applications. In those applications, interpreting the scene and visualization of information become easier, as they offer various interaction mechanisms. People look for familiar objects around them to find out where they are when they are visiting a place [Neumann 2010]. It is observed that 3D maps provide users with a better recognition of their position and orientation, since they can observe the resemblance of the landmarks [Oulasvirta *et al.* 2009]. In order to make a landmark based, visually

oriented instruction approach, it is found necessary to provide all the buildings in 3D, not only the landmarks [Bogdahn & Coors 2009]. In this sense, 3D Geographical Information Systems (GIS) have become important location-based sources for mobile guides since they provide geometric and semantic data. GIS data consists of spatially enabled information sources and, in that sense, they can be used in AR applications as 3D geographical data of the surroundings. On the other side, mobile devices allow to access the physical location, orientation and direction of movement of the user in real-time. This information, combined with GIS data, improves the user experience, such that she can point her mobile device to the objects around and 3D GIS content of the surroundings can be downloaded on the device.

3D models provide novel means to visualize and organize the spatial content [Fröhlich *et al.* 2011]. Merging these two fields, 3D GIS and mobile AR, promises a simpler, yet more appealing way to address outdoor applications [Schall *et al.* 2013]. Different application scenarios can be thought of using 3D GIS by visualizing the 3D models on-site in real-time, overlaid on the real view, such as visualizing some future architectural projects, historical buildings, underground infrastructures, *etc.*

While retrieving data about the surrounding objects, the primary information that allows users to access location-based information sources in real-time is the *spatial component* of such data. For this, the data should be stored on the Web or on a local database with geo-location information (such as latitude, longitude and altitude coordinates or a street address, *etc.*), so that it can be queried with location cues. During the surroundings discovery, data can be queried by acquiring the location information of the mobile device. Mobile Augmented Reality deals with visualizing this location-based data and providing interaction mechanisms to the user. As the user moves around (*i.e.* as her location changes), the change of her position in real world affects the content of this layer dynamically. How this data is used, such as a Point of Interest (POI) service, navigation, location-based game, *etc.* depends on the application logic. Various applications can be built using the same location-based data with different application logics [Suomela 2006].

Augmented Reality offers many interesting research challenges. From a technological point of view, we can categorize these topics into three main research areas [Hugues *et al.* 2011]. Subjects related to 1) hardware technology, such as increased performance of embedded sensors; 2) software technology, such as image analysis, sensor data-merging algorithms, *etc.*; 3) data and access technology, *e.g.* ubiquitous computing, cloud computing, query methods and techniques, *etc.* Regarding data

and access technology, some of the questions that we try to answer in this thesis is "*how can the system determine which data is to be displayed (i.e. superimposed) ?*", "*where this data is coming from ?*" and "*to which extent this data is related to the user's preferences*". Organizing data related to the AR objects is one of the important steps to enable the system to decide upon what this data corresponds to (*e.g.* a description, an old image, an architectural information, *etc.*) and access it when needed. On the other hand, easily accessible and extensible data sources are more appealing for information retrieval in an AR application. In that sense, the Semantic Web seems to be the perfect source for content retrieval since it provides these kinds of mechanisms. These points of view will be discussed in detail in this dissertation.

## Problem Definition and Thesis Contribution

Nowadays, users want to access information by just touching the screen no more than a few times on their mobile devices [van Aart *et al.* 2010]. Therefore, mobile applications should not demand too complex operations while accessing a piece of information. If an AR application provides an excessive amount of information, the cognitive effort of the user would be demanding to pick the necessary information according to her interests (for instance, art museums, different architectural styles, shopping stores, restaurants, objects belong to a particular century, *etc.*). Information about a real-world object may be related to art, history, architecture, *etc.* or all of these at once. Proposing all this categorical information to the user may generate some complexity while accessing the information of her interests. In order to solve this problem, the displayed information should be formalized somehow, so that the user can find her way in a large amount of data space and harvest the one she is interested in. Therefore, if objects (or information related to these objects) can be categorized and displayed using AR mechanisms, the user would efficiently select and be provided with the information she looks for while visiting the place. The transformation of complex information into manageable sources is the ultimate goal for *information visualization* [Farman 2012].

With this objective in mind, our proposition includes an approach for conceptualizing the knowledge related to a real-world object (*e.g.* in terms of the role of the object or in terms of temporal components of knowledge related to it, *etc.*). Using these concepts, we aim to provide an access to multiple representations of this object, because a real-world object is an evolving entity in time and may have se-

veral representations and information related to these representations (geometrical, temporal, thematic *etc.*). Using this conceptualized knowledge base, we propose an architecture which can be used by various domain-specific mobile AR applications, for instance, cultural heritage, tourism, architecture, *etc.* The applications that are built using our architecture can provide the user with an access to these knowledge bases in a systematic way (by displaying information in separate categories and with user-friendly interaction mechanisms) and allow her to gather information about her surroundings depending on the matching between her interests and available knowledge categories.

In some situations, the user may not have an explicit demand for information from the AR application. In this case, the application is responsible for organizing the data according to her interests, instead of presenting everything available, so that the user can find her way among information.

AR applications may also be used for revealing the stories behind the objects that people may not have the time to access them on the Web, in the libraries, or by talking to experts related to these objects. A visitor of a place may seek for answers to the following questions about that place: "*What do I see ?*", "*What is the history ?*", "*Who is the architect ?*", "*What kind of events have taken place here ?*", *etc.* The answers to these questions may be held in databases somewhere on the Web. Hence, Augmented Reality provides interaction mechanisms for accessing these data sources, so that the user may better perceive her surroundings. AR improves her interaction with that space. One of the purposes of our proposition is to create mechanisms to access such information sources, and then enable AR applications to reveal stories about a place. Today, there are many examples of site-specific AR applications. However, these applications are developed to provide story-telling specific to one or a few selected places, but not for several sites.

At this point, we come to the conclusion that the semantic web technology is not fully explored to response these needs. For instance, linked data sources (discussed in Chapter 2) available on the Web can be used to explore information related to a real-world object on mobile devices. This can be achieved by interlinking 3D objects with the available information on the Linked Open Data Cloud (LOD). Interlinking means publishing these 3D geometry information on the Web by creating relations with the available linked data sources. This way, we can benefit from the semantic web technology for discovering information in an AR application, as it is designed to store, process and access data easily.

For this purpose, we propose a data model for transforming 3D models into semantic data sources which can be accessible over the Web. These 3D models will be overlaid on the captured scene to get an augmented reality view, thus allowing the user to access information by interacting with them. This data model helps conceptualizing these 3D models with thematic and temporal components, hence creating different representations of one model. We believe that in order to represent a 3D model that will be used in an AR application, temporal aspects are important to allow for a temporal navigation, showing the evolution of the 3D geometry of real-world object through time. The role of the object is also important to be able to group and add additional attributes. We also enrich the data model with a thematic aspect to provide thematic navigation (showing the change of roles of objects through time) for Augmented Reality. This way, the information related to the surrounding objects can be grouped according to their type of the objects and the application designer can create further experiences for these objects (we refer *AR experience* as the involvement/experience of the user with the mobile augmented reality application upon using it actively for a moment and completing some tasks that the application proposes). For instance, for the object type *castle*, while the 3D evolution of its towers, guardhouse or walls may be accessible through AR, for the object type *hospital*, its opening times, type or specialization, as well as when it was built or by whom can be made available to the user. Thus, we think that these three components, which are temporal, thematic and spatial (3D) information, are major cornerstones and indispensable for a data model that is designed to represent real-world objects on the Web and facilitate information access. This way, real-world objects can be located in space and time associated with their roles which will allow capturing of changes of an object. Spatial attributes can be used to understand how the object looks/looked like and where it is/was located; temporal attributes are used to know the time spans where the object has the same spatial and thematic attributes; and lastly, thematic attributes are used in representing semantics related to the role of the object. These attributes will also allow application designers to create various sorts of *AR experiences* by accessing the complete state of an object.

On the other hand, modeling spatial, temporal and thematic information about real-world objects is not enough to explain the dynamics of accessing these data. The Semantic Web, on the other hand, provides information access standards that allow the same 3D model and information attached to it to be accessed across different applications. In order to create this knowledge base and make it available

on the Semantic Web, we use ontologies, which help us to express various concepts with relations and attributes (such as the temporal and thematic knowledge related to a 3D object). Ontologies facilitate accessing, sharing and re-using some specific and formalized knowledge. Once the information is expressed as a knowledge base on the Web, it can be accessed systematically.

In this doctoral dissertation, we focus on the following topics:

1. the way of visualizing the surroundings using mobile AR;
2. the selection and retrieval of information contents from structured data sources on the Web;
3. the design of an architecture that can be used by various AR application scenarios;
4. the delivery of 3D AR content and their access by different applications as a structured data source on the Web.

## Outline

The first part of thesis report consists of a state of the art that helps to position and further understand our research questions and approach.

In Chapter 1 (*Surroundings Discovery Applications*), we discuss the meaning of *context* and its importance for providing relevant information to the user. We explain how it is captured using mobile devices and exploited to access location-based information. Several methods for visualizing location-based content are proposed in the literature, including 3D maps and Augmented Reality. We review these methods and point out the requirements to have an effective surroundings discovery application using the available mobile technology.

Chapter 2 (*Semantic Information in Mobile Applications*) analyses the evolution of the Web and introduces the notion of *semantic* information. In this chapter, we describe the *linked data* paradigm, which encompasses principles and methods to publish structured data on the Web using semantic web technologies. Then, we review mobile applications that exploit the Semantic Web and use linked data as their first source of information for surroundings discovery.

In the second part of the thesis, we present our approach and detail our proposition and implementations.

Chapter 3 (*Overview: A New Augmented Reality Approach*) introduces the overall idea and key elements of our proposition.

In Chapter 4 (*3D2T: A Data Model Mixing 3D, Thematic and Temporal Information*), we introduce the 3D data model that we propose to make 3D models available on the Web by associating them with linked data sources and temporal and thematic information to identify different representations of a real-world object.

Chapter 5 (*From LOD to Augmented Reality based Interaction*) discusses how this data model can be exploited by a mobile AR application for information retrieval. We also introduce different application modules that can be seen as software components to get through several tasks in an AR application, such as displaying 3D objects, providing interaction mechanisms to the user, managing the communication with information sources according to these interactions (in order to search data on the LOD cloud that is queried by the user), *etc.* Lastly, we describe the implementations that we have made in order to validate our approach.

Chapter 6 (*ARCAMA-3D*) presents our architecture which we implement with a use case scenario concerning a real-world edifice, the Royal Monastery of Brou, in France, before finally we conclude in the last chapter (see Conclusion).

Chapter 1

# Surroundings Discovery Applications

**Sommaire**


1.1 Introduction . . . . . **11**
1.2 Information processing: from sensing and querying to visualization . **14**
1.2.1 Sensing the World with Mobile Devices . . . . . 16
1.2.2 Location-based Content Retrieval . . . . . 18
1.2.3 Visualizing Location-based Content during Surroundings Discovery . . . . . 22
1.3 Augmented Reality . . . . . **33**
1.3.1 Mobile AR systems . . . . . 35
1.3.2 Tracking and Registration . . . . . 37
1.4 Conclusion . . . . . **40**


## 1.1 Introduction

With the increasing number of mobile devices, such as smartphones, tablets, PDAs and optical head-mounted displays (wearable glasses), context-awareness, location-based services (LBS) and ubiquitous systems have become popular keywords these days. The terms '*Ubiquitous computing*' or '*pervasive computing*' are proposed for the first time in 1991 by Mark Weiser as referring to the technology that seamlessly integrates itself into human environments where computers know the location and surroundings of their users in order to adapt their behavior in significant ways without requiring even a hint of artificial intelligence [Weiser 1991]. Weiser envisions computers as pushed into the background.

> *"The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." Mark Weiser, 1991.*

In this context of *invisibility*, Weiser expands his description of ubiquitous computing with a new field of computer science that can speculate on a physical world with its sensing capabilities, actuators, displays and computational elements, embedded seamlessly in the every object of our lives and connected through a continuous network [Weiser *et al.* 1999]. As in 2001, the results in this field are grouped into the following areas: *mobile networking*, *mobile information access*, *support for adaptive applications*, *system-level energy saving techniques* and *location sensitivity* in [Satyanarayanan 2001]. Today, these fields and Weiser's definition are still popular subjects and they are re-discovered with the capabilities of mobile devices with cameras, accelerometers, GPS and so on. Mobile computing has been a rapidly developing technology since then.

The ability of identifying user's location at a certain time is one of the most promising steps in developing applications that collect and capture user's context in order to deliver relevant information to a moving user. *Context*, as it is stated in [Paul 2012], from a general perspective, is about location and enriching the specifics of that particular space; and in a local perspective, *context* is about the ability to engage with that location. Therefore, one can claim that *context-awareness* is about an enhanced experience with a particular location in concert with awareness of self (*i.e.* preferences, profile, *etc.*). Today, context-aware systems are able to adapt their functioning to the current context of the user in order to increase their usability and effectiveness by taking into account the environment itself [Baldauf *et al.* 2007].

As the user's surroundings change rapidly while she moves around, mobile devices adapt to the changing location, time and other environmental elements rapidly. In this regard, location-aware mobile devices have enabled a wide range of LBS [Brush *et al.* 2010], such as vehicle monitoring, tourist guidance, emergency assistance, location-dependent advertising, friend-finder, points of interest services, *etc.* These services use contextual information to provide relevant services, adaptive user interfaces and personalized results [Mokbel & Levandoski 2009].

One of the most challenging issues in LBS is the *location-based queries*, which are considered as continuous queries [Ilarri *et al.* 2010]. The answer to these queries change as it depends on the geographical location where the query originates. Therefore, the semantics of a set of data in location-based queries are tightly coupled with a particular location [Dunham & Kumar 1998]. Numerous advances in semantic web technologies facilitate the systems to understand and respond to human requirements and context [Ilarri *et al.* 2011]. There has been a growing interest in

using semantics in LBS. Some of the early solutions were based on using ontologies for user's semantics [Yu *et al.* 2005], or an architecture that enriches user's preferences and location [Kim *et al.* 2005]. Since then, various applications with different scenarios have been developed by linking LBS with semantics. Some of these approaches are based on semantic annotations and descriptions of specific places for effective retrieval and recommendation of Points of Interests (PoIs) [Ye *et al.* 2011]; or semantic-based personalization in LBS by representing physical entities and people profiles with ontologies [Haav *et al.* 2009], [Viktoratos *et al.* 2014]. One of the drawbacks of these systems is that their semantically annotated data sources are discouraging their adaption and broad re-use by the Web community since they are not interlinked with other vocabularies. Thus, it is cumbersome to extend their knowledge base with other data sources, and make them work on various other devices for different LBS.

Some other recent studies are based on the rapid emergence of Web of Data, or *Linked Open Data* (LOD) (see Chapter 2 for more details on the subject), which enables transforming information into machine readable and semantically annotated structured data. By linking semantics of the contextual data models to standard vocabularies of LOD, richer concepts will be used to express the context related to a visitor on the move [Costabello *et al.* 2012], [Buján *et al.* 2013].

With this approach, more intelligent services have been enabled in LBS, such as using urban-related data in location-based gaming applications [Celino *et al.* 2012], using different data sources from the Web of Data to query the PoIs around the user [van Aart *et al.* 2010], or to deliver background information about a specific PoI [Schmachtenberg *et al.* 2014].

In parallel to these advancements, Mobile GIS, a term to describe the deployment of Geographical Information Systems on mobile devices, has been a rapidly growing research field. Mobile GIS and GPS technologies make it possible to improve the guiding process by providing users with advanced visual and graphic representations of their surroundings.

Satellites, street images and 2D views have been vastly used in mobile GIS, as in some widely known applications Google Maps [1], Bing Maps [2], ArcGIS [3], OpenLayers [4], *etc.* On the other side, mobile 3D GIS were mainly restricted by several hardware aspects of mobile devices (memory restrictions, slow processing speed,

1. http://www.maps.google.com/
2. http://www.bing.com/maps/
3. http://www.arcgis.com/
4. http://openlayers.org/

*etc.*) [Stoter & Zlatanova 2003]. Some of the first prototypes have used a 3D model for route instructions [Kray *et al.* 2003]. However, the advent of mobile graphics units (GPUs) in recent years has boosted the graphics capabilities of mobile devices [Noguera *et al.* 2012]. Mobile 3D maps with more features have been developed [Nurminen 2008], [Oulasvirta *et al.* 2009].

Thereafter, some research activities have focused on mobile ubiquitous applications using 3D GIS, such as recommender systems [Noguera *et al.* 2012], displaying PoIs on 3D maps [Trapp *et al.* 2011] and car navigation systems [Nedkov 2012], to name a few. Some of the applications have adopted the Augmented Reality (AR) to display the 3D data [Schall *et al.* 2013].

In the following of this chapter, we first introduce some representative works that are based on the processing of location-based information in order to visualize it on mobile devices. This process consists in gathering sensor information (*i.e.* physical location and orientation of the mobile device), as well as querying context related to the surroundings of the user. As we briefly mention in this introduction, the surroundings discovery mobile applications may vary in their use of semantics (ontologies), the Semantic Web or linked data sources. We introduce and examine each of these topics in detail in Chapter 2. In this chapter, we focus on the way of discovering information using mobile devices. Afterwards, we present the concepts and principles of Augmented Reality and illustrate how it has been used for surroundings discovery applications.

## 1.2 Information processing: from sensing and querying to visualization

With the growing computational power of mobile devices, applications designed for these environments take advantage of the context information to offer greater services to the mobile user. In order to use context effectively, we should understand what context is, how it can be used and what context to use in our applications [Dey 2001].

The definition of *context* depends also on the field of application. In [Schilit & Theimer 1994], authors define *context* as location, identities of nearby people and objects, and changes to these objects over time. [Chen & Kotz 2000] define it as a set of environmental states and settings that determines an application's behavior (active context) or in which an application event is interesting to the user (passive context). They also emphasize the importance of *time* as an index

to other sources of contextual information. According to [Joshua *et al.* 1998], the context is based on the situation where the user's actions occur. It will have different meanings as the same action takes place in another situation. [Verbert *et al.* 2012] propose a mobile recommender system and defines context as an aggregate of various categories that describe the setting in which the recommender is deployed (such as location, current activity and available time of the learner). In a Human Computer Interaction (HCI) design point of view, [Rodden *et al.* 1998] claim that the context needs to be reconsidered in terms of the interaction possibilities that a mobile device offers to its users. Depending on the design of this interactivity, the relationship of the device with the technical infrastructure, the application domain, the socio-technical system in which it is situated, the location of its use and its physical nature will differ, thus the meaning of context as well. [Hull *et al.* 1997] list location, time, identity, companions, computing resources and physical environment as context elements according to which a range of devices will be communicated in a ubiquitous computing infrastructure.

As it can be seen through these examples, the scope of the definitions of *context* changes from one to another. However, there are many similarities and the main idea remains the same: a context-aware mobile system aims to provide users with relevant information and services based on their current situation. With this point of view, [Dey & Abowd 1999] represent the main idea of context through this generic definition:

**Definition 1 (Context).** *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*

This definition of Dey is referenced extensively within various application domains. On the other hand, in order to better define the role of context in wearable computer systems, the author in [Pascoe 1998] identifies four generic context capabilities: *sensing*, *adaptation*, *resource discovery* and *augmentation*. According to this definition, a context aware system should sense the environmental states (such as location), adapt its behavior to integrate more seamlessly the user's environment, discover and exploit resources within the same context, and augment the environment with additional information by associating digital data with a particular context. Even though there are numerous definitions about what *context* is, the capabilities of context information defined by Pascoe are still valid.

Since our research aims at discovering surroundings using mobile devices in outdoor environments, *location* as a context information plays an important role in our approach. There have been many attempts to build context-aware systems that provide users with LBS (Location-based Services), such as personalized recommender systems [Rinner & Raubal 2004], [Savage *et al.* 2011], [Verbert *et al.* 2012], decision making systems [Espeter & Raubal 2009], navigation services [Saeedi *et al.* 2014], mobile tour guides [Cheverst *et al.* 1999], [Chou *et al.* 2005], [Baltrunas *et al.* 2011], *etc.*

In the following section, we examine previous works on how the location information is detected and served to provide users with relevant information, what kind of resources is queried and transmitted to the user through a mobile interface and lastly, how this information is visualized.

### 1.2.1 Sensing the World with Mobile Devices

Mobile devices nowadays include a variety of sensing components embedded in their hardware and most of the context can be collected using these sensors, such as cameras, accelerometer, gyroscope, GPS, microphone, Bluetooth, *etc.* For instance, iPhone4 includes eight different sensors: accelerometer, GPS, dual cameras, dual microphones, compass, gyroscope, ambient light and proximity sensors (a representation can be seen in Figure 1.1). Using their data, applications are able to answer some important questions that enable them to deduce the user context, such as *where is the user ?*, *what is the user doing ?*, *who is she with ?* [Yang 2009]. These heterogeneous sensor data can be complemented by software that retrieves information from the Web, location-based services or various data sources [Clear *et al.* 2010].

The GPS is used for positioning and navigation purposes. It triangulates its own position by getting precise microwave signals from at least three satellites. The result is provided in the form of a geographical position - latitude and longitude. Software applications can use these coordinates to provide walking or driving instructions. Accelerometer, gyroscope and digital compass compose the inertial measurement units of the mobile device. They are used to detect the translational and rotational movements which provide an improved position estimation of the device with direction and orientation detection. The accelerometer is used to sense the linear acceleration of the device and how it is tilted. The gyroscope uses the Earth's gravity and determine the orientation. The digital compass provides mobile

devices with an orientation related to the Earth's magnetic field, so that the device knows where north is. The built-in cameras are used for image recognition or to detect the lighting conditions of the surroundings or reading 2D barcodes to get information about a location. These built-in sensors are very often used in mobile applications to improve the user's perception of her surroundings.

For instance in *Laκsa* project [Lim & Dey 2011], researchers have designed an intelligent social awareness application that uses mobile sensors (GPS, Wi-fi connection, accelerometer and microphone) to infer the context of the user, such as where the user is (home, office, library, café, *etc.*), what she is doing (walking, cycling, placing the phone flat, *etc.*) and sound activities (listening to music, talking, *etc.*). In *Geopointing* project, the location-based application exploits the orientation information to allow users to access information by physically pointing their device towards objects in real world [Simon *et al.* 2008].

**Sensing the Location and Orientation** Location is the most important context information especially in outdoor mobile surroundings discovery applications. The term "mobile" makes us think of changing location and these changes in location can affect the system [Rodden *et al.* 1998]. User's location change is the change of user's situation, such as her activities and surrounding environment which may affect the user's information needs [Iwata *et al.* 2011].

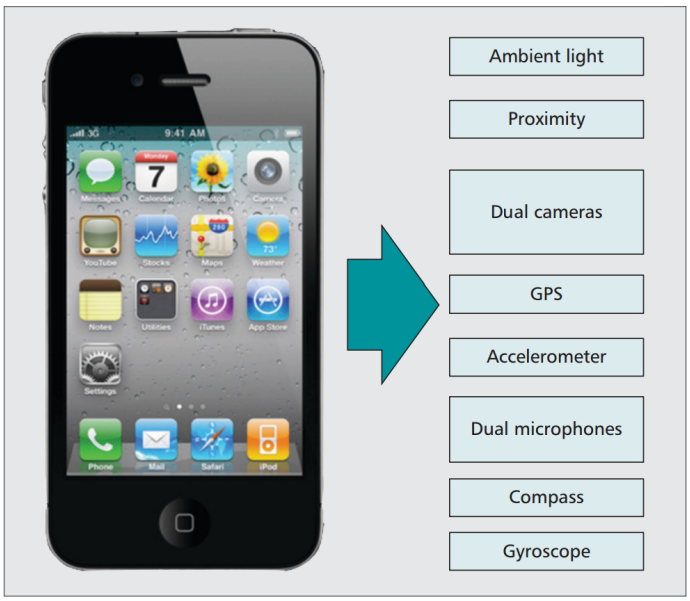



Figure 1.1 – A representation of the growing class of sensor-enabled phones: iPhone 4 (from [Lane *et al.* 2010]).

The key design of providing location-aware information or interaction with the environment using the mobile screen depends on the degree to which the mobile application is coupled with the location and orientation data [Rodden *et al.* 1998]. In the Guide project, for instance, the mobile unit receives location messages from wireless access points in order to determine the location of the device [Cheverst *et al.* 2000]. In [Christoph *et al.* 2010], the authors state that by using GPS, the position of the mobile user can be detected and additional information related to user context, *e.g.* street name, building name or name of a point of interest can be obtained.

Although, positioning technologies has been widely used, building a system with a seamless navigation is still an open issue. Many researches have been adapting fusioning of two or more sensor data to provide more accurate and pervasive location and orientation information [Samadzadegan & Saeedi 2005]. Fusioning the data of two or more sensors means combining information made available by various knowledge sources, such as sensors, to provide better judgement. Thus, fused information is richer and more accurate than data obtained from one data source [Saeedi *et al.* 2014]. In capturing context of the user, several mobile sensor data, such as location data using GPS and acceleration using accelerometer or direction of movement using gyroscope, are fused using mathematical algorithms in order to construct the movement of the user in 3D space. Kalman filter [5] [Kalman 1960], a set of mathematical equations that minimizes the errors by estimating the past, present and future states of the data, is often used to combine GPS and inertial sensor measurements to provide high accuracy, such as in [Schall *et al.* 2009] and [Retscher 2005]. It is often used to eliminate the noise resulting from sensors. In [Schall *et al.* 2009], the location and orientation detection is also supported by the visual tracking.

### 1.2.2 Location-based Content Retrieval

Mobile devices allow users to access information on-the-move. They allow them to search location-based contents (such as information about landmarks around them, reviews about some places or ways to get there, *etc.*) anytime and anywhere. Wireless access with 3G/4G technologies or wireless networks has eliminated the geographical limitations and mobile devices are commonly used by the general public today. [Church & Smyth 2009] state that 75% of the geographical searches

5. see [Welch & Bishop 1995] for more information

tend to happen when the user is moving. Thus, they conclude that location is a key factor in prompting information needs in a mobile situation where location-based services and local search can be very fruitful.

Using the context, mobile applications are able to acquire the information that best suits the mobile user. For example, if a mobile guide is designed to give information about the surroundings of the user, the location and orientation information can be used as *context* to provide *content* to the user. This application scenario might be enriched using other software support.

In one of the early location-based mobile information retrieval examples, the GUIDE project, wireless communication infrastructures are used to broadcast dynamic information and current location of the user to portable GUIDE units on a customized Web browser [Cheverst *et al.* 2000].

Even though mobile information retrieval is widely used, it has been suffering from three major challenges, as it is stated in [Lee *et al.* 2009b]:

1. Mobile devices have a small display area. Thus, users are more interested in precision than in recall.
2. Mobile device portability means users frequently change their interest with location.
3. Though wireless development is progressing; its response rate and mobile device processing ability are still lower than for a wired network and PC.

Facing these difficulties, some research activities have focused on the query processing to retrieve the most relevant location-based information. A *location-based query* is a query whose answer depends on the location of the object, where *location* is described by an attribute that is used in order to determine if a particular information (*e.g.* an object) is a part of the answer or not [Ilarri *et al.* 2010], whereas a *traditional query* is a query whose answer does not depend on the location.

In [Ilarri *et al.* 2010], location-based queries are classified according to the following criteria:

- the evaluation time and past states;
- whether they refer to past, present, or future states;
- the way uncertain locations must be handled;
- their semantics regarding a certain time interval;
- the dynamics of the query.

In some of these approaches, the user interests/preferences or the degree-of-interest of the objects around the user affect the query procedure. For instance, in

[Iwata *et al.* 2011], the system decides the importance of each location-based category (whether it is useful for the user or not). Only the high-importance categories are displayed on the menus and maps. Then, user makes her selections according to her interests.

> *"The information that clients are actually interested in is a subset of all the location information that might be available to them." [Schilit & Theimer 1994]*

One important point in location-based content searches is that the changing location of the user may affect the content that can be queried using location cues [Lee *et al.* 2009b], as well as the user interests may be altered as the content is changing. Therefore *continued sensing* is a basic requirement in mobile context-aware systems [Priyantha *et al.* 2011]. As authors in [Ilarri *et al.* 2006] mention, the queries in mobile contexts are continuous queries since the set of objects that satisfy the conditions may vary due to the changing location of the user (in their proposition, the set of objects may be moving as well). Updating the answer continuously at a certain frequency, as in [Tao *et al.* 2003], may overload the wireless mobile devices with query processing. Thus, they propose using location-dependent queries where only the set of interesting objects within a certain diameter are queried for certain attributes.

In [Baltrunas *et al.* 2012], Points of Interests (PoIs) were aggregated into categories in order to avoid the sparseness of the collected data. They define 10 categories for the PoIs, *e.g.* castles, nature wonder, church or monastery, museum, musical event, theatre event, *etc.* Then, the application proposes these categories to the user to gather her preferences according to her selections. It calculates the relevance of the objects in her surroundings with a complex calculation using many context factors, such as distance to PoI, crowdedness, weather, budget, season, knowledge of surroundings, time availability *etc.* Even though their approach seem novel in the sense that suggestions are given according to a relevance degree (see Figure 1.2 (b)), the context used for relevance calculations seem so vast that the user has to complete a survey to define the context among a long list of contextual factors (see Figure 1.2 (a)). The user is also constantly notified in the case of a context change, (see Figure 1.2 (c)) which may cause an information overload on the user side. Also, the list of categories (object types) are very few, which makes the usability of their proposal poor. Despite these drawbacks, their proposition is interesting in terms of

using categories (castles, church, monastery, *etc.*) and allowing the user to select among them according to her interests, as well as automatically retrieving PoIs based on these selections and sending information about them to the user.

mSpace mobile [Wilson *et al.* 2005] is one of the early examples that use the Semantic Web to support the exploration of location-based information resources. It is a London city guide with connections to various domains or sources of interests (movies, music, *etc.*). Using the location information of the user, the application queries different knowledge bases (IMDB (a movie database), video archives of the BBC, *etc.*) and displays the results in an information box with textual descriptions. Although, its user interface does not allow a fast and efficient exploration, it remains interesting for the use and exploration of multiple interconnected knowledge databases. [Liagouris *et al.* 2011] propose *task computing* using ontologies, in order to deliver rich content according to user's needs. They use an ontological task model that they have designed. Using the concepts of this ontology (*i.e.* vocabulary and additional knowledge), the user can accomplish basic tasks such as finding a theater, a museum or a restaurant. The mobile application provides user with general tasks

Figure 1.2 – POI recommendations according to context [Baltrunas *et al.* 2012]: (a) list of all the contextual factors shown to user; (b) recommendations according to the user's preferences; (c) user's notification in the situation of a context change.

such as “find a museum”, “find an attraction”, “find an event/activity”, which are also divided into subtasks (see Figure 1.3). The information is stored in a database which can also be used as an offline source of content in the mobile device. Using the Semantic Web and ontologies in ubiquitous solutions facilitates the classification of resources, reuse of knowledge and access to various data sources. We introduce in detail the Semantic Web and Ontologies in the next chapter (Chapter 2).

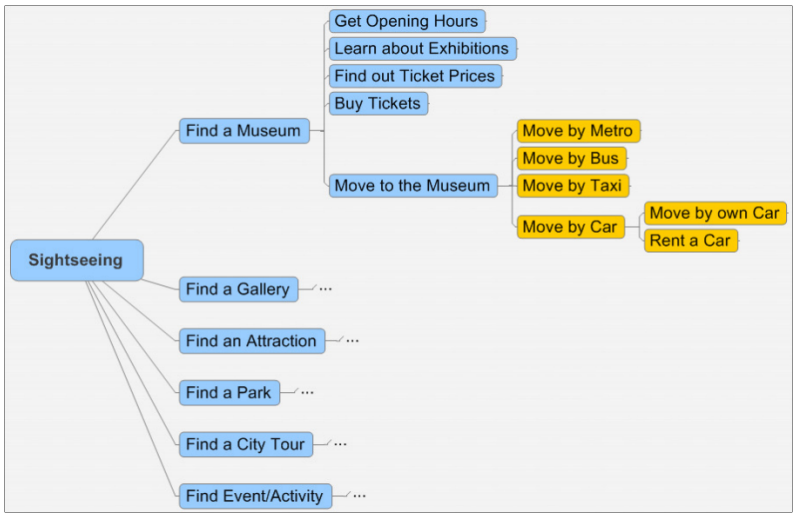



Figure 1.3 – The tasks and sub-tasks that are proposed to the user in [Liagouris *et al.* 2011].

### 1.2.3 Visualizing Location-based Content during Surroundings Discovery

With the developments in GPS services, as well as the graphic and memory cards of mobile devices, displaying location-based information over a map of the surroundings has become a popular mode of visualization. Real-time monitoring of GIS data on these devices, such as PDAs and smartphones, has created a market and a research area on location-based services and context-awareness. Since then, pervasive computing systems allow not only thin clients, which are PCs and computer programs, to interact with the Web servers, but also other devices, such as mobile phones, networked vehicles, personal digital assistances, *etc.* [Division 2000]. Later on, the advancements in map-based technologies have opened up new cartographic possibilities such as 3D maps.

**Emergence of Digital Mobile Maps** Information technology has expanded the potentials of mobile electronic maps. They offer advantages by displaying user’s

location on the map, zooming as well as linking with location-based services and personalizing using profile information [Kaasinen 2003]. There have been many research studies from all directions to improve presenting information over a background of digital maps on the mobile device. Adaptive map rendering and dynamic generation of maps are part of a vivid research field.

Early propositions focus on taking into account user context, profile and preferences in order to adapt the digital map accordingly. Such as in [Reichenbacher 2001], the authors propose a conceptual framework based on recommending spatial items on digital mobile maps by exploiting user interests and preferences. In another early work, [Baus *et al.* 2002] describe a navigation system that adapts the presentation of route directions according to the location of the user and the characteristics of the mobile device. In LoL@ (Local Location Assistant) project [Pospischil *et al.* 2002], researchers propose a mobile tourist guide for the city of Vienna. The application provides a map of the city in two zoom levels (overview and detail), plus textual and multimedia information screens. The CRUMPET system offers a simple user interface and interactive maps highlighting the sites of interest according to the preferences of the user and it notifies user with tips about the nearby sites [Belz *et al.* 2003] (see Figure 1.4(a)). In a more recent work, the RecoMap project [Ballatore *et al.* 2010], in order to avoid spatial information overload, the authors propose to deduce user interests by monitoring user interaction and context to provide personalized spatial recommendations. The PoIs are represented as complex geometries (*polygons*) and stored in a spatial database (see Figure 1.4(b)). Some of them are associated with relevant resources from the Web. An interest score is assigned to each of them according to the current interest of the user (*i.e.* previous interactions) and proximity. The layered approach proposed here is interesting since it intends to minimize the information overload on users. However, the semantic layer which holds the PoIs in a spatial database is not elaborated. The authors does not explain how the Web resources are associated with these PoIs and used in defining the role of these real-world items.

Besides the spatial information of PoIs, some applications are designed to display temporal information, such as information about the PoI that is associated with a certain instant or a time interval and duration. In the STEVIE application, a single time slider widget is placed at the lower part of the screen [Schmeiß *et al.* 2010] (see Figure 1.4(c)). It allows the user to select a time interval and displays the events and PoIs within the selected dates or periods. Users also can create, modify or delete

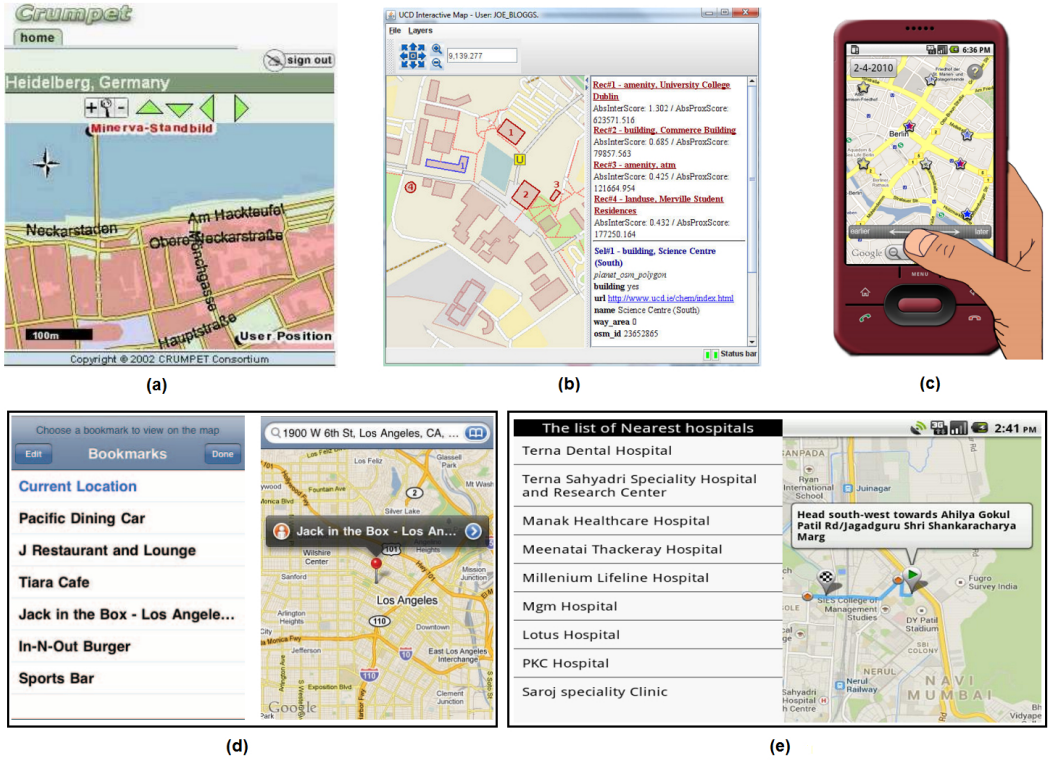



Figure 1.4 – Different 2D mobile map examples: (a) The CRUMPET system highlights the nearby interesting sites [Belz *et al.* 2003]; (b) The RecoMap monitors user interests to provide personalized recommendations [Ballatore *et al.* 2010]; (c) The STEVIE application offers time-based PoIs [Schmeiß *et al.* 2010]; (d) In [Ashikaga *et al.* 2011], users can share their maps and bookmarks; (e) [Mate *et al.* 2014], a database stores some basic information related to PoIs and the system provides a list of nearby PoIs to user.

PoIs and events in a Wikipedia style manner (*i.e.* collaborative).

In [Ashikaga *et al.* 2011], collaborative use of mobile maps is examined. In the proposed mobile application, users can bookmark locations on the map and share their map and bookmarks view with other users (see Figure 1.4(d)). Then, users can move to a certain point on the map, based on the information obtained from other users. In another map-based visualization approach, the current location and direction of the user is used as a primary contextual parameter to automatically adjust visualization (*e.g.* maps, arrows, directions, *etc.*) [Schmidt 2013]. In [Mate *et al.* 2014], the proposed mobile system searches for the nearby attractions (*e.g.* schools, colleges, hotels, hospitals, *etc.*) and displays a list of these PoIs, which are within a specific distance, as a list (see Figure 1.4(e)). The device continuously queries as the user is moving in her environment. In their approach, a database server stores the information about various places, such as their latitude, longitude,

address and name. The application also provides a navigation path from the current location of the user to selected destination. They use Google Map [6] as a background to display all this information.

Mobile map-based visualization is not only limited to projects of research institutions. There are also commonly used in commercial applications, such as Foursquare [7]. Foursquare is a location-based social networking service which permits its users to *check-in* some places. The user would manually tell the application about her current location with the help of the GPS of the mobile device. This is achieved by either text messaging, interacting with a mobile version of the Foursquare website, or using a mobile application. Since May 2011, Foursquare offers place recommendations and displays them on a map, considering user's location, check-in history, user's friends' check-in history and popular sites near the user. Map-based visualization has become a part of everyday lives of mobile device users with Web-based mapping applications like Google Maps, Yahoo Bing Maps [8], OpenStreetMap [9], Nokia Here [10], Apple Maps [11] (see the two Web-based map applications in Figure 1.5). In another Web-based approach, DBpedia Mobile [Becker & Bizer 2008], users can explore structured information extracted from Wikipedia on a map view. PoIs displayed on the map are Wikipedia pages of the real-world entities. Designed for the exploration of a touristic place, it displays information on a map about the nearby locations that can be found in the DBpedia dataset. The user can then explore information associated with these locations through links between DBpedia and other data sources (for instance, the associated Wikipedia pages). It also allows users to publish their location, photos and comments on the Linked Data. We present DBpedia in detail in Chapter 2.

In the examples above, we observe that 2D maps are commonly used in surroundings discovery applications. LBS (Location-based Services) are extensively used in these applications in order to show the user what is interesting within a specific distance on the map and navigate her towards an interesting object. With the technological advances, 3D maps visualization has become an easy task on mobile devices. Since 3D representations provide a realistic view of the surroundings, interpreting the scene has become easier for the user, contrary to the 2D maps

6. https://www.google.fr/mobile/maps/
7. https://www.foursquare.com/
8. http://www.bing.com/maps/
9. http://www.osmand.net/
10. http://www.here.com/
11. https://www.apple.com/ios/maps/

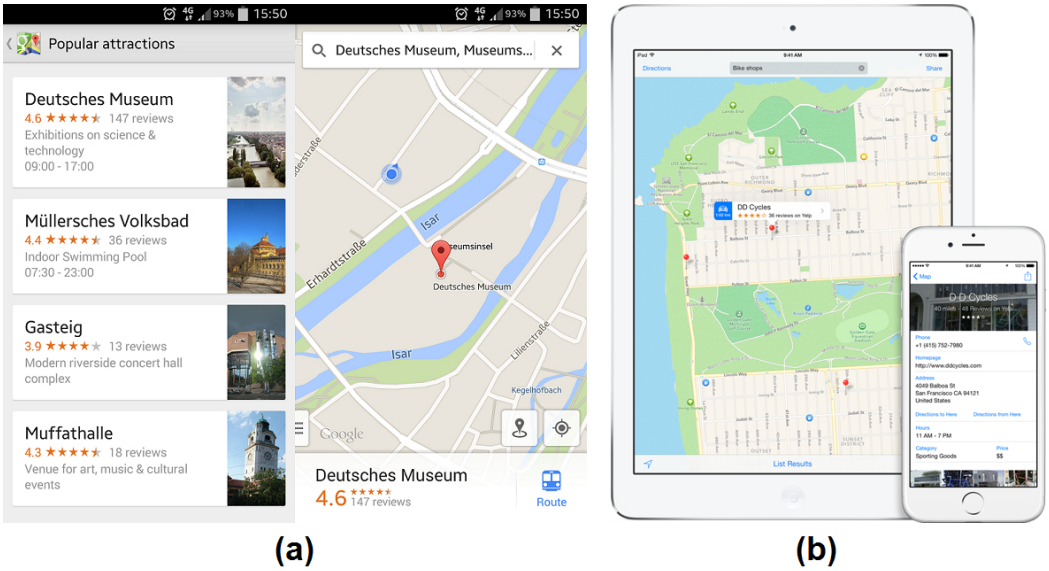



Figure 1.5 – Points of Interest visualization on mobile devices using (a) Google Maps; (b) Apple Maps.

[Rakkolainen *et al.* 2000]. In the following part, we discuss the use of cartography on mobile devices for surroundings discovery.

**A Cartographic Renaissance: 3D Mobile Maps** The TellMaris project, developed at Nokia Research Center, is one of the first prototypes that use a 3D model rather than a symbolic 2D map for route instructions on mobile devices [Schilling *et al.* 2003] (see Figure 1.6(a)). Information displayed on the map is modified according to the user's location, orientation, speed, acceleration and knowledge of the environment. The application presents both 2D and 3D maps on mobile screen to provide easier orientation for tourists. The authors also emphasize the risks of adding to much text, which increases the cognitive load of the user, as well as the importance of turning the map according to the user's orientation (*i.e.* the direction in which the user is holding her mobile device) for better perception.

Another preliminary work, the m-Loma project [Nurminen 2006], focuses on creating a 3D city model with a rich set of location-based information provided through *push and pull* functionalities (see Figure 1.6(b)). This system renders textured 3D models in order to represent a photorealistic city, but not much information is elaborated about the push-pull model. Authors state that a database holds touristic information, 3D data and user data, but give no detail about how this information is accessed, linked with the objects or visualized. However, it is a signi-

ficant early approach about accessing information using 3D models. In [Burigat & Chittaro 2005], the LAMP3D system for the location-aware presentation of VRML content (a 3D file format) on mobile devices is introduced (see Figure 1.6(c)). The relevance of this approach lies in the fact that the user points her device towards the real-world object and the application uses position and orientation information in order to render the 3D model of the objects that she is looking at (*i.e.* the object(s) that are captured by the device). Then, an interaction with the objects using a stylus (a small pen-shaped instrument used to tap on the touch screen in order to send input commands to the device) allows users to get some information.

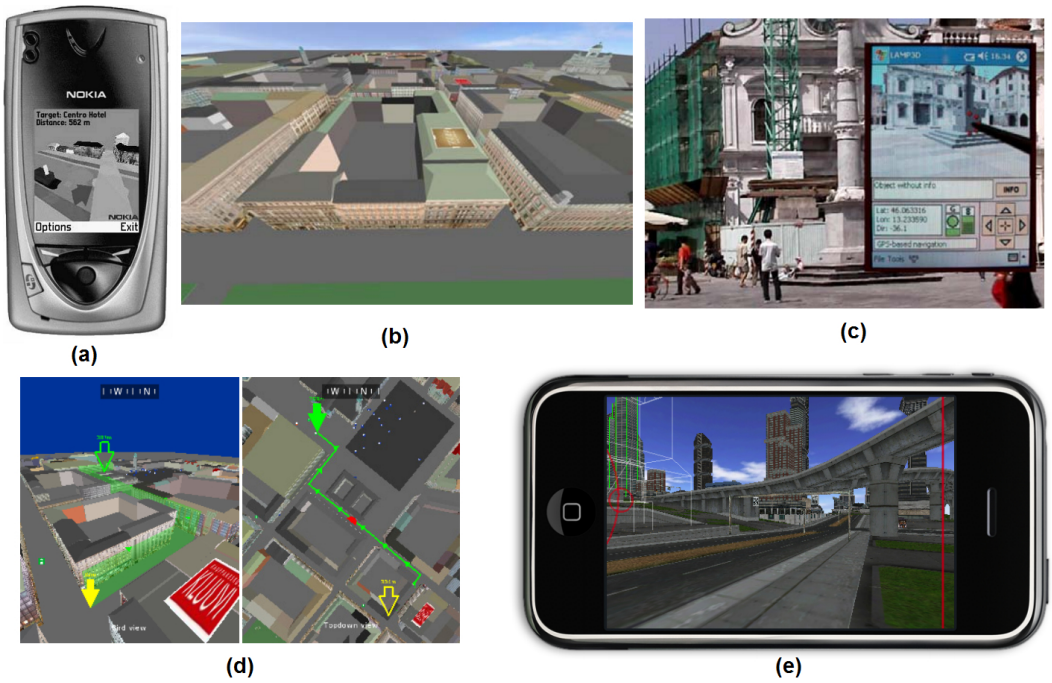



Figure 1.6 – 3D maps on mobile devices for surroundings discovery: (a) TellMaris tourist information system [Schilling *et al.* 2003]; (b) m-Loma project, real-time 3D map running on a mobile device [Nurminen 2006]; (c) LAMP3D system for the location-aware presentation of VRML content on mobile devices [Burigat & Chittaro 2005]; (d) a 3D map with four degrees of freedom [Nurminen & Oulasvirta 2008]; (e) PoIs visibility is maintained by rendering object as wire-frames in [Trapp *et al.* 2011].

3D models provide novel means to visualize and organize the spatial content [Fröhlich *et al.* 2011]. New remote sensors allow highly detailed 3D topographic mapping with sub-meter accuracy. Some of the common methods used for automatic generation of urban 3D models are aerial laser scanning (also referred to as LIDAR or mobile mapping) and aerial photography [Früh & Zakhor 2003]. The method of creation depends largely on the area of implementation of these 3D models.

They can also be generated using other methods such as image-based modelling via passive sensors (using imagery), using architectural plans, terrestrial laser scanning, or using unmanned aerial vehicles *etc.* [Jazayeri *et al.* 2013]. A number of researches propose to generate 3D models by combining some of these methods, since multiple data fusion from a number of sources often enhances the reliability of the data acquisition process. A combination of laser scanning and image-based modelling (*e.g.* [Grussenmeyer *et al.* 2008]) or an integration of LIDAR and satellite images (*e.g.* [Chen *et al.* 2009]) are examples of these approaches. These high resolution point clouds can be converted to wire-frame models and textures by using specific functions and programs, which then can be displayed on mobile devices as the models become more light-weight files.

In addition to this high quality but costly sensor data, collaborative Web 2.0 projects (*e.g.* wikis or social networks) have contributed in collecting freely available user-generated geographical content, such as OpenStreetMap (OSM) [Höfle 2012]. Generating 3D models has become much easier using these contents with the help of the recent tools and their increasing accessibility. Early mobile 3D experiments were limited by computational resources. However, today, mobile devices have improved a greater computational power and memory storage, while graphics hardware has made it easier to render 3D geometries in real-time. One of the 3D model developing tools is Sketchup [12]. It makes 3D object modeling possible even for non-professionals and provides plugins for generating 3D geometries in more interoperable file formats (*e.g.* CityGML). OSM-based 3D City Models can be easily generated using the polygon data of the buildings [Neubauer *et al.* 2009]. This facilitates the automatic 3D generation of large quantities (districts, cities, *etc.*) by extracting polygon data from OSM and extruding them into 3D using a modeling software like Sketchup. OSM-3D [13] is a project that aims to generate web-based 3D city models from OpenStreetMap. Geometries are generated from the available information, *i.e.* crowd-sourced OSM geodata (*e.g.* key-value attribution *etc.*), such as building height, building roof shape, building level, building façade color, building roof material, *etc.* [Goetz & Zipf. 2012]. It is also possible to generate buildings in various levels of detail and OSM ensures interoperability by taking into account the existing open web service standards of the Open Spatial Consortium (OGC), *e.g.* by generating geometries in CityGML file format [Over *et al.* 2010]. An exemplary view of a 3D city model generated using this method can be seen in Figure 1.7.

12. http://www.sketchup.com/
13. http://www.osm-3d.org/home.en.htm

Figure 1.7 – The 3D city model of Munich generated by the OSM-3D project [Goetz & Zipf. 2012].

The step into the third dimension adds value by providing realistic representations of the surroundings [Schilling *et al.* 2005]. Companies like Google, Nokia, Yahoo or Apple have also integrated 3D city models into their map services. On the other hand, 3D data are maintained in numerous formats, in different systems with various representations of the geometry and they also lack semantic notions (*i.e.* common definitions of objects, attributes, and their interrelationships with respect to a specific domain, *etc.*) [Kolbe *et al.* 2005]. Therefore, interoperability in accessing and using these models is often very difficult. In order to facilitate the exchange of 3D models, OGC has introduced the CityGML [14] open data model, an XML-based format for 3D city models. CityGML represents semantics, geometry, topology and appearance of 3D city models. It defines five Levels of Detail (LoD) [15] in the representation of 3D model: LoD0 is a two and a half dimensional digital terrain model; LoD1 is the well-known blocks model without any roof structures; LoD2 has distinctive roof structures and larger building installations like balconies and stairs; LoD3 denotes architectural models with detailed wall and roof structures, doors and windows; LoD4 completes a LoD3 model by adding interior structures like rooms, stairs, and furniture (see Figure 1.8). On the semantic level, real-world entities are represented with attributes, relations and hierarchies (part-whole relations), such as building parts, boundary surfaces (surfaces, roofs, *etc.*), building installations (dormers, stairs, *etc.*), openings (doors, windows, *etc.*), rooms and furniture [Stad-

14. http://www.citygml.org/
15. in order to avoid confusion, we use the abbreviation LoD for "Level of Detail", and LOD for Linked Open Data in the document.

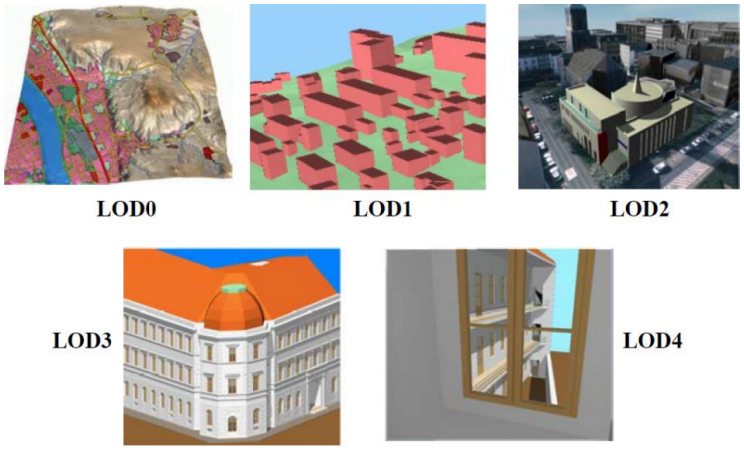



Figure 1.8 – Levels-of-detail introduced by CityGML to represent 3D city models [Kolbe *et al.* 2005].

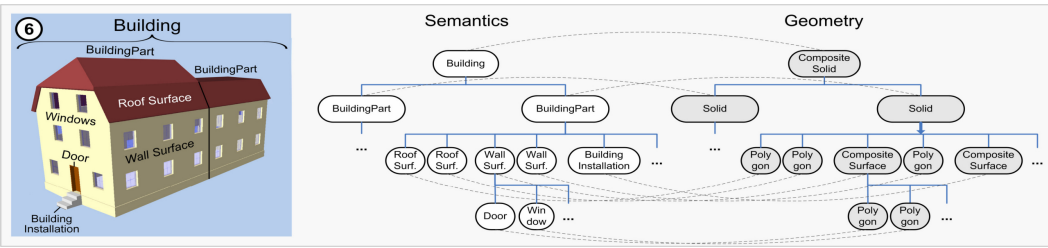



Figure 1.9 – Complex object with fully coherent spatio-semantic structure [Stadler & Kolbe 2007].

ler & Kolbe 2007] (see Figure 1.9). Since the introduction of CityGML, researchers have worked on the visualization of these models on mobile devices according to the user's viewpoint (*i.e.* the orientation in which the user holds her mobile device). Some of these works are based on using plugins installed on a web browser as in [Willmes *et al.* 2010]; or without a plugin, by exporting CityGML models towards a more efficient file format (such as X3D [16]) and visualizing them through browsers using WebGL [17] technology as introduced in [Prieto & Izkara 2012].

[Fröhlich *et al.* 2011] explore mobile-spatial interaction and show target (object) selection as one of the core challenges since users sometimes have trouble to recognize to which physical object a digital object refers. Way-finding is another challenge according to the authors, since when physically moving to a target; users should be continuously updated with available route choices. Other obstacles are

16. X3D is a 3D file format that represents 3D scenes and objects using XML. http://www.web3d.org/x3d/what-x3d
17. WebGL is an open standard software library that uses JavaScript to generate interactive 3D graphics on any compatible browser without plugins. https://www.khronos.org/webgl/

listed as performance limitations and personalization of content.

According to [Döllner *et al.* 2006], the type of information that is displayed on the virtual 3D model depends on the purpose of the application. For instance, a real-estate portal may want to see the vacancy, year-of construction and monthly rent of buildings within the virtual 3D city model using different color and texture indicators on the buildings. Some other researchers focused on pedestrian navigation support in urban environments [Nurminen 2008], [Bogdahn & Coors 2009]. [Bogdahn & Coors 2009] divide the digital urban space into '*walkable*', '*semi-walkable*' and '*non-walkable*' spaces to facilitate the guidance for pedestrian navigation. They propose providing 3D city models with a landmark-based approach claiming that the user can compare the real world with the 3D representation and determine her position accordingly.

In a user experiment conducted by [Nurminen & Oulasvirta 2008], a 3D map with four degrees of freedom (forward, backward, rotate left and rotate right), running on a PDA, is provided to the user (see Figure 1.6(d)). The user is asked to navigate with this map to find a landmark in the environment. For this simple task, the user performs many rotations and forward movements on the map by interacting with it. This example illustrates that four degrees of freedom are too much for the majority of users who are trying to navigate in this case study. Also, the time that the user spends interacting with the 3D map avoids her to observe the environment, which is a drawback for surroundings discovery applications. In a study where cognitive resources of a mobile user are examined, the researchers observe that users' capability to invest uninterrupted attention to a mobile device is limited [Oulasvirta *et al.* 2005]. They conclude that the interface should display whatever is currently needed as quickly and easily as possible, without complicated manoeuvring that requires all of the user's attention. Therefore, mobile applications that require a cognitive effort and complicated operations might be cumbersome for a user who has time constraints and less attention.

There have been some research projects conducted to compare the usability of 2D with 3D maps. According to [Rakkolainen *et al.* 2000], photorealistic 3D model of the environment is more recognizable than in 2D maps. Therefore, 3D representation of a real place makes it easier to perceive the proportions and distances, and to recognize landmarks. According to the CityGML developers, the visualization of 3D models facilitates the presentation of information in a more understandable way than on a 2D map, such as touristic information or routes [Prieto & Izkara 2012].

However, there is no widely agreed design approach in accessing location-based information using 3D maps. According to [Trapp *et al.* 2011], it is unnecessary to give too much detail about the scene to the user since it is overloading her cognitive resources. They introduce a PoI visualization concept to maintain the visibility of the objects. If, for instance, the 3D object occludes another landmark in the scene, it is rendered as a wire-frame (see Figure 1.6(e)). In another approach, [Coors & Zipf 2007] also discuss how much realism is needed when using 3D landmarks to allow navigation support. Their approach adopts high-detailed representations for landmark buildings, and not detailed photorealistic representations for all the other buildings with some visual similarity. [Klippel & Winter 2005] and [Nurminen & Oulasvirta 2008] mention that if the user can perceive her position with respect to a landmark on the 3D map, this will facilitate the exploration of the environment.

**Augmented Reality for Surroundings Discovery** Another interesting visualization method that helps users to experience their surroundings is the Augmented Reality (AR) technology. In one of the early works [Pascoe 1998], Pascoe regards the Augmented Reality as contextual augmentations that are achieved by associating digital data with a particular context to which it is related, such as a flood message attached to a location. He also gives the example of Cyberguide [Abowd *et al.* 1997], a mobile context-aware tour guide that uses vision techniques to inform the user about what she is looking at. In another early approach, [Feiner *et al.* 1997], Feiner and his colleagues introduce *touring machine*, a prototype of a 3D mobile augmented reality system that displays information (in text form) about a campus. Even though the system is limited by hardware and software resources (a combination of displays and interaction devices with a PC carried in a backpack is used without a precise registration of the 3D graphics), it has been one of the promising approaches for future use of AR technology. These preliminary works indicate the interest in using AR on mobile devices, even though it was not possible to implement them using the available mobile technologies back then.

According to [Meng 2008], the ideal geo-information representation should possess the largest adaptability and least interaction. AR facilitates visualization, perception, way finding and interacting with the environment. It is used not only with 3D models of the environment, but also with other approaches. We examine all these approaches in a separate section in the following.

## 1.3 Augmented Reality

Augmented Reality (AR) technologies have become very popular in recent years, as a consequence of the developments in handheld device technologies. Since its introduction by a Boeing's researcher Thomas Preston Caudell in 1992, "Augmented Reality", has grown attraction in technology and media domains. AR relates to enriching, but not replacing, the real world with virtual objects in order to improve the perception of the user. The computer-aided graphics are added as an additional content and information layer in order to interact and understand the immediate surroundings through the mobile screen.

On the other hand, GIS offer an abundant source of digital information that can be exploited by AR [Hugues *et al.* 2011]. GIS present spatially referenced data which can be easily understandable using visualisation techniques of AR. For now, several years since the late 90s, wearable mobile devices have been an interest for mobile GIS community, such as in [Coors & Wiedmann 1998], [Feiner *et al.* 1997]. In [Coors & Wiedmann 1998], authors foresee the future of wearable computers like *smart horizons* that allow the user to look beyond her field of view; magic wands which point devices to identify geographic objects; smart glasses that facilitate the AR by superimposing a digital image into the field of view. All these predicted technologies are available today. Nowadays, smartglasses [18] are entering the market, such as Microsoft HoloLens [19] or Google Glass [20] (see Figure 1.10).

Milgram has introduced the well-known *reality-virtuality continuum* which defines a taxonomy for differentiating the techniques that are used to visualize spatial information in Mixed Reality (see Figure 1.11) [Milgram & Kishino 1994]. According to this taxonomy, if the world is completely modeled, then the method belongs to the Virtual Reality (VR) side of this continuum. A real scene is a completely unmodeled environment. If the 3D model is superimposed on a real scene, then it belongs to the Augmented Reality side.

> *By extending this vision (ubiquitous computing) to include small, lightweight, comfortable see-through (and hear-through) wearable displays, we can craft a personalized (and, where appropriate, private) augmented reality whose user interface is not restricted to the displays and interaction devices embedded in the surrounding world or held in the user's hands. [Fei-*

18. optical head-mounted and internet enabled displays
19. http://www.microsoft.com/microsoft-hololens/en-us
20. https://www.google.com/glass/start/

*ner et al. 1993]*

Based on this continuum, we can say that most of the preliminary 3D location data visualization studies that we have introduced in the previous section (Section 1.2.3) have focused on virtuality, where the surroundings are discovered through fully modeled 3D scenes, such as in m-Loma [Nurminen 2006]; or on Augmented Virtuality (AV), where the camera position and orientation in the 3D scene is constantly updated on the camera view according to the location of the user in real world as in the LAMP3D project [Burigat & Chittaro 2005]. However, the fact that the user is *moving* is often ignored. As we state previously in this chapter, a user who is moving around with her mobile device is often occupied to discover her environment and her attention span might be very short.

In [Iwata *et al.* 2011], authors introduce some design policies for surroundings discovery systems with mobile devices since users tend to have low attention in these conditions: *Easy to use*; *A small number of operations*; *Narrowing of search scope based on context*. Augmented Reality seems to remarkably provide these design aspects. AR is easy to use since information is seamlessly overlaid on the real-world view and this enables users to pay sufficient attention to their environments. Also, AR interfaces reduce cognitive effort of the user by linking information with the objects around her [Hile *et al.* 2010]. In low attention situations, users are likely looking for immediate answers and AR can narrow down the search scope according to the user's context (such as the direction in which the user is moving and her viewport (*i.e.* the objects seen through the mobile screen)).

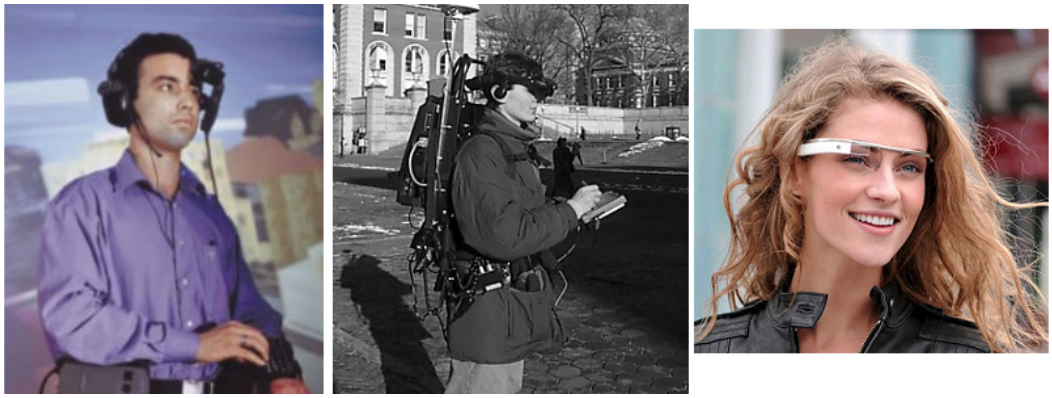


Figure 1.10 – Head-mounted displays in late 90s are a combination of displays and interaction devices that are tied to the head and the body: *on the left*: [Coors & Wiedmann 1998]; *in the middle*: [Feiner *et al.* 1997]. *On the right*: Google Glass, A new trend of head-mounted display with integrated sensors.

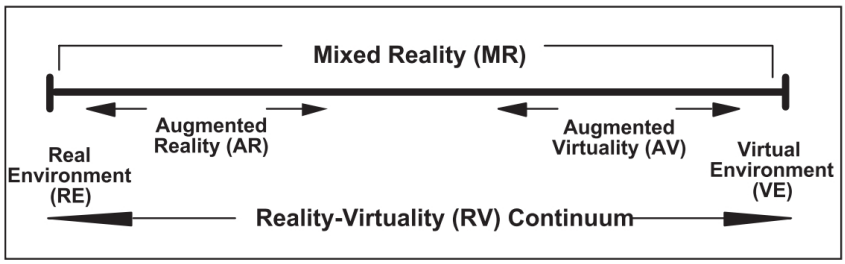



Figure 1.11 – The reality-virtuality continuum introduced by [Milgram & Kishino 1994].

### 1.3.1 Mobile AR systems

Several applications have been proposed in the literature that allow users to navigate through information presented using AR techniques, such as embedding multimedia presentations (images, 3D graphics, audio, *etc.*) overlaid on the surrounding world by a see-through head-worn display [Höllerer *et al.* 1999], or improving tracking techniques for better accuracy [You *et al.* 1999]. Many of these applications are generally developed for the purpose of *tourist guidance* or *navigation and path finding*. In the Patrimonio Móvi project [Díez-Díaz *et al.* 2007], users can receive and consult information about the cultural and historical-artistic patrimony of a place when they enter inside the perimeter of coverage of a PoI. This is achieved by editing the knowledge about the environment with the Context Markup Language (CML) documents that are used to keep the references to different kinds of multimedia files. The mobile guide provides information about main monuments and buildings of the town describing their history using text, images and sounds. The authors also emphasize that the metadata used in these CML documents can also provide extra semantic meanings for multimedia files (such as images) that certain kind of users (such as blind people) cannot perceive. In [Schinke *et al.* 2010], in order to guide the tourists, the application uses arrows directly embedded into an augmented reality scene which point at the PoIs that are off-screen. [Reitmayr & Schmalstieg 2004] proposes a mobile application that displays location referenced icons on the captured scene and allows users to interact with them to access cultural information about sights in the city of Vienna which are stored in a database. In DiscovAR project [DunedinProject 2011], authors overlay graphics onto the live videostream, *e.g.* old photographs, video and audio narratives related to the objects, in order to tell the story of the New Zealand city, Dunedin, from the 1840s to present day.

With the improvements in 3D mobile rendering, some researchers have focused on overlaying 3D location-based objects on the user's viewport (*i.e.* the area of vision seen on the screen). Earlier examples of 3D augmented reality are achieved using a backpack test bed (hand-held computer, interaction and visualization devices, orientation and position trackers carried in a backpack). Such as in [Höllerer *et al.* 1999], using this test bed, a simplified 3D buildings are overlaid on the view by using a see-through head-worn display. Recently, 3D objects are often used to perform AR on mobile devices, without using additional interaction or sensor devices.

In Smart Vidente [Schall *et al.* 2013], the authors propose a 3D AR interface with tracking accuracy in order to visualize underground pipes for civil engineering tasks. The proposed solution addresses the challenge of visualizing not only the geometric attributes of the 3D models, but also the semantic ones. The user interactively selects semantic features that she wants to visualize and the 3D model is generated according to these selections. Annotating semantically interesting entities of an AR scene is also mentioned in Hydrosys [Nurminen *et al.* 2011], an interactive platform for 3D visualization of the terrain. Similarly, in an earlier work, using the topological relations of different parts of the 3D model, a semantic data model for tourist guidance is introduced [Schmalstieg & Reitmayr 2007]. In order to indicate that further information is available, different parts of the studied building are highlighted. In [Reitmayr & Schmalstieg 2004], the 3D building geometry is overlaid on the real-world object and different parts of the selected building are highlighted to show additional information. The data is handled using an XML-based database that holds references to geometric representation of the environment, as well as semantic and contextual elements. Using common XML tools, data are transformed and imported from different sources.

A layer-based augmented reality model is proposed in [Lee *et al.* 2009a] for integrating several types of layers into a unified model (sensing layer, object-identification layer, web search layer, media layer, *etc.*) (see Figure 1.12(a)). A similar model is proposed by the commercial AR-browser application Layar [21]. Layar allows users to experience their immediate surroundings by displaying digital content (*layers*) overlaid on the physical world as seen through the mobile device camera (see Figure 1.12(b)). Layers are web services serving geo-located PoIs developed and maintained by the third parties using a free API. Another commercially available and

21. https://www.layar.com/

widely used application is Wikitude [22]. In Wikitude, the user discovers information about some interesting places, famous landmarks and other PoIs. In order to get such information, the application communicates with some servers such as Wikipedia, Twitter, Instagram, world heritage list, *etc.* (see Figure 1.12(c)). Layar also integrates information from social networking platforms such as Twitter and Qype, as well as with photo sharing web services such as Flickr. In both Layar and Wikitude, location-based PoIs are overlaid on the real view as an augmented reality scene.

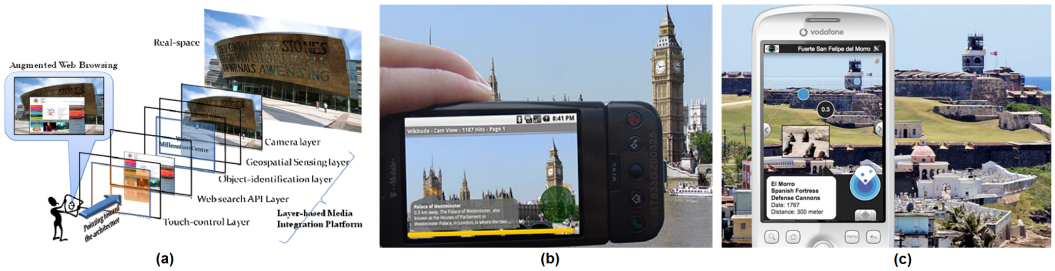



Figure 1.12 – (a) Layer-based augmented reality [Lee *et al.* 2009a] (b) Geo-located PoIs displayed using Layar (c) Information about interesting places, famous landmarks and other PoIs displayed using Wikitude.

### 1.3.2 Tracking and Registration

One of the biggest challenges for mobile AR is tracking and registration. Registration is the process of analyzing images of a video stream in order to insert visual objects on this real-world view by aligning them correctly. One of the key challenges in creating a realistic AR application lies in maintaining accurate registration between the real world and virtual objects. In order to achieve great accuracy for aligning these objects on the captured real scene, one of the options is to track the location, orientation and direction of movement of the user's mobile device in real-time. Thus, the changes can be properly reflected on the viewer's display by rendered graphics (*i.e.* registration). This means that the system should be able to do a pose estimation (*i.e.* position and orientation estimation) of the device with 6 degrees-of-freedom (6DOF): three variables (x, y and z) for the position and three angles (yaw, pitch, and roll) for the orientation. If the two worlds (digital and real) are not properly aligned, the AR loses its ability to allow users to point and select the real-world objects and to provide information at the correct real-world position [Suomela 2006].

22. http://www.wikitude.com/

**Vision-based tracking:** Vision-based methods rely on the scene features and consist in constant processing of video input from the camera of the device [Marimon & Adamek 2011].

– **Marker-based Systems** Tracking based on fiducial markers is a highly efficient vision-based method. The use of markers increases robustness and reduces computational requirements [Genc *et al.* 2002]. Artificial markers are added in the scene to determine the orientation and position of the camera within the environment. Mostly used fiducial design is 2D square fiducials where one may replace some barcode-type patterns that can generate thousands of different codes and can be used in wide area tracking [Naimark & Foxlin 2002]. The popular ARToolKit library is developed by Hirokazo Kato in 1999 as an AR conferencing system that allows overlaying virtual images on multiple users' real environments by using head-mounted displays [Kato & Billinghurst 1999]. It is based on tracking size-known square markers and extracting the coordinates of the camera relative to the marker (see Figure 1.13(a)). It has been modified since then and now is a very widely used open source AR tracking library [ARToolkit 2013a]. For instance, [Wagner & Schmalstieg 2003] carry the ARToolkit onto the PocketPC platform with some optimizations in order to be able to use it in handheld AR projects (see Figure 1.13(b)). [Henrysson *et al.* 2005] have developed an AR game (AR Tennis) by creating a custom ARToolkit library on Symbian mobile phone operating system. In AR Tennis, two players hold their mobile phones towards ARToolkit markers and hit to a virtual ball by moving their phone (see Figure 1.13(c)). Nowadays, developers use ARToolkit open source library to create their own markers and

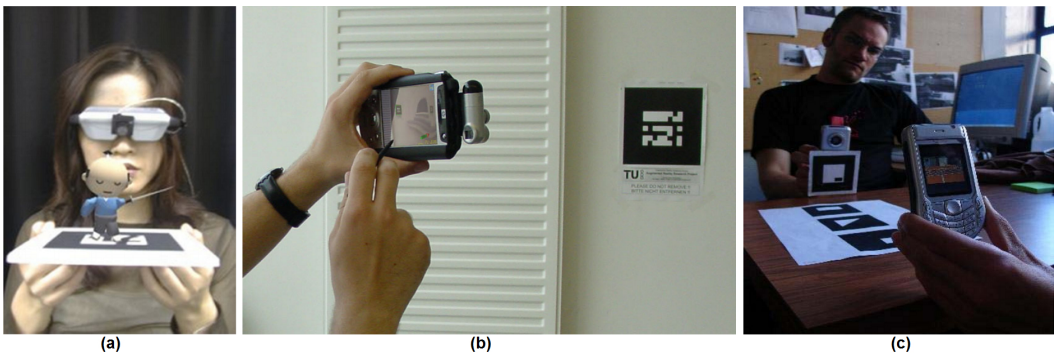



Figure 1.13 – *(a)* ARToolkit [ARToolkit 2013a]; *(b)* ARToolkit is used on hand-held platform by [Wagner & Schmalstieg 2003]; *(c)* ARToolkit is used to create an AR game [Henrysson *et al.* 2005].

train the system to recognize them [ARToolkit 2013b].

However, marker-based tracking is a specialized use case where it is possible to add markers in the scene. Hence, it needs a controllable environment, such as indoors like factories and museums. It is generally not suited for general purpose outdoor applications, such as suggesting nearby restaurants [Gotow *et al.* 2010].

– **Markerless Systems** Rather than using fiducial markers in the scene, camera pose can also be determined using the natural features occurring in the scene, such as points, edges, lines, planar surfaces or textures, which is called *feature-based tracking*. Once image features are extracted, they are tracked frame-to-frame for camera pose estimation. In [Lee & Hollerer 2006], with the assumption that the tracked object has planar surface, which makes it easy to calculate homographies between the reference frame and the other following frames, corners are tracked throughout the video. A similar approach is applied for outdoor environments in [Aydin 2010]. Edges are one of the most used features in markerless tracking, since they are relatively easy to detect due to their robustness to illuminations changes in video frames [Zhou *et al.* 2008]. Another markerless tracking method is model-based approach, in which the knowledge about geometrical features of the scene is used to estimate the camera motion. [Behringer *et al.* 2002] propose an algorithm that relies on knowledge of a 3D model of the building and determines the optimal motion vector in 3D (translation and rotation) by extracting and tracking 2D image features (edges and corners). Model-based tracking is often used in hybrid tracking approaches.

  Vision-based tracking is accurate even in uncontrolled environments, but it suffers from occlusion and it is computationally expensive.

**Sensor-based tracking:** Sensor-based tracking are based on sensors like optical, magnetic, mechanical, inertial, *etc.* Today, mobile devices are built with these embedded sensors which enable to track the position, orientation and direction of movement of the device sufficiently. Outdoor augmented reality requires tracking in unprepared environments. For a global outdoors tracking, GPS is currently the most used technology, but it does not provide enough precision for a high quality tracking [Schall *et al.* 2009]. In some situations, the performance of GPS may deteriorate significantly in urban environments, as buildings may reflect off the signals that come from the satellites. Magnetic sensors have high update rates and are used

to detect the orientation of device with respect to the magnetic field of the earth. However, they are susceptible to electromagnetic interferences and can be distorted by any nearby metallic substance around them [Zhou *et al.* 2008].

A convincing solution for sensor-based tracking is using the advantages of all these embedded sensors and combining (*i.e.* fusioning) multiple sensor data in order to obtain a more accurate and stable system. Combining (fusioning) sensor data on mobile devices is not new and several approaches can be found in previous works, such as in [Caarls *et al.* 2003], [Hol *et al.* ] and [Persa 2006]. Most of the multi-sensor fusion systems use a Kalman filter ([Kalman 1960]) in order to avoid the errors caused by inertial sensor measurements. The strategy of using a Kalman filter consists in merging data from different sensors to localize the camera following a prediction/correction model. For instance, in [Schall *et al.* 2009], the authors propose a system that uses Kalman filtering for fusion of GPS, gyroscope, compass and accelerometer to improve the accuracy of the user's 3D position estimate. Their application domain is the augmented reality visualization of underground infrastructures, such as waterways and electricity lines.

**Hybrid tracking:** Hybrid systems try to combine the advantages of two or more technologies. It is very common for AR systems to combine vision-based tracking with sensor-based tracking [Azuma *et al.* 1999], which is frequently achieved by using a Kalman filter [Kalman 1960]. In [You *et al.* 1999], inertial gyroscope sensor data is combined with vision-based tracking methods to achieve more robust tracking results. [Ribo *et al.* 2002] introduce a hybrid tracking system for outdoor augmented reality using a database of feature points established offline from a 3D model combined with gyroscope data in order to improve the robustness. A model-based hybrid tracking algorithm is introduced in [Reitmayr & Drummond 2006] combining edge-based tracking for robustness and fusion of gyroscope and accelerometer sensor data by using a Kalman filter.

## 1.4 Conclusion

In this chapter, we have reviewed some of the most relevant works related to surroundings discovery using mobile devices. Location is the primary context for exploring the place around. The meaning of *context* changes from one application to another. Context is used to provide content to a moving user and to help her to discover a special place or landmarks. At this point, what is important in these

applications is to use context actively and retrieve PoIs and information related to them in real-time. The changing location of the user results in changing context and possibly, the change of provided content.

We have observed that *semantics* provide richer concepts and this approach is often used to conceptualize context, as well as the content (semantic annotations and descriptions) in LBS. The World Wide Web has also evolved towards this dimension. Nowadays, vocabularies and ontologies make data retrieval more effective and provide richer content. Following these evolutions, surroundings discovery mobile applications have recently started to implement these data repositories available on the Web for content retrieval. Richer information through the exploitation of interlinked data between various resources can be provided to mobile device users. We examine these resources in the next chapter (Chapter 2) in detail. In our research, we are particularly interested in spatial and temporal content integration into surroundings discovery applications using these web resources.

Visualization of location-based data has also changed with the advancements in mobile technology. 2D and 3D maps are often used to display PoIs on mobile devices. Another noteworthy technological progress in mobile visualization methods is Augmented Reality. AR has been a popular research topic since 90s, and nowadays it is possible to use it on tablets, smartphones and wearable glasses without additional hardware devices. While the choice of these visualization techniques is open to discussion and mostly depends on the desired task, looking at the existing works, we can conclude that the designed system should meet the following requirements for an effective location-based discovery experience:

1. The user should be able to easily relate the information with the objects around her.
2. The 3D models should be extensible for different scenarios so that the same 3D object can be accessed and re-used by various domain-specific applications (architecture, tourism, navigation, cultural heritage, real estate, risk management, *etc.*). They should be modelled considering the cognitive resources of the user in order to avoid information overload. Thus, they should contain *minimum, but necessary* details.
3. The system should allow the user to interact with her surroundings through the mobile device while she keeps observing her surroundings, instead of totally focusing on her mobile device's screen. In other words, for instance, while navigating or accessing to a piece of information, the user should not be re-

tained from observing her surroundings by means of demanding interaction mechanisms or detailed visualisation methods.

4. The quality of location-based information should be increased, probably by using semantic web technologies so that information can be semantically categorized and data related to an object can be retrieved easily.
5. The user should be able to discover the information related to the surrounding objects through these categories, according to her interests and preferences, during her visit of the location.

Today, people are much more involved with their surroundings and interactive with location-based information sources. AR technologies have started to be used as a complement to the cognitive process of users. As the digital information is seamlessly overlaid on the captured real scene, they can instantly relate a piece of information with their surroundings which results into reduced cognitive activities. However, most of the proposed AR approaches are domain specific and the digital data that is overlaid on the real view are maintained in local databases. Moreover, at times, the superimposed 3D data may overload the user with details or require complicated interaction mechanisms to interact and find their ways among available information. However, a semantic web approach, where the access to these 3D models is centralized and empowered with linked data mechanisms, in which the models are interlinked with necessary concepts to be easily retrieved by the AR application should be thought of as a more standardized approach.

In the following chapters, before describing our approach, we first examine the semantic information available on the Web and how these resources can be exploited by mobile devices in Chapter 2. Then, we synthesize these two domains, namely surroundings discovery and semantic information on mobile devices, and point out some remarks that help us to construct our proposition, before discussing our proposition.

Chapter 2

# Semantic Information in Mobile Applications

## Sommaire


2.1 Introduction . . . . . **43**
2.2 Semantic Information . . . . . **44**
2.2.1 Web x.0 Evolution . . . . . 45
Web 1.0 . . . . . 45
Web 2.0 . . . . . 46
Web 3.0 . . . . . 46
2.2.2 Ontologies . . . . . 50
2.2.3 Ontology Description Languages . . . . . 53
XML(S) . . . . . 54
RDF and RDFS . . . . . 55
OWL . . . . . 61
2.2.4 Linked Open Data . . . . . 62
2.2.5 DBpedia . . . . . 68
Querying DBpedia . . . . . 71
2.2.6 Publishing Data on the LOD Cloud . . . . . 74
2.3 Different Ways to Use Semantic Information . . . . . **76**
2.3.1 Extracting Information from the LOD Cloud . . . . . 76
2.3.2 Examples of Applications based on the LOD Cloud . . . . . 78
2.3.3 Semantic Information on Mobile Devices . . . . . 83
2.4 Conclusion . . . . . **86**


## 2.1 Introduction

The World Wide Web is the universe of accessible digital information and is continuously changing. New generation Web technologies are constantly appearing and the current ones have evolved over the time. Today, the Web provides a vast range of information. From biology to humanities, or from the list of museums to

the bibliography of academic works, an extensive amount of information can be extracted from the Web. With these evolutions, information modelling and knowledge representation have become important topics and challenges.

In order to use the Web as a source of information, we need to understand the architecture and the principles of this technology, as well as the best practices that are currently being used. In this chapter, we first focus on the evolution of the World Wide Web and the technologies that have been used along the path. Afterwards, we focus on the Semantic Web which is today the key asset to discover the information sources and extract the relevant information from the entire linked data space.

New web and mobile technologies enable accessing information from different devices, such as tablets, notebooks or mobile phones. The positioning technologies of these portable devices enable developers to acquire (and provide) an extended knowledge about (and to) users by querying location-based information in order to answer questions like *who*, *where*, *what* and *when*. These developments have opened up new ways of application design on mobile devices such as *location-based cartography*, *digital story telling* or *augmented reality* in order to return extracted information to the user.

In the following of this chapter, we first introduce semantic information available on the Web and the notion of ontology. Then, we describe the Linked Open Data initiative which is the largest interconnected source of structured data that uses semantic web technologies to make data open and accessible over the Web (see Chapter 2.2.6). After that, we present the ways to extract information from the Linked Open Data and finally, we give examples of web and mobile applications that use it as a source of information.

## 2.2 Semantic Information

The amount of data published on the Web has tremendously increased [Trotman & Zhang 2013]. This rapid growth has also brought some difficulties related to the extraction of information from the existing distributed data repositories on the Web [Brin 1998]. In 2006, the Semantic Web is introduced in order to meet the emerging needs of the information retrieval [Shadbolt *et al.* 2006]. It extends the current World Wide Web by adding machine-processable descriptions using ontologies and provides an account of *meaning* which establishes interoperability among different systems. We examine this evolution of Web in the following.

### 2.2.1 Web x.0 Evolution

When Tim Berners-Lee introduced the Web in 1989 [Berners-Lee 1989] in order to tackle the problem of loss of information about complex evolving systems at CERN. The goal was to enable workers to easily access, communicate and share information within the organization. This problem also concerned the rest of the research community who faced the similar problems at the same time. In 1989, Berners-Lee described the *linked information systems* where the interrelationships between the things are represented with circles and arrows. In this diagram, nodes (circles) and links (arrows) constituted the "web" of nodes.

"Hypertext" [1] is defined as human-readable information linked together in an unconstrained way by Ted Nelson [Nelson 1965]. In his article of 1989, Tim Berners-Lee remarked that a standardization would be necessary in order to allow for the access to existing databases using distributed hypertext systems. In such a standardized model, hypertext allows documents to be linked to "live" data (where the data is temporary and may change); such that every time the link is followed, the information is retrieved. This hypertext project which was designed to allow people to work together by combining their knowledge on Web, led to what is known today as the World Wide Web, a common information space in which people communicate by sharing information [Berners-Lee 1998].

> *"Only by understanding the Web at this deeper level will people ever truly grasp what its full potential can be." Tim Berners-Lee*

#### Web 1.0

The early version of World Wide Web (WWW) was dominated by hyperlinks between web pages. Hyperlinks allow a web resource to point to any other Web resource by embedding the URL (Uniform Resource Locator) within an HTML page. Web 1.0 has enabled the author-driven content delivery with page navigation, linking, search and browse facilities. However, it did not provide any client-side interaction to participate by reviewing, commenting, sharing or creating information. The publishers were experts with skills to create web pages. To the same extent, Web 1.0 technology has provided a *read-only* web with static pages used mainly for scientific publications and businesses to spread information. The linking structure was weak and the user's role was limited to reading the information published.

1. http://www.w3.org/WhatIs.html

### Web 2.0

Web 2.0 has changed the whole idea about interactivity on the Web. It aimed at facilitating collaboration and sharing between users and introduced the "Social Web". The term was first used by Darcy DiNucci in her 1999 article as an interactive space that will be used by a variety of devices and platforms [DiNucci 1999]. However, it was not until 2004 that the term was officially defined at the first Web 2.0 conference by Timothy O'Reilly with the notion of a *read-write* Web where users get involved when publishing their content. It has taken its power from the collaborations of communities of users with key manifestations such as blogs, wikis, social networks, instant messaging *etc.* Therefore, the publishing experience has given rise to a participating activity over the Web with the freedom to create and circulate content through all the connected devices. Some examples include Wikipedia [2] which creates a dynamic content of an Internet-based encyclopaedia edited by users considered as collaborators all around the world, or Flickr [3] which forms a collaborative image database of categorized photos by enabling the tagging of photos published on an image hosting web service, or Delicious [4] which is a social bookmarking service that allows users to tag and share links, collaboratively group them under similar topics. As can be seen through these examples, data management became an important issue with Web 2.0. However, it has also become difficult to find relevant information from the massive amount of unstructured content generated by millions of users.

### Web 3.0

Web 3.0 is also known as the "Web of Data" or the "Intelligent Web" which is a more pervasive Web with the ability given to machines to access information. Tagging does not solve the problem of organizing the data on the Web and requires a vast amount of human efforts. However, if the machines (computers) were capable to "*understand*" the meaning of data, they would automatically crawl over Internet and find the relevant information by using different reasoning techniques, which will transform the Web into a *read-write-execute* Web. That is the key idea of the Web 3.0.

*"People keep asking what Web 3.0 is. I think maybe when you've got*

2. http://www.wikipedia.org/
3. http://www.flickr.com/
4. http://delicious.com/

*an overlay of scalable vector graphics - everything rippling and folding and looking misty - on Web 2.0 and access to a Semantic Web integrated across a huge space of data, you'll have access to an unbelievable data resource" Tim Berners-Lee, 2006 [Shannon 2006]*

The idea of Web 3.0 originated with the idea of the "Semantic Web", which was introduced as a concept back in 2001 [Berners-Lee *et al.* 2001]. Following the new trends and developments in web technologies, the Semantic Web came out with the purpose to automate the processing of content by bringing machine-readable descriptions that can be applied on top of the Web (to existing data resources). With this approach, machines can deduce information such as *Lyon is a city in Rhône-Alpes region* and *Rhône-Alpes is in France, which is a country in Europe*; or *the number of inhabitants of the commune Saint Martin d'Hères* by automatically using different distributed knowledge bases.

The content represents valuable assets that may be reused in different contexts and therefore has to remain accessible for a long time [Zettsu *et al.* 2010]. The key point of the Semantic Web is that the semantics characterizes the data by giving it a meaning explored by the machines in order to establish interoperability between systems using these logical connections. Web 1.0 and Web 2.0, on the other hand, are not encoded in a semantic form, therefore are not transparent and *meaning* is not accessible to a variety of reasoning engines. The Semantic Web appears to push the limits of the Web by aiming at the automation of document processing and information retrieval.

Before the Semantic Web was introduced, knowledge representation systems were centralized and required everyone to use the same definitions for common concepts such as "parent" or "vehicle" [Berners-Lee *et al.* 2001]. With the Semantic Web, formal standards were introduced to process data automatically. This is done under the coordination of the World Wide Web Consortium (W3C) [5], an international organization devoted to the creation of standards and guidelines for the Web. W3C has introduced standard languages for the metadata: eXtensible Markup Language (XML) [6] and the Resource Description Framework (RDF) [7]. XML allows everyone to create their own tags that annotate Web pages or sections of the page, whereas RDF encodes the meaning in sets of triples [Berners-Lee *et al.* 2001]. RDF is a simple language to state the relation between things in the form of subject-

5. http://www.w3.org/
6. http://www.w3.org/XML/
7. http://www.w3.org/RDF/

attribute-object triples. It uses URIs (Universal Resource Identifier) to ensure that the concepts are uniquely defined and has the same meaning. By using a global naming convention, everyone can find these concepts on the Web and re-use them (further details can be found in Section 2.2.3).

Tim Berners-Lee proposed a layered architecture, which is also known as the *Semantic Web Stack* or *Layer Cake* (see Figure 2.1) [Berners-Lee 2000]. According to this architecture,

- The URI/IRI (Internationalized Resource Identifier) layer provides a simple way for identifying resources. A resource can be a document, a people, a web site or an image. This identification also enables interaction over a network with the existing protocols on this layer. With these protocols, web pages are transmitted over the Internet.
- The XML layer provides a base syntax to structure data on the Web without communicating the meaning of the data. It facilitates the interoperability for other technologies developed for upper layers of the Semantic Web.
- The RDF layer supports representing, exchanging and reusing metadata at the semantic level. It uses URIs to identify web resources and a graph model beyond hierarchies, which is defined by XML, to describe the relationships between resources. Therefore, it is a generic language that can be used by

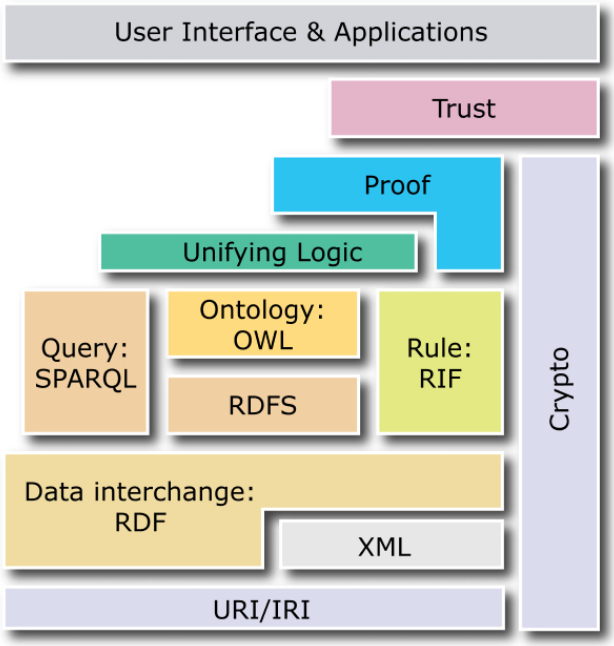



Figure 2.1 – Semantic Web Layers (Layer Cake) [Berners-Lee 2000].

other applications. The benefit of this is that it is only necessary to learn a generic language rather than learning a new language for each application [Wilson & Matthews 2004].

– The next layer contains basic and more descriptive ontologies as well as a flexible query language. RDF Schema (Resource Description Framework Schema) describes classes of resources, properties and relations between them. It intends to structure RDF resources and improve the expressiveness. OWL (Web Ontology Language) is a more descriptive language used for ontological knowledge representation. SPARQL (SPARQL Protocol and RDF Query Language) is a query language that enables to retrieve information depending on a syntax, whereas RIF (Rule Interchange Format) creates a standard for interchanging the rules in rule-based systems on the Semantic Web. It expresses constraints and bindings that a static schema could not express.
– The Logic layer enables writing rules, whereas the Proof layer executes the rules and the Trust layer provides an assurance of confidence whether to trust the given proof or not. However, there are no standards yet for these layers. The Cryptography layer relies on the Proof layer to certify the origins of the information and verify that they are coming from the trusted sources. It is a difficult task since the challenges include how to trust the information sources on the Web, how to negotiate between different parties, *etc.* [Thuraisingham 2008].

The Web has changed our businesses, social and scientific practices. It is still evolving over time. The Semantic Web seeks to organize the mass of data available on the Web by analyzing the content of the data and the transactions between users and machines in order to create a personalized system. This also includes providing a mobile environment with distinct elements, such as location awareness, real-time execution, ubiquity, *etc.*, which supports personalization in a greater extent. The Semantic Web provides infrastructures for the extensive use of the distributed knowledge. Therefore, providing a shared and common domain structures becomes essential [Fensel 2003]. Ontologies respond to this need by providing structures and semantics for the information exchange. These shared machine-readable representations are the key asset for the Semantic Web. In the next section, we present ontologies with their current technologies and the projects that are related to our proposition.

### 2.2.2 Ontologies

The notion of *ontology* has been originated from metaphysics, a branch of philosophy, in which it emerged as to know what there is to be represented and how to represent it [Wyssusek 2004]. In this philosophical point of view, *ontology* has been concerned with the question "what kind of things are there ?" and categorizing these things [Domingue *et al.* 2011].

Thus, in the Semantic Web, given a domain, *ontologies* correspond to the structural knowledge about what exists in that domain. Ontologies are the main building block of the Semantic Web and they represent a subset of the world with *shared* and *common* understanding that can be communicated among people. Therefore, in the Semantic Web point of view, we can define an ontology as in the following, based on [Gruber 1993]:

**Definition 1** (**ontology**). *An ontology is a formal, explicit specification of a shared conceptualisation.*

A *conceptualisation* is an abstract model of a phenomenon of the world that interprets that phenomenon by forming concepts. *Explicit specification* refers to defining the concepts and the constraints of their use explicitly. *Formal* means that the ontology is machine readable. A *shared conceptualisation* states that the ontologies are intended to be used by a group. In summary, an ontology provides a shared vocabulary that defines the concepts, properties and relations involved in intelligent information integration and knowledge management processes in order to make machines capable of understanding the semantics of the languages that humans use.

*Knowledge representation* is modelling data of a given domain so that it can be manipulated by the machines. Recently, computers systems have evolved to perform reasoning on these machine-interpretable forms of knowledge. In a web environment, ontologies play a key role in the modelling of a particular domain of interest. They make the hidden knowledge explicit, logically tractable, and provide tools for deductive reasoning [Duží *et al.* 2011].

The main components of an ontology are the concepts, relations, axioms and instances. *Concepts* are the entities or 'nodes' of a graph that represent relevant categories of a domain. *Relations* connect the concepts by specifying their interrelations. *Axioms* contain rules for fixing the semantic interpretation of the concepts and the relations of an ontology. *Instances* are the concrete objects or 'things' in

the domain of interest that are represented by the ontologies. These components can be visualized in Figure 2.2 based on [Keet 2004].

The ontology itself should not contain any instances, because it is the conceptualization of a domain knowledge [Stevens *et al.* 2000]. The combination of the ontology with the associated instances is called the *knowledge base*. An ontology, therefore, describes the knowledge on a schema level, i.e. in terms of conceptual taxonomies and general statements [Grimm *et al.* 2007], which is used by a knowledge based application. The ontological vocabulary is made up of these concepts and relations with their definitions in an informal text or in a formal language.

**Types of Ontologies:** Ontologies exist in many levels of formalism and for different types of purposes. [Guarino 1997] categorizes ontologies according to their subject of conceptualization:

1. *top-level ontologies* describe very abstract and general concepts that can be shared across many domains, such as the general notions describing time, space, events *etc.*

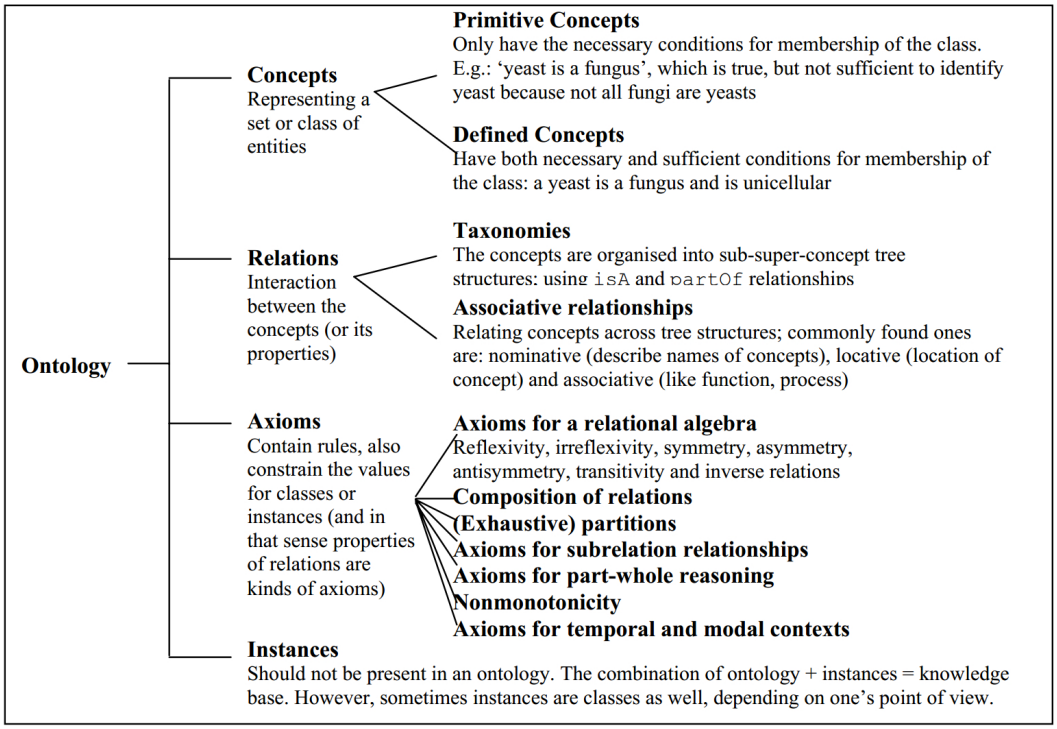



Figure 2.2 – Components of an ontology [Keet 2004] (based on [Stevens *et al.* 2000]'s descriptions; with the addition of axiom categories as discussed by Staab and Mädche (2000)).

2. *domain ontologies and task ontologies* express specific domain knowledge that has a narrower scope than top-level ontologies, such as the geographic information, tourism or cultural heritage ontologies, *etc.*
3. *application ontologies* provide a specific vocabulary for a certain task. They typically use domain and task ontologies in order to describe the role of a domain entity in a specific application, such as an ontology of architectural structures ontology derived from a cultural heritage ontology which describes ancient buildings (materials, shape, ...), *etc.*

In this categorization, the lower-level ontologies inherit from the top-level ontologies and specialize where a narrower application scope is necessary rather than general notions. However, top-level ontologies have more potential to be re-used.

According to [van Heijst *et al.* 1997], the ontologies are classified in two dimensions: the amount and the type of structure of the conceptualization and the subject of the conceptualization (see Figure 2.3). The first dimension contains three categories: *Terminological ontologies* specify terms that are used to represent knowledge, such as lexicons; *Information ontologies* specify the record structure of databases, such as an ontology that is used for modelling medical records of patients; *Knowledge modelling ontologies* specify conceptualizations of the knowledge and are often used to describe a particular use of the knowledge. In the second dimension, there are four categories: *Application ontologies* and *Domain ontologies* are as described in [Guarino 1997]'s categorization above; *Generic ontologies* are similar to domain ontologies, but the concepts that they describe are more generic across many fields;

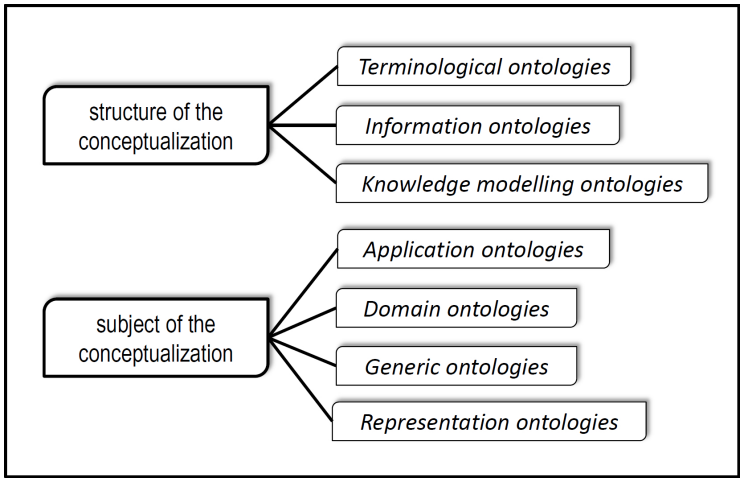



Figure 2.3 – The two dimension classification of ontologies according to [van Heijst *et al.* 1997].

*Representation ontologies* provide a representational framework without making explicit statements and allow domain ontologies to be specified with this representation.

### 2.2.3 Ontology Description Languages

In practice, ontologies can be regarded as hierarchies of concepts with attributes and relations, which establish a terminology to construct a semantic network of concepts, relations and instances into which some domain-specific knowledge is stored. Ontologies specify rich semantic domain knowledge and allow reasoning by relying on *logical formalisms* and, therefore, facilitate expressing complex statements.

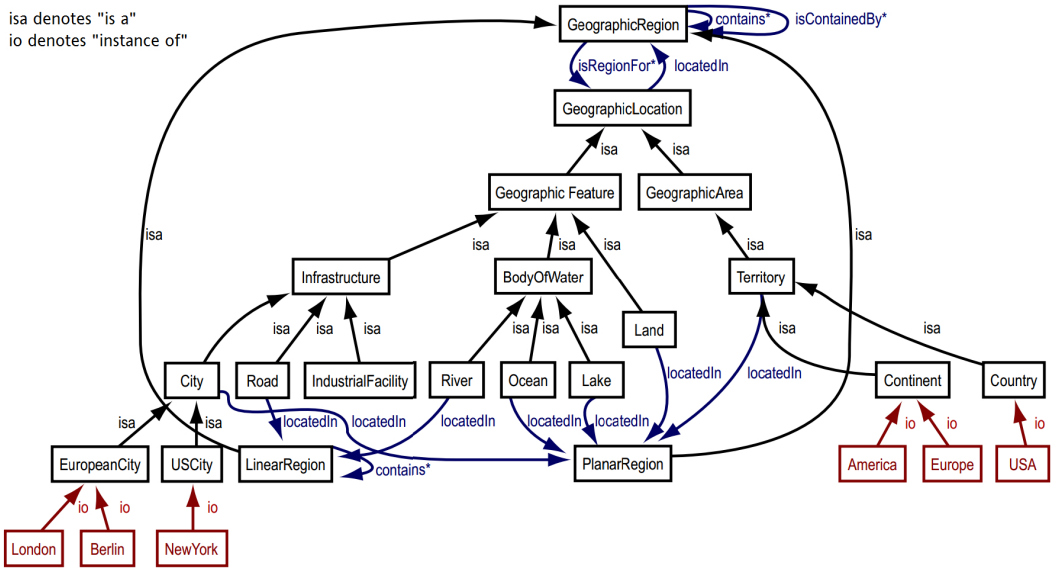



Figure 2.4 – An example of geographical data expressed as a graph [Grimm *et al.* 2007].

An ontology language is used for the storage and transfer of a knowledge base with a machine processable serialization format. A *graphical* representation of this base is used to display an ontology. For example, in Figure 2.4 [Grimm *et al.* 2007], the geographical knowledge which includes geographical locations and features with the regions of these features, as well as their relations are depicted in a *graph* (where the black rectangles are *concepts*, the red ones are *instances* and the lines represent the *relations* between them). In this geographical knowledge base, some statements can be deducted, such as a "*river is a body of water*", "*country is a territory*", *etc.* Using these statements, reasoning systems can make many other deductions

automatically, such as "*the geographic region of a European city is contained in Europe*", "*lake is a geographic location and located in a planar region*", *etc.* This graph also represents a *semantic network* where nodes are the concepts and arcs are the relations between these concepts. The concepts can be either general concepts (*e.g.* EuropenCity) or individual concepts (*e.g. London*), where the latter represent objects in the domain of interest, the former serve as classes that group these objects together.

As shown in Figure 2.1, on top of the term identification layer represented by a URI or an IRI, there are XML and RDF as commonly used technologies that define a syntax and a structure for ontologies.

### XML(S)

XML is a simple text-based format for representing structured information for documents or data on the Web [8]. Even though it is not a knowledge representation language, its syntax is used for knowledge exchange in other knowledge representation languages such as RDF and OWL. It generalizes HTML by allowing user-defined tags [Domingue *et al.* 2011]. An XML document consists of a nested set of tags in human readable format where each tag can have attribute-value pairs. The way these tags should be combined is not predefined, which gives a flexibility to the applications and allows them to define their own tags to give additional information about the text in the document [Terzi *et al.* 2003]. However, in order to exchange an XML file between two systems, a standardized structure that contains attributes to describe the content enclosed by the tags is necessary

XML Schema (XMLS) is a language for expressing constraints about XML, also referred to as XSD (XML Schema Definition). It provides grammar specifications for XML tags and allows for the definition of complex data types, which assures a human-readable and machine processable documentation. For instance, the Geography Markup Language [9] (GML) defined by Open Geospatial Consortium (OGC) [10] expresses geographical features using XML grammar. The key GML geometry object types are *Point*, *LineString* and *Polygon*, expressed with XML syntax as <gml:Point>, <gml:LineString> and <gml:Polygon> respectively. A GML document is described using a GML Schema (which is based on XML Schema) which allows users and developers to define specific application schemas that are specia-

8. http://www.w3.org/standards/xml/core
9. http://www.opengeospatial.org/standards/gml
10. http://www.opengeospatial.org/

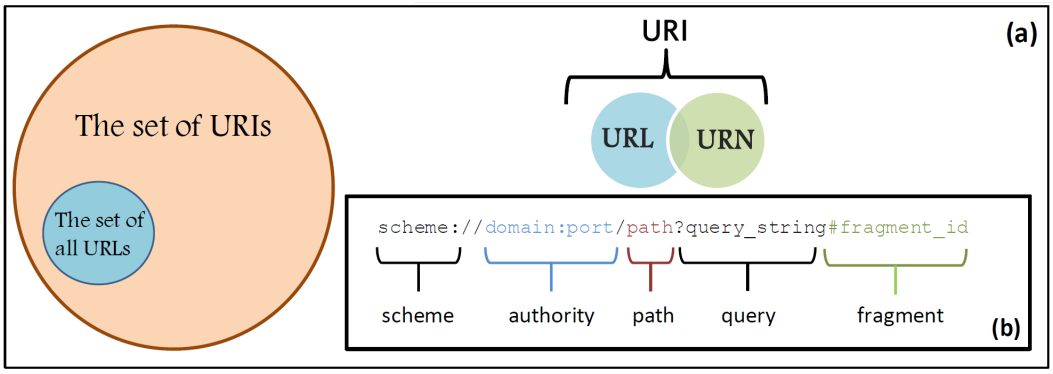



Figure 2.5 – (a) URLs are a subset of URIs (b) components of a URI syntax.

lized extensions of GML, such as roads, highways, bridges *etc.* CityGML [11], for example, is implemented as an application schema for GML in order to easily exchange data and define 3D urban objects with related feature types (*i.e. Window, Door, WallSurface, etc.*)

### RDF and RDFS

The Resource Description Framework (RDF) provides a standard way to express a graph of data on the Web and interoperability for the exchange of metadata between Web applications. Initially, RDF is introduced with the purpose of associating collections of distributed information, particularly for the representation of metadata, about identifiable web resources, such as the author, creation date, subject, and copyright of an electronic document on the Web. It provides a secure and reliable mechanism by expressing these resources in the form of *subject-predicate-object* expressions, called *triples*. Such triples correspond to statements which are commonly written as *p(s,o)*, where *p*, *s* and *o* are the representations of *predicate*, *subject* and *object*, respectively [Terzi *et al.* 2003]. RDF makes assertions that particular things (people, web pages, *etc.*) have properties (is the author of, is created by, *etc.*) with certain values (name of a book, name of a person, *etc.*) [Berners-Lee *et al.* 2001].

**URIs:** RDF assigns unique identifiers, namely URIs, in order to name the entities so that they can be referred to constantly across the triples. In graph representations, the disambiguation is provided by using slightly different strings for the same entities (*e.g.* "Notre-Dame Cathedral" or "Notre-Dame de Paris"). When multiple da-

11. http://www.citygml.org/

tasets are used in larger applications, it is difficult to guarantee the use of the same (unique) string identifiers for resources, therefore RDF removes the ambiguity when transmitting semantic data between machines by identifying resources with URIs. The predicates can also be identified by URIs. This enables to define a new concept and re-use it just by defining a URI for it somewhere on the Web [Berners-Lee *et al.* 2001]. URIs can be further classified as locators (URLs), as names (URNs (Uniform Resource Name)), or as both. URLs are a particular type of URIs, where URIs identify the document and URLs locate them (see Figure 2.5(a)). URNs are used to identify something, rather than implying its location or how to access it. The ISBN system that is used to identify books can be an example for URN (such as ISBN 3540929142 cites the book Handbook of Semantic Web Technologies of [Domingue *et al.* 2011], without implying its location). URI syntax consists of a hierarchical sequence of components referred to as *scheme*, *authority*, *path*, *query* and *fragment identifier*: scheme://domain:port/path ?query_string#fragment_id where the *scheme* identifies the *protocol*, *authority* gives the exact destination location, *path* specifies this location, *query* contains data (mainly name/value pairs) to be passed to the web application and *fragment identifier* (introduced by a hash mark #) is used to identify a portion of a particular document (see Figure 2.5(b)).

A URI expressed with a fragment identifier preceded by the character "#" is called URI *reference* (URIref) (*e.g.* http://www.w3.org/People/EM/contact#me identifies an individual). The part before the character # denotes the resource and the fragment identifier refers to some portion of that resource. Triple notations composed of subject, predicate and object URIs might result in long lines and confusion for an RDF reader. XML *namespace*, or simply *namespace*, is introduced in order to provide a simple method for qualifying element and attribute names by associating them with namespaces identified by URIrefs [12]. Some of the commonly used namespaces are shown in Figure 2.6.

XML also introduces *Qualified Names* (or QNames) for nodes which contains a *namespace prefix* that identifies a *namespace URI*, followed by a colon, and then a local name for an *XML tag* or an *attribute name* [13]. It is represented in the form P:L, with a single colon separating name into *namespace prefix P* and a *local part L* [Breitman *et al.* 2007].

For instance, RDF vocabulary provides a predicate type, which is used to relate resources to a category or class of resources, defined with the namespace

12. http://www.w3.org/TR/REC-xml-names/
13. http://www.w3.org/TR/rdf-primer/#documents

| Namespace | Namespace URI | Prefix |
|---|---|---|
| RDF | http://www.w3.org/1999/02/22-rdf-syntax-ns# | rdf |
| RDF Schema | http://www.w3.org/2000/01/rdf-schema# | rdfs |
| OWL | http://www.w3.org/2002/07/owl# | owl |
| Dublin Core | http://purl.org/dc/elements/1.1/ | dc |
| XML Schema | http://www.w3.org/2001/XMLSchema# | xsd |

Figure 2.6 – Examples of commonly used namespaces.

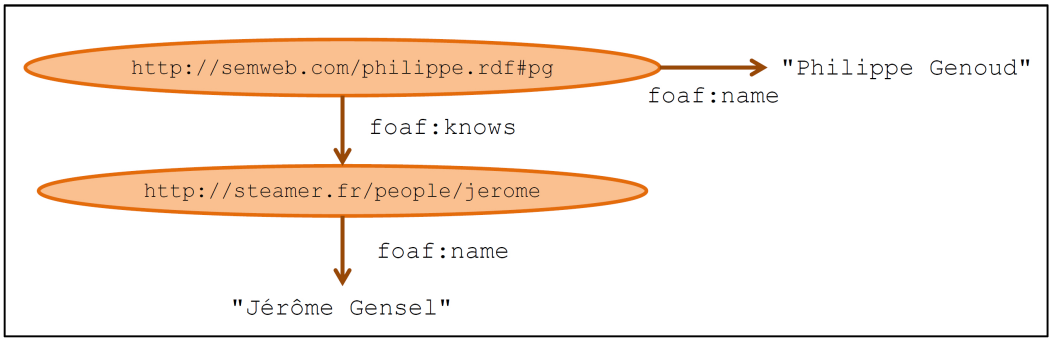



Figure 2.7 – A simple RDF graph showing foaf (friend of a friend) relations between two persons.

URI http://www.w3.org/1999/02/22-rdf-syntax-ns#. Therefore, the predicate can be identified with the QName rdf:type which has namespace prefix rdf and local part type instead of the longer representation as a URIref http://www.w3.org/1999/02/22-rdf-syntax-ns#type.

In the graph representation in Figure 2.7, a node-link-node (*subject-predicate-object*) structure is shown [Segaran *et al.* 2009]. It is a simple graph introducing *Jérôme Gensel* as a friend of *Philippe Genoud*. The graphs are useful to visualize data but sometimes an appropriate visualization paradigm is not possible if a more detailed view is desired to be displayed (such as restrictions of concepts or complex information about axioms). Therefore, data are stored in a serialized format in files. The common formats for serializing an ontology and the instance data are N-triples, N3 notation and RDF/XML notation. We will not go through details of these serialization formats, yet an example graph is given in Figure 2.7. In order to see the use of the QNames, we will analyze the RDF/XML serialization of this graph in the following:

```
1. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
2.        xmlns:foaf="http://xmlns.com/foaf/0.1/">
3.    <rdf:Description rdf:about="http://semweb.com/philippe.rdf#pg">
4.      <foaf:knows>
5.        <rdf:Description rdf:about="http://steamer.fr/people/jerome">
6.          <foaf:name>Jérôme Gensel</foaf:name>
7.        </rdf:Description>
8.      </foaf:knows>
9.     <foaf:name>Philippe Genoud</foaf:name>
10.    </rdf:Description>
11. </rdf:RDF>
```

Line 1 starts with rdf:RDF element which indicates that the following XML content represents RDF until the closing tag </rdf:RDF>. Following this, at the same line is an XML namespace declaration xmlns. In XML, a namespace is defined using the *xmlns* as a name, or the xmlns: as a prefix with a URIref identifying the namespace. This declaration on line 1 specifies that in the rest of the document, the XML tags prefixed with "rdf:" are the part of the RDF resource denoted by its URIref. Line 2 introduces another namespace, namely foaf. FOAF (Friend Of A Friend) is a collection of well-known predicates forming a vocabulary that is used to describe a number of "*things*" available on the Web [Segaran *et al.* 2009]. These *things* do not only describe relationships between individuals over a social network as the name of the vocabulary may evoke, but also what people create and do, such as web pages, photographs, organizations, *etc.*

The resource RDF nodes are usually specified with rdf:Description elements which is followed by an rdf:about attribute. Therefore, the rdf:Description tags in Line 3 and Line 5 indicate the beginning of the *description* of a resource with the rdf:about attribute specifying the URIref of that resource.

On the other hand, the property nodes contain only the properties, namely the foaf:knows in Line 4 and foaf:name in Line 6 and Line 9. Each property node represents a statement. Thus, in this example, we have two statements (also see Figure 2.7). Objects can be either resources or literals. If the object is a literal, then it is expressed within a property node, as in Line 6 and Line 9. However, if the object is a resource, then it is expressed with a rdf:Description tag. The two predicates in this example are expressed with Qnames foaf:knows foaf:name where the prefix *foaf* is defined with the namespace URI http://xmlns.com/foaf/0.1/.

The QName rdf:type is often used in RDF documents where, in the triple form of

*Subject* rdf:type *Object*, expresses that the *Object* is a *category* or a *class* of resources and that the *Subject* is an *instance* of it [Breitman *et al.* 2007]. In our example, we can add this *type* information to indicate that *Jérôme Gensel* and *Philippe Genoud* are instances of the category (class) foaf:Person, which can be represented as:

```
<rdf:Description rdf:about="http://semweb.com/philippe.rdf#pg">
  <rdf:type><foaf:Person>
```

The rdf:type predicate is used to indicate that the class/category of the resource http://semweb.com/philippe.rdf#pg is foaf:Person. Since rdf:type is used quite often, instead of creating a property node for this new statement, it can be abbreviated replacing the rdf:Description element in the following form:

```
<foaf:Person rdf:about="http://semweb.com/philippe.rdf#pg">
```

Then the graph represented in Figure 2.7 will be represented in RDF/XML format as:

```
1. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2.       xmlns:foaf="http://xmlns.com/foaf/0.1/">
3.   <foaf:Person rdf:about="http://semweb.com/philippe.rdf#pg">
4.     <foaf:knows>
5.       <foaf:Person rdf:about="http://steamer.fr/people/jerome">
6.         <foaf:name>Jérôme Gensel</foaf:name>
7.       </foaf:Person>
8.     </foaf:knows>
9.    <foaf:name>Philippe Genoud</foaf:name>
10.   </foaf:Person>
11. </rdf:RDF>
```

**RDFS:** Once there is a data model, the vocabulary language needs to be defined in order to allow users and developers to define their own terms that they intend to use in their RDF documents. Just as XMLS provides a vocabulary definition for XML, RDF Schema (RDFS) provides a way to express simple statements about resources. RDFS is used to define a *vocabulary* to be used in an RDF graph. These schemas are written and exchanged using the RDF vocabulary, so they are valid RDF graphs. It enables to define class and property hierarchies and domain and range of properties.

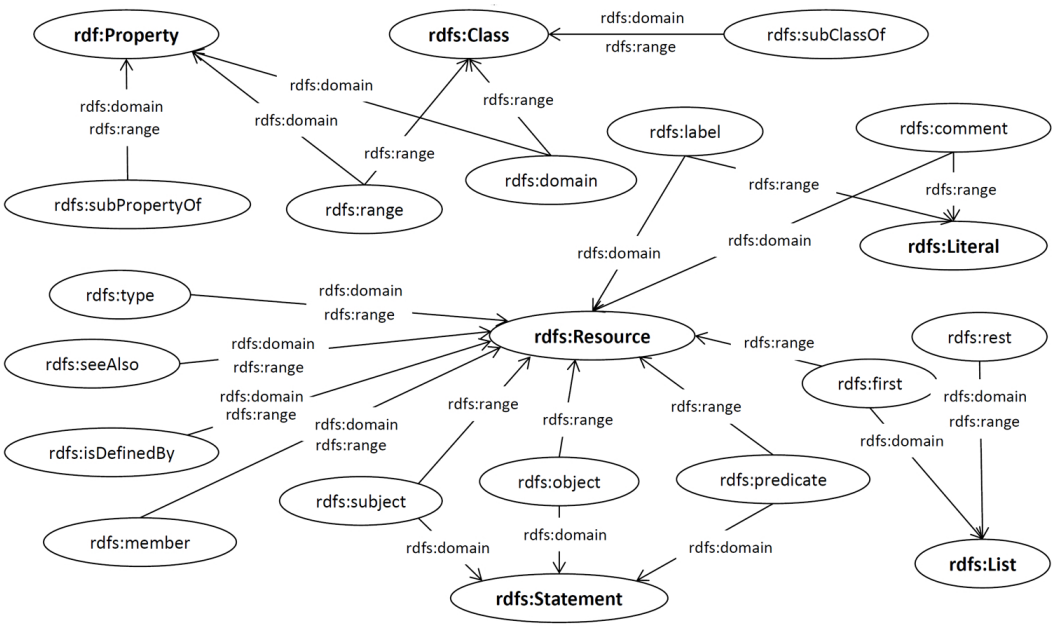



Figure 2.8 – Domain and range of RDFS properties [Gandon *et al.* 2012].

RDF already provides the membership property with rdf:type class that indicates which class the resource belongs to. Other descriptions related to classes are achieved using RDFS resources rdfs:Class and rdfs:Resource. In RDFS, a class is a generic concept defined with a rdf:type property. Resources that belong to a class are called its instances and a resource may be an instance of more than one class [14].

The RDFS predicates are described using the RDF class rdf:Property and RDFS properties rdfs:domain, rdfs:range, rdfs:subClassOf, rdfs:subPropertyOf, *etc.* rdfs:subClassOf and rdfs:subPropertyOf properties provide a hierarchical structure for other classes and properties. The type of a resource declared by the rdf:type propagates through the hierarchy defined by rdfs:subClassOf. This property also allows multiple inheritance [Domingue *et al.* 2011]. For instance, using a fictional semantic model with namespace URI http://example.org/ represented by the namespace prefix ex, we could declare the following relations:

```
ex:Journalist rdfs:subClassOf foaf:Person
ex:PhotoJournalist rdfs:subClassOf ex:Journalist
```

then RDFS makes it possible to infer the statement ex:PhotoJournalist rdfs:subClassOf foaf:Person.

14. http://www.w3.org/TR/rdf-primer/#schemaclasses

RDFS also makes a class definition for literal values with rdfs:Literal. These literal values are raw texts that can be used as objects in RDF triples. They can be plain text, which can be qualified with a natural language, or typed literals that are associated with a *datatype*, rdfs:Datatype, which indicates how to interpret the text (such as *date*, *time*, *name*, *etc.*). For instance,

```
"2013-11-25T19:23:03Z"^^xsd:dateTime
```

where xsd:dateTime is a datatype that describes instances identified by the combination of a date and a time.

Figure 2.8 shows the RDF graph of the domain and range of RDFS properties (RDFS metamodel) [Gandon *et al.* 2012]. RDFS extends the RDF classes and properties using RDFS vocabulary. Properties are linked with classes via the '*domain*' construct and restricted to one or more classes with '*range*' construct. For instance, P rdfs:range C states that P is an instance of the class rdf:Property, that C is an instance of the class rdfs:Class.

### OWL

The Web Ontology Language (OWL) [15] is an RDF language that is built as a much more expressive extension of RDFS which only allows defining hierarchies of classes and properties. OWL, on the other hand, allows to state definitions about the equivalence of classes and properties, equality and difference of resources, cardinality restrictions, *etc.* This provides powerful reasoning and inference over relationships [Segaran *et al.* 2009].

OWL is a W3C standard for defining semantic web schemas. OWL1 is recommended by W3C in 2004 and OWL2 in 2009. OWL1 is composed of three sub-languages depending on the expressiveness of increasing order: OWL Lite (designed to express simple constraints for reasoning systems), OWL DL (provides maximum expressiveness with some restrictions) and OWL Full (supports maximum expressiveness and syntactic freedom without assuring the completeness for reasoning systems). We will not cover in detail these three sub-languages and OWL2 in this manuscript, but present just enough of OWL in order to comprehend the rest of this document.

The definition of an ontology written in OWL language is composed of class and property definitions, a meta-description of the ontology and the descriptions of

15. http://www.w3.org/2001/sw/wiki/OWL

instances [Gandon *et al.* 2012]. The following classes are some of the most important ones in order to create an ontology:

- owl:Class designates the class of OWL classes and all these classes are instances of rdfs:Class.
- owl:Thing is the class of all *things* in OWL. All the OWL classes are sub-classes of it. The property rdfs:subClassOf is re-used in OWL to express that the instances of a sub-class are also instances of the super-class.
- owl:DataTypeProperty regroups the properties that have *literals* as their ranges.
- owl:ObjectProperty regroups the properties that have resources (*i.e.* instances of owl:Class) as their ranges.

OWL also reuses some other RDF(S) properties such as rdf:type, rdfs:subPropertyOf, rdfs:domain, rdfs:range, *etc.* OWL introduces a more complete set of functionalities with a broad type of class definitions such as class intersections and unions defined with properties such as owl:intersectionOf, unionOf, property restriction (*e.g.* owl:Restriction) represented by owl:onProperty, class annotations (*e.g.* owl:AnnotationProperty) used with RDFS properties such as rdfs:label, rdfs:comment, rdfs:seeAlso, *etc.* [Gandon *et al.* 2012].

In this section, we have introduced the semantic web technologies. Using these technologies, several connected datasets are published on the Web which constitute the Linked Open Data (LOD) initiative. LOD is a project started in 2007 with the aim to connect related data over the Web and provide an easy access. In the following section, we present the LOD, which is currently the best practice to share, connect and access pieces of data over the Web using URIs and RDF technology.

### 2.2.4 Linked Open Data

Linked data is a method of publishing structured RDF data on the Web using semantic web technologies, (*i.e.* URIs as resource identifiers, HTTP protocol to retrieve the resource descriptions, *etc.*) and setting links between data from one source to another [Auer *et al.* 2007], [Bizer *et al.* 2007]. These web-accessible resources can return meaningful descriptions and additional information related to that resource.

- Resources identified by URI references can be real-world objects, as well as abstract concepts [Heath & Bizer 2011]. This constitutes the first principle of Linked Data. It extends the Web from being an online resource of tangible objects to encompassing *any* concept in the world.
- The HTTP protocol enables to dereference these URIs and identify the des-

criptions of objects (*e.g.* a painting, a landmark, *etc.*) and abstract concepts (*e.g.* a poem, a review of a movie, a film movement, *etc.*). Thus, the second principle of Linked Data recommends to use the HTTP URIs.

- The Web content is intended to be shared among various applications, so it is important to use the standardized formats. Therefore, using a single data model, such as the RDF serialization format RDF/XML or N3 and SPARQL query services in order to acquire data constitutes the third Linked Data principle.
- The fourth and last principle of Linked Data advocates for the use of hyperlinks to connect any type of things. For instance, a FOAF graph may relate *two people* with foaf:knows relation, but it can also relate a *person* to a *place* with foaf:based_near.

With these principles, linked data aims to create a shared single global data space - a Web of Linked Data - where applications are able to navigate through URIs and retrieve data. Some benefits of creating such data space are as in the following [Heath & Bizer 2011]:

- Using HTTP URIs as unique global identifiers for data items and for the vocabulary enables requesting easily this data in other formats thanks to the HTTP content negotiations.
- The data relies on a single data model (RDF) which makes it possible to capture all kinds of information adhering to the RDF data model, such as taxonomies, vocabularies, ontologies, *etc.*
- By following the links, applications can retrieve more information at run-time, instead of preparing off-line data in advance.
- The integration of data from other sources is easy since the vocabulary can be shared and the terms from different vocabularies can be connected to each other.

Linked Data architecture makes it easier to store, process, access and integrate data. It enables the implementation of generic applications that run over the complete data space, query the RDF data model and retrieve entity descriptions [Bizer 2012]. Generic Linked Data applications are developed as browsers and search engines in order to process domain-related data. Linked Data browsers can navigates through the RDF links in order to merge data from different sources, such as the *LinkSailer* browser [16], the *Tabulator* browser [17] or the *Marbles* Linked Data brow-

16. http://linksailor.com/
17. http://www.w3.org/2005/ajar/tab

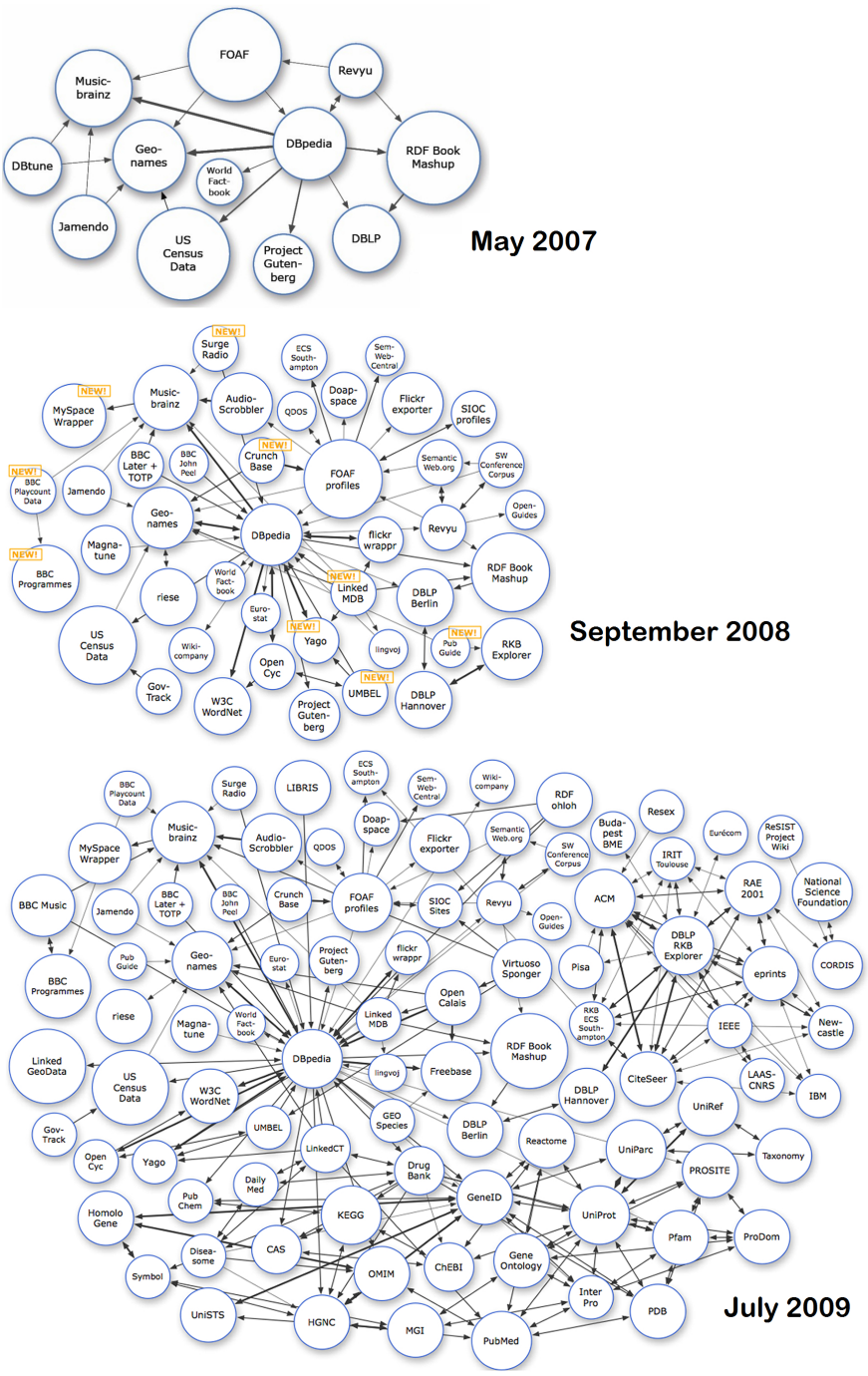



Figure 2.9 – The evolution of Linked Open Data from 2007 to 2009.

ser [18]. On the other side, a number of search engines have been developed to provide query capabilities over the aggregated data by using the RDF links of the Linked

18. http://mes.github.io/marbles/

Data. Some examples of search engines are *Sig.ma* [19], and SWSE [20]. They not only provide a keyword-based search service, but also interactions through which users can select additional structured data to be visualized. Other search engines, such as *Sindice* [21], *Swoogle* [22], provide a basic layer for the applications that are built upon Linked Data RDF documents [Heath & Bizer 2011]

Linked Data started to be deployed on the Web by the W3C Linked Open Data (LOD) project, which was founded in 2007. Since then, its principles have applied to various domain data such as e-science, e-government, geographic data, library and scholarly communication data, *etc.* The aim of the project is to identify existing datasets and facilitate the publication of data under open licenses by combining knowledge representation techniques (such as RDF, RDF Schema and simple ontologies) with traditional Web technologies (such as HTTP and REST) [Bizer 2012]. These data may also contain non-textual material such as maps, genomes, mathematical formulas, medical data *etc.*

Figure 2.9 demonstrates the evolution of the number of datasets published on the Web. Each node in the figure represents a distinct dataset and the arcs show the existence of links between items of these datasets. The thicker arcs correspond to a greater number of links between those datasets. As it can be seen, the Linked Open Data project has initiated the publication of different domain data on the Web by interlinking them with each other. It covers a wide range of different topical domains. A document providing statistics about the content of Linked Data published in September 2011 [23] and it shows that the cloud includes 295 datasets in the form of a connected cloud with over 31 billion RDF triples which are interlinked by around

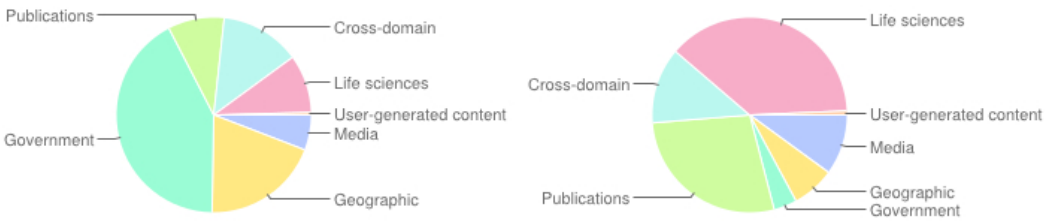



Figure 2.10 – *On the left:* the distribution of triples by domain. *On the right:* the distribution of links by domain (according to the statistics published in September 2011).

19. http://sig.ma/
20. http://www.swse.org/
21. http://sindice.com/
22. http://swoogle.umbc.edu/
23. http://lod-cloud.net/state/

504 million RDF links. The distribution of triples by domain and the distribution of links by domain are shown in Figure 2.10, according to this document. A more recent document published in April 2014 [24] indicates that the number of datasets has doubled since 2011 (from 295 to 571 datasets), with an overall growth of 271%. Figure 2.11 shows a broader view of the cloud according to the 2014 statistics, with different colors corresponding to different domain knowledge.

Datasets in Linked Open Data cloud are published using widely deployed vocabularies in order to make data retrieval easier for the applications. Some of the most widely used vocabularies are as in the following:

- dc (Dublin Core): a set of vocabulary terms used in order to describe the resources, such as videos, images, web pages, *etc.* (http://purl.org/dc/elements/1.1/)
- foaf (http://xmlns.com/foaf/0.1/)
- skos (Simple Knowledge Organization System): an OWL ontology used to represent controlled vocabularies, taxonomies and thesauri. (http://www.w3.org/2004/02/skos/core#)
- geo : an RDF vocabulary for representing latitude, longitude and altitude information in the WGS84 geodetic reference datum. (http://www.w3.org/2003/01/geo/wgs84_pos#)
- geonames : an ontology describing more than 10 million geographic features all over the world with information such as names of places in various languages, elevation, population, postal codes, *etc.* (http://www.geonames.org/ontology#)
- time : an ontology of temporal concepts such as instants and intervals, together with information about durations, and about datetime information (http://www.w3.org/2006/time#)
- dbpedia (http://dbpedia.org/resource/)

In the next two sections, we introduce the DBpedia which is at the core of the Linked Open Data Cloud. Also, we explain the rules that make it possible for everyone to publish data in the LOD cloud.

24. http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/

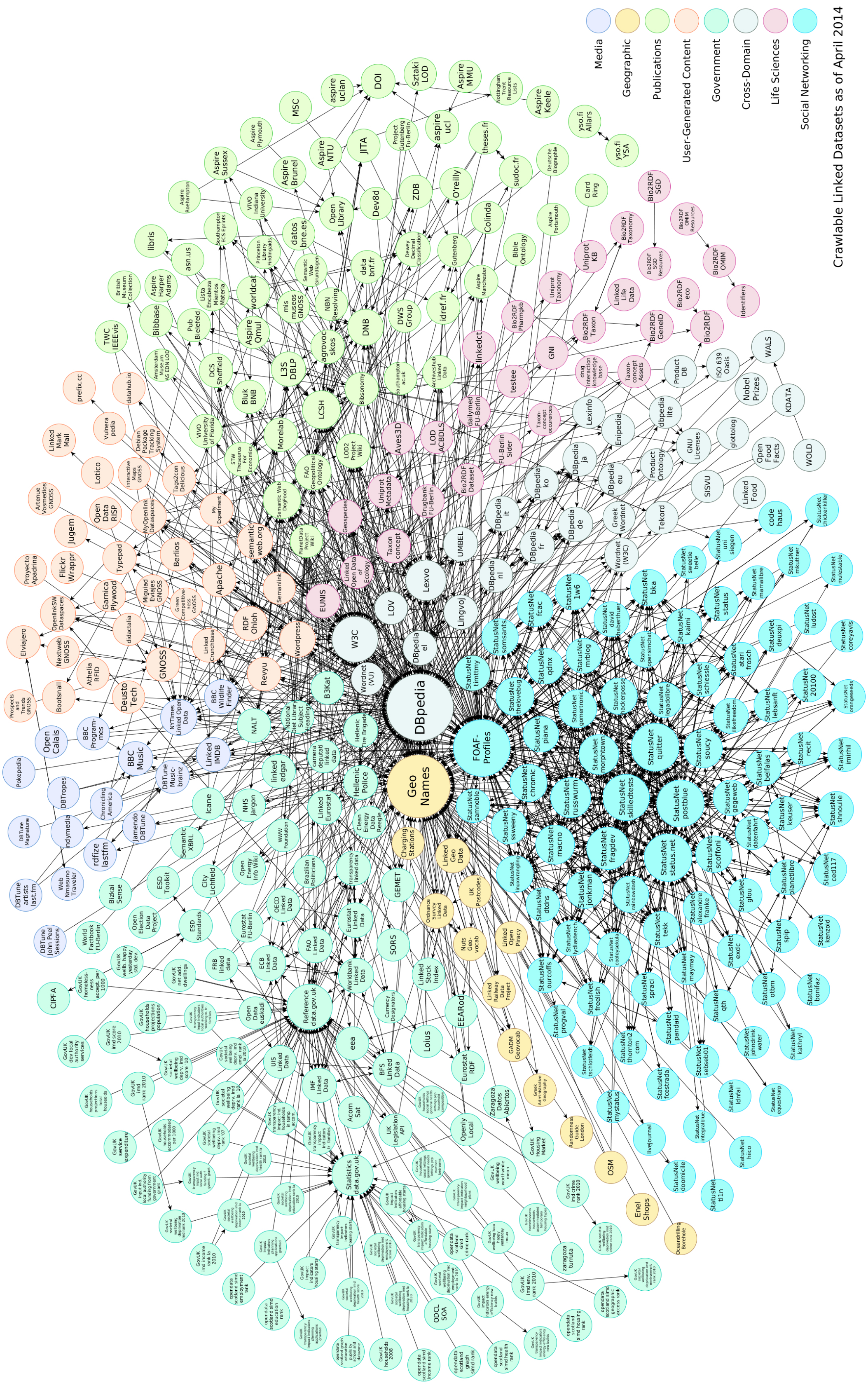



Figure 2.11 – Linked Open Data project as of April 2014 (taken from http://lod-cloud.net/).

### 2.2.5 DBpedia

Some of the datasets in the LOD cloud cover multiple domains. One such dataset is DBpedia, which is a multilingual cross-domain knowledge base [Mendes *et al.* 2012]. The DBpedia [25] is at the center of the LOD cloud (see Figure 2.11). It contains the structured information from Wikipedia, and thus constitutes a base for general world knowledge. Wikipedia is constantly updated with information and is heavily visited. It is a free and open encyclopaedia with the most recent knowledge. However, its search capabilities are limited to full text search [Auer *et al.* 2007]. With the DBpedia project, for every page of the Wikipedia, a URI is created in order to identify an entity or a concept described by a Wikipedia page.

The full DBpedia dataset features labels and abstracts for 12.6 million unique things in 119 different languages; 24.6 million links to images and 27.6 million links to external web pages; 45.0 million external links into other RDF datasets and 67.0 million links to Wikipedia categories. It consists of 470 million RDF triples extracted from the English edition of Wikipedia. [26]

During the extraction process, structured information from the Wikipedia, such as infobox templates, categories, images, geo-coordinates and page links are extracted as RDF triples and added to the dataset as properties of the corresponding entity (thus, its URI) [Mendes *et al.* 2012]. For instance, the extraction of the semantics of an infobox template on a Wikipedia page is achieved via the extraction algorithms that look for pattern matching. Then, the template is parsed and transformed into RDF triples as properties of this entity.

An example of an infobox of an historic landmark located in Paris, in France is shown in Figure 2.12. The Wikipedia article of this landmark is

`http://en.wikipedia.org/wiki/Notre_Dame_de_Paris`

and the corresponding DBpedia URI is

`http://dbpedia.org/resource/Notre_Dame_de_Paris`

The name in the DBpedia URI (Notre_Dame_de_Paris) is derived from the URI of the Wikipedia article. This RDF graph can be visualized on a browser as an HTML page which is available at

`http://dbpedia.org/page/Notre_Dame_de_Paris`

25. http://dbpedia.org/
26. more statistics can be found at http://wiki.dbpedia.org/Datasets39/DatasetStatistics

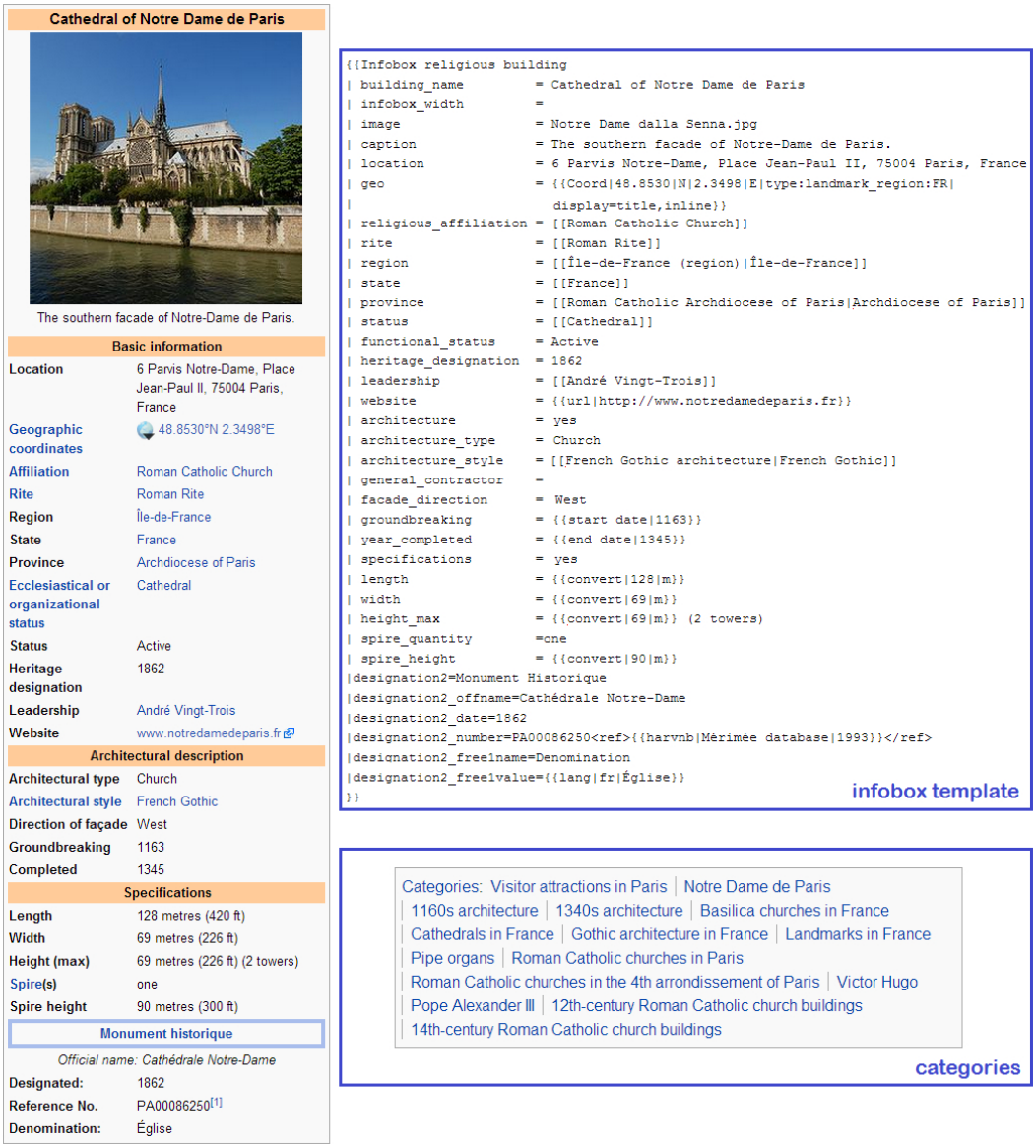



Figure 2.12 – *On the left:* Infobox of a Wikipedia article. *On the right: top*: Religious building infobox template for this article, *bottom:* Categories for this article.

**The Infobox Dataset** Wikipedia infoboxes are important sources of information about the *things* that they describe. They contain structured data that can be retrieved in order to get some summary information about the *thing* that is being presented on the page. They are defined using the templates that are re-used for *things* that belong to similar subjects (*i.e.* organizations, events, buildings, *etc.*). Therefore, modifying the style of a template allows to modify the displayed information at once [Yu 2011].

There are three DBpedia datasets that are extracted from the infoboxes [27]:

1. Mapping-based Types: this dataset contains rdf:type of the instances extracted from the infoboxes, such as for the article *Notre Dame de Paris*:

   ```
   Notre Dame de Paris is a Roman Catholic Church in Paris:
   <dbpedia:Notre\_Dame\_de\_Paris> <rdf:type> <yago:RomanCatholicChurch-
   esInParis>

   Notre Dame de Paris is a Religious Building:
   <dbpedia:Notre\_Dame\_de\_Paris> <rdf:type> <dbpedia-owl:ReligiousBui-
   lding>
   ```

2. Mapping-based Properties: it contains the data values of the instances extracted from the infoboxes. Properties are represented with the http://dbpedia.org/ontology/propertyname naming scheme. Some examples are:

   ```
   Notre Dame de Paris is a Cathedral:
   <dbpedia:Notre\_Dame\_de\_Paris> <dbpprop:status> <dbpedia:Cathedral>

   Notre Dame de Paris has geographic coordinates 48.853, 2.3498:
   <dbpedia:Notre\_Dame\_de\_Paris> <grs:point> "48.853 2.3498"@en
   ```

3. Mapping-based Properties (Specific): it contains the data values of the instances that have specific classes of Dbpedia ontology, such as for the article *Paris*:

   ```
   The area of Paris is 105400000 in meter square:
   <dbpedia:Paris> <dbpedia-owl:area> "105400000.000000"(xsd:double)
   ```

**The Infobox Ontology** The DBpedia ontology is created manually using the infoboxes within the DBpedia. It is generated by mapping the Wikipedia infoboxes to a newly created DBpedia ontology. In the last version of DBpedia (released in April 2015), the ontology covers 735 classes with more than 4 million instances [28]. The instance data in the infobox ontology is better organized than the infobox dataset since some weaknesses in the infobox structure is solved by avoiding having different infoboxes for the same class or using different property names for the same property, *etc.* The classes of the DBpedia ontology can be found at dbpedia.org/ontology/. Some of the classes are Architect, Cardinal, Company, Space Shuttle, Film Festival, Book, Lighthouse, Skyscrapper, *etc.*, to name a few. Each of these classes includes a collection of *property-value* pairs. For instance, some of the properties of Museum class are

27. http://wiki.dbpedia.org/Datasets
28. http://wiki.dbpedia.org/dbpedia-data-set-2015-04

```
<http://dbpedia.org/ontology/collection>
<http://dbpedia.org/ontology/curator>
<http://dbpedia.org/ontology/museumType>
```

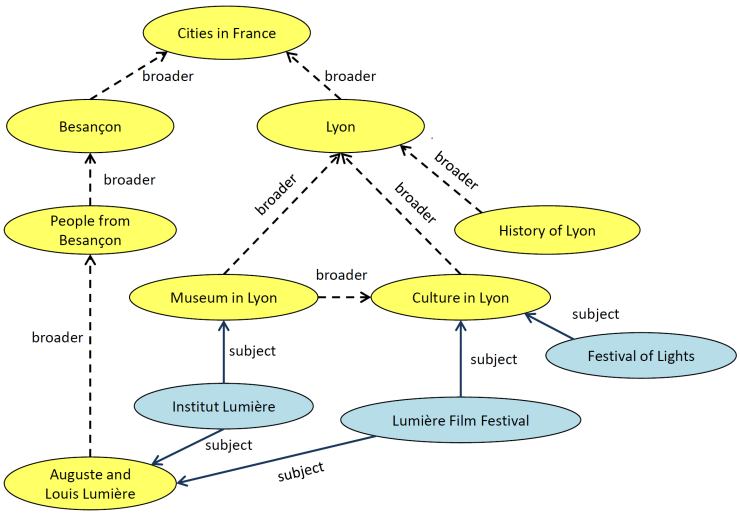



Figure 2.13 – An example of SKOS vocabulary representing the relations between some Wikipedia categories.

**DBpedia Categories** Wikipedia articles are organized in categories, which are represented by the SKOS vocabulary. The thematic concepts of DBpedia datasets rely on the Wikipedia category system. They are organized with two distinct properties, dcterms:subject and skos:broader. The former relates an article to its category and the latter relates a category to its parent category. An RDF graph example that represents some relations between Wikipedia categories is shown in Figure 2.13.

### Querying DBpedia

The DBpedia dataset provides a query possibility with a SPARQL (SPARQL Protocol and RDF Query Language) endpoint. SPARQL is based on *graph patterns* and *subgraph matching*. It is a key element in semantic web technology that can also be used to manage the RDF document [Karmacharya *et al.* 2012]. SPARQL is able to query OWL ontologies that use RDF graph structures. It is recommended by W3C since 2008 as an official query language to retrieve RDF data [Prud'hommeaux & Seaborne 2008]. The SPARQL endpoint of DBpedia is hosted using Virtuoso Universal Server [29].

29. http://virtuoso.openlinksw.com/

SPARQL provides four distinct forms of query: SELECT query, CONSTRUCT query, DESCRIBE query and ASK query [Stadler 2010].

- SELECT returns the result as a table. The results can also be serialized into XML or RDF graph.
- CONSTRUCT constructs a graph from the results.
- DESCRIBE is used to fetch data about individual resources.
- ASK is used to decide whether a query contains any solution or not.

A SPARQL query can be in the following form:

```
PREFIX(Namespace Prefixes)
SELECT(Result Set)
FROM(Dataset)
WHERE(Query Triple Pattern)
ORDER BY, DISTINCT etc.
```

In the following example, the SPARQL query interrogates a specified default graph (by using FROM clause) in order to find names of friends of a person using the property foaf:knows and orders these distinct names alphabetically:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
FROM <http://example.org/foaf/aliceFoaf>
WHERE {
    <#alice> foaf:knows ?friends.
    ?friends foaf:name ?name.
}
ORDER BY ?name
```

Tokens starting with a question mark ( ?) in the query identify the variables which will be substituted with the values from the database that has been queried. SPARQL is based on matching parts of RDF graph with *triple patterns* as in ?friends foaf:name ?name. This is basically called "*query by example*" where the query engine tries to match this pattern with the triples in the database. Each triple is separated with a full stop (.). The difference between an RDF triple and a SPARQL triple pattern is that a SPARQL triple pattern can include variables at any part of the triple, *i.e.* any or all of the subject, predicate, and object values in a triple pattern can be a variable [Yu 2011].

Here is an example of a SPARQL query that can be run on the Virtuoso SPARQL endpoint (*i.e.* dbpedia.org/sparql) of DBpedia in order to ask for the list of capital cities that have more than 2 millions of inhabitants:

```
PREFIX prop: <http://dbpedia.org/property/>
PREFIX dbponto: <http://dbpedia.org/ontology/>
SELECT ?city
WHERE {
   ?country rdf:type dbponto:Country;
    dbponto:capital ?city.
   ?city prop:populationTotal ?population;
     rdfs:label ?city_name.
   OPTIONAL{?country dbpedia-owl:dissolutionYear ?date}.
   FILTER(?population > 2000000 && langMatches(lang(?city_name), "en")
         && (!bound(?date))).
}
```

This query returns only one variable back, which is denoted by ?city. OPTIONAL is a binary operator that is often used with FILTER operator in order to combine several graphs patterns. Here, it gets the dissolution dates of the cities, if that piece of information is available. Then, FILTER is used to add some value constraints, such as the minimum population and language match. It also eliminates the cities that are selected with the OPTIONAL operator, that is to say the cities which are not capitals any more and whose population was less than 2 million. As these constraints are logical expressions, they are combined with logical expression &&. Finally, the query returns the current capital cities in a table.

Recently, the Open Geospatial Consortium (OGC) has introduced an extension to SPARQL, namely GeoSPARQL [30]. The spatial data in the LOD cloud has increased tremendously over the past decade. GeoSPARQL aims at easing the access to geospatial data, exploit and express spatial relations. In order to provide geospatial reasoning and querying in a triple store, GeoSPARQL defines a small ontology for representing features and geometries with some SPARQL query predicates and functions [Battle & Kolas 2012]. Parliament [31] is a triple store compatible with GeoSPARQL (also with other W3C standards, such as RDF, RDFS, OWL, SPARQL, *etc.*). Using the geospatial index of GeoSPARQL, Parliament can answer

30. http://www.opengeospatial.org/standards/geosparql
31. http://parliament.semwebcentral.org/

queries like "find all items located within region X". Besides the geospatial index (such as points, polygons, *etc.*), it also includes a temporal index (for indexing intervals and points in time), and a numeric index (for optimizing range queries on numeric property values) [Battle & Kolas 2012]. As for DBpedia, a query like "Show all the museums within 3 km of the City Hall of Lyon" will be possible. However, geospatial data on DBpedia is expressed with the WGS84 ontology which provides basic terms in RDF to describe point locations (*i.e.* their latitude and longitude). Parliament provides also a conversion function, which takes a latitude and longitude pair and converts it to GeoSPARQL geometry representation [Battle & Kolas 2012]. When Parliament is implemented to a geospatial dataset in the LOD cloud, such as to GeoNames [32] or LinkedGeoData, GeoSPARQL can be easily used to retrieve data. This way, topological relationships between two geometries can be known, such as intersections (containing, touching, crossing, within, overlapping, *etc.*). Thus, interesting queries taking advantage of the geospatial features can be answered by the LOD. For instance, geof:sfWithin predicate of GeoSPARQL (given that geof: <http://www.opengis.net/def/geosparql/function/>) can be used to query religious landmarks within a boundary (*e.g.* a city); or all the monuments or information available within the boundary of a park.

### 2.2.6 Publishing Data on the LOD Cloud

Many governments today initiate the publication of governmental data for all. The motivation of this intention is to provide efficiency, support transparency and accountability, and ease the re-use of published data. Open data government initiatives such as Data.gov (launched in May 2009) and Data.gov.uk (launched in September 2009) have started with the idea of providing access to the linked government data and open ways to convert, enhance, and use government datasets, as well as to generate resources on how datasets were derived or used [Ding *et al.* 2010]. Raw data catalogues are made available in formats such as CSV, Text, Microsoft Excel, XML, KML, *etc.*, which are popular as they archive data as a clear structured text that is easy to manipulate by software programs [33]. In order to consume and visualize these data, Linked Data principles introduce some international standards (*i.e.* data format requirements and data descriptions, *etc.*) in order to achieve the goals of the open government initiative [34]. Generating governmental linked data

32. http://www.geonames.org/
33. http://www.w3.org/TR/gov-data/
34. http://www.w3.org/2011/gld/wiki/Linked_Data_Cookbook

consists of some tasks such as finding a suitable ontology or vocabulary to model the data, naming things (ontologies, vocabularies, real-world things, datasets, *etc.*) with URIs, converting data to RDF, linking with other datasets, *etc.*

Linked Data relies on the RDF data model with globally unique identifiers as entities and it allows different schemata to be used in parallel to represent data [Heath & Bizer 2011]. The access mechanism for datasets is standardized as HTTP protocol and these datasets are interlinked to each other using shared vocabularies that are also connected with vocabulary links. These properties of Linked Data architecture make it possible to integrate new data.

Tim-Berners Lee described a 5-star rating system in order to publish data as Linked Data:

- No star: Data is not available under an open license.
- 1 star: Data is available on the Web with an open license, but it cannot be read easily by a machine, so it cannot be re-used (*e.g.* image or pdf file).
- 2 star: Data is accessible on the Web as a structured and machine readable format, thus it is interpretable by the machines with a proprietary software (*e.g.* excel file).
- 3 star: Data is available on the Web in a non-propriety format (*e.g.* CVS file)
- 4 star: Data uses W3C open standards (RDF, SPARQL, *etc.*) and thus accessible via a URI (*e.g.* RDF file that uses URIs as a set of triples)
- 5 star: Data is interlinked to other data, which makes it easier to find through other datasets (*e.g.* Linked Open Data).

If we sum up the benefits of a 5-star Linked Data, on the consumer side: (1) more (related) data can be discovered while consuming data; (2) the ease of data integration and re-use makes is cost effective way of sharing data; (3) data schema is accessible; on the publisher side: (1) it increases the expressiveness, quality and value of the data; (2) data becomes visible and discoverable.

In order to be added to the Linked Data Cloud, the dataset should meet the following requirements [35]: (1) There must be resolvable URIs; (2) They must resolve to RDF data in one of the RDF serialization formats (RDF/XML, Turtle, N-Triples, *etc.*; (3) The dataset must contain at least 1000 triples; (4) The dataset must be interlinked via RDF links to a dataset on the LOD cloud; (5) At least 50 links are required; (6) The entire dataset must be accessible via RDF crawling, via an RDF dump, or via a SPARQL endpoint.

35. http://lod-cloud.net/

Interlinking of two OWL ontologies can be done in various ways according to [Stadler 2010]:

- *Class-level* interlinking related two classes with relations such as owl:equivalentClass or rdfs:subClassOf.
- *Instance-level* interlinking relates instances with each other using for instance owl:sameAs.
- Instances of one ontology can be interlinked with the classes of another ontology.

In the following section, we present the methods for extracting information from the LOD cloud and different ways to use semantic information on web and mobile platforms.

## 2.3 Different Ways to Use Semantic Information

Since the introduction of the Semantic Web, the Web is used as a different medium: a medium for information search and discovery. The LOD project has become very popular in the last years from an Information Retrieval (IR) point of view. Several recommender systems have been designed to extract the best piece of information from the cloud, but the LOD cloud has also been used for recommending interesting related information by following the interlinks. These systems are applied to Web-based media, as well as mobile ones.

> *"The Web, they say, is leaving the era of search and entering one of discovery. What's the difference ? Search is what you do when you're looking for something. Discovery is when something wonderful that you didn't know existed, or didn't know how to ask for, finds you."*[36] *Jeffrey O'Brien*

### 2.3.1 Extracting Information from the LOD Cloud

In order to consume and process the data available on the LOD cloud, there has been a growing number of systems developed. Some of them focus on the approaches based on *recommender-systems* for information filtering. They calculate a similarity value between two items (*i.e.* resources, properties, names, item's features, user's interests, preferences *etc.*). In [Noia *et al.* 2012], in order to calculate similarities between two movies in three different datasets, *i.e.* DBpedia, Freebase and LinkedMDB (semantic-enabled version of the Internet Movie Database (IMDB),

36. http://money.cnn.com/magazines/fortune/fortune_archive/2006/11/27/8394347/

authors adapt the Vector Space Model (VSM) (see [Salton *et al.* 1975]) which is one of the most common models in classical information retrieval. By representing the whole RDF graph as a 3D matrix where each dimension refers to an ontology property, the authors semanticize the VSM. Then, each movie is seen as a vector and the similarity degree between two movies is calculated with the correlation between the two vectors (*i.e.* cosine of the angle between the vectors). This allows the system to recommend movies to users according to their profile and preferences.

The LOD cloud has also been used as an information repository thanks to its RDF structure and semantic relations. In [Mirizzi *et al.* 2010], a ranking algorithm *DBpediaRanker*, which analyzes the DBpedia graph and calculates a similarity value is introduced. The authors calculate the similarity value using the internal links (Wikilinks) and external information sources that appear as references on a DBpedia resource (Google, Yahoo !, Delicious, *etc.*). During this graph exploration, resources are ranked according to their relatedness, which is a common process in information retrieval. This and similar approaches (such as in [Franz *et al.* 2009], [Toupikov *et al.* 2009],[Blanco *et al.* 2011]) can be used to rank resources in any RDF-based web documents and improve the search results. When a new dataset is introduced in the LOD cloud, it is interlinked with one of the existing datasets on the cloud that have similar and trustable data. DBpedia is the most important datasets on the cloud right now. As the number of datasets linked with the DBpedia, its credibility also increases since direct links can be used for rank calculation [Mulay & Kumar 2011].

[Stankovic *et al.* 2011] also proposes a method that explores the RDF graph of DBpedia in order to suggest related topics. This approach is based on combining relevant DBpedia concepts (*i.e.* ontology types and categories) and filtering them out based on their type. The authors identify the relevant concepts by calculating the proximity of topics related to each other with the skos:broader relation. Therefore, an algorithm calculates the proximity of concepts and accordingly, the importance of nodes at higher distances drops exponentially. Then, they filter out the unwanted concepts by using a pre-created list to avoid the execution for each concept. Another relation discovery approach for DBpedia is introduced in [Heim *et al.* 2010] through the *RelFinder*[37] algorithm which extracts relations between two fields. The user first enters texts into two text fields as input. The algorithm uses Jaro-Winkler distance measure in order to extract the matching properties in the dataset and rank them according to their popularity (*i.e.* by counting the mat-

37. http://www.visualdataweb.org/relfinder.php

ching properties, the popularity of the field is determined). The user then chooses the property strings among the list. The relationship between these two properties is calculated by using two parameters: *length* and *directionality*, which correspond respectively to the number of intermediate objects between two nodes and to the number of changes in the directions of the links.

There are also works focusing on a multimedia information retrieval approach applied to the LOD. Complementing a textual description with audio and visual information improves user experience according to [García-Silva *et al.* 2011]. However, searching for a multimedia content via search engine is still an unresolved problem. Then, [Roa-Valverde 2011] presents *playence Media* for multimedia retrieval by content augmentation. This approach is based on interlinking domain ontologies with DBpedia ontology in order to extend the query (with other related information) for retrieving multimedia documents with the same annotations. Therefore, the information that is readily available on LOD is used to enhance media annotation of a document. The aim of the approach is twofold: (1) the interlinking of domain knowledge complements the existing annotations of a multimedia document, thus results in a query expansion; (2) such added information allows to query additional concepts related to the document, which is known as content augmentation. In [García-Silva *et al.* 2011], the authors introduce a system called *Multipedia* that uses the Wikilinks in order to collect context information about each Wikipedia article. Here, the semantic neighborhood between a DBpedia resource is expanded, and then, the relatedness between a DBpedia resource and an image is calculated based on the overlapping terms in the set composed of the context words and image tags. Thus, these context words are used to retrieve images with similar tags. The similarity is calculated using a Vector Space Model as in [Noia *et al.* 2012].

### 2.3.2 Examples of Applications based on the LOD Cloud

In [Mirizzi *et al.* 2012], the authors introduce a *Facebook* application, *MORE* (MORE than MOvie REcommendation) that exploits LOD and computes semantic similarities among movies using a semantic version of the Vector Space Model (called sVSM) (inspired by [Franz *et al.* 2009]). This system relies on DBpedia and LinkedMDB for allowing a metadata navigation and exploration. It extracts the favorite movies of the user that are available on her Facebook profile. The DBpedia properties dcterms:subject and skos:broader, which organize the hierarchical structure of the categories, are used to calculate the similarities between movies. The

adapted representation for the RDF graph can be seen in Figure 2.14. Thus, for a selected *property*, a *movie* is represented with a vector containing all the nodes related to the movie with this property.

Another project that makes use of the Linked Open Data is carried out by the BBC, where the online content of the BBC (in text, audio and video formats) is linked to the LOD cloud [Kobilarov *et al.* 2009]. The aim of the project is to provide meaningful navigation paths for users that include semantic links across different domains, since it is difficult to present a coherent website [38] with the lack of structured and linked data. That way, users will also be able to traverse the graph of the BBC content towards other data sources on the cloud, and the content will be enriched with additional data available (short descriptions, temporal and geo-location data, *etc.*) and the rich hierarchical relationships of DBpedia. The authors use web identifiers to denote *things*, as described in Linked Data principles, and interlink them to the corresponding entities on DBpedia with owl:sameAs relations. This solves the problem of cross domain linking of concepts and content since the domain-specific information (such as music, events, food, tv programmes, *etc.*) has been classified separately.

LOD is broadly used also in the tourism domain. In [Sabou *et al.* 2012], the authors explore the use of Linked Data by transforming a dataset of European tourism statistics, TourMIS, into RDF and combine it with data from other sources to support decision-making scenarios in tourism applications. For the triplification process, the authors design an ontology to describe tourism related measurements and concepts. DBpedia is then exploited to identify cities and countries. Cultural

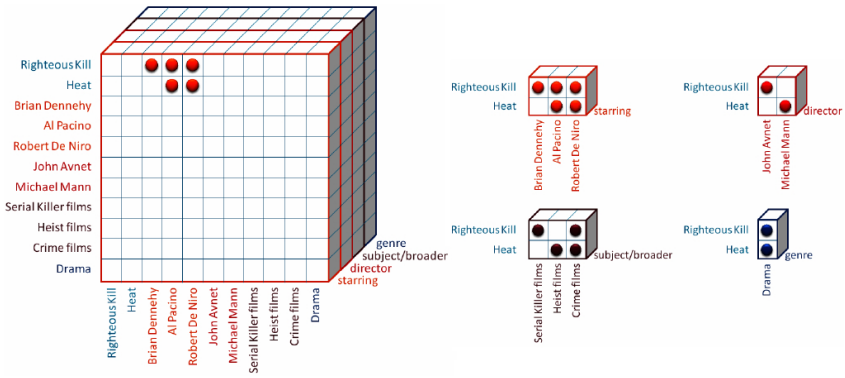



Figure 2.14 – The whole RDF graph is represented as a 3-dimensional tensor in [Mirizzi *et al.* 2012] where each component refers to an ontology property.

38. http://www.bbc.com/

heritage data is also tripled with the 5-star method of Tim-Berners Lee, in order to augment the user experience. The mobile applications based on libraries, museums or archive collections appeal to visitors as they offer new interfaces that enhance the interactivity during the exploration of cultural heritage items and stories related to them. In [Törnroos *et al.* 2013], the authors interlink the historical collections related to the First World War (WW1) with Linked Data. This allows the 261 wartime events (occupations, resistances, wartime hunger, *etc.*) provided by the Imperial Museum (with terminologies related to battle) to be connected to DBpedia with owl:sameAs relations. Similarly, place names are interlinked with the GeoNames dataset. While the wartime events dataset is regularly enriched, the authors provide RDF dumps for exporting, SPARQL endpoint functionalities (Browser and Editing View: `http://purl.org/ww1lod/saha` and a SPARQL endpoint `http://sparql.onki.fi/ww1lod/sparql`).

In Japan, several actions have been conducted under the Linked Open Data for ACademia (LODAC) project [39] (which results in over 40 million triples extracted from 114 museums and research institutes in Japan) as one of the activities of the open government data initiative. Museum data, biodiversity and species data, location data dealing with addresses, zip codes, *etc.* are exposed with this project. As one of the open government data initiative projects in Japan, the LODAC Museum integrates museum information across multiple resources, in order to establish a cycle of *Publish*, *Share*, *Collect*, *Use* and *Create* [Kamura *et al.* 2011]. Different types of information related to artists, artworks and museums are identified and integrated as in Figure 2.15. The authors provide a SPARQL endpoint for querying the RDF data. As a part of this project, a collaborated research activity has been conducted by LODAC Museum, Yokohama Art LOD (which provides event information from museums and galleries and maintained by a local community) and PinQA (a local Q&A dataset maintained by a company) [Matsumura *et al.* 2012], as shown in Figure 2.16. As a result of this collaboration, a web application, Yokohama Art Spot [40], is developed. It provides three different views: 1) map view (the locations of current events), 2) institution view (basic information about the institution with event and collection information), and 3) administration view (addition or correction of data by the staff who is not familiar with RDF) embedding data obtained with SPARQL into templates of web pages.

39. `http://www.ontotext.com/case/Lodac`
40. `http://lod.ac/apps/yas/` (in Japan)

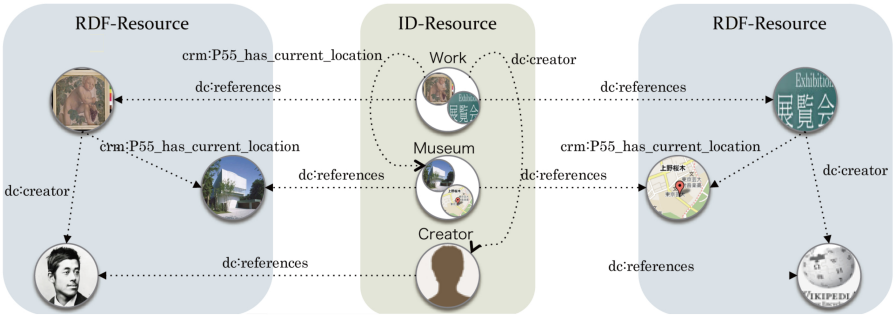



Figure 2.15 – Data Integration in LODAC project [Kamura *et al.* 2011].

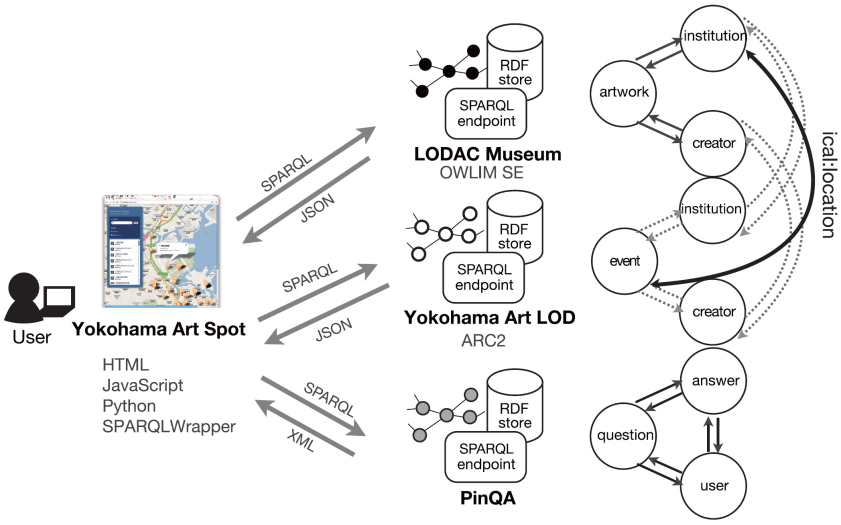



Figure 2.16 – The system overview of Yokohama Art Spot that integrates three data sources [Matsumura *et al.* 2012].

Another web-based cultural heritage service for tourists is *CultureSempo*[41] which consists of the collections of 22 Finnish museums, libraries, archives, and other memory organizations. The web application is developed in order to present heterogeneous cultural heritage data as they are represented in Figure 2.17. It provides a content creation process with distributed content and ontologies. The major distinction of the proposition is that the end-user can access resources with different thematic views: map views, relational search (how people and organizations are related), collections view, Finnish history view, biographies, *etc.* [Hyvönen *et al.* 2009b]. The history view delivers semantic recommendations to other contents (such as biographies of people participating an event) and this creates narrative stories. As described in a previous work of the project [Junnila *et al.* 2006], stories give bet-

41. http://www.kulttuurisampo.fi/?lang=en

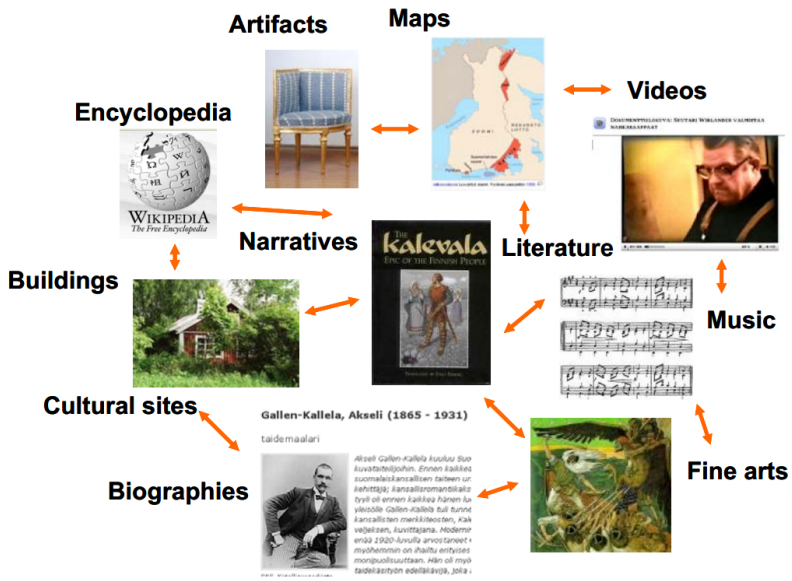



Figure 2.17 – Cultural heritage is semantically heterogeneous and mutually linked as stated in [Hyvönen *et al.* 2009a].

ter views to the end-user. The authors use ontologies to create a logic for binding situations together in order to construct stories. For instance, a situation ontology describes events related elements, such as *actions*, *actors*, *time*, *place*, *etc.* The goals of the project are to create a single repository for distributed collections, to provide content-based information retrieval based on ontological concepts, to interlink the collection items together by using semantic links, and to ease the local content publication.

In [Kauppinen *et al.* 2011], researchers discuss the importance of spatio-temporal references to cultural objects that have been implemented in the CultureSampo project. Spatial information is widely used in cultural resources as a geographic reference attached to the document. Such as in [Gadolou *et al.* 2010], a digital storytelling project for a historical event (a battleship) is introduced. The authors enrich the multimedia documents with spatial information and annotations in order to automatically generate a series of virtual visits that narrate historical events. For the CultureSampo project, a spatio-temporal ontology is used in order to enrich the cultural object collections with such information [Kauppinen *et al.* 2011]. This ontology contains historical Finnish locations with the relations between them, and temporal properties (if the location still exists or not) as well as spatial properties (*e.g.* center points, polygon boundaries, *etc.*). For instance, a location can be a union of three temporal parts as in the following:

```
sapo:Imatra
  sapo:unionof
    sapo:Imatra(1948-1973),
    sapo:Imatra(1974-1976),
    sapo:Imatra(1977-).
```

In this example, the temporal parts of the place Imatra have different spatial extensions such as boundries defined by polygon coordinates. The topological and geometrical relations are expressed with spatial attributes such as isPartOf, equals, contains, intersects, isCenterPointOf, *etc.* The recommendation engine can then query information to enrich the historical maps with different time periods and locations using these relations such as topological ones (overlaps, disjoints, *etc.*) or directions (north, south, *etc.*).

### 2.3.3 Semantic Information on Mobile Devices

Mobile devices are interesting tools for the evolution of the Web, as they represent a medium for the new generation of context-aware applications. In such applications, information about location enables users to access location-based contents as they are on-the-move. The Semantic Web contains large amount of location-based information which is an interesting data source for these applications.

Figure 2.18 – The map view of DBpedia Mobile [Becker & Bizer 2008].

One of these applications, the *DBpedia Mobile*[42], is introduced as a mobile application that renders a 2D map indicating nearby locations from the DBpedia dataset [Becker & Bizer 2008], [Becker & Bizer 2009]. Starting from this map, users can explore the related content following data links towards other web data sources (see Figure 2.18). This also allows users to publish their current location, pictures and reviews which are interlinked with a nearby DBpedia location. When the user

42. http://wiki.dbpedia.org/DBpediaMobile

selects a resource, DBpedia Mobile incorporates Linked Data from Geonames, the flickr wrapper, Revyu (a publicly accessible reviewing Website that generates RDF metadata interlinked with the Linked Data) and DBpedia to display information. If there are some RDF links in those resources, users may interact with them to get further information. The user may also filter the map and set conditions such as *resource types* (*museums*, *train stations*, *etc.*), *ratings*, *etc.*

It is stated in [Costabello 2011] that mobile devices provide a deeper awareness of the surrounding physical environment and that delivering effective user interfaces for RDF resources is a well-known problem in Linked Data applications. With this context, the *PRISSMA* vocabulary is proposed so as to provide dynamic adaptations across mobile applications consuming Linked Data (see Figure 2.19). The vocabulary consists of three core classes: Environment, User and Device. Environment models the physical world where the Linked Data resources are accessed. Some dimensions involved are *Points of Interest (PoI)*, *time* and *activity*. According to the activity of the user (*e.g. running*, *driving*, *shopping*, *etc.*), the further sub-properties (*e.g. shopping purpose*, *shopping environment*, *etc.*) can be associated with the user representation in order to present context-aware information.

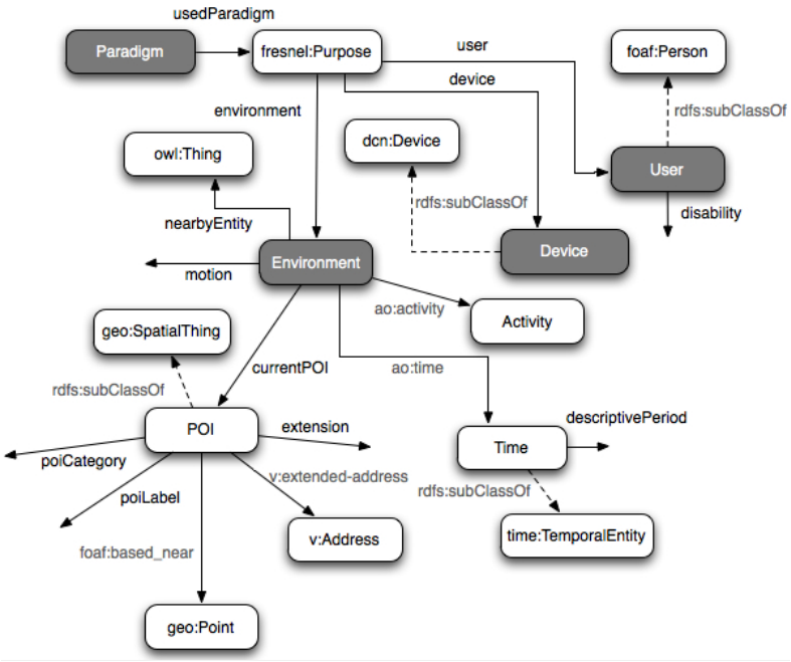



Figure 2.19 – The PRISSMA vocabulary proposed in [Costabello 2011].

In [Celino *et al.* 2012], a mobile and location-aware game for the city of Milano, *UrbanMatch*, is introduced (see Figure 2.20(a)). The authors point out the

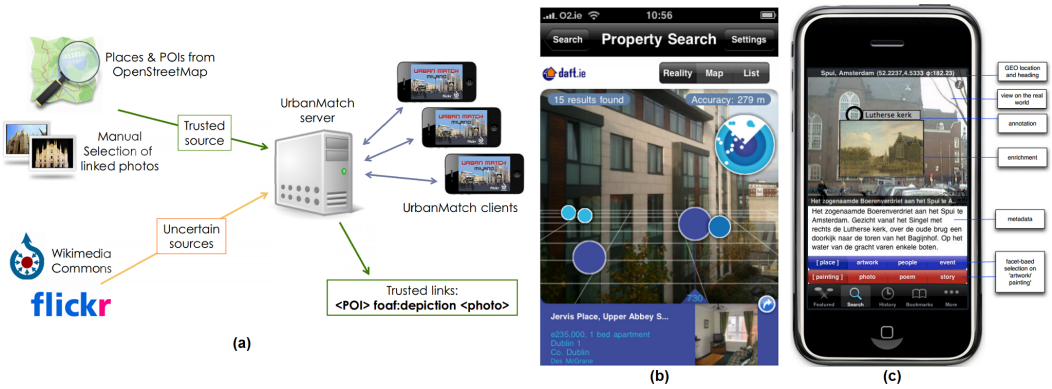



Figure 2.20 – (a) Input data sources and output links in the AR application *UrbanMatch* [Celino *et al.* 2012]; (b)a screenshot from the AR application proposed in [Reynolds *et al.* 2010]; (c) AR view of a church in [van Aart *et al.* 2010]

.

advantages of using Linked Open Data as it provides information quality, because its accessibility over the Web and the use of shared vocabularies make it easier to verify data integrity and infer implied data. They also remark that the quality of urban Linked Data can be at unpredictable ranges and at times, it is impossible to detect missing information and inconsistencies. However, this can be improved by a small amount of manual work according to the authors. They propose improving this with a mobile game that requires *physical presence* of the user. The game provides user with eight photos of the PoIs in the vicinity and ask to relate the photos with the PoIs and rank them with a *confidence value*. As a result, they intend to provide a valuable dataset that contains linked and ranked photos of places.

Cinemappy is a context-aware mobile application that recommends movies based on the location of the user [Ostuni *et al.* 2012]. It recommends movies to watch in nearby cinemas and closeby places such as *restaurants*, *bars*, *etc.* [Reynolds *et al.* 2010] present an Augmented Reality mobile application that displays PoIs (*historical sights*, *nearby bus stops*, *cafes*, *etc.*) obtained from the execution of spatial queries on some datasets of the LOD cloud (see Figure 2.20(b)). Another Augmented Reality project that incorporates LOD cloud is SmartReality [Nixon *et al.* 2012]. The project focuses on a music scenario where the music posters are enriched with information about *music artists*, *albums*, *events*, *etc.* [van Aart *et al.* 2010] explores cultural heritage repositories by using the location of the user and uses Augmented Reality-based interfaces to present these information (see Figure 2.20(c)). The user can interact with the landmarks by pointing her device towards them. The system uses the ontologies of Linkedgeodata, DBpedia and some datasets related to Dutch

historical buildings. Then, it analyses the ontological characterization of the surroundings using the geolocation of the user. The retrieved PoIs are combined if they refer to the same physical location by using spatial reasoning techniques. A general information and a specialized information about the PoI is presented to the user. The general knowledge is collected from datasets such as Freebase, DBpedia, GeoNames, *etc.* Specialized knowledge about the cultural heritage domain is retrieved from art and architecture thesauruses and Dutch cultural heritage institutions. A facet hierarchy is pre-defined which can be also modified by the user. This hierarchy refers to the types of information that will be gathered by the application. Some facets are *artwork*, *painting*, *people*, *etc.* and some subfacets are *painting*, *photography*, *book*, *artist*, *etc.*

As it can be seen from these examples, semantic information in LOD sources has been exploited by mobile applications in various scenarios. LOD is constantly growing with newly added datasets. These resources are promising as they can be integrated easily in mobile applications by exploiting the structured knowledge sources and provide higher reusability, which makes them well-suited for organizing content in mobile applications.

## 2.4 Conclusion

As the Semantic Web evolves, the way we share and publish knowledge on the Web also changes. Documents are linked to each other using *hyperlinks* that allows readers to immediately access other related information. Web crawlers retrieve information traversing these hyperlinks. The Web has also evolved from being a *read-only* medium to *read-write-execute* environment, where readers also contribute by creating and for sharing their own content. Following these developments, the *Semantic Web* has been introduced by the international organization, W3C, which has designed the standard languages for the creation of the machine readable content and sharing and reusing it across various applications. Ontology languages have become the main building blocks of the Semantic Web which provide more expressive knowledge representation. Detailed, accurate, consistent and meaningful distinctions can be made using classes, properties and relations of ontologies [43]. They facilitate exchanging data and enable machines to perform meaningful reasoning tasks.

In recent years, as the Web has become a global information space where one

43. http://www.w3.org/TR/webont-req/

can easily access to knowledge, some challenges have also aroused. Initially, the documents published on the Web were linked to each other, but the data within was not structured, thus the semantic meaning in the document was not easily accessed. The Linked Open Data project has enabled the transition from "*Web of Documents*" to "*Web of Data*" and has extended the Web with structured data that can be crawled easily. Linked data has been adopted globally by data publishers in variety of knowledge domains such as *geographic*, *media*, *libraries*, *user-generated content*, *cross-domain datasets*, *etc.*

Linked Open Data has further enabled establishing new data spaces easily by using data links and vocabularies available on the cloud. Spatial information has played an important role in the development of location-based applications both on the *Web* and *mobile* environments. Mobile technologies have enabled developing applications with a spatial point of view, as well as temporal information has been manipulated and presented to the user in different interactive mediums (with Augmented Reality and/or in audio, video, textual forms, *etc.*)

In this chapter, we have analyzed the development of the Web as the needs evolve. We have also mentioned the current principles in order to publish data on the Web and analyzed commonly used Web technologies. LOD has improved the quality of data on the Web and established a trusted environment for data publishers. We have introduced current methods that use the LOD as the first source of information for the discovery of the surroundings, and finally the location-based applications and the popular trends that have been used on the mobile.

In the next chapter, we make a synthesis of the surroundings discovery approaches and the use of semantic information on mobile devices. We mention the importance of these topics for our approach, before giving an overview of our proposition in Chapter 3.

# Synthesis

In the previous two chapters (Chapter 1: *Surroundings Discovery Applications* and Chapter 2: *Semantic Information in Mobile Applications*), we have analyzed the existing works related to surroundings discovery mobile applications (from sensing to visualization with Augmented Reality) and different approaches that use semantic information on mobile devices.

Even though these two research domains (Augmented Reality and the Semantic Web) may seem very distinct from each other, both of them have been explored by researchers to develop methods to access information on mobile devices. Augmented Reality (AR) helps to seamlessly integrate a piece of information by overlaying some digital content on the real view and facilitate visualization. The Semantic Web, on the other hand, is used effectively to facilitate information access since it provides standards for the representation and retrieval of knowledge. The concepts, properties and relations between these concepts are defined using ontologies which provide shared vocabularies to structure the data. This common knowledge space has been extensively used on mobile devices as well.

The Linked Open Data (LOD) cloud has a great potential as an information base, as it embodies easily accessible public datasets which are continuously extending. These data models are used for capturing all kinds of information (such as object descriptions) using query algorithms on mobile devices. The cloud also consists of location-based data, associated with geo-location attributes, names of places, *etc.* Developers may publish their own datasets, provided that they follow the Linked Data principles. This brings visibility and re-usability to their data, as these can be accessed as a structured data source on the Web and exploited by any other application which uses the LOD cloud as an information source. Instead of using one local database as the source of information, linked data sources can improve the presentation of information in a meaningful and dynamic way, thanks to interlinked information sources and semantic concepts. This way, knowledge management (actuality of information, efficient storage and retrieval) is performed collaboratively based on linked data standards. Used in a location-based mobile application, the available structured data may be exploited in querying and grouping Points of Interests (PoIs) according to different criteria (such as the types of objects, relatedness with each other, *etc.*) and accessing information associated to them.

On the other hand, mobile devices nowadays are extensively used for ubiquitous access to information, due to their computational power, the various sensors that they embed and their ability to send and receive data with wireless communication. Information can be processed taking into account the *context* from sensor data, further adapted to user needs, preferences, location, *etc.* and visualized in various different ways. Thanks to the advanced mobile graphics, computer vision and high speed connections (3G/4G, Wi-fi, bluetooth, *etc.*) available in outdoor environments, 3D map visualization has become possible on mobile devices. New applications allow displaying objects around the user by using geo-location and orientation as context elements for updating the 3D view. Augmented Reality, on the other side, has opened up a broader perspective by presenting information in a way which makes it easier to interpret the environment as the information is directly inserted (overlaid) over the objects of the captured scene.

At the frontier of these three domains (Semantic Web, Augmented Reality and surroundings discovery using mobile devices), we develop an approach that facilitates location-based information discovery through an architecture that is adapted for various mobile application scenarios. We consider the superimposition of the objects as overlaying a digital layer on the captured view to facilitate the interaction with the real-world objects. The AR objects that we refer to here are the lightweight 3D geometries of the surrounding real-world objects. These transparent 3D objects are overlaid on the captured scene and the user can interact with them to access further information. We are interested in providing answers to the following questions: *What information to present ?*, *Where is this information coming from ?*, *How to filter this information ?*, *etc.* At this point, the Semantic Web plays a great role. The LOD cloud can provide numerous opportunities to surroundings discovery AR applications. The seamlessly superimposed 3D objects can act as a bridge to the information sources available on the LOD cloud. In order to achieve this goal, we focus on how to manipulate this knowledge base such that, by interacting with the 3D objects on the mobile device's screen, necessary information can be queried from the cloud and displayed to the user. To achieve this, we analyze how to interlink these 3D models with the LOD resources (namely DBpedia), hence, publish these geometries as a LOD dataset. Once they are available on the cloud, by following the links, information related to these objects can be retrieved. While designing the architecture with this approach, we focus on its re-usability and adaptability to various application scenarios. We develop a data model that we use for crea-

ting the interconnection of the 3D geometries with the LOD resources. This data model contains necessary information that we think is interesting to manage and manipulate while designing an AR application (such as the temporal validity of the 3D object, the type of the building, the level-of-detail of the 3D geometry, *etc.*) The availability of different representations of these real-world objects on the LOD cloud (such as *how a building looked like 30 years ago* and *what was its function*, *etc.*) enable application designers to tell stories related to these objects by attaching these geometries with some multimedia documents (discussed in Chapter 6). This way, for instance, the user can have access to what a church looked like 30 years ago by looking its 3D model, as well as a photo from that time. In the rest of this dissertation, we present our approach which is based on these ideas.

The second part of thesis is then divided into four chapters and organized as in the following: Chapter 4 introduces our data model and ontology, that we use to interconnect 3D AR objects with the data sources from the LOD cloud. We emphasize a 5-dimension information model to represent these objects, which are spatial (3D), temporal and thematic data, to provide information for any kinds of AR applications. In Chapter 5, we further analyze this model in order to show how it can be used extensively by an AR application. We also describe the implementation that we have done in order to demonstrate how this data model is served to query information from the Semantic Web through a mobile AR application. Chapter 6 describes the general architecture and its components, while a use case scenario shows how to build and tell a story related to a POI. Finally in the last chapter, we conclude (see Conclusion).

# Part II

# Proposition

Chapter 3

# Overview: A New Augmented Reality Approach

**Sommaire**


3.1 Introduction . . . . . 95
3.2 Overview . . . . . 98
3.3 Conclusion . . . . . 103


## 3.1 Introduction

Augmented Reality (AR) is a powerful tool to improve visualization for the user and helps her to perceive her environment. As it is seamlessly integrated on the real-world view, by just pointing the mobile device, the user can easily access to information related to her surroundings. The AR content is overlaid on the real-world view and, sometimes, further interaction provides additional information to the mobile user. Technical improvements in mobile technology have allowed the integration of Augmented Reality in location-based information discovery. Using AR, multiple users can share the same physical space and perceive a digital experience which is tied to a specific location.

On the other hand, we observe that the Linked Open Data (LOD) cloud has become an important source of information over the last years, and it is constantly growing with additional datasets and community efforts. The LOD is the source of vast amount of structured data which are referenced with geolocation properties, *i.e.* latitude and longitude, place names, *etc.* These location data are easily exploitable by location-based AR interfaces which can be an adapted solution for the discovery of surroundings by providing access to these information resources through mobile devices.

We focus on the use of AR in outdoor environments. Many scenarios can be imagined in these spaces, such as the user may have no idea about where to start her

visit of the surroundings, or she may demand domain specific information, if there are any, related to her environment, or she may want to access *any* information related to an object within her vicinity. Each of these scenarios are separate and might result in developing different AR applications (different repositories, access and interaction methods, *etc.*). Today, digital location-based information (including 3D models) available in data repositories over the Web can be adapted to different user profiles and application scenarios. However, there is no widely accepted standard and approach that can be embraced by different application domains. On the other side, location aware mobile devices are continuing to offer countless possibilities.

We consider the possible advantages of building a generic model that allows us to exploit these structured data repositories with 3D objects in order to generate application scenarios for AR applications. By providing greater access to 3D AR models, interlinking them with linked data resources and exploiting these links using intelligent services, location-based data could be presented with much meaningful descriptions. If the interface is user-friendly, accessing these data through AR interfaces would be a trivial but also pedagogical, playful and instructive task of our daily lives. However, to the best of our knowledge, there is no such architecture that constitutes a backbone to create LOD-based mobile AR applications that use 3D geometries for surroundings discovery.

We develop our approach on how mobile augmented reality can be used for surroundings discovery in urban environments. The 3D augmented space can be *programmed* to retrieve location-based information on the LOD cloud and to visualize these information with a user-friendly interface [Aydin *et al.* 2012a], [Aydin *et al.* 2012b]. We propose an architecture that involves specific components in order to facilitate AR application development for various domains. We suggest that the location-based digital objects can be identified using resources on the LOD cloud and the user can ask for information according to her preferences by interacting these objects.

AR is introduced by its creators as a form of *virtual reality* allowing a clear view of the real view [Milgram *et al.* 1994]. On the contrary to Virtual Reality, AR uses real world as a background and adds virtual objects on that view. Therefore, superimposition is one of the important key points. However, if these virtual objects are the duplicates of the real ones, then the digital augmentation will get closer to Virtual Reality on the Reality-Virtuality Continuum, as we mention in Chapter 1. We argue that in order to create an AR experience, the scene does not necessarily

have to duplicate the real view [Aydin *et al.* 2013b]. However, 3D can also help as an interaction layer and to manage the information overload.

We illustrate our approach in Figure 3.1. On the left of this figure, the user is observing her surroundings with her smartphone. The application connects to a server to demand the 3D representations of the surrounding objects (light-weight 3D models) using the location information of the user. The server retrieves these objects from a triple store where 3D objects are interlinked with LOD resources. These objects are then displayed in augmented reality view on the smartphone. Thanks to these triples, the application can follow the links to provide information to the user. Finally, the user interacts with the surrounding objects through these 3D view to access to further information on the LOD cloud.

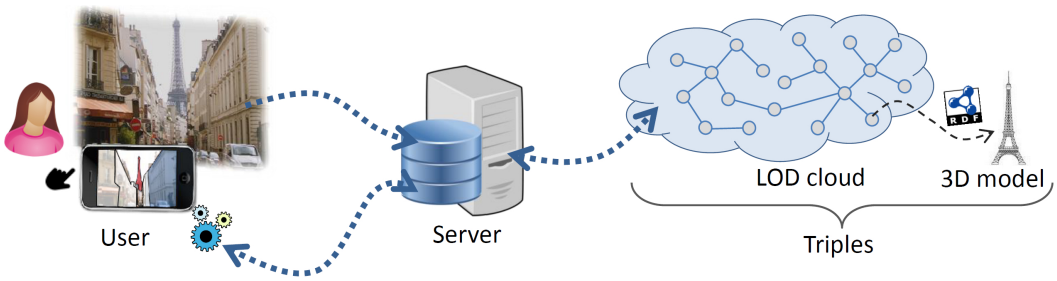



Figure 3.1 – An illustration of our approach.

In order to help the user to recognize her surroundings, these 3D models can be used interactively, especially in urban zone, in identifying the real-world objects around her. This interaction is performed by superimposing these 3D geometries on the real-world view which is captured through the camera of the mobile device. These 3D objects are geo-referenced, have a temporal validity and can be interacted by the user to access to further information. Since they have location and temporal information, they can be annotated (interlinked) with available LOD resources, which will be queried to access these additional data related to the objects. Further on, these objects can also be interlinked with multimedia documents, such as static images, texts, temporal 3D models with architectural details, *etc.*, or to tell a site-specific story upon the interaction of the user [Aydin *et al.* 2013a]. Our approach is an informative AR approach where the digital AR layer is interactive and upon the interaction of the user, the relationships between the models and information resources are analyzed in order to allow her to further explore the surroundings.

The advantage of AR over virtual reality is that it enables users to interact with their surroundings in real-time and virtual objects take place of their corresponding

real counterparts in the scene as a layer of interaction. AR provides content to the user related to her environment without replacing the real world entirely with digital world. However, there are some aspects that should be considered for developing AR applications on mobile devices, such as the amount of information displayed on the screen, level-of-detail or transparency of 3D objects, *etc.* When a user walks in an environment, the quality of information gains importance, because displaying any *unrelated* information or overloading her by displaying *all* the information on the screen may be distracting. Also, the mobile user might have time constraints or limited attention in outdoor environments. With these remarks in mind, we point out the challenges for our proposition that aims at developing an architecture for AR applications for the discovery of surroundings:

1. Determining the needs of a mobile augmented reality application user for surroundings discover in urban environments.
2. Considering the 3D object modelling for the purpose of mobile augmented reality (levels-of-detail, transparency, part-whole hierarchy, *etc.*) in order to increase their re-usability in various scenarios.
3. Designing a generic data model that will be used to interlink 3D objects with the information sources on the LOD cloud to access structured data sources.
4. Extending this data model with necessary relations, for instance 3D data and their *temporal validity* or *thematic information.*
5. Designing the methods for publishing (interlinking) these 3D objects on the Semantic Web using this data model and enabling developers to easily proceed this interlinking using a Web application.
6. Studying the exploitation of this data model and spatio-temporal, thematic information related to 3D objects for the design of mobile augmented reality application dedicated to the discovery of surroundings.
7. Developing an architecture with a use case scenario which will be based on the exploitation of this data model.

## 3.2 Overview

Our primary concern is to provide the user with information during her interaction with the real world through Augmented Reality. In order to achieve this, an additional interactive content is provided to the user which will enrich her experience with her surroundings. We bring a conceptual approach to location-based

information access for mobile Augmented Reality by using readily-available structured data resources on the Web. We propose an architecture for mobile AR applications and a data model to bind information (several 3D models, images, text, URLs,*etc.*) and make them accessible on the LOD cloud. As mentioned previously (see Section 2.2.4), the LOD cloud has a big potential to be the future's unifying technology. A substantial amount of structured data exists on the LOD cloud, waiting to be discovered and exploited. In our proposition, we use the Linked Open Data mechanisms to combine LOD and 3D data for mobile augmented reality.

In a typical AR application, 2D or 3D digital content (*e.g.* 3D models, text, photos, *etc.*) are uploaded to a repository (*i.e.* server, database...) which is accessed by mobile devices in order to download this content. In general, this 3D content, or Points of Interests (PoIs), are uploaded with geo-location tags, and using the inertial sensors of the mobile device, they are superimposed on the captured scene with respect to the physical location and orientation of the device (*i.e.* the direction in which the user holds her mobile device). In order to retrieve information related to a specific real-world object, these PoIs are annotated and saved in relational databases. However, big repositories might be necessary to store different media documents. Making these systems scale to large data volumes and numbers of users might be challenging.

On the other hand, the data defined by URIs are easily accessible since they have unique identifiers over the Web. In this case, publishing data on the Web and interlinking it with structured data sources bring many advantages as it defines relations and allows the convenient aggregation of data by following these links [Auer & Lehmann 2010]. Knowledge can be discovered from the interlinked data over the Web with adequate inference mechanisms. Information can be combined from several knowledge bases using the links among each other and various types of search and information provision can be made available by querying semantic properties of these structured datasets.

With these perspectives in mind, the key point of our approach consists in publishing our 3D AR objects on the LOD cloud using our data model (see Figure 3.2). This way, the AR objects take place as an exploitable dataset on the LOD cloud and this allows inferring existing data and generating knowledge from LOD that can feed an AR application as information sources. Once the 3D AR objects are interlinked with the LOD resources, the information related to that landmark can be collected following the associations (links) attached to this object. Knowing the

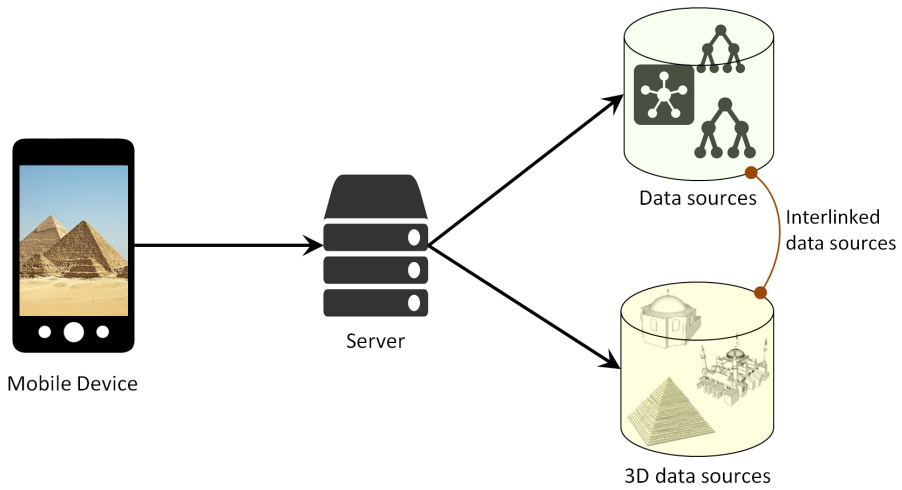



Figure 3.2 – Overview of the general approach.

RDF grammar of the data helps to identify its linking structure and, thus, to explore its relationships with other information. We can use these triples to build a graphical representation of these relationships and look up data from multiple distributed sources across many systems, being compatible with web standards, open and extensible. The combination of RDF, OWL and SKOS allows us to easily link our data with other datasets. Moreover, describing new objects and classes using semantic web grammar increases the added value of our data.

The data model that we propose is used in order to bind 3D models with the information from LOD, and also, it aims at increasing the usability of 3D objects which can be interlinked with thematic and temporal aspects. The final goal of proposing this data model is to be able to automatically find the paths between a 3D model and the information related to it available on the semantic web graph. The key elements of this data model are the 3D models that are designed to be used by various AR applications (see Figure 3.3). As it can be seen in this figure, each element in our approach has a relation with the 3D models. For instance,

- The data model that we propose has 3D (spatial), temporal and thematic elements (this is why we call it *3D2T*). It is used to bind 3D objects with their corresponding DBpedia page, as well as the thematic and temporal elements related to them. We describe these elements in Chapter 4 (see Section 4.2 for more details).
- Thematic elements correspond to the type (*i.e.* role, function) of real-world ob-

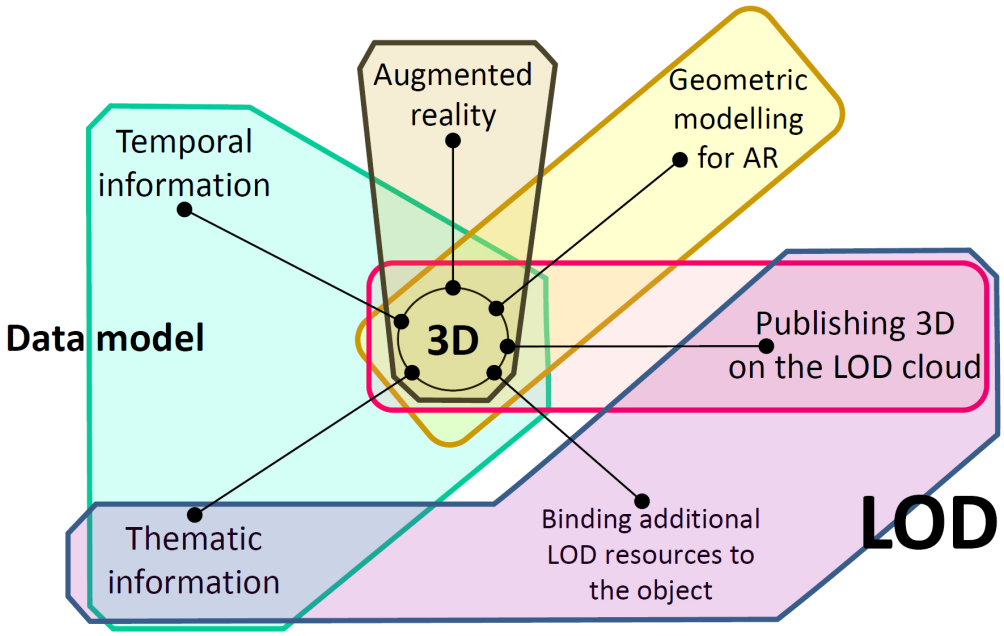



Figure 3.3 – The key elements in our approach.

jects (museums, chapels, schools, apartment blocks, *etc.*)[1]. This information is linked to the corresponding LOD classes in order to retrieve similar buildings or monuments available on the cloud, according to the selection of the user. It also allows designing AR applications that offer various interactions for a specific type of buildings (*e.g.* history of religious buildings, architectural style of historic buildings) (more details can be found in Section 4.2.1).

– Temporal information is used in order to be able to show users the 3D evolution of the buildings through time in an augmented reality view. Therefore, it designates the validity of the 3D model (*i.e.* the temporal interval during which the 3D geometry is valid) (temporal elements are described in Section 4.2.3).
– Further information related to an architectural structure (a PoI) may be available on the LOD cloud. It can be retrieved to tell a story related to a place during the AR experience of a user. We propose to interlink these data with the 3D model (or its parts) with which user can interact during her visit (detailed in Section 4.3). Following these links, enriched PoIs allow application developers to feed their AR application by retrieving additional data related to the object. This additional information may exist on the LOD cloud or

1. We mainly focus on urban structures in our approach, but this can be extended with scenic elements (mountains, rivers, *etc.* in a future work.

stored somewhere else on the Web. Such information may also be specific to a general theme or a topic of interest (culture, history, architecture, *etc.*) which allows the creation of domain specific applications. With this element in our design, we aim at creating an experience between the physical real world and its related digital content through the interaction with 3D AR objects.

- 3D models are published using the data model which makes them easily accessible on the Web and reusable by other applications. We describe the process of publishing 3D objects on the Web in Section 4.4.
- The geometric modelling of 3D models should be thought thoroughly in order to deliver a user-friendly AR experience to the user. If the models are the replicas of the real-world counterparts (with textures and details), then it might not be very interesting for the user to use it in the AR mode. Besides, they might overload the user with too much information as well and hamper her concentration upon her surroundings. The geometries should also be modelled in a part-whole hierarchy in which smaller models are bound to each other (using the data model) to create the entire geometry. The whole geometry represents the real-world object and the smaller parts are modelled so that a geometric change in the object can be represented easily. Thereby, additional information can be attached not only to the entire geometry, but also to object parts. Details about the geometric modelling of our 3D models can be found in Section 4.2.2.
- All these elements of our approach are designed with Augmented Reality in mind. The 3D models are used in an AR application and they are retrieved using the information linked to them using our data model. In order to accomplish the Augmented Reality, the application detects the physical location and direction of the mobile device in order to display the 3D models superimposed on the real view. It renders the 3D objects which has interesting information in various colors (*i.e.* according to the selections and choices of the user). We describe how we use LOD for AR purposes in Chapter 5.

We have based our approach upon an architecture which comprises several components, such as 3D models and their repository (a triple store which keeps the triples that interlink 3D models with LOD cloud resources), principles of interaction with the user, elements of a real-time detection of mobile sensor data, *etc.* and the communication among these components.

The major advantage of using LOD resources in the design of our architecture is

that they increase the possibility of using 3D objects in many different application domains (touristic, cultural heritage, history-related, architecture-related applications or AR games, *etc.*). Moreover, interlinking these objects with the LOD data sources offers an extensible application design for which numerous scenarios can be imagined (applications about castles, descriptions of the buildings, structural evolutions, *etc.*).

In the last chapter (Chapter 6), we introduce the general architecture of our approach and define its components based on the key elements we mention in this chapter.

## 3.3 Conclusion

Our research focuses on proposing an architecture for mobile augmented reality application development in order to facilitate information search and discovery by bringing a new approach for location-based mobile applications using 3D models and the LOD cloud. The architecture can be also used by various domain-specific mobile AR applications (cultural heritage, tourism, architecture, *etc.*) thanks to the extensible data model that we use in order to publish the 3D models on the LOD cloud as a dataset. By *extensible*, we mean that additional information can be interlinked with 3D models and new augmented reality experiences can be generated by exploring the links over the LOD cloud. Giving 3D models unique identifiers on the Web enables interlinking them further with other datasets and letting users access various contents according to their interests by following the links between data.

Open data is already available on the Web as a structured knowledge base with semantic properties. Our architecture uses these resources in order to be able to manipulate and selectively provide users with data through their experience with an interactive mobile augmented reality application. Using the data model, we provide the basic information about real-world objects (temporal and thematic and 3D geometry-related information). Moreover, using the unique identifiers of these 3D geometries (or the identifiers of the geometry parts), further information from vast repositories of open data can be interlinked with them and explored by the users through an AR interface. This way, for instance, rich cultural or historical background of these objects that are less known and not easily accessible will be proposed to the user as a digital media upon her interaction with these 3D objects. Knowing the location of these objects also helps to create a logical flow of

information transfer, where the user receives data that corresponds to a site-specific story.

In the following chapters, we first introduce the data model that we propose to publish 3D models on the LOD cloud attached with some basic information (see Chapter 4). We then present an overview of the design of a mobile AR platform based on our approach (see Chapter 5). We introduce a prototype that performs the communications between different application modules (such as querying 3D information, superimposing these models on the captured scene, listening the interaction of the user on the device's screen and retrieving the information that she demands, *etc.*). We detail our architecture (ARCAMA-3D) with a use case scenario related to a landmark, the Royal Monastery of Brou, in France, (see Chapter 6); and finally, we conclude (see Conclusion).

Chapter 4

# 3D2T: A Data Model Mixing 3D, Thematic and Temporal Information

**Sommaire**


4.1 Introduction . . . . . **105**
4.2 3D, Thematic and Temporal Data Modeling . . . . . **108**
4.2.1 Thematic Layer . . . . . 109
Mediator Ontology . . . . . 112
4.2.2 3D Information Layer . . . . . 122
4.2.3 Temporal Layer . . . . . 126
4.3 3D Data Model . . . . . **128**
Arcama-owl Ontology . . . . . 129
4.4 Publishing 3D Models on the LOD . . . . . **134**
4.5 Conclusion . . . . . **138**


## 4.1 Introduction

This chapter introduces a conceptual data model designed for conceptualizing 3D models that can be exploited by surroundings discovery applications using augmented reality. This model is linked with the structured information on the Web (Linked Open Data) in order to facilitate the information retrieval using resources that describe an object. With this interlinking approach, we enrich the 3D geometries (and their parts) using these available data repositories on the Web and allow developers to create applications for various domains with different augmented reality application scenarios (*i.e.* architecture-related, history-focused, theme-based, 3D evolution-based *etc.*).

3-dimensional modeling has becoming remarkably popular in several different domains such as urban planning, architecture, geographical information systems, digital preservation, *etc.* Even though there are various modeling software programs

that facilitate their use in these fields, there is no such commonly accepted remote database that connects these 3D data with the information on the Web. Hence, these 3D data may lack thematic and temporal conceptualization of the readily available structured data on the Web. For instance, the temporal interval during which these models are valid (on the sense that they represent 3D view of the real world) or the roles (functions) of these objects during that interval might not be easily accessible. Therefore, a conceptual model that interlinks the 3D models with these data should facilitate the information access on the Web and the integration of these data in various Web-based applications.

Setting up a 3D data repository is one of the first stages of our project. The main purpose is to, somehow, bind these 3D data with the structured data on the Web, so that they can further be related with the conceptual information that will bring additional semantics to the 3D objects, such as the geographical coordinates of the building, the level-of-detail of the 3D geometry, the architectural style of the structure, the temporal interval referring to a past period of time in the life of existence of the building, *etc.* We believe that these are the interesting aspects from an aug-

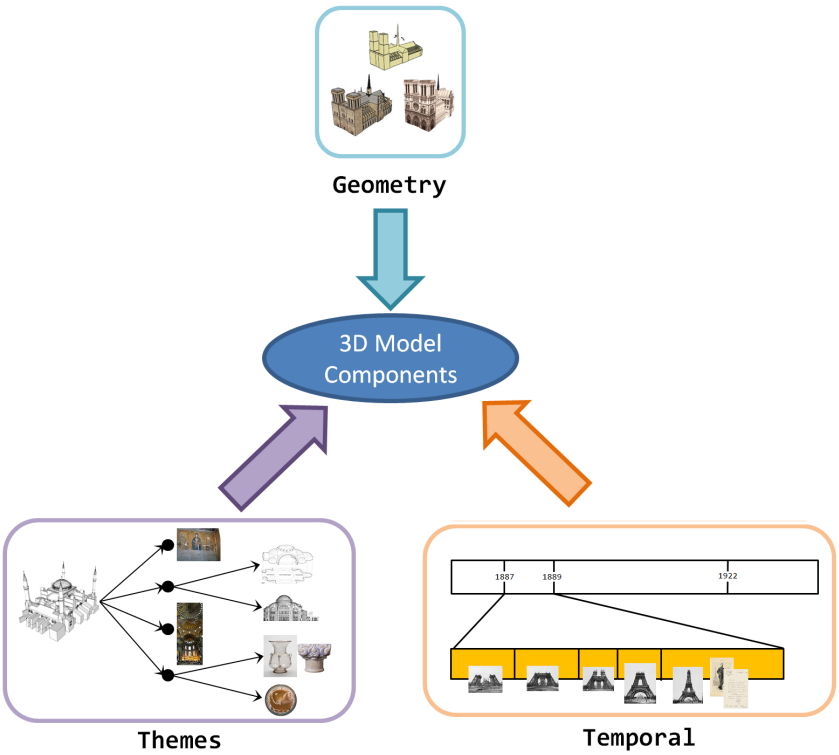



Figure 4.1 – 3D data model is conceptualized with geometry, thematic and temporal components.

mented reality point of view. These data can be exploited by various domain specific AR applications. Thus, we conclude that we need an optimal conceptual data model for the 3D objects to support the integration of these geometries in augmented reality applications. Some of the key elements in our augmented reality approach are spatio-temporal-thematic information access through 3D objects, light-weight geometries for visualization, making these geometries interactive and support for different level-of-details. The 3D objects represent and interaction layer that is used to indicate the existence of different information. We design a conceptual data model that allows us to interlink the smaller geometric parts of the 3D objects with the information available on the LOD (*i.e.* whole-part relation, explained later in this chapter).

We adapt a similar approach to the one described in [Parent *et al.* 1997], in which the authors propose a spatio-temporal conceptual model focusing on the orthogonality of spatial, temporal and structural instances in application modeling. In their approach, the modeling mainly has a topographic approach. For instance, spatial components may define a road, a vineyard, or a forest; structural instances correspond to a vast amount of object types (such as 'avalanche event', 'hazard zone', 'erosion zone', *etc.*). We integrate this approach in our conceptual data model with the purpose of allowing the exploitation of spatial, temporal and thematic information related to real-world objects during the AR experience of the user (See Figure 4.1). This approach allows us to combine the strengths of temporal analysis of spatial and thematic evolution of the objects and reveal the changes that have occurred during their lifetime.

During the lifetime of an object, its geometry and its attributes may change. In order to operate with these data, the model should permit their conceptualization. Meanwhile, we also want to emphasize that conceptualizing these dimensions also allows binding other information from the Web or local repositories to these components of the 3D model. Our objective is to introduce a data model which binds the 3D models with information sources on the LOD. This data model shall be used in order to interrogate data sources and provide user information during the execution of the AR application. Today, one can find 3D models of some landmarks over the Web, but no access is given to information about the lifetime of the objects (temporal information), the name of the architect, the architectural style, what it looked like before (*i.e.* how their 3D geometries evolved (spatial information), if there are photos or videos taken before, *etc.*) and if it has got different roles over

time (thematic information). In order to explain our point, let us give an example of a real-world object. The Hagia Sophia is an edifice in Istanbul, Turkey that got different roles and geometries (shapes) over time. Indeed, this landmark has served as a cathedral, mosque and museum; and in the meantime, its geometry has constantly evolved over time. Therefore, one can represent this edifice with several geometries with different roles. These geometries, then, can be used in mobile augmented reality in order to tell a story about the building and its evolution. In the following section, we introduce our data model that allows us to represent such real-world entities over the Web by interlinking their geometries with LOD cloud data sources and provides access to several AR applications which interrogate this linking to retrieve information related to the surroundings of the user.

## 4.2 3D, Thematic and Temporal Data Modeling

The main issue with the existing data models is that they are mostly project-based and do not support the 3D geometry for our purposes - *i.e.* their integration into Augmented Reality. Therefore, we define a generic ontology that describes general concepts which are independent from a specific domain. This ontology is also extensible with knowledge about any new specific domain and can serve as an application ontology (as defined in Section 2.2.2).

We would like to make the 3D models accessible on the Web and re-usable for various domain specific applications. Binding these models with the structured data on the Web increases the variety of application scenarios for web and mobile platforms where these 3D models will be used. Therefore, we introduce a *data model* for 3D geometries of real-world entities which we enrich with necessary information and interlink them to the LOD sources. The main interest of this approach (*i.e.* publishing 3D models on the LOD cloud) is to increase the re-usability of these models in various scenarios and use this access point to these geometries and the information bound to them in augmented reality applications.

In our data model, which contains spatial, temporal and thematic concepts, each attribute is defined using either pre-existing semantic web vocabularies. The relations between attributes of this data model and LOD cloud resources are created by binding these 3D geometries with the concepts available on the LOD cloud (*i.e.* publishing data according to the Linked Data principles, as described in Section 2.2.4).

### 4.2.1 Thematic Layer

The Thematic layer encompasses concepts that describe the semantics related to the themes (*i.e.* roles/functions) of an entity. We use DBpedia [1] dataset as an entry point to the LOD cloud. DBpedia categories and classes have properties to describe the role/function of a real-world entity. However, since DBpedia is a crowd-sourced community effort, the dataset is neither complete, nor homogeneous.

DBpedia ontology currently holds 685 classes [2], but it lacks coverage in terms of the encyclopaedic knowledge of Wikipedia, which means that the information available on Wikipedia is not entirely modeled in DBpedia ontology, therefore is not represented as a structured data on the LOD cloud. DBpedia ontology is a representation of the Wikipedia infoboxes, but a piece of information found in the text of the Wikipedia page may not be found in its infobox. As for the Wikipedia category system, the hierarchy can sometimes be very complex [Pattuelli & Rubinow 2013]. Not all the entities are interlinked with corresponding categories nor with the related class information. As a consequence, when, for instance, *all the monuments in the world* are intended to be retrieved using the class information dbpedia-owl:Monument via the SPARQL endpoint of DBpedia, the following query can be used:

```
SELECT ?name WHERE{
?entity rdfs:label ?name;
    rdf:type dbpedia-owl:Monument .
FILTER(langMatches(lang(?name),"en")) .
}
```

However, looking at the query result, we would expect that there are more than 355 monuments in the world [3]. On the other hand, if we query the monasteries using the class information (dbpedia-owl:Monastery), we observe that this class has not been instantiated, therefore the query does not return any results.

In another example, we observe that the *Royal Monastery of Brou*, a religious building located in Bourg-en-Bresse, France, is classified as a *monument* with class information dbpedia-owl:Monument in the English DBpedia [4]; whereas it is classified as *religious building* and *architectural structure* in the French DBpedia [5]. Moreover,

1. http://dbpedia.org/
2. http://wiki.dbpedia.org/Ontology (last access in March 2015)
3. the query is run on 26 November 2014
4. dbpedia.org/page/Royal_Monastery_of_Brou
5. http://fr.dbpedia.org/

if we look at the category hierarchy in order to define the roles of the entities, we observe that the categories express them with more details than the ontology classes. For instance, if we look at the Category:Monuments_and_memorials, we can see that the sub-hierarchy of this category is richer (*e.g. cemetery art*, *natural monuments*, *royal monuments*, *shrines*, *pyramids etc.*). However, as a hierarchical disorder, we also observe that Category:Monument_types or Category:Lists_of_monuments_and_memorials are also sub-categories of Category:Monuments_and_memorials, even though they represent a broader concept of other monument types.

In addition, if we look at the entities, we observe that they may have been interlinked to sub-categories expressed with the *location* and *time* annotations. For instance, some of the monuments are interlinked only with one of these following categories: Category:Monuments_and_memorials_in_France, Category:Monuments_and_memorials_in_Paris or Category:16th-century_architecture. Therefore, the role of these monuments cannot be extracted by querying the dataset simply with these general categories. Sometimes, a monument might have been interlinked with some other category which does not directly express the role of the entity, such as Category:Visitor_attractions_in_France or Category:Landmarks_in_France.

For instance, for the Royal Monastery of Brou, the DBpedia resource of this edifice has categories depicting *time* and *location*: *Gothic architecture in France*, *Christian monasteries in France*, *Churches in Ain*, *16th century architecture*, *etc.* When we read these categories, we can make many deductions, such as the Royal Monastery of Brou is a Gothic church constructed in 16th century and located in the Ain department of France. However, this category hierarchy makes it difficult to query these knowledge, as they are not well-organized. The meanings of types of structures, temporal aspects, architectural types and locations should be differentiated from each other and organized taxonomically.

As it is stated in [Pattuelli & Rubinow 2013], Wikipedia categories are organized hierarchically, but are not grounded in a strict taxonomic structure. For instance, in our example of Monastery of Brou, the edifice is assigned to several categories which should be in an inclusion relation: *Christian monasteries in France*, *Churches in Ain* and *Gothic architecture in France*. As [Suchanek *et al.* 2008] point out, the DBpedia category system merely mirrors the "thematic structure" of a Wikipedia article and the hierarchy is of little use from an ontological point of view. [Bizer *et al.* 2009] address this lack of consistency as a "rather loose relatedness between articles".

Therefore, automatic extraction of DBpedia entities with a particular role is not a trivial task. Despite this drawback, DBpedia has an important volume of semantic data which represents a very precious source of information for creating web and mobile applications. For this reason, we have to consider another approach to effectively use this vast amount of knowledge in a mobile augmented reality application.

In order to interlink 3D models with their corresponding thematic roles available on DBpedia, and to easily access this information later on, we use a mediator ontology between the thematic attribute of our entity and the thematic information on DBpedia. The thematic information sources of DBpedia correspond to the category hierarchy and the ontology classes (explained later in this section). The purpose of this ontology is to mediate between DBpedia ontology, category hierarchy and concepts of another existing ontology that we use in order to interlink our 3D models with structure roles.

With this approach, we aim at providing a larger coverage of information related to man-made (urban) structures available on DBpedia and leverage the inconsistencies caused by the collaborative organizational system of Wikipedia (as it is pointed in [Pattuelli & Rubinow 2013]). If the user is, for instance, interested in archaeological structures, the related class in the thematic ontology is selected on the client side (mobile device). The mediator ontology acts as a bridge to access all the DBpedia entities with categories and classes related to this selection made by the user. These DBpedia entities are then queried using the matching classes of thematic ontology with the corresponding DBpedia classes and categories. This matching with the mediator ontology is also used in order to provide the user with a thematic list (*e.g.* national monument, museum, government building) of objects in her surroundings and present only the information related to the selected object type.

In order to perform this matching, we need to first define the thematic roles of objects that might be worth presenting in an AR application. In our approach, we currently focus on the discovery of urban structures using Augmented Reality, but as a future work, we would like to extend it with other features such as agricultural regions (wheat belts, corn belts, *etc.*), hydrographic features (rivers, lakes, mill ponds, moats, estuaries, *etc.*), biogeographic regions (deserts, forests, *etc.*) and so on (discussed in Conclusion). We re-use an already existing ontology, FTT-v01.owl [6]

6. FTT-v01.owl ontology developed by University of Muenster, http://ifgi.uni-muenster.de/~janowicz/downloads/FTT-v01.owl

(Feature Type Thesaurus), and exploit it for our purpose. FTT-v01 models geographical concepts that cover the administrative regions, parcels of land, physiographic and hydro-geographic features. At first sight, it might be considered as a very large ontology for our purpose, since it includes classes such as *military areas*, *seafloor features* or *cotton plantations*, which one may find less interesting as far as an AR approach is concerned. However, it also provides a large coverage of classes for *type* (role) information of man-made objects (*e.g.* monument, library, space center, church, *etc.*).

### Mediator Ontology

Ontologies are in general defined for specific applications and it is not easy to find and adapt an entire ontology without making any changes for different needs. We have decided to use a subset of the FTT-v01.owl ontology that covers urban geographical concepts (*e.g. temples*, *castles*, *government buildings*, *airports*, *hotels*, *museums*, *post offices*, *hospitals*, *mosques*, *cemeteries*, *cisterns*, *etc.*), which we want to provide access as thematic knowledge (types of structures) in AR applications. Therefore, we have first reduced the size of the ontology to cover the classes related to urban objects, as we want to provide access to users through a mobile AR application. Then, we have made some modifications in the case if there was some missing type information by comparing the DBpedia ontology classes and category hierarchies (such as adding *abbey*, *basilica*, *madrassa* and *shrine* as sub-classes of *religious site* class). The reason for these modifications is that we want the ontology to cover as many object types as it can. This way, the ontology can be broadly used in different domain applications.

The mediator ontology can also serve users to limit their location-based experience to a specific thematic experience by allowing them to choose the type of objects that they want to get information about (such as *historical sites* or *abbeys and chapels*, *etc.*). These selections constitute the user's preferences and the user receives information related to only these types of objects.

The FTT-v01.owl ontology covers 1262 concepts with a depth-of-level 6. Figure 4.2 shows an excerpt of this ontology. From this ontology, we have kept some of the classes and enriched or omitted some others in order to create our mediator ontology: MediOnto.owl, which can be found in Appendix A.

MediOnto is described using RDFS and OWL ontologies, which enable richer queries and new types of inferences, whereas DBpedia category hierarchies are de-

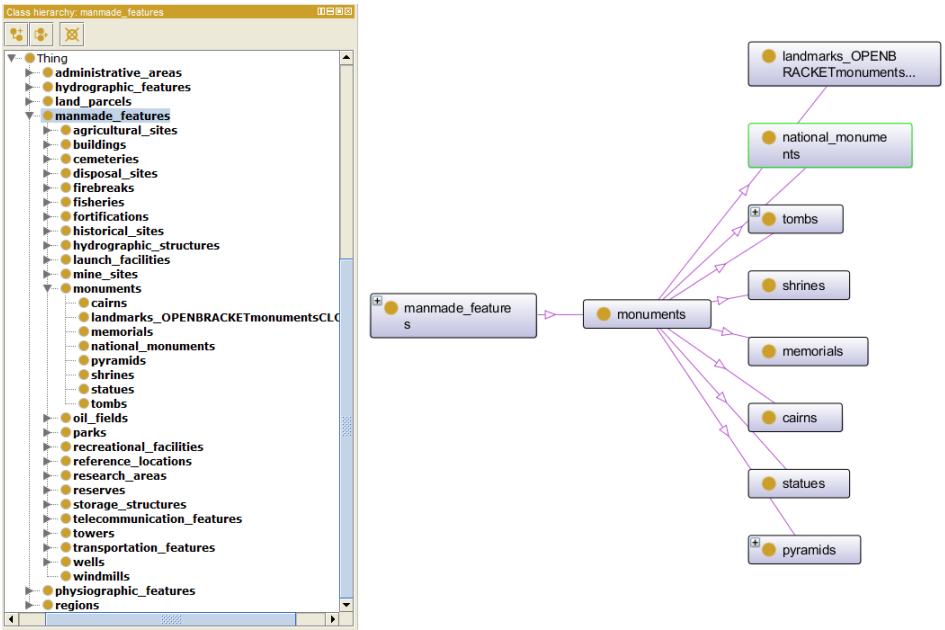



Figure 4.2 – An excerpt of the FTT-v01 ontology: *On the left*: the class hierarchy. *On the right*: the ontology expressed as a graph.

fined using SKOS (Simple Knowledge Organization System) vocabulary. SKOS is a data model described with the vocabularies of OWL and RDF. For instance, every concept is defined as a skos:Concept which is an OWL class itself (skos:Concept rdf:type owl:Class). SKOS focuses on concepts and how they are organized (*i.e.* their relationship and hierarchy).

DBpedia ontology, on the other hand, is written with OWL and RDFS vocabularies. The classes are owl:Class and further described with rdfs:label and rdfs:comment properties which are instances of rdf:Property. They provide, respectively, human readable version and description of classes in different languages. The class hierarchy is defined with another RDF property which is rdfs:subClassOf. Additional properties are introduced as *object* and *datatype properties*.

We use MediOnto as a mediator ontology, in order to create correspondences with the DBpedia classes and DBpedia category hierarchy. We apply ontology matching between MediOnto and DBpedia classes, as well as between DBpedia categories (an illustration of this can be seen in Figure 4.3 and further examples can be found later in this section).

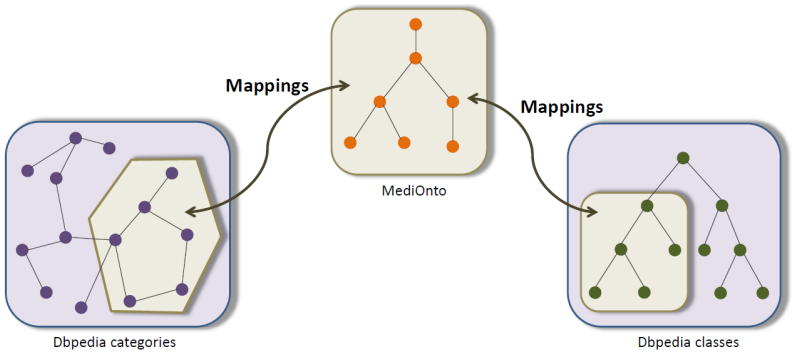



Figure 4.3 – MediOnto classes are mapped with DBpedia ontology classes and category hierarchy.

**MediOnto and DBpedia ontology** The DBpedia ontology is a shallow, cross-domain ontology which currently covers 685 classes [7]. Our goal is to provide access to a portion of this ontology, which contains mainly the classes related to man-made (urban) objects. We need to make a bridge between (*i.e.* match) the classes and categories of DBpedia and the MediOnto classes, so that when we retrieve objects that belong to a MediOnto class (*i.e.* an object type or theme), we will be covering and retrieving all the objects of that theme that are available on DBpedia. Another advantage of using a mediator ontology is that it provides a better hierarchy of classes than DBpedia, in which some objects which should belong to the same thematic concept (*i.e.* class or category) are annotated with two different concepts (this subject is further analyzed in this section).

In order to match MediOnto and DBpedia ontologies, we perform an ontology mapping which consists in basically finding the correspondences between two ontologies and storing these correspondences separately from these ontologies (details about ontology mapping can be found in [de Bruijn *et al.* 2006]). Then, these correspondences can be used to query heterogeneous knowledge bases composed of different ontologies and ontology languages (in our case DBpedia). The matching of ontologies can be done automatically or manually, yet manual mapping is more suitable for our purpose since we aim at maximizing quality for a small quantity of concepts (as suggested in [Damova *et al.* 2010]). We have compared the lexical representation of concepts in order to compute and generate mappings. In Figure 4.4, an example of this mapping is shown.

In order to express that two classes correspond to the same meaning (*i.e.* two classes denote the same concept), the class descriptions can be linked with

7. http://wiki.dbpedia.org/Ontology

owl:equivalentClass property. For instance, $<C1\ owl:equivalentClass\ C2>$ means that both class extensions contain exactly the same set of individuals. This relation does not mean that these classes are equal (*i.e.* their properties are the same), but it denotes that the meaning of two terms (*i.e.* $C1$ and $C2$) are unique.

In this mapping, a triple that binds a class of MediOnto with a class of the DBpedia ontology is described as in the following (MediOnto.owl classes are expressed with the "mediOnto" prefix):

```
mediOnto:monastery owl:equivalentClass dbpedia-owl:Monastery .
```

which denotes that any instance (individual) of DBpedia class *Monastery* will be also an individual of MediOnto class *monastery*. Therefore, if one wants to query all the instances of mediOnto:monastery (*i.e.* all the 3D objects), this relation shall be used to retrieve the entities annotated with this DBpedia class.

**MediOnto and DBpedia category hierarchy** DBpedia categories are defined using the SKOS vocabulary. However, SKOS is not an ontology. The category structure is a concept hierarchy (a schema) and it does not provide explicit semantics. The categories are instances of skos:Concept. skos:Concept is defined as an OWL

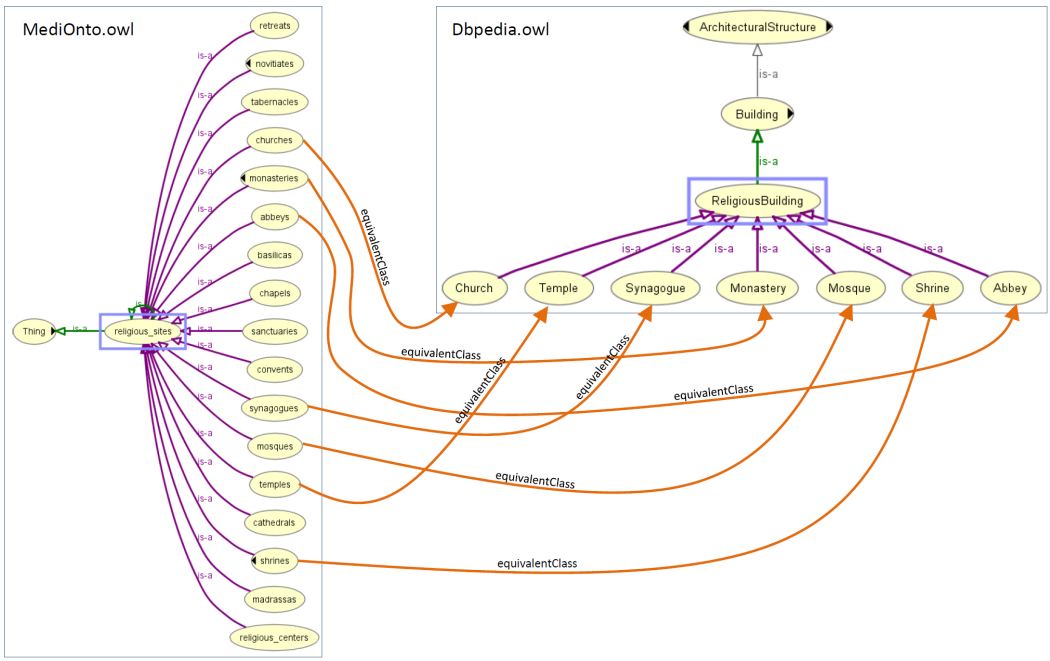



Figure 4.4 – A mapping example between MediOnto and DBpedia class ontologies (subclasses of dbpedia-owl:ArchitecturalStructure).

class. Then, DBpedia categories becomes individuals in OWL terms. On the other hand, it is important to be able to query DBpedia entities by bridging this category hierarchy with our mediator ontology in order to create correspondences between the two. This allows us to access the DBpedia entities that are not covered with classes, but with categories. The solution to this problem lies in the various ways of using the OWL language. OWL can be used without restrictions (OWL Full) or with some constraints (OWL-DL or OWL Lite) which has desirable computational properties for reasoning systems (more details can be found on the page of W3C [Bechhofer *et al.* 2004])

This situation can be solved by treating instances of SKOS concepts explicitly as classes, thus by bridging an OWL class with an instance by using owl:equivalentClass. For instance,

```
mediOnto:church owl:equivalentClass category:Church_buildings
```

where category is the prefix for DBpedia category hierarchy, *i.e.* http://dbpedia.org/page/Category:

However, this statement above is valid for OWL Full which does not require the sets of classes and individuals to be disjoint. On the other hand, OWL-DL requires the disjointness of classes, properties, individuals and data values, and retain desirable computational properties for reasoning systems. Therefore, in order to make the matching between an OWL class and a SKOS concept, we adapt the approach described in the SKOS Primer published by W3C working group [Isaac & Summers 2009]. We use dedicated *OWL annotation properties* for this purpose and define "mediOnto:correspondingConcept" to bridge them.

In this mapping, a triple that binds a class of MediOnto to a SKOS concept is described as following (with the existing DBpedia triple Category:Monastery rdf:type skos:Concept):

```
mediOnto:monastery rdf:type owl:Class .
mediOnto:correspondingConcept rdf:type owl:AnnotationProperty .
mediOnto:monastery mediOnto:correspondingConcept Category:Monastery .
```

We also prefer to keep the hierarchical relations (*i.e.* broader, narrower) that exist between some DBpedia categories in order to be able to suggest users entities *related* to their selection by using these links in the query. For example, if the user is interacting with an object of type *fortress*, the application should be able to deduce

that she might be further interested in other objects of type *castle*. Therefore, in order to keep this hierarchy, we have introduced two annotation properties that define the broader (more general concept) and narrower (more specific concept) relations between MediOnto OWL classes and SKOS concepts: "mediOnto:broaderConcept" and "mediOnto:narrowerConcept".

An example of similar concepts of the DBpedia category hierarchy and MediOnto classes can be seen in Figure 4.5. The concepts are related to a *religious object* with object type *church*. If we intend to match these concepts, we observe that it is not trivial to perform such a mapping because of the complex hierarchy structure of DBpedia categories. For instance, in DBpedia SKOS hierarchy, there is no indication that *Church* is a narrower concept of *Religious building*. Additionally, we observe that some of the narrower concepts of *Church* category do not refer to a narrower meaning in the category hierarchy, but they have similar meanings. For example, the DBpedia category *Church buildings* has two narrower categories, namely *Church buildings by city* and *Church buildings by century*, but some of the individuals belonging to the latter one are not instantiated using the broader category *Church buildings* (see Figure 4.5). However, an individual belonging to the category *Church buildings by city* or *Church buildings by century* should also be accessible using the broader category *Church buildings*. On the other hand, some other narrower categories, such as *Cathedrals*, *Wooden churches* or *Christian monasteries* refer to a specific type of *Church building*; therefore they represent narrower concepts in the category hierarchy.

Our mediator ontology helps us to access the individuals of DBpedia covered

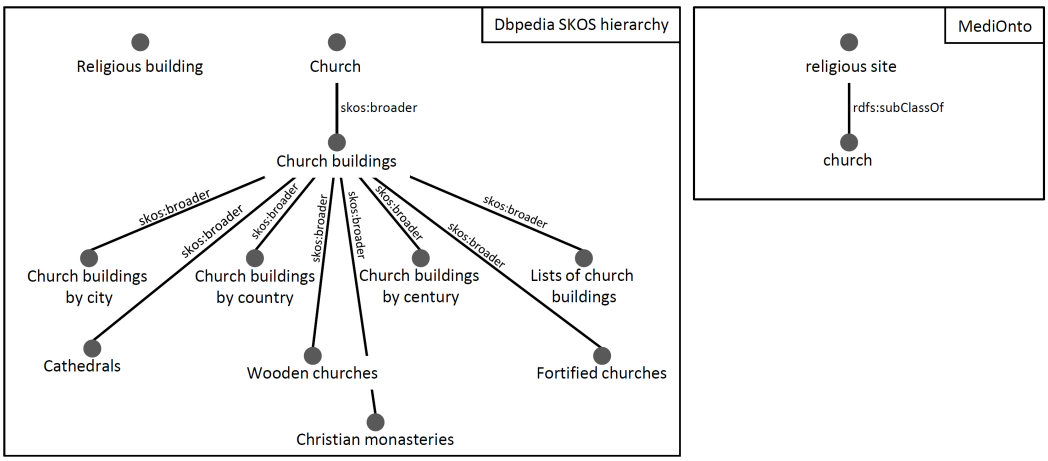



Figure 4.5 – An example of similar concepts in DBpedia and MediOnto.

not only by classes, but also by categories. However, categories by themselves do not provide a well-written hierarchical knowledge. Then, in order to perform a matching between MediOnto and DBpedia categories in an automatic manner, we use syntactic-based methods to measure the *similarity* between concepts as described in [Gamallo *et al.* 2001]. For instance, let us express a DBpedia category with '$c$' (such as *castles*), for which we observe that the sub-categories of '$c$' are expressed with the following syntaxes: *'c'+_by_country*, *'c'+_by_century*, *_by_century+'c'+_by_country*, *'c'+_by_city*, *'c'+_by_heritage_register*, *Types_of_+'c'*, *Lists_of_+'c'*, *etc.* The entities (*i.e.* individuals) that are annotated with these sub-categories have the same thematic concept '$c$' (*i.e.* castles), but two entities of this concept may be annotated on DBpedia with two different categories (for instance, *'c'+_by_country* and *'c'+_by_century*). Therefore, when one queries the dataset with the category '$c$', these two entities that are annotated only with the sub-categories may not be in the result. On the other hand, these instances both have the same thematic concept. However, with the DBpedia category hierarchy, they appear to be a sub-category of '$c$'. Using MediOnto, we aim to bring these two individuals to the same level and match them with the corresponding MediOnto class using mediOnto:correspondingConcept predicate (see Figure 4.6). With this approach, when we query the thematic concept '$c$', the individuals annotated with *'c'+_by_country* and *'c'+_by_century* will also be retrieved.

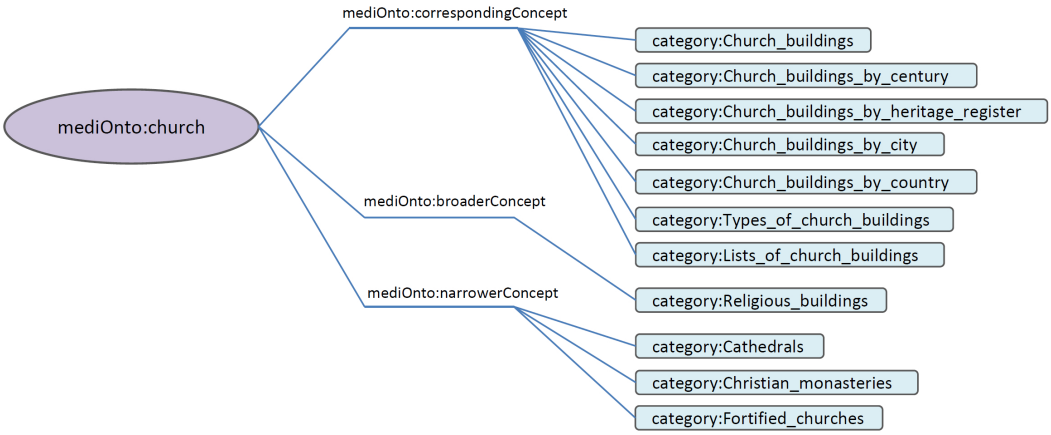



Figure 4.6 – A mapping between MediOnto classes and DBpedia SKOS concepts using mediOnto:correspondingConcept, mediOnto:broaderConcept and mediOnto:narrowerConcept annotation properties.

We use the *Jaccard measure* (see [Niwattanakul *et al.* 2013] for more details) in order to calculate the syntactic similarity between two concepts. As a first step,

we match the classes of MediOnto manually with some broader DBpedia categories (such as mediOnto:museum with category:Museums). Then, we use this as a pre-set in order to analyze the sub-categories of these DBpedia categories (*i.e.* category:Museums) in order to find the syntactic matches. The Jaccard similarity measure (or the *Jaccard index*) computes the word similarity. We introduce $C_1$ and $C_2$ as two sets of concepts (classes, categories, *etc.*). In order to find the corresponding concepts between these two sets, we calculate the Jaccard index using the following equation:

$\text{Jaccard}(c_{1i}, c_{2j}) = \frac{|c_{1i} \cap c_{2j}|}{|c_{1i} \cup c_{2j}|}$ where

- $c_{1i} \cap c_{2j}$ : the intersection of two set of words, in which the common words are included only once. It contains all the words that are in both the first and the second set.
- $c_{1i} \cup c_{2j}$ : the union of two set of words. It contains all the words in the first chain, as well as the second one, except the words that are in both sets.

If there is a perfect correlation between the two concepts, $c_{1i}$ and $c_{2j}$, the equation returns the value '1'; if there is no co-occurrence, the measure will be 0.

Let us now call $c_{1i}$ a concept (class) from MediOnto ontology and $S = \{s_1, s_2, s_3, ...\}$ a pre-determined collection of words, which is the set created by concatenating some pre-determined strings (such as *by_city*, *by_century*, *Lists_of*, *etc.*) with the class $c_{1i}$. The different combinations of $c_{1i}$ with the words from the collection $S$ give us the set of concepts $C_{1i}{}'$ that we intend to compare with DBpedia categories using the Jaccard index to compute the syntactic similarity.

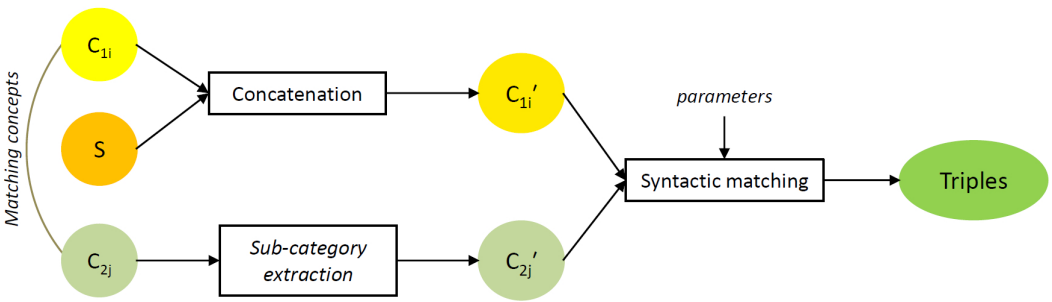



Figure 4.7 – A representation of mapping MediOnto classes to DBpedia categories.

A representation can be seen in Figure 4.7, where $c_{1i}$ represents a class of MediOnto that can be matched to $c_{2j}$, a DBpedia category. $S$ represents the set of pre-determined collection of words. The concatenation of $c_{1i}$ with the set of words

from $S$ gives us a set of concepts ${C_{1i}}'$. These concepts are compared with the sub-categories of the DBpedia category ${C_{2j}}'$. If the Jaccard index is equal to '1' (*i.e.* if there is equality of these sets of words), then the triple is generated using the URIs of the resources binding these two concepts with the OWL property mediOnto:correspondingConcept. If the Jaccard indice is not '1', it means that the sub-category is not a corresponding concept, but a narrower one, so it is bounded with mediOnto:narrowerConcept. This automatic process is described in Algorithm 1.

In this algorithm, we use $C_1$, a set of MediOnto classes, and $S$, a set of predetermined words as inputs. The algorithm runs through all the classes of MediOnto and for each, it tries to find the corresponding DBpedia concepts by running through the sub-categories of the matching concepts. If it finds a corresponding concepts, then it interlinks these two concepts with correspondingConcept relation, otherwise with narrowerConcept relation. The complexity of the algorithm increases linear with the number of sub-categories of each corresponding concept. Hence, we can denote the running time (complexity) of the algorithm as $O(N)$ using big O notation. We can iterate this algorithm again with the corresponding concepts that the algorithm finds in order to verify if these concepts have sub-categories (depth-level 2) that match. However, we stop at depth-level 1 and make the assumption that the sub-categories of these concepts are also corresponding concepts. We achieve this with a simple algorithm that runs iteratively to generate these correspondingConcept relations.

Let us explain the matching of MediOnto with DBpedia classes and categories with an example. Given the prefixes,

```
prefix mediOnto: <http://steamer.imag.fr/ontology/mediaOnto#>
prefix dbcategory: <http://dbpedia.org/page/Category:>
```

the triple that matches these two concepts will be as in the following:

```
mediOnto:castle mediOnto:correspondingConcept  dbcategory:Castles_by_type
```

An example of this mapping can be seen in Figure 4.8. Here, the concepts related to the object type castle are shown. On the left, we see the DBpedia category hierarchy and on the right, the related DBpedia ontology classes. The arrows with red color indicate the mapping between MediOnto ontology classes and DBpedia categories. The blue arrows show the mapping with narrower classes (*i.e.* sub-categories of DBpedia, which are narrower concepts compared to corresponding concepts). In this figure, the MediOnto classes that are mapped with DBpedia categories

**Algorithm 1** Automatic mapping of MediOnto classes with DBpedia classes

**Input:** $C_1$, $S$ $\{C_1$ is a set of MediOnto classes and $S$ is a set of pre-determined words as inputs$\}$

**Output:** $\gamma \leftarrow \emptyset$

1: **for each** $c_{1i} \in C_1$
2: **repeat**
3: **for each** $c_{2j}$ where $correspondingConcept(c_{1i}, c_{2j})$
4: **repeat**
5: advance on the set $S$
6: create a set $C_{1i}{}'$ that contains the concatenations of $c_{1i}$ with $S$
7: **until** the end of $S$
8: create a set $C_{2j}{}'$ that contains all the sub-categories of $c_{2j}$
9: **repeat**
10: advance on the set of $C_{1i}{}'$
11: **if** $JACCARD(c_{1i}{}'_k, c_{2j}{}'_l = 1)$ **then**
12: add triple $correspondingConcept(c_{1i}, c_{2j}{}'_l)$ to $\gamma$
13: **else**
14: add triple $narrowerConcept(c_{1i}, c_{2j}{}'_l)$ to $\gamma$
15: **end if**
16: **until** the end of $C_{1i}{}'$
17: **until** the end of $C_1$

and classes are mediOnto:building, mediOnto:fortification and mediOnto:castle. mediOnto:building and mediOnto:castle are manually linked with DBpedia ontology classes using the owl:equivalentClass property, since they are directly corresponding (shown with the yellow arrows in the figure). However, there is no equivalent class for fortification, therefore it doesn't have an equivalent DBpedia class. The mappings with DBpedia category hierarchy are performed using syntactic matching algorithm, as explained (Algorithm 1), in order to find the corresponding and narrower concepts in the DBpedia categories.

In this interlinking method, DBpedia categories behave as a classification scheme for the mediator ontology. They take part in organizing conceptualized entities which do not have a formal semantics like DBpedia classes, therefore cannot be interpreted unless being interlinked. This serves us to organize and find entities

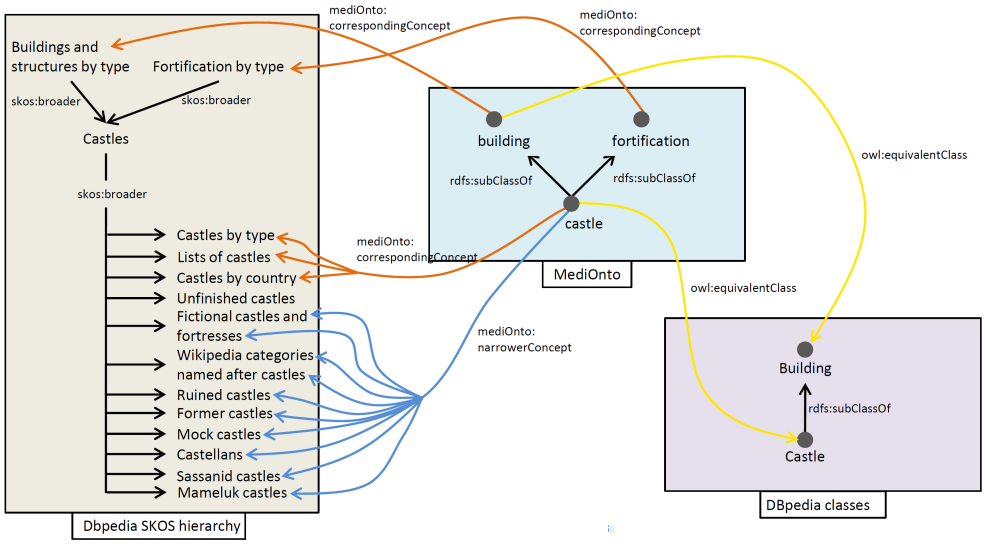



Figure 4.8 – The mapping of thematic concept *castle* to the DBpedia ontology and categories.

using DBpedia categories, as well as DBpedia classes.

In a schematic view, the interlinking of DBpedia categories and classes with MediOnto.owl enables us to organize the conceptual hierarchy of categories. DBpedia categories are well-defined, but they do not provide explicit semantic meaning. MediOnto also acts as a bridge between these categories with DBpedia classes (*i.e. DBpedia ontology*). This increases the accessibility of DBpedia entities and allows their filtering according to their thematic information (*i.e. object roles*).

In this section, we have described the MediOnto ontology and mappings between its classes and DBpedia concepts in order to provide a thematic layer in our data model. In the following, we introduce the 3D information layer which corresponds to the spatial layer of our data model.

### 4.2.2 3D Information Layer

The 3D information layer corresponds to 3D geometric models that represent real-world entities. During its existence, a real-world object might have experienced several transformations (*e.g.* construction of an entire entity, demolition of some parts, reconstruction of some parts, *etc.*). We aim at representing these changes by generating the 3D representations of these transformations.

The 3D models are generated as light-weight (wire-frame and semi-transparent) geometries for AR visualization and interaction purposes. With this approach, users can observe their surroundings instead of focusing on the virtual (3D) scene and

it appears as if they are interacting with the real-world objects due to the transparency of the geometries. In order to allow users to visualize a more detailed 3D model or parts of it and to access information related to the real world entity that they represent, we have also added a *Level-of-Detail* (LoD) component to our data model. This way, a user may interact with the parts of a 3D model and observe a more detailed geometry with, for instance, texture, zooming and interaction options. Users may also be invited to visualize the evolution of an object through time by displaying the 3D models in different LoD. Increasing LoD corresponds to a more detailed geometry. The LoD notion in our approach is based on the LoD description of CityGML, as mentioned in Section 1.2.3. To recall, LoD1 corresponds to building blocks without a roof (the AR objects in our approach are visualized in LoD1, unless the user demands a more detailed model); in LoD2 distinctive roof structures and balconies are added; and LoD3 denotes architectural models with detailed wall and roof structures.

The 3D models can also be composed of different 3D parts (individual geometric features) that constitute the entire 3D geometry. This notion is also introduced in CityGML, as 3D geometry may have attributes, relations and hierarchies that describe part-whole relations.

In order to generate 3D wire-frame models, we need to know the 2-dimensional footprint geometry of the object. Using this 2D source, 3D data can be generated through a process known as *extrusion*. There are several available Web-based and open-source tools such as SketchUp [8], Blender [9], Autodesk 123D Catch [10], *etc.* to develop 3D models. Accessing the 2D footprint content has become easier using crowdsourced geographic data made available by OpenStreetMap (OSM), as we have mentioned in Section 1.2.3. OSM is an affordable way to generate 3D data based on the 2D polygon data of the objects. It is a crowdsourced data that is made available under an open database licence. Map data is collected by volunteers using basic tools such as handheld GPS, digital camera or voice recorder and then, this data is entered to its database. It is also reviewed, edited and corrected by the OSM community, in a manner similar to Wikipedia.

To the best of our knowledge, there is no such warehouse where we can find all the downloadable 3D building geometries to be used in our approach. However, recently many tools have been introduced for generating 3D models from vast amount

8. http://www.sketchup.com/
9. http://www.blender.org/
10. http://www.123dapp.com/catch

of building polygon data. For instance, using *OSM2World* [11], it is possible to generate 3D models based on OSM and export them in different file formats. *Glosm* is a hardware-accelerated renderer based-on OpenGL [12] (Open Graphics Library). Web-based 3D rendering has also become popular recently, in which interactive 3D content is rendered on a web browser using HTML5, WebGL [13] (Web Graphics Library), JavaScript, *etc.* For example, *OSM Buildings* is an open-source library to visualize 3D buildings using HTML5. It provides extensions for both Leaflet and OpenLayers javascript libraries. *Three.js* [14] is a javascript library which allows displaying 3D graphics on a Web browser. Another similar project is *xml3d* is based on WebGL and runs on most of the browsers without any plug-ins.

Web browsers are nowadays often used for displaying 3D content (*e.g.* Google MapsGL [15] which shows 3D buildings on a Web browser), but AR recently technologies are also evolving towards web platforms. WebGL and JavaScript have already been used with tracking technologies using a marker in the scene (*e.g.* JSARToolKit [16]). 3D Points of Interests (PoIs) are displayed on the browsers for location-based AR using device compass, accelerometer and GPS.

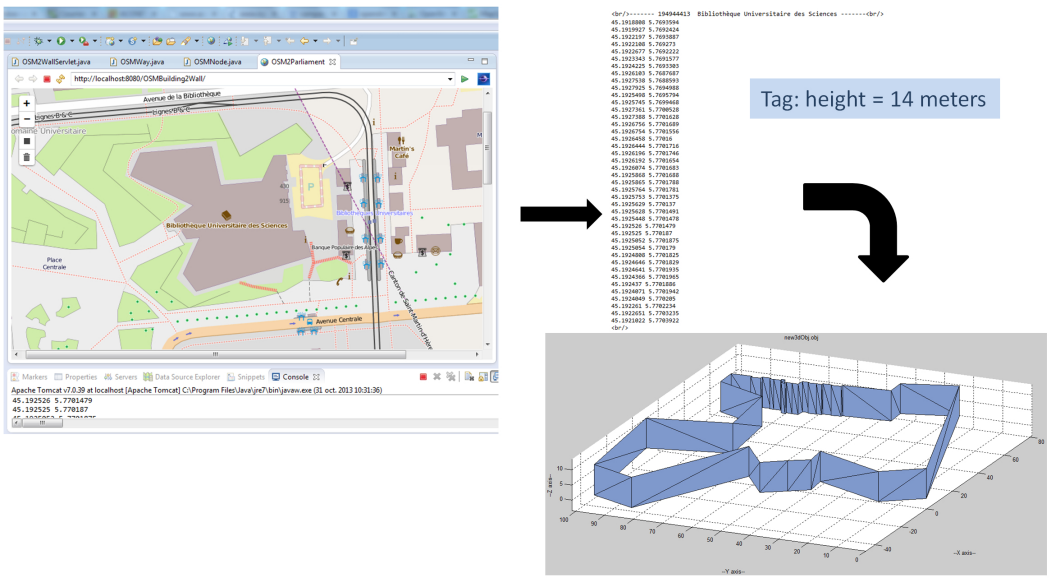



Figure 4.9 – The 3D extrusion of a library building: OpenStreetMap is used to select the region and Matlab is used to extrude it.

11. http://osm2world.org/
12. a multiplatform API for rendering 2D or 3D graphics - https://www.opengl.org/
13. http://www.khronos.org/webgl/
14. http://threejs.org/
15. http://maps.google.com/
16. http://fhtr.org/JSARToolKit/

These technologies are in development. However, in order to create a test bed of 3D models (with different levels-of-detail) representing some real world objects within an area, and to publish these geometries on the Web, we follow these five steps:

1. OpenStreetMap [17] is used to extract the polygon shaped footprint geometry, which constitutes the 0 (zero) level-of-detail in the modeling (see Figure 4.9). The vertices of this 2D geometry are real-world coordinates retrieved from the polygon information of OSM.
2. The height information of real-world objects available in OpenStreetMap through the building:height key is used to set the height of the 3D models. If this information is not available, a default height of 10 meters is chosen.
3. The footprint geometry and height information are sent to Matlab and the 3D object is extruded using a basic triangulation algorithm to create a triangulated *Wavefront* (a file format with .obj extension) wire-frame model.
4. In order to generate objects with higher levels-of-detail (according to CityGML LoD definitions) or 3D models that belong to a time interval (*i.e.* to display the evolution of the objects through time during the AR experience of user), *Sketchup* is used (see a representation in Figure 4.10). CityGML plugins for SketchUp to import, edit and export CityGML files are available and facilitate the conversion from one format to another, such as GeoRes [18] or CityGML-Editor [19] plugins for SketchUp. This step concerns designing 3D models and is manual unless the application designer is using some pre-generated

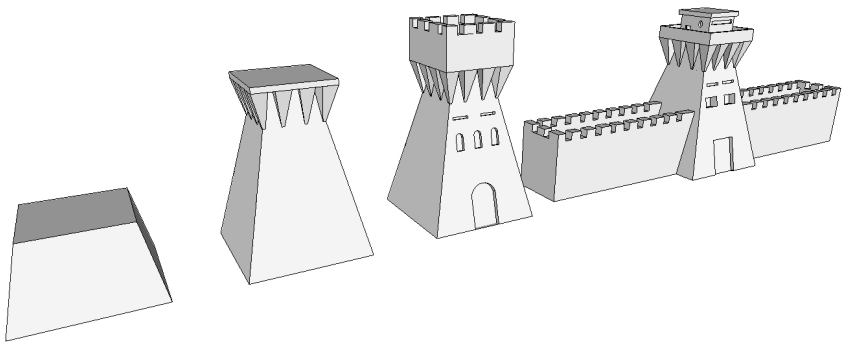


Figure 4.10 – From left to right: a 3D object evolution in time (represented in LoD2).

17. http://www.openstreetmap.org/
18. http://www.geores.de/geoResPlugins.html
19. http://www.citygml.de/index.php/Download.html

3D models with different levels-of-detail.

5. In order to generate sub-parts from a whole 3D object, the object is divided into smaller parts using SketchUp, and each part is exported individually in a separate file, so that the parts can be manipulated (changing colors, transparency, interaction methods *etc.*) by the AR application.

The different 3D models belonging to an entity represent the historical evolution of this object and the generation of these models is not automated, as the information about the evolution of the 2D footprint of the real-world object is seldom, if not never available. However, using Matlab algorithm, we can automatically extrude different 3D object files at the same time. One of the limitations of this extrusion process is that it is time consuming. However, the purpose of our proposition is to make 3D representations available on the Web. Therefore, it can be seen as a digitization process of 3D representations of real world objects, which can be used in great extend in other projects by being accessible on the LOD cloud.

### 4.2.3 Temporal Layer

Time can be considered as the 5th dimension besides the 3D geometric information and thematic data. For a complete representation of 3D objects, the data model should be able to conceptualize the evolutions of geometric and thematic information through *time*. The temporal layer, for this purpose, has a very important role. This dimension is necessary to capture the changes that have occurred at a particular moment in the past. However, it should be noted that the geometric evolutions that we mention in this part only concern stationary (*i.e.* motionless) geographical objects, so the translation of one object from one site to another is beyond the scope of our research (such as the relocation of a library from one place to another side of the city, *etc.*).

Temporal information itself have various dimensions. It can be a moment in time (one dimension), an interval (bi-temporal), or composed of several intervals (multi-temporal). However, when considering our use case for the 3-dimensional modeling (*i.e.* for mobile augmented reality applications), the instant temporal information will be very rare since the spatial evolution is not translational. On the other hand, queries might include instant-based requests, *e.g.* "*what was the state of this building at time t ?*". Time can exist in different granularities (days, years, centuries, *etc.*). This granularity can be manipulated in order to visualize both the spatial and thematic changes related to an object in an AR application. For instance, queries

such as "*how did the object evolved during the 17th century ?*" is interesting to tell stories about a historical site of interest to a mobile user.

Then, we need to be able to represent the interval-based temporal information belonging to a 3D model using temporal attributes. We use an existing ontology to describe these intervals. There are several time ontologies, such as OWL-Time [20], an ontology of temporal concepts that provides a vocabulary with topological relations of instants and intervals. It is appropriate for our modeling of temporal information related to 3D objects. Its TemporalEntity class provides two subclasses: Instant and Interval and properties to define relations between these instants and intervals: hasBeginning and hasEnd. The duration of an interval can be described in different granularity using temporal descriptions defined in *years*, *months*, *weeks*, *days*, *hours*, *minutes*, and *seconds*. We define the property hasTimePeriod to bind it with a temporal interval whose *beginning* and *end* instants are defined with OWL-Time properties. For instance, for the real world landmark the Eiffel Tower in Paris, France, its construction which describes the change (*evolution*) of the geometry can be expressed as following:

```
:construction
      a       :Interval ;
      :hasBeginning :constructionStart .
      :hasEnd :constructionEnd .

:constructionStart
      a       :Instant ;
      :inXSDDateTime
              1887-01-26 .

:constructionEnd
      a       :Instant ;
      :inXSDDateTime
              1889-03-31 .
```

In the following section, we introduce the 3D data model that we propose for representing the three layers of information (thematic, temporal and spatial) related to a 3D object in order to publish it on the Web by interlinking with LOD resources.

20. http://www.w3.org/TR/owl-time/

## 4.3 3D Data Model

We describe here a data model that contains spatio-temporal and thematic information related to a 3D model. Our data model focuses on the orthogonality of spatial (portion(s) of space the 3D model covers), temporal (time period(s) the 3D model belongs to) and structural (role/function) of that real world object instances. It supports the representation of spatial and thematic changes over an interval of time. Thus, the model allows the exploitation of this information during the AR experience of the user.

Our goal is to publish 3D models on the Web using this model in order to allow representing these objects with their thematic and spatial (*i.e.* geometrical) changes over different time intervals. In order to achieve this, we design the data model so that any changes occurring on this orthogonality (spatial, temporal and thematic) can be instantiated on the LOD cloud. If, for instance, an object with a specific role and geometry for a given certain time interval can be represented and accessed on the LOD cloud, this representation can also be linked with additional information later on. Moreover, information related to a real-world object can be examined further on these dimensions.

In order to better explain these concepts, let us give some examples about a real world entity and the information available on the Wikipedia page dedicated this entity: the *Hagia Sophia*, an edifice in Istanbul, Turkey. This edifice is suitable for illustrating our approach, since the architecture and the role (function) of this building have both significantly evolved through time. Between the years 360-532, the structure served as a cathedral with a rather simple structure; between 532-1453 the structure evolved and became immense; in 1453, it started serving as a mosque

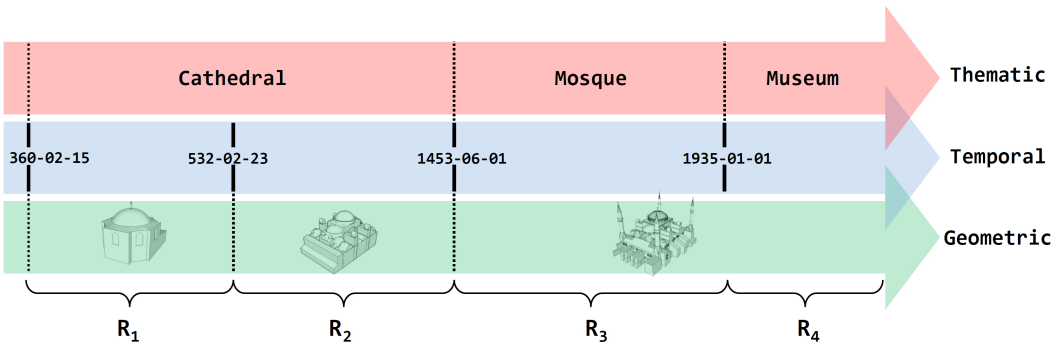



Figure 4.11 – The thematic and geometric evolution of the real world entity *Hagia Sophia*.

and four minarets were added to the structure; and finally, since 1935, it serves as a museum (this evolution is represented in Figure 4.11).

On the infobox of the Wikipedia page dedicated to this monument [21], the information related to this thematic evolution over time can be found. Since the information on an infobox is extracted as RDF triples using infobox templates and represented in DBpedia with its ontology properties (this process is described in Section 2.2.5), such information related to *Hagia Sophia* is transferred with the following attribute-value pair: dbprop:type: "* *Eastern Catholic Cathedral * Eastern Orthodox Cathedral * Roman Catholic Cathedral * Eastern Orthodox Cathedral * Imperial Mosque * Museum*". However, the temporal information related to the duration of these roles is not transferred and cannot be extracted from DBpedia. In order to retrieve only the thematic (*type*) information related to the entity, this data should be parsed and analyzed by an algorithm. This requirement seems against the purpose of linked open data, which is to facilitate access and integration of structured data. However, type descriptions of infoboxes are not easy to manipulate and they are not entirely transferred to DBpedia with proper attributes (*e.g.* time information described in the infobox of *Hagia Sophia* disappears in its DBpedia resource while extracting them as RDF triples).

The model that we propose takes into account both the temporal evolution in space (*geometry*) and thematic information (*role/function*) that describe the entity. Therefore, for a given entity, we can have multiple representations of the object, each of them corresponding to a change in the *thematic* and/or the *geometric* dimension. Figure 4.11 shows the evolution for the Hagia Sophia, and the different representations (*i.e. R1, R2, R3, R4*) that the 3D model should be able to handle for this entity. As expected, each of these four representations correspond to an *architectural* and/or a *functional* change of the entity.

### Arcama-owl Ontology

In order to construct these multiple representations, we define an OWL ontology, called arcama-owl, that describes a generic conceptual model for an entity. Such a model allows us to define a 3D model (spatial component) with temporal and thematic attributes in order to increase its usability by various AR applications. The thematic attributes of arcama-owl consists of the classes of the MediOnto ontology. Our data model is designed using a Unified Model Language (UML) diagram as it

21. http://en.wikipedia.org/wiki/Hagia_Sophia/

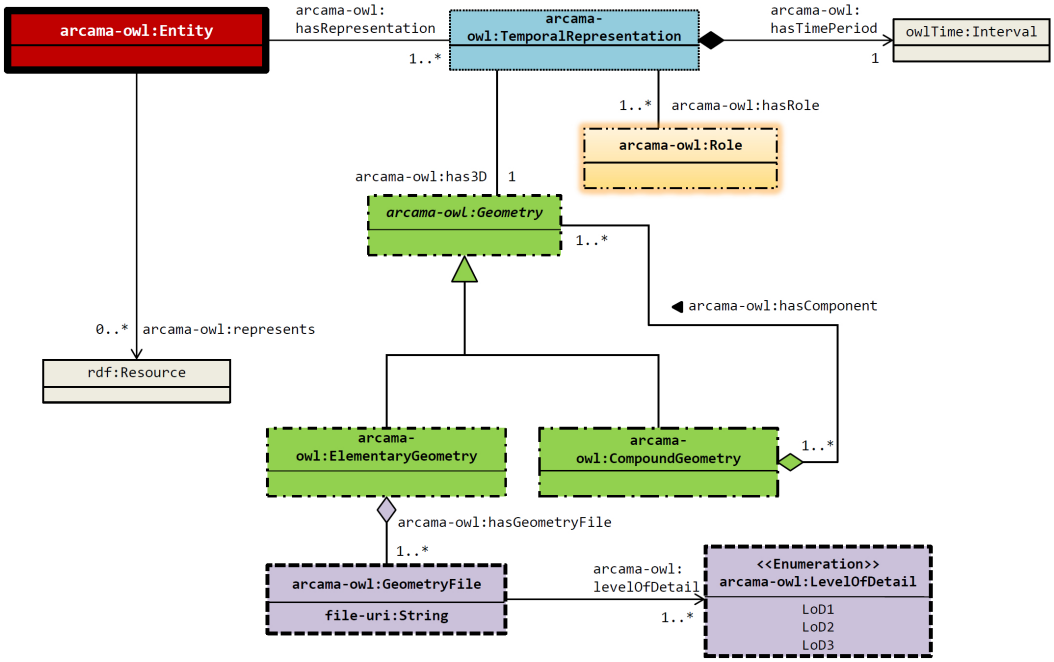



Figure 4.12 – UML class diagram of the arcama-owl ontology.

can be seen in Figure 4.12. This diagram represents the interlinking of 3D geometry of real-entities with the information that can be found on the LOD cloud. The concepts (classes) and relations between these concepts (*properties*) can be seen in the UML class diagram and are defined using arcama-owl ontology.

For our system, this class model (and its instantiations) plays an important role. It allows the interconnection between the historical 3D representations of real-world objects (even the non-existing ones) and the information related to these objects found on the LOD cloud. It also allows enriching this knowledge base with additional structured data (temporal, thematic and spatial) and increases the re-usability of these models in AR applications. We describe the main classes and relations of this model (as shown in the UML class diagram of Figure 4.12).

- The Entity class corresponds to a real world object (the edifice *Hagia Sophia*, for instance). This model allows combining several temporal representations of the same entity (see class TemporalRepresentation). The represents association allows interconnecting this model with other resources described in other datasets and publish it on the LOD cloud.
- The TemporalRepresentation class is at the heart of our model. It corresponds to a representation of the entity depicted with temporal (interval), thematic (role) and 3D geometry attributes. This representation has a temporal vali-

dity defined by OWL-Time [22] with the hasTimePeriod association. A temporal occurrence aggregates spatial and thematic attributes that describe the entity during this time interval. These attributes are described respectively by the Geometry and the Role classes. Any change in the Geometry and/or Role information requires (and results in) the creation of another TemporalRepresentation associated with a new time interval.

- The hasRole association sets the role(s) of the TemporalRepresentation of an object through the Role class which holds the properties about the function of the object (*museum*, *monument*, *hall*, *church*, *etc.*). All the roles presented here are defined by the MediOnto.owl ontology that we have introduced in the previous section. It has classes corresponding to different types of architectural structures, for example, *religious sites* (*cathedral*, *mosque*, *temple*, *etc.*) or *historical sites* (*castle*, *monument*, *pyramid*, *etc.*). To facilitate interlinking resources from other datasets defined in their own ontology, a mapping between categories and classes of DBpedia is established (as explained in Section 4.2.1).
- In order to describe the geometry associated with a temporal representation instance, we use the composite design pattern [Gamma *et al.* 1995] that combines the geometry in a tree structure that represents a part-whole hierarchy. A 3D model might be composed of smaller parts (individual geometric features). In order to represent this, the CompoundGeometry class is used to define a complex geometry composed of several different geometries that can be either complex geometries or elementary geometries. An elementary geometry represents the smallest 3D object part. An entire object can be represented with one geometry as well. Geometry is an abstract class with two subclasses ElementaryGeometry and CompoundGeometry.
- The ElementaryGeometry class describes a leaf node in a tree structure of geometries. As for a specific purpose, we need to offer different 3D models at different levels-of-detail. To this end, ElementaryGeometry class aggregates one or more GeometryFile. Each GeometryFile is described by a URI that provides access to a physical file containing a description of the 3D object in a common format (KML, CityGML, VRML, COLLADA *etc.*).
- The LevelOfDetail class describes the level-of-detail of a GeometryFile. We use the three levels-of-detail defined by the CityGML (see Section 1.2.3). LoD1 re-

22. http://www.w3.org/TR/owl-time/

presents a rectangular prism without any roof structures. LoD2 corresponds to a much detailed 3D geometry with no texture. Lod3 corresponds to architectural models with detailed roof, door and window structures, defined. These 3D objects can also have textures in LoD3 (*e.g.* a laser scanned object, textured objects with aerial photography, *etc.*).

Let us give some use case examples of our data model for AR experiences in specific situations. If a real world entity has only one geometric representation, a role (*i.e.* a class of MediOnto) associated with that geometry and a time interval association with a beginning date, but with no end (it indicates that the entity still exists), then there is only one temporal representation for that entity. It means that the object has one depiction. Therefore, during the augmented reality experience,

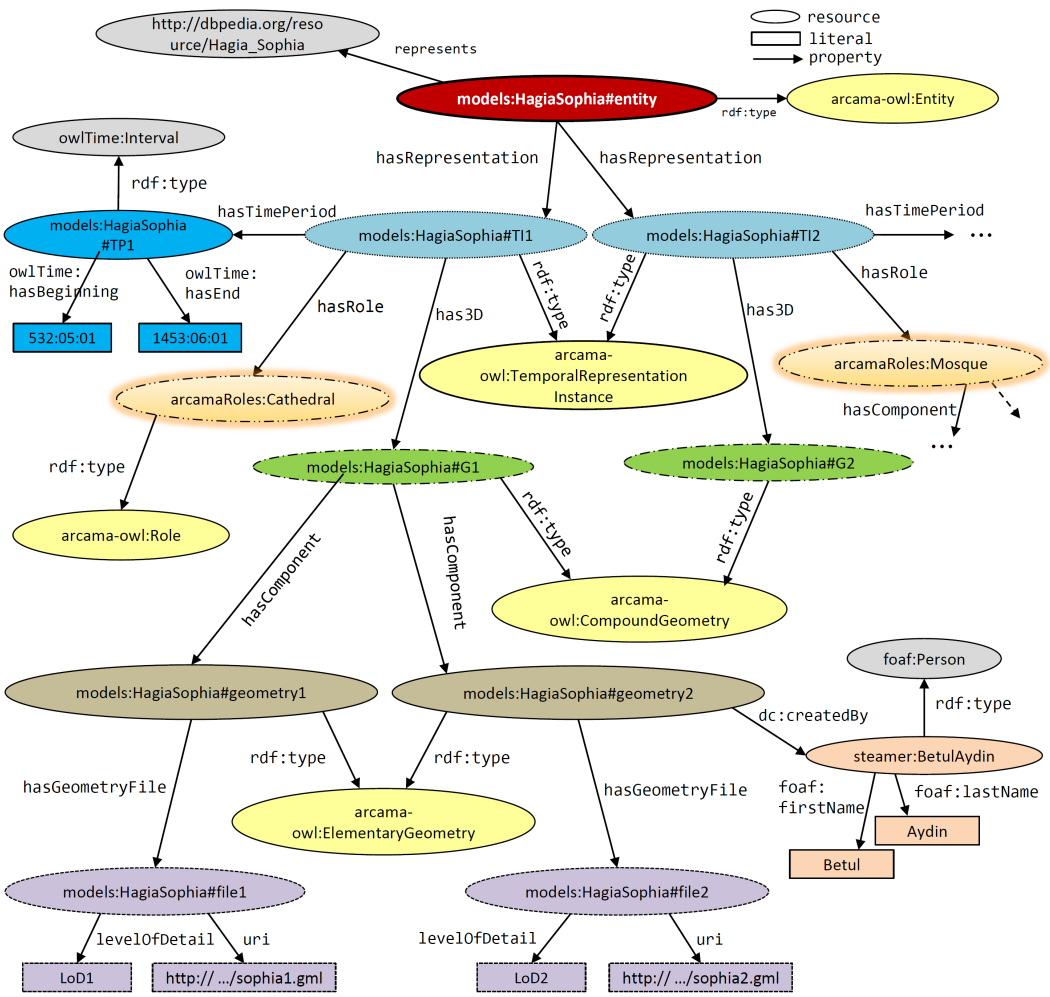



Figure 4.13 – Representations of *Hagia Sophia* according to both its thematic and geometric evolutions.

the user will observe a transparent 3D block without any further geometric interaction, since only one temporal representation exists. On the opposite, if an object has several other temporal representations, then a *temporal visit* is proposed to the user, where she can observe the same entity at different time periods. The hasRole attribute is used to classify the temporal representations depending on the thematic criteria chosen by the user. It also allows filtering temporal representations of the objects according to different roles. This provides user with an access to objects which had these roles in the past, *i.e.* not only to current real-world objects with the thematic information selected by the user (such as *religious buildings*). These 3D models of the past can be also displayed as interactive AR objects and the user can access additional information related to them within time intervals where they had that specific role (such as a sanctuary turned into a ruin later).

The purpose of describing 3D objects as compound geometries with part-whole hierarchies is to allow publishing changes related to the parts of a 3D model in time. Evolutions such as adding a bell tower to a church or demolishing the roof of a city hall can be represented with separate temporal representations and accessed by the AR application in order to reveal the stories behind it (3D evolution, photos, *etc.*).

Figure 4.13 shows a part of the RDF graph corresponding to an instantiation of this model for the example of *Hagia Sophia*. In order to facilitate the interpretation of this RDF graph in relation to the generic model presented above, we choose to represent the resource nodes that belong to the same `arcama-owl` class in Figure 4.12 with the same color and line format.

In this RDF graph (Figure 4.13), *Hagia Sophia* is represented by two temporal representation instances: `model:HagiaSophia#TI1` and `model:HagiaSophia#TI2`. These instantiations are depicted in Figure 4.11 with intervals $R_2$ and $R_3$, respectively. As it can be seen through the evolution in time of the real world entity, the first temporal representation instance represents a change in the geometry of the entity, whereas the second one represents a change both in its role and its geometry. Additionally, in this graph, the creator of the geometry file is associated to the file using Dublin Core property dc:subject.

In the following section, we detail the process of publishing these 3D models by interlinking them with LOD cloud using our data model.

## 4.4 Publishing 3D Models on the LOD

In order to make 3D objects visible in the LOD cloud and bind them with the resources from the other datasets (*i.e.* creating RDF triples), we have developed a Web-based application. This application automatically creates the RDF triples and adds them to the triple store (described in Section 6.2.1). We have followed the 5-star rating system as introduced by Tim-Berners Lee (details can be found in Section 2.2.6) to interlink our data with the LOD cloud. Therefore, they can be used by any linked data content further on.

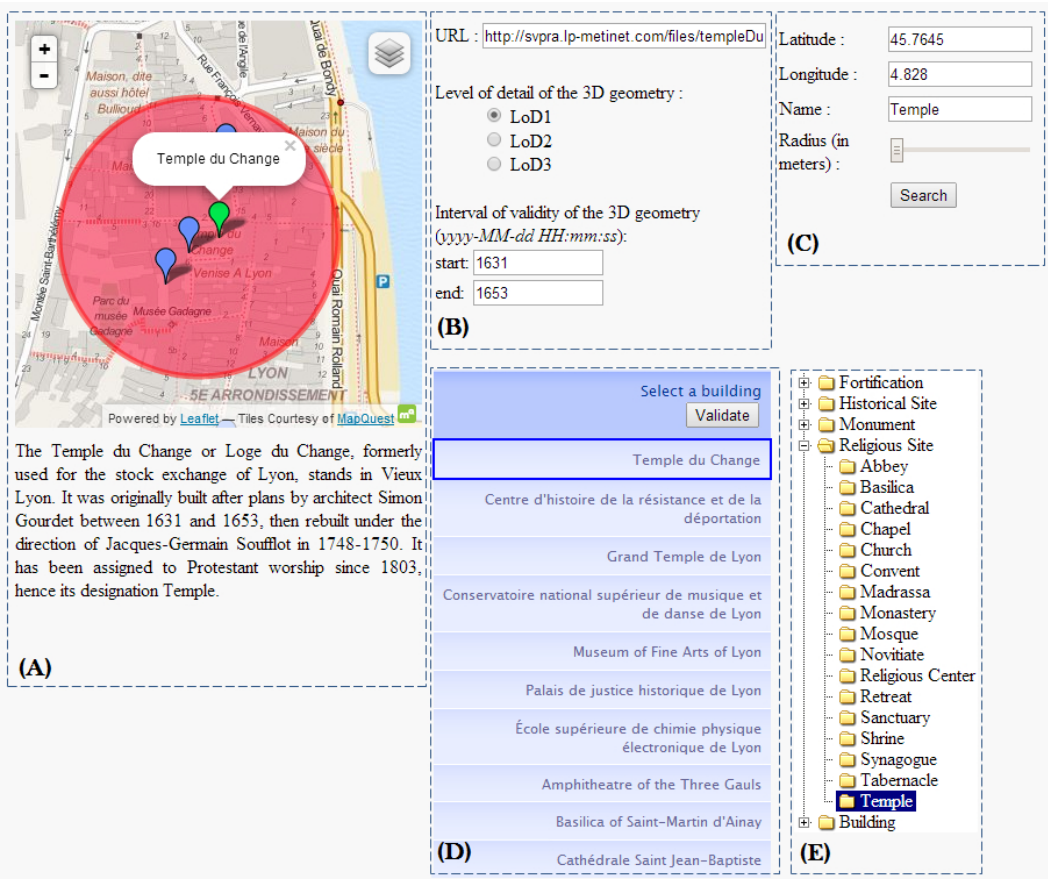



Figure 4.14 – The web-based application for interlinking 3D models with data sources on the LOD cloud.

When the 3D model is ready and available on the Web through a resolvable URI, we publish them on the LOD cloud by using the interface of our Web application (see Figure 4.14). During this process, the following steps take place:

– Shown on Figure 4.14 (B): The developer enters the URL of the 3D block that she wants to publish on the cloud, as well as the *level-of-detail* of the 3D struc-

ture: *LoD1* for transparent wire-frame blocks, *LoD2* for detailed transparent wire-frame blocks, *LoD3* for objects with textures. The interval of validity of the 3D geometry is needed and required in order to develop applications that deal with the temporal evolution of the entity. If the developer does not enter any temporal interval, then the web application warns her with an error message.

– Shown on Figure 4.14 (C): In order to find the geo-referenced resources on the LOD cloud (DBpedia for our case), she also inserts the geo-location of the model (*i.e.* one of the coordinates of the polygon vertexes or by clicking on the location of the building on the map) and if it is within her knowledge, the name (whole or partial name) and the role (function) of the entity from the available list (the roles in this list are based on the classes of MediOnto ontology). Lastly, she can also insert the radius in *meters* in order to limit the search in an area. Then the search can be initiated.

According to the knowledge of the developer about the entity, the provided information might vary. The purpose of this web interface is to allow developer to find the corresponding DBpedia resources for the instances of temporal representations and be able to publish them on the LOD cloud. Then according to her selections, the triples will be generated after validation.

The following scenarios might take place in order to find the corresponding resource on DBpedia.

**Scenario A)** If the developer provides only the *geo-location* (*i.e.* latitude and longitude) information:

1. The application first queries the DBpedia resources using the geo-location to have instances within a diameter, and then, uses the categories in MediOnto in order to eliminate the type of objects that is not relevant.
2. This result is then sorted so that the closest (distance) DBpedia resource appears on top of the list.

At this point, our interest is to focus on providing the user with the man-made structures. We can observe at this stage that MediOnto helps reducing the results that will be displayed on the map (in Figure 4.14 (A)). If we only query all the geo-located entities (*i.e.* those with *latitude* and *longitude* information) of DBpedia, we may end up with entities representing administrative divisions displayed on the map as well (*e.g. 5th arrondissement*, *Munich*, *Saint-Martin-d'Hères*, *Avenue Foch*, *etc.*). Therefore, in order to be able to search only for urban structures, we query

the DBpedia resources that have a matching class or category with MediOnto, as explained in Section 4.2.1.

**Scenario B)** The developer can also provide a radius in addition to the *geo-location* of the entity, following these two items above, in order to limit the search in a certain region,

3. The resources are queried using this radius (with geo:lat and geo:long attributes) and displayed on the map. If the resources are outside of this area, then they are eliminated from the query result.

**Scenario C)** If the developer enters the *geo-location* and the *name* of the building on the interface (or partial name, such as "*Theatre...*", "*Tour...*", "*Monastery...*"), the results are ordered by the spatial proximity, as well as the lexical similarity (we use the Jaccard index, for measuring the overlap of two lexical sets).

After the search, query results of DBpedia resources are shown as a list (Figure 4.14 (D)). The developer can select each of them to see the abstract related to these objects, which is displayed under the map view (Figure 4.14 (A)). This helps her to select the correct object, if it is within her knowledge.

When she selects the corresponding DBpedia resource, she then inserts the *url* of the 3D model, *the temporal validity*, *level-of-detail* of that model (see Figure 4.14 (B)). Finally, she also selects the *role* of that object from the list of MediOnto classes (see Figure 4.14 (E)) and validates. This process publishes the object interlinked with the corresponding DBpedia resource with the information described by our data model (see Figure 4.14 (B)).

**Generating the triples:** When the developer finds the corresponding DBpedia resource of the 3D model, she selects it and clicks on the Validate button (as can be seen in Figure 4.14 (D)). The application then constructs the RDF triples and stores them in our Parliament triple store [23] (a local triple store that we have in repository http://steamer.imag.fr/ontology/ where we publish our triples that interlinks 3D models to LOD using our data model) and allows the application to stock and retrieve information. We have used Jena [24], an opensource architecture to add, remove and find triples in the Parliament triple store and query DBpedia dataset. Parliament provides a query engine that supports spatial SPARQL queries. Using this support, the 3D models can be retrieved according to their geo-locations.

23. http://parliament.semwebcentral.org/
24. http://jena.apache.org

An example of triples that is generated by the application is shown below (in this example, we examine the temporal representation instances of Royal Monastery of Brou as we will further examine it in Chapter 6 in a use case scenario):

```
@prefix : <http://steamer.imag.fr/arcama/resource/RoyalMonasteryOfBrou#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix time: <http://www.w3.org/2006/time#> .
@prefix dbpedia: <http://dbpedia.org/resource/> .
@prefix mediaOnto: <http://steamer.imag.fr/ontology/mediaOnto#> .
@prefix arcama-owl: <http://steamer.imag.fr/ontology/arcamaOnto#> .

:entity rdf:type arcama-owl:Entity ,
                 owl:NamedIndividual ;
        arcama-owl:represents dbpedia:Royal_Monastery_of_Brou ;
        arcama-owl:hasRepresentation :TR1 .

:G1 rdf:type arcama-owl:CompoundGeometry ,
             owl:NamedIndividual ;
    arcama-owl:hasComponent :geometry1 ,
                            :geometry2 ,
                            :geometry3 .

:TP1 rdf:type owl:NamedIndividual ,
              time:Interval ;
     time:hasBeginning :constructionBegining ;
     time:hasEnd :constructionEnd .

:TR1 rdf:type arcama-owl:TemporalRepresentation ,
              owl:NamedIndividual ;
     arcama-owl:has3D :G1 ;
     arcama-owl:hasTimePeriod :TP1 ;
     arcama-owl:hasRole mediaOnto:monastery .

:constructionBegining rdf:type owl:NamedIndividual ,
                               time:Instant ;
                      time:xsdDateTime "1506-05-25T00:00:00"^^xsd:dateTime .

:constructionEnd rdf:type owl:NamedIndividual ,
                          time:Instant ;
                 time:xsdDateTime "2004-04-12T00:00:00"^^xsd:dateTime .

:file1 rdf:type arcama-owl:GeometryFile ,
                owl:NamedIndividual ;
       arcama-owl:levelOfDetail arcama-owl:LoD1 ;
       arcama-owl:fileUri <http://svpra.lp-metinet.com/files/monasterbrou
```

```
       .obj.672> .

:file2 rdf:type arcama-owl:GeometryFile ,
                owl:NamedIndividual ;
       arcama-owl:levelOfDetail arcama-owl:LoD2 ;
       arcama-owl:fileUri <http:/svpra.lp-metinet.com/files/monasterbrou
       .obj.673> .

:file3 rdf:type arcama-owl:GeometryFile ,
                owl:NamedIndividual ;
       arcama-owl:levelOfDetail arcama-owl:LoD1 ;
       arcama-owl:fileUri <http:/svpra.lp-metinet.com/files/monasterbrou
       .obj.682> .

:geometry1 rdf:type arcama-owl:ElementaryGeometry ,
                    owl:NamedIndividual ;
           arcama-owl:hasGeometryFile :file1 .

:geometry2 rdf:type arcama-owl:ElementaryGeometry ,
                    owl:NamedIndividual ;
           arcama-owl:hasGeometryFile :file2 .

:geometry3 rdf:type arcama-owl:ElementaryGeometry ,
                    owl:NamedIndividual ;
           arcama-owl:hasGeometryFile :file3 .
```

These RDF triples provide AR applications with an access to 3D models on the Web. The LOD resources interlinked with these models can also be discovered as additional information sources. Furthermore, using our data model, application developers can create various AR experiences. Since 3D models are published with temporal representations which concern their spatial and thematic evolutions over time, several application scenarios can be created, such as story telling during an AR application (see Chapter 6).

## 4.5 Conclusion

In this chapter, we have introduced our data model which we propose to conceptualize and publish 3D information as a dataset in the LOD cloud. We emphasize that publishing these *3D* geometries attached with *temporal* and *thematic* data (which we call 3D2T: spatio-temporal and thematic) has many advantages. This 5-dimensional information increases the usability of these models by augmented reality applications and provide them with the possibility to query the LOD cloud to access certain information related to these entities. The data model is also used

to create representations of the real-world entity with different temporal intervals when there is a spatial (a change of 3D geometry) or thematic change (a change of role/function of the enity) occur. These representations can be instantiated separately and further information can be attached to them (*i.e.* photos, text, videos, *etc.*) (explained in Section 6).

Publishing data on the LOD cloud requires interlinking it with at least one resource available on the Web (see Linked Data principles in Section 2.2.6). For this purpose, we have chosen DBpedia dataset as it is one of the most prominent knowledge bases in the LOD cloud. Our approach also takes advantages of the DBpedia ontology (classes) and its category hierarchy while annotating these 3D models with different roles. We use the variety of object roles available in DBpedia in order to maximize the number of objects with various thematic information that would be accessed using an AR application. However, since DBpedia is a community effort, at times the same data might be expressed in several different ways. Hence, extracting information from DBpedia on-the-go might be complicated. Therefore, we have re-used another ontology (MediOnto.owl) by mapping it with some DBpedia classes and categories in order to limit the search to a concise section of the DBpedia dataset. This ontology also constitutes the list of the *types* (*thematics*) of the buildings that will be used in order to generate an interactive mobile augmented reality application and deliver information about these objects.

Moreover, publishing the 3D models with temporal information that represents the periods of time during which the observed real world entity keeps the same form (geometry) and function (role) has further advantages. This way, application developers can create different experiences that is specific to the type of the building (*e.g. time travel through the construction periods of religious buildings*, *architectural information related to castles in a region*, *precise information related to the parts of a building etc.*) (details can be found in Chapter 6).

Making our data available on the Web results in re-using the existing structured data, being able to categorize our 3D models according to these semantic data, providing users with additional information using the Web of data and creating different experiences through mobile applications. The fundamental idea behind the LOD is that the value of the data is maximized if it is interlinked with other datasets. Therefore, binding our 3D models with LOD also increases the re-usability of these models.

In this chapter, we have mainly focused on publishing 3D models by binding

them with LOD resources and using our data model. In the following chapter, we discuss about what kind of LOD resources can be accessed using our data model through a mobile augmented reality application. We present different AR application layers that we need to achieve our purpose. We also describe the transitions and communications between these layers, as well as the development steps.

Chapter 5

# From LOD to Augmented Reality based Interaction

## Sommaire


5.1 Introduction . . . . . **141**
5.2 System Overview of the AR Application . . . . . **143**
5.2.1 Application Modules . . . . . 144
5.2.2 The Application Flow based on the User's Interaction . . . . 146
5.3 AR Objects as Points of Interests . . . . . **147**
5.3.1 A Warehouse for 3D Objects . . . . . 149
5.3.2 Tracking and Sensor Module . . . . . 149
5.4 Querying the LOD Cloud . . . . . **153**
5.4.1 Thematic and Temporal Visits of the Surroundings . . . . . 153
5.4.2 Creating Narratives using Multimedia Documents (Site Specific Storytelling) . . . . . 157
5.5 Conclusion . . . . . **161**


## 5.1 Introduction

Augmented Reality (AR) constitutes a digital layer superimposed on the view of the real world. Reality is *augmented* by adding this layer on top of the objects of the scene. Looking at the previous mobile AR research projects (see Chapter 1), we observe that they mainly focus on:

1. visualizing Points of Interests (PoIs) as textual or image annotations superimposed on the real view of the mobile device;
2. visualizing interactive 3D models (*i.e. laser scanned objects* or *detailed 3D objects with textures*, *etc.*) whose rendering requires high performance computing platforms;
3. using markers that solve the major problem of mobile AR, which is "*tracking the position (location and direction) of the user (the mobile device) with respect to the objects in the environment*".

With the recent developments, new approaches have focused on markerless tracking systems for AR in outdoor environments (see Section 1.3.2 for more details) by using: (i) only the inertial sensors of the mobile device (*i.e.* inertial tracking); (ii) image features and tracking between subsequent frames in order to model the camera movement; (iii) a hybrid approach, *i.e.* enhancing the sensor information with the help of image cues gathered from the real view in real-time (*i.e;* visual-based tracking).

Our AR approach is based on using light-weight and transparent wire-frame 3D objects (*i.e.* LoD1) superimposed on the real view of the surroundings of the user as an interaction layer when the user is observing her surroundings. In our approach, the user interacts with the environment through this adaptive virtual representation of the real world through her mobile device. Therefore, real-world objects in the surroundings themselves become the interactive source of information. Through this interaction, the user can access to a more detailed (*i.e.* LoD2 or LoD3) geometrical representations of the real-world object. AR makes this experience more realistic and provides efficiency in retrieving pieces information that interests the user and sends them to her via *push-pull notifications*.

In the following sections, we describe the design of a mobile Augmented Reality (AR) platform based on our approach. The prototype is an outcome of an Augmented Reality project that is developed in collaboration of the engineering school *IUT Lyon 1*[1]. The 3D objects, first, are published on a server accessible through the Web. We contribute to the project by interlinking these 3D objects with resources on the LOD cloud using our data model. Therefore, the 3D geometry can be thought of as an interface (or an entry point) for accessing to information on the LOD cloud. Then, the LOD cloud serves as an information base for the AR application that uses our data model.

After introducing the AR platform, we mention what kind of *additional* information can be provided to the user according to her interaction. We also describe the sources of such information and the steps that we follow in order to make them accessible. This additional information will help the user to explore the surroundings and to discover the hidden stories related to the observed site or structure.

1. http://iut.univ-lyon1.fr/

## 5.2 System Overview of the AR Application

During her AR experience with the application, the user is proposed a list of themes and interests related to the environment. This list is constructed by sending location-based information queries to the triple store. Triple store contains instances of arcama-owl ontology, thus information related to the temporal representations of the real-world objects (as explained in Section 6.2.1). The application initially gathers the types of structures around and supplies them to the user as a list. If, for instance, there is no religious building around the user, the related theme, such as "*religious heritage structures*", is not proposed to her in this list. The user selects the objects that she is interested in learning further. Her selection is stored in the application on the mobile device.

The user's interests with respect to her surroundings are inferred from her selection through this list of themes. According to the choices she made, the AR application can propose her different interactions, such as *architectural evaluation of a historical landmark*, *3D evolution of a castle*, *3D view of a city in previous decades*, *etc.* Using the data links associated with 3D geometries that are available on the LOD, as well as the information that is made available through the data model (temporal, thematic, level-of-detail and so on), these kinds of AR experiences can be created by the application developers. Furthermore, by knowing to which object the queried data belongs, these objects can be displayed in interactive mode. The application can also deduce the interests of the user by looking at her previous choices and construct a profile, which can be re-used to make suggestions to the user during her visit.

Our approach in displaying the interactive 3D AR objects relies on the following aspects:

1. Publishing the 3D models on a Web server and establishing a connection to this server from a mobile device in order to import and use these models in an augmented reality application.
2. Using the embedded sensors of the mobile device in order to retrieve the location and direction of the user.
3. Displaying the 3D model on the mobile device's screen superimposed on the real-world object.
4. Adding opacity and colors to the 3D object according to the existence of information and its types.

5. Achieving the super-imposition of the 3D object while the user is moving at the site.
6. Design of the user interface.

In order to be able to change the appearance of the 3D objects, the application relies on several modules. These modules need to communicate with each other in order to ensure the visualization of augmented reality views, wait for the interaction and display (or render) the information. We now describe these modules.

### 5.2.1 Application Modules

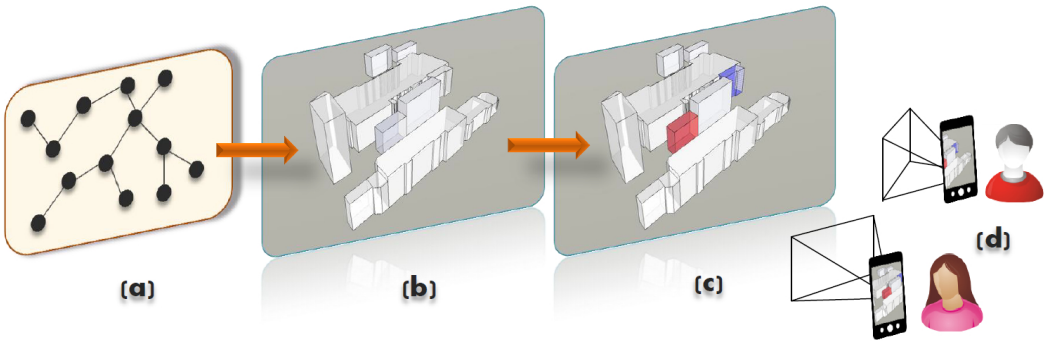



Figure 5.1 – Four modules of the mobile augmented reality application: (from left to right) information module, object importer module, object renderer module (visualization module), and interaction module.

In our system design, the AR mobile application is composed of four modules as illustrated in Figure 5.1 (a) The first module is the InformationModule. (b) The second module is the ObjectImporterModule. (c) The third module is the ObjectRendererModule. (d) And the fourth and last module is the InteractionModule.

The communication between these modules during the execution of the AR application is shown in Figure 5.2 (the SensorModule that is seen in this figure is described in the next section).

The InformationModule represents all the information sources available on the LOD cloud. The instances of temporal representations are published on the Web with resolvable URIs and interlinked with temporal and thematic information using our data model. Information managed by this module can be queried by the upper module (ObjectImporterModule) in two ways. In the first scenario, the application might initially have some pre-set themes to propose to the user (*architecture*, *history*, *first world war*, *middle age*, *black and white photographs from the past etc.*).

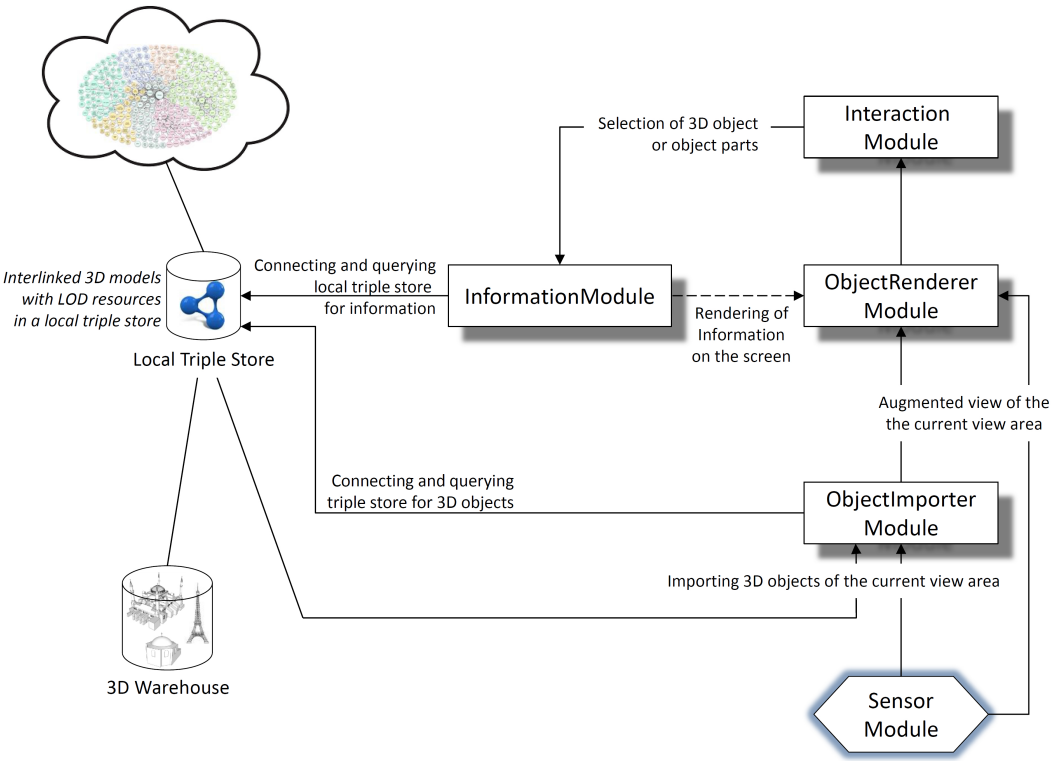



Figure 5.2 – The flow diagram during the execution of the AR application.

Then, a location-based query is executed in order to seek for some information belonging to these themes that are related to real-world objects around the user within a certain distance. If such information is found, the theme will be proposed to the user. In the second scenario, when there is no pre-set themes, the application gathers the interests of the user at the beginning by proposing her some object types (roles) that are available within a current diameter (*fortifications*, *religious buildings*, *monuments*, *etc.*). Only the information related to objects fulfilling the selected themes are displayed.

The ObjectImporterModule is the module responsible for communicating with the server in order to import the 3D models (see Figure 5.2). In this module, using GPS, the geo-location information of the mobile device is sent to the server. The 3D objects which are within the distance defined by the radius, which is set by the application developer (500 meters, for instance) and modifiable by the user, are retrieved from the local triple store and downloaded on the mobile device.

The ObjectRendererModule is responsible for rendering the 3D scene objects superimposed on the real view, as well as the 2D information templates such as lists of selections, images, texts, *etc.* When the user moves around a real-world

structure, using the sensors of the mobile device, the ObjectRendererModule re-renders the scene accordingly in order to keep the superimposition.

The InteractionModule handles the user interaction with the objects on the mobile device screen. This module communicates with the InformationModule through the ObjectRendererModule. If the user interacts with one of the interactive objects or object parts in order to access the information, this module generates a trigger. The trigger tells the application to communicate with the InformationModule and to construct a query composed of the object URI and the user's preferences to access to further information related to that object (photos, videos, text, *etc.*).

### 5.2.2 The Application Flow based on the User's Interaction

Based on the choices made by the user (*i.e.* her preferences and interests), the application can go through different states. Let us explain it through a scenario. Let us suppose that the user arrives in a city she doesn't have much knowledge about. Moreover, she doesn't have a map to visit the city or doesn't know where to start either. She turns on the AR application on her mobile device. The application uses the geo-location information in order to retrieve the 3D objects around her. Then, using the objects within a given distance, the application queries a list of thematic visits to the user, as well as the types (*i.e. roles, functions*) of the entities around her. The user decides to select one of the thematic visits (*e.g. history*). Then, the ObjectRendererModule filters out the object that match with the choice of the user and displays the corresponding transparent 3D objects on her smartphone screen, some of them being highlighted in different colors. The user can interact with these highlighted objects in order to access information. This information will be displayed differently by the ObjectRendererModule depending on its format (*i.e.* text, photo, 3D, *etc.*). The user can switch from viewing these information back to the augmented reality view. Therefore, in this scenario, information is delivered to the user continuously as long as she wants to be guided by the application. This mode of interaction corresponds to push notifications.

In a second scenario, the user doesn't make any selection from the list of thematic visits (*i.e.* stories) or roles of architectural structures. Instead, she wants to ask a question to the AR application as she passes by near a historical site. Initially, the 3D objects retrieved by the ObjectImporterModule are displayed being superimposed on the real view by the *ObjectRendererModule*, as in the previous scenario. Since she did not make a choice, the objects are transparent but are, in this case, not

highlighted with colors. She interacts with one of the objects by clicking on it on the mobile screen. A query is sent to the InformationModule and the available data, if any, interlinked with this object on the LOD are queried. The application returns a set of information to the user; for instance, an abstract about the object available on DBpedia, the architect of the building, the 3D evolution of the object, some images and audio files, *etc.* Since the user queries information from the application, the mode of interaction corresponds to pull notifications. These two scenarios can be observed in Figure 5.3.

## 5.3 AR Objects as Points of Interests

As a result of our collaboration with the engineering school *IUT Lyon 1*, a Web-based application has been developed to upload the 3D augmented reality objects on a Web server and thus, let them have unique identifiers over the Web. It forms a

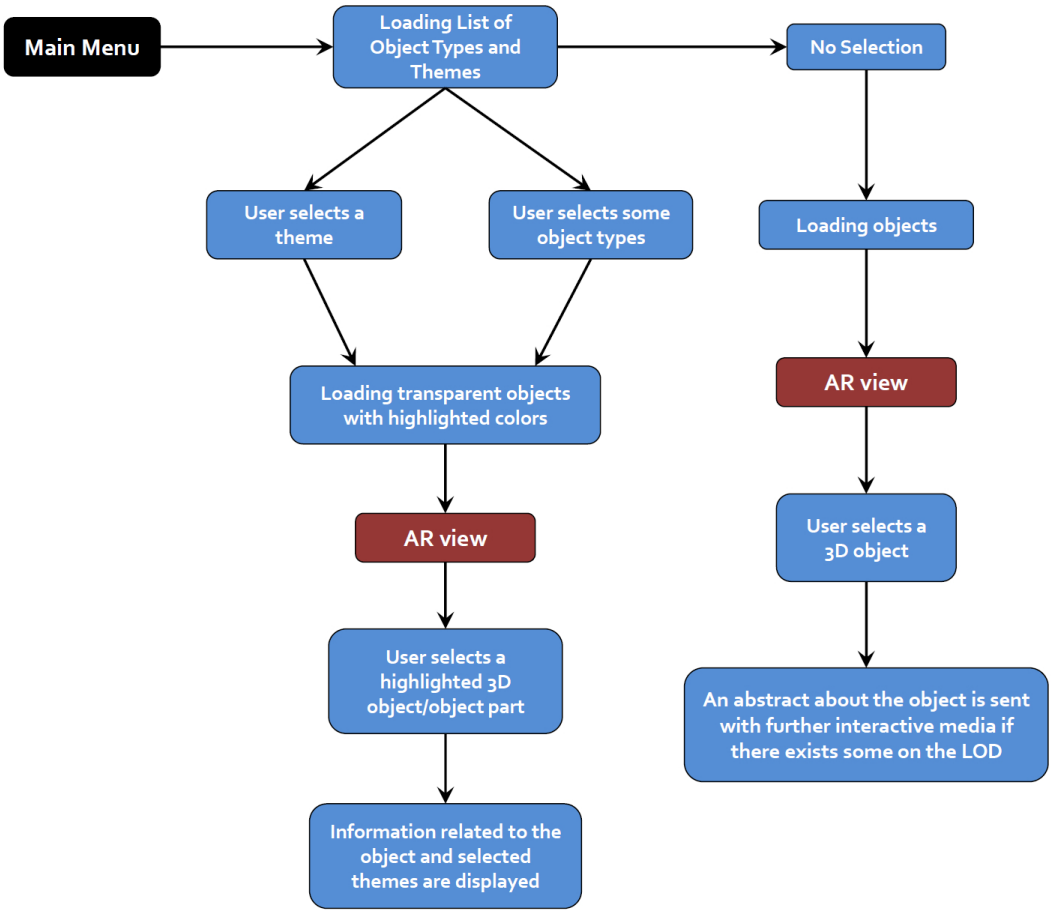



Figure 5.3 – The transitions based on the choices made by the user.

warehouse that keeps the 3D geometries and other information by associating them with these objects. Once the geometries are available with resolvable URIs over the Web, we publish them on the LOD cloud with the related thematic, temporal and 3D semantics (levels of detail, which 3D object parts belong to which object, *etc.*) as explained in Section 4.4.

The 3D geometry files have the information of vertex data (geometric vertices, texture vertices, vertex normals, faces...) and the geo-location of the real-world object. The application connects to this Web server, and using the GPS information of the mobile, imports the 3D objects of the surroundings from that warehouse to the mobile device. As the 3D models become available locally on the device, they can be displayed using the hardware and software resources of the device. OpenGL ES [2], the standard for embedded accelerated 2D or 3D vector graphics, which is already available on most of the mobile devices, is used to display the 3D objects on the screen.

The objects which have interesting information for the user (according to the interactions of the user, *i.e.* her selection, interests, *etc.*) are displayed in different colors, in order to gather objects according to their types or to stories that can be told about them. The identifiers of these objects are registered on the mobile device and this means that these objects have information which may interest the user and that they are available for further interaction. Then, the AR application displays these objects using specific colors (according to the theme or story), thus highlighting them in order to draw the attention of the user visiting the place.

All the information, including the 3D objects (since we publish them on the cloud as well), comes from the LOD resources and therefore has unique identifiers (see Section 2.2.4 for further explanations). User can interact with the highlighted objects in order to access these data. Therefore, when, for instance, the user selects a type of building that she might find interesting, a query is formed and sent to the LOD cloud (*i.e.* SPARQL endpoints of the selected resources). When a 3D object identifier is returned, it corresponds to an information source on the LOD cloud. Depending on the type of the information, the 3D objects are displayed in different colors. The color information is embedded within the application itself, which means that it is assigned by the application developer, but also modifiable by the user. For instance, the developer may assign different colors for specific types of structure, such as *religious buildings* in *blue*, *monuments* in *red*, *museums* in *yellow*, *etc.*

2. http://www.khronos.org/opengles/

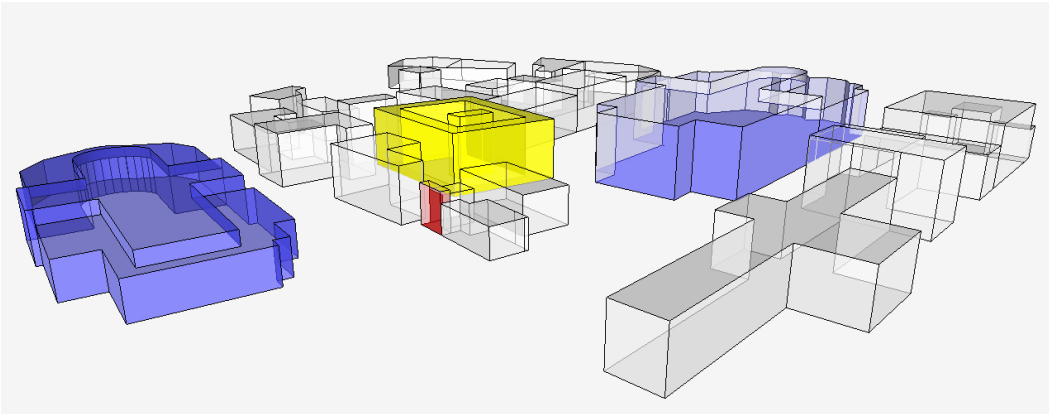


Figure 5.4 – According to the *types* of objects, semi-transparent 3D models are displayed in different colors, such as *religious buildings* in *blue*, *monuments* in *red*, *museums* in *yellow*, *etc.*

These objects which are highlighted in different colors form the Points of Interests (PoIs) of the AR application. They are the entry points to access further information (videos, texts, photos or pieces of information from LOD resources such as architect, date of construction of the building, *etc.*), or a story about the real world object told by displaying a sequence of multimedia documents (a representation of highlighted 3D objects can be seen in Figure 5.4).

### 5.3.1 A Warehouse for 3D Objects

In order to upload the 3D geometry to the warehouse, the developer provides some basic information about it: *name of the object* (the name of the real world object represented by the 3D model), *type/role/function of the object*, *latitude*, *longitude*, *transparence* and *a short description* are some of such pieces of information (see Figure 5.5 and Figure 5.6).

### 5.3.2 Tracking and Sensor Module

In order to superimpose the 3D models on the real-world objects of the captured scene, the physical location and direction of movement of the mobile device are tracked in real-time all along the run time of the application. We have chosen to develop a sensor-based tracking using the GPS, accelerometer and gyroscope that are embedded in the mobile device. A hybrid approach would have been more robust, but visual-based tracking necessitates more time to develop and robustness in tracking was not our initial priority. We have given the priority to creating a

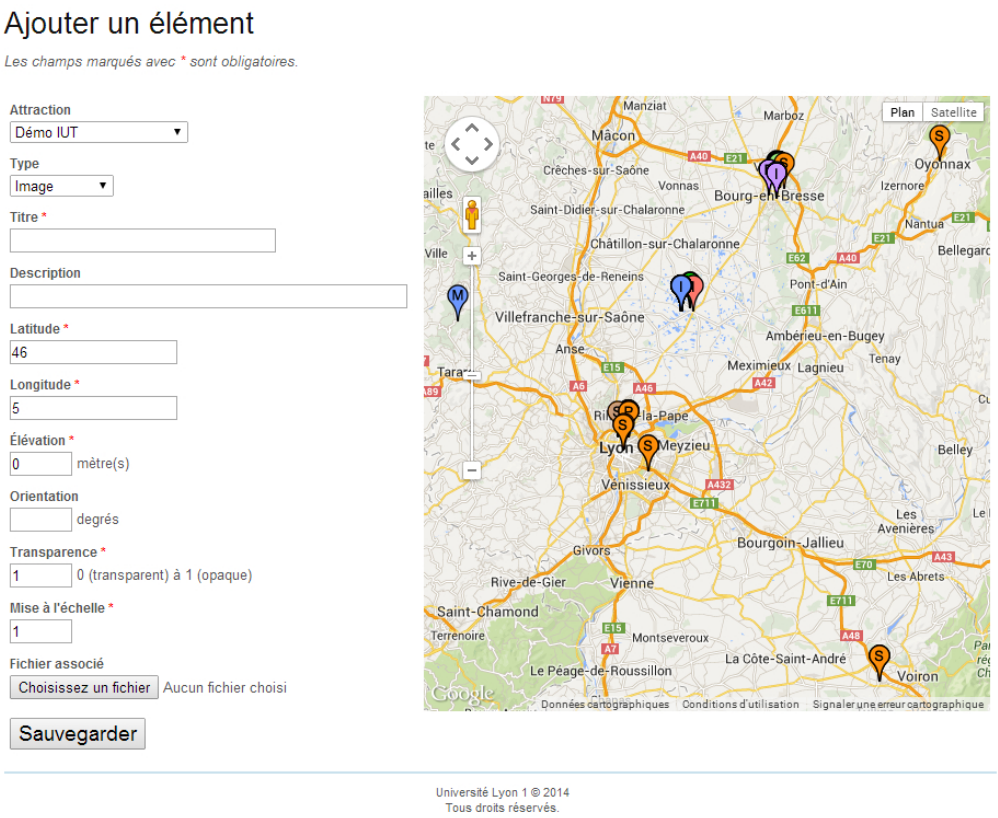



Figure 5.5 – The interface for adding 3D objects to the warehouse.

mobile platform that basically support interaction and rendering of 3D objects. By the way, the sensor-based tracking has turned out to be robust enough, but we have also added an interaction layout where user can manually correct the misalignment, if it occurs.

The communication between the Sensor Module and other modules can also be seen in Figure 5.2. As shown in this figure, the Sensor Module running on the mobile device gives feedback to the ObjectRendererModule for the superimposition of the objects on the captured view. As represented in Figure 5.7, the sensor data is refined and merged (*i.e.* fusion), in order to correctly estimate the position and direction of movement of the user in real-time. This step is necessary to ensure the alignment of the 3D model with the real-world scene. In the meantime, geo-location and selection/interests of the user are used to retrieve 3D models and determine the ones to be rendered in different colors. Then, the 3D virtual objects are rendered on the captured frame of the screen.

When we have started the development of the AR application, the objects were

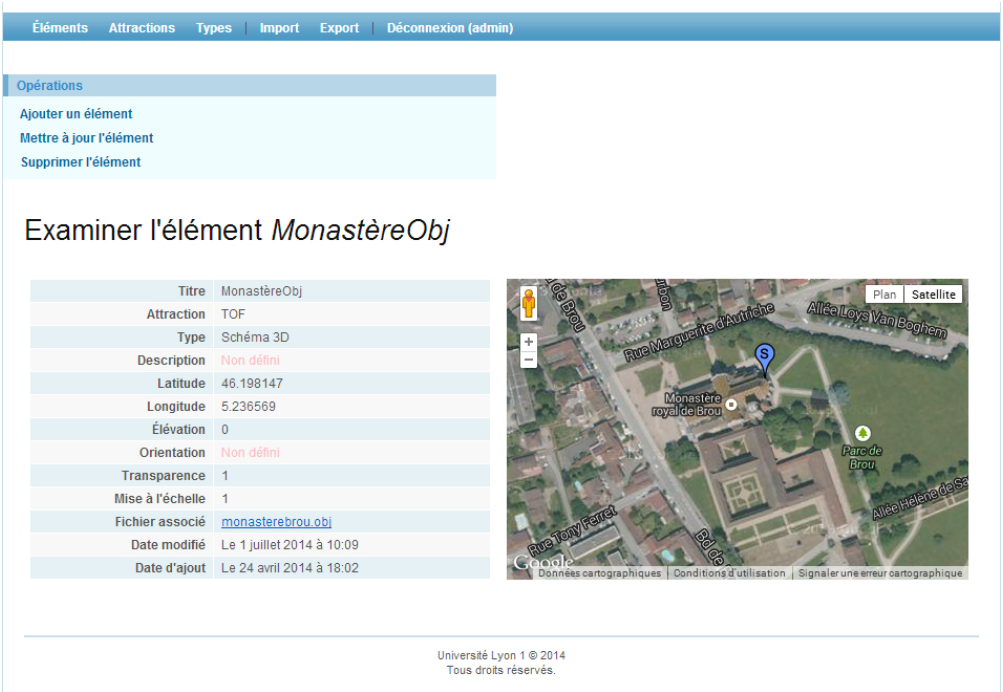



Figure 5.6 – One of the 3D objects of the warehouse with the piece of information about it.

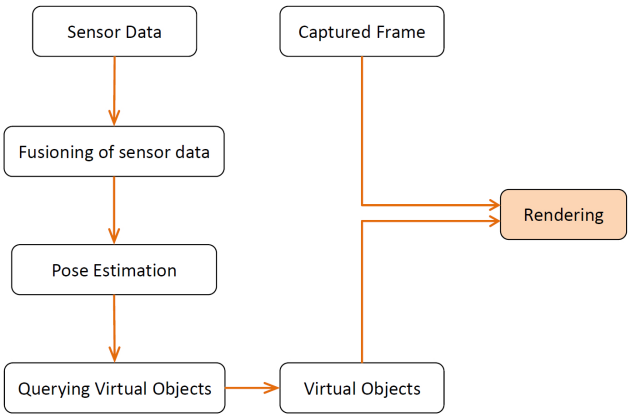



Figure 5.7 – Sensor data is fusioned in real-time in order to render the virtual objects with superimposition on the real objects.

not displaying stable, and they were slightly oscillating (*i.e.* jitter problem). We have improved the stability by applying a Kalman filter to the sensor data [Kalman 1960] and we have fusioned the data in order to detect the user's movement in 3D space (*i.e.* phone's position, heading and acceleration). Since we haven't used a hybrid approach, sometimes the objects do not overlay perfectly on the real scene, yet further optimization and improvements will enable a robust superimposition of these

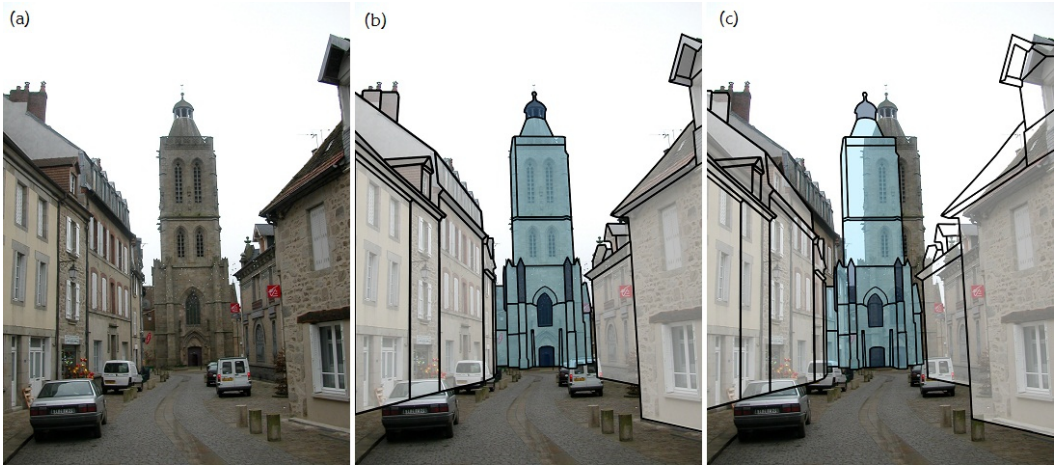



Figure 5.8 – The AR view with misalignment: (a) the captured real-world view; (b) the representation of the aligned AR models; (c) the representation of the misalignment due to the errors in sensor data.

objects for future work. Then, in order to solve possible misalignments, besides AR and information view, we have added a *menu* button on a lower corner of the screen. When the user interacts with this button, she has access to classical 3D tour features, such as moving the 3D object on $x$, $y$ and $z$ axis, rotating the model around its center or moving the model closer or farther (see Figure 5.9). Hence if the misalignment is quite noticeable (such as in the example in Figure 5.8), the user can access to these features and re-align manually the object and lock the screen. Then, the Sensor Module detects user's movement by fusioning gyroscope and accelerometer data (*i.e.* phone's heading and acceleration).

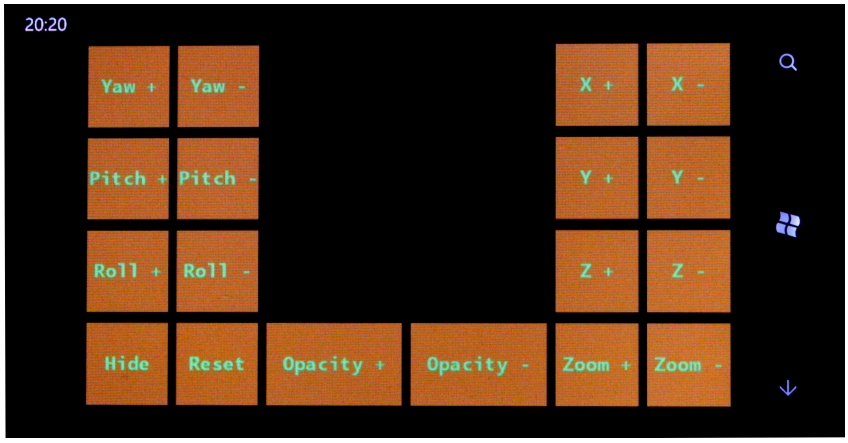



Figure 5.9 – Screen capture of the interface for allowing users to correct the misalignment of the 3D model.

## 5.4 Querying the LOD Cloud

When the 3D augmented reality objects are displayed, the ones highlighted with colors indicate the existence of interesting information depending on the choices of the user. We provide the users with two lists:

(i) one list for the types (*role, function, etc.*) of the structures which is built using the MediOnto classes (*e.g.* temple, museum, statue, *etc.*) (see Section 4.2.1 for further details). These building types constitute the thematic layer of the objects in our data model. Therefore, when a 3D object is published in the LOD cloud using that data model, it is associated with one or more roles.

(ii) the other list consists of the site specific stories that are dedicated to a 3D object or its parts. The *story* aims at providing the user with a narrative through digital media when she accesses information about the real world object or site (see Section 6.2.2 for further information).

### 5.4.1 Thematic and Temporal Visits of the Surroundings

In this section, we describe how we manipulate thematic and temporal information attached to a 3D model using the data model. By *thematic*, we mean providing user with some information related to certain types of structures according to her preferences. The 3D models that are superimposed on the real view have identifiers that describe them uniquely over the Web. These identifiers are interlinked with additional information (such as thematic and temporal information, and 3D semantics as described in Section 5.3) which is basically stored in our Parliament triple store (as mentioned in Section 4.4). Using these links, the AR application delivers additional information to users. In our approach, thematic information (*i.e.* types/roles of the structures) is one of the main sources of data since a particular type or role associated with an object can be queried for specific information depending on the role of this object (for instance, monasteries can be queried about their *architect*, *architectural style*, whereas commercial sites may be queried about their *opening and closing hours*, *etc.*).

When the AR application is launched, it first queries the Parliament store which has triples that makes temporal representations of real-world objects available on the LOD cloud using arcama-owl ontology. The application uses the geolocation information of the user and her preferences, if she has any. The user can state her preferences in advance (*i.e.* before the AR mode is launched) by looking at the list of *all* the object types, or explore her surroundings on-the-go by making selections

from the list of *available* object types that lie within a certain distance from her. If there are no preferences made in advance, the application provides the user with a list of object roles (*i.e.* thematic list) present in the vicinity of the user by querying the Parliament triple store with geolocation and types of structures as can be seen in the following pseudo-code:

```
queryString="
    SELECT ?entity WHERE {
    ?entity arcama-owl:represents ?rdfResource.
    ?rdfResource geo:lat ?lat.
    ?rdfResource geo:long ?long.
    FILTER(?lat <= X + dLat && ?lat >= X - dLat &&
        ?long <= Y + dLong && ?long >= Y - dLong)
    }";
resultSet=do_query(queryString);
foreach(result in resultSet){
    queryString=SELECT DISTINCT ?role WHERE {
    ?entity arcama-owl:hasRepresentation ?tempRep
    ?temRep arcama-owl:hasRole ?role
    }
}
```

In this code, the SPARQL query first searches for the entities which are within a certain distance from the user's location. The physical location of the user is indicated by coordinates X and Y (*i.e.* the latitude and the longitude). In order to define the search window in spherical coordinates [3], dLat and dLong are calculated using the Haversine formula by the application. This formula basically converts meters to spherical distances in latitude and longitude (the details related to this formula can be found in [Robusto 1957]).

The roles of the structures are basically the classes of MediOnto ontology as described in 4.2.1. When the roles of structures around the user are retrieved, they can be displayed on the screen in a categorical order (see Figure 5.10). The roles can also be ordered according to the number of real world entities that fall into each category. The user can then make her selection according to her interests.

3. Since DBpedia currently does not allow running GeoSPARQL queries, our location-based queries are based on rectangular regions defined by putting user's location at the center of the rectangle, instead of circular regions defined by a radius.

Only the information related to the entities within the category of her selection will be delivered through a push notification. An example of such a thematic list is given in Figure 5.10 where the user selects to get information only about religious structures. Although, it is not detailed in the pseudo-code above, initially the top classes of MediOnto are displayed on the screen. For instance, if an object with role *church* is retrieved, it falls into the upper category *Religious sites*. The user can choose *Religious sites* category, or only *Churchs* category if her preferences are more precise.

As the user changes her physical location by walking around her surroundings, the application sends queries to the triple store to check if there are other types of structures around. If there are other interesting categories in her surroundings, another list is proposed to the user. The selection of the user defines the objects that will be rendered in different colors. Each 3D object that belongs to the same category is therefore rendered with a specific color.

The temporal visit consists in visualizing the temporal evolution of a 3D struc-

Figure 5.10 – The representation of a thematic list displayed on the screen based on the types (*i.e.* roles) of entities that are detected around the user. *Above:* Four general categories of type of structure around the user. *Below:* When the user selects "Religious sites" as her center of interests, sub-categories are displayed to be selected.

ture. This is achieved by sending a query to the triple store to acquire the 3D geometries with some temporal information attached to them. Then, the application, based on these time stamps, renders the 3D models consecutively. The user observes the evolution of the 3D structure like an animation. A bar at the bottom of the screen indicates the time interval over which a slide can be moved by the user in order to visualize the temporal evolution with the desired pace. A representation of this is shown in Figure 5.11.

As shown in this figure, the 3D model that depicts an edifice (*e.g.* the Hagia Sophie) at a given period of time is superimposed on the real view of the structure. We observe that the geometry of the real-world entity has evolved and this 3D geometry represents it over a time interval in the past. If there is another temporal model available, the application will display it. This model can also be manipulated (rotated, zoomed into, *etc.*) by the user to explore its composition and structural details. In this mode, the application will switch from the augmented reality view

Figure 5.11 – A representation of a temporal visit of an edifice (Hagia Sophia, Istanbul, Turkey): *Above:* the temporal evolution of Hagia Sophia is displayed on the mobile screen in AR view. *Below:* the user visualizes only the 3D view and interacts with the object in order to observe it in details.

to the 3D visualization view.

### 5.4.2 Creating Narratives using Multimedia Documents (Site Specific Storytelling)

Once the superimposition of the 3D models on the real world is achieved, and the preferences of the user are collected, the AR application enables a wide range of surrounding exploration experiences for the user. The interaction of the user with one of those models can trigger the AR application for looking up for some extra information using the unique identifiers of these geometries over the LOD cloud. The 3D virtual models of the environment will act as an interaction layer while the user walks through the urban structures. Moreover, the temporal dimension of the 3D models makes it possible to attach information to that interval of the object which allows the data to be displayed with a time component as well.

To this regard, we propose to integrate multimedia documents in the scene in order to tell the location's history in addition to the temporal 3D evolution of the site. These documents may be historic media retrievable from archives, or up-to-date information about the building. Depending on the application, they can be stored from the LOD cloud or Web sites related to that structure. This can be achieved in various scenarios. For example, when the user interacts with one of the objects that she selected from the list of object types (as described in Section 5.4.1), the 3D parts of the object can be associated with various information related to them (*e.g.* the historical drawings of a church's spire or crossing, or the wooden galleries that belongs to a 14th century castle, *etc.*).

In order to inform the user about the presence of additional information related to the entity, the multimedia documents will be accessible by interacting with the corresponding parts of the building. In order to indicate where they are, only these parts of the building will remain highlighted (see the representation in Figure 5.12). These colored 3D object parts indicate the user that interesting information can be reached through interaction.

As a matter of fact, the Linked Open Data cloud is not only composed of DBpedia, but it embodies various data sources ranging from archived photos, posters, newspaper articles, films, to library collections, museum resources, *etc.* As mentioned in Section 2.3.2, it is widely used in tourism and cultural heritage domains. One of these datasets, Europeana is a portal for European cultural heritage data which promotes more open data to be published. `data.europeana.eu` is an ongoing pro-

Figure 5.12 – A representation: the parts of the 3D object are rendered in different colors to tell the user that there is some additional information related to these parts of the real world entity. The user interacts with these parts on the screen to access this information.

ject started in 2012 with the aim of making Europeana metadata available as Linked Open Data on the Web [Haslhofer & Isaac 2011]. Currently, it includes about 20 million texts, images, videos and sounds, gathered by Europeana, transformed into linked data. data.europeana.eu provides unique identifiers for the Europeana resources (for instance, http://data.europeana.eu/item/92056/BD9D5C6C6B02248F187238E9-D7CC09EAF17BEA59), as well as the downloadable dump files [4] and a SPARQL endpoint [5]. Europeana items are connected with external links to other LOD cloud datasets [6], such as places as provided by Geonames, persons as provided by DBpedia, time periods from an *adhoc* time period vocabulary, *etc.* It also enriches the items with metadata using Dublin Core properties, such as dc:subject, dc:type, dct:temporal and dc:publisher, which makes it easy to find a particular piece of information related to the multimedia item.

Even though Europeana portal is currently under development and it is not yet fully integrated into the LOD cloud, we claim that data from such a portal can contribute to reveal the historical data related to a location. As a proof of concept, we extend our approach to integrate storytelling aspect using multimedia items about the real-world entities. Once Europeana resources are fully interlinked with DBpedia or another dataset from the LOD cloud, these documents can be

4. http://data.europeana.eu/download/2.0/
5. http://europeana.ontotext.com/sparql
6. http://labs.europeana.eu/api/linked-open-data/data-structure/

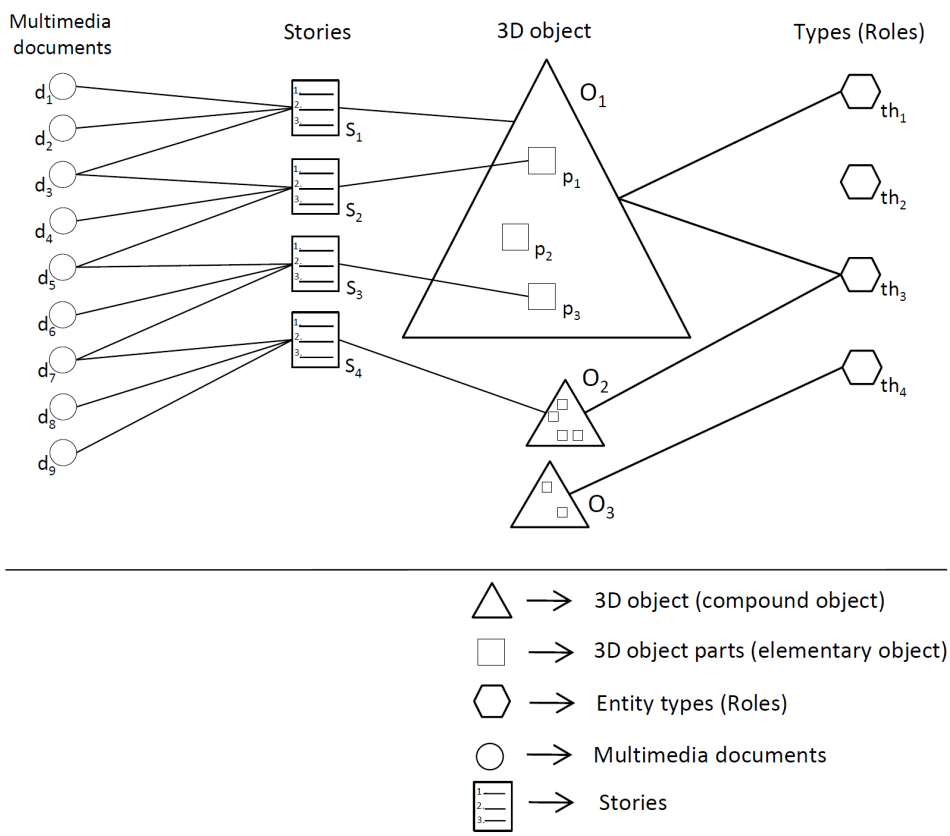



Figure 5.13 – The relationships between types (roles) of structures, multimedia documents, site specific stories, 3D objects and object parts.

widely used to explore the hidden information related to a location. As shown in Figure 5.13, each 3D object is attached with at least one type (role) information in our triple store. The 3D objects are compound geometries (see Section 4.3 for details related to our 3D data model) and can be made up of several smaller object parts (*i.e.* elementary geometries). Multimedia documents can be attached to the temporal representations of the 3D objects (in our data model). However, in order to provide a fluent narrative to the user, these documents should be displayed in certain logical order which constructs a story related to that entity.

We express our approach in a graphical representation in Figure 5.13. As illustrated in the figure; a 3D object $O_1$ may be composed of several smaller object parts: $O_1 =< p_1, p_2, ..., p_m >$ ($p$ denoting a 3D object part). We call each of these object parts, an *elementary geometry* in our data model (the arcama-owl ontology (see Section 4.3)). These elementary geometries compose the *compound geometry* $O_1$. The object can also be depicted with one whole geometry instead of several smaller parts. Each object $O$ has to have at least one type information $th_i$ ($th$ de-

noting the type (role, theme) of the object). Each object part, $p_i$ of $O_1$, (or the compound object itself) can be attached with one or more stories $S$ denoting a past event, a historical figure lived in the area, an architecture-related anecdote, *etc.*: $<S_1, O_1>$, $<S_2, p_1>$, $<S_3, p_3>$, $<S_4, O_2>$; where $<S_1, O_1>$ indicates that $S_1$ is a story related to the compound object $O_1$. Every story is a file which holds a sequence of multimedia documents available on the Web or on the LOD cloud. For instance, $S_1$ is composed of the multimedia documents $<d_1, d_2, d_3>$. These multimedia documents are displayed to the user in order to communicate a narrative for the purpose of storytelling. Therefore, a story is the set of multimedia documents presented in a logical sequence $S_i = d_k \rightarrow d_l \rightarrow d_m \rightarrow ...$ (our storytelling approach is detailed in Section 6.3).

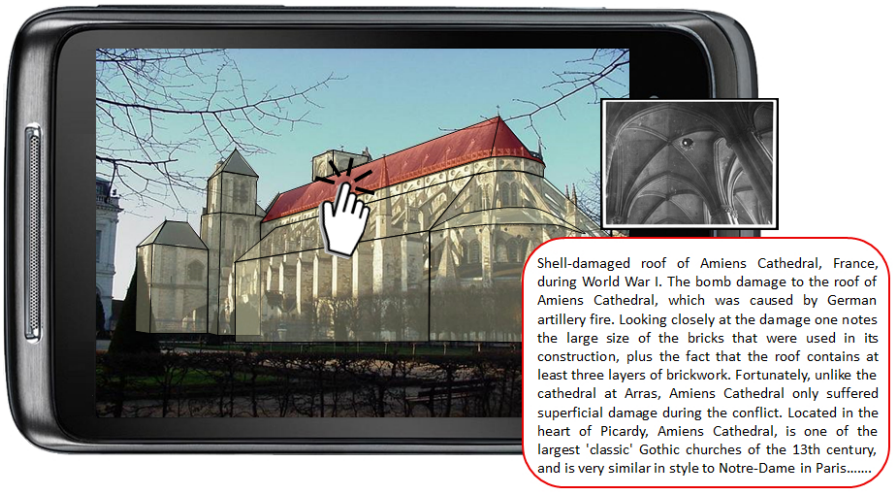



Figure 5.14 – The user interacts with the highlighted object parts in order to access information from Europeana portal to learn more about the structure.

In Figure 5.14, the user interacts with the highlighted part of an object, which is the roof of the Amiens Cathedral in France. To access the information (story) related to that part proposed by the application. This interaction triggers a search towards interlinked documents to that part of the model. The query returns a photograph from Europeana dataset [7] (as its metadata indicates dc:type photograph) and displays it with a textual description (indicated with dc:description metadata). We can also imagine that the description is delivered to the user as a voice recording, as some AR application may propose audio visits. In the next chapter (Chapter 6), we give examples related to our storytelling approach through a use case scenario.

7. http://europeana.eu/portal/record/9200316/BibliographicResource_3000092751362.html

## 5.5 Conclusion

In this chapter, we have introduced an overview of an AR application that uses our data model to communicate with the LOD cloud and provide users information related to their surroundings. In order to realize this, we explain the application modules that are necessary to communicate with the LOD resources, import the necessary information on the mobile device, render the 3D objects and realize the interaction through these objects. Our data model allows developers to create thematic visits or present site specific stories through digital media. We explain with examples how our data model helps to exploit the LOD cloud in order to achieve these.

In our augmented reality approach, the 3D models are seamlessly integrated into the real-view of the user's surroundings. By interacting with this digital content, the user can access information as if she is directly interacting with the real-world entities through their mobile devices. The superimposition of the 3D transparent models is achieved using sensor-based tracking technology. Since the superimposition is not the primary concern of our proposition, we have not conducted extensive research upon different augmented reality methods (*i.e.* based on embedded sensors, visual tracking or hybrid solutions, *etc.* - please refer to Chapter 1.3.2 for more details). However, the fusioning of embedded sensor data has been satisfactory enough to develop our approach.

The information sources are LOD cloud datasets and our Parliament triple store (introduced in Chapter 2.2.6). Mobile users access information by interacting with 3D AR objects. By selecting the 3D model through the mobile screen, a query is generated on the mobile device and sent to the SPARQL endpoint of our local triple store. Our proposition is based on a generic data model which can be extended by several AR applications in different domains. The main purpose of our approach is not to help users find their ways, but rather to allow them exploring their surroundings, identifying and pointing their smartphone towards objects around them and discover information and stories behind these objects according to their interests. To the best of our knowledge, there are virtually no studies conducted about this purpose which uses LOD cloud as an information base and 3D objects at the same time for augmented reality.

In our proposition, the user can focus on her navigation process, rather than trying to find an interesting piece of information among all the available data. This is achieved using our triple store which holds the triples that interlink the 3D models

with the sources from LOD (as introduced in Chapter 4). Time is considered as another dimension and it has a considerable importance in our model. Describing the geometric evolution of a 3D object (how it looked in the past, but also how it will look in the future) by using temporal attributes in a data model makes it possible to attach information to that object (or its parts) for a given time interval.

Another factor is the preferences of the user. In order to avoid some information overload in our approach, instead of presenting *all* the information available related to *all* the surrounding objects, we use LOD mechanisms to present the information in a more organized manner so that the user can choose among them according to her interest (*e.g.* list of available categories, stories, temporal information, *etc.*). The application queries the information sources available within a certain diameter and asks for the user's preferences.

These are the general aspects of our architecture. The more the 3D models are published using our data model on the LOD cloud, the more the usability of these models will increase. Future perspectives for these kinds of applications that use our architecture are numerous. Various application scenarios can be imagined, such as travelling in time, thematic navigations in a city or site specific storytelling, *etc.* Using the LOD cloud resources as a knowledge base has many advantages. The cloud is rapidly growing. Finding some information among such vast amount of data sources is a difficult task. However, the LOD cloud comprises semantically annotated structured data accessible on the Web. It provides access mechanisms for its datasets. Therefore, we claim that by coupling these 3D models with LOD resources and publishing them on the cloud with our data model, we make information search an easier task for AR applications. Using our approach, further effort to integrate new data into AR applications is not necessary. Once these data are available in the LOD cloud (as the LOD cloud expands everyday), they can automatically be queried by the application, since they are published according to the principles of Linked Data. For instance, let us assume that the DBpedia page of a church building is updated. The application developer does not need to make additional effort for these data to be accessible through the AR application.

Developing storytelling applications using our architecture is also one of our goals. We apprehend the augmented reality concept as an interface and opportunity to access and provide the story which may be told by using multimedia documents. In this regard, museum or archive collections can be integrated when telling a story about a real world object, which also increases the potential of these archival data

to be managed actively in user friendly applications.

In the following chapter, we introduce a use case scenario based on an edifice, the Monastery of Brou in France, and give details about how the cultural heritage or architectural stories related to this edifice can be told using our architecture. The scenario results from a continuing collaboration between the Monastery, and the engineering school IUT Lyon 1 [8].

8. This collaboration has been established during the scholar year 2013-2014, while the author of this PhD dissertation was working as a research and teaching assistant at the IUT Lyon 1. Other researchers from different domains (history, history of art, *etc.*) have also been involved in the project.

Chapter 6

# ARCAMA-3D

**Sommaire**


6.1 Introduction . . . . . **165**
6.2 General Architecture and its Components . . . . . **166**
6.2.1 Triple store . . . . . 166
6.2.2 Providing Further Information from the LOD Cloud Depending on Different Application Scenarios . . . . . 169
6.3 Use Case Scenario: Royal Monastery of Brou . . . . . **172**
Stories Related to the Monastery . . . . . 173
6.4 Conclusion . . . . . **180**


## 6.1 Introduction

In the previous chapters, we have introduced our approach of AR in which we use 3D object as an interactive information layer on top of the real-world view and our proposition related to how the LOD cloud can actively be used in accessing information through AR. Using our data model, we make 3D objects available as a dataset and interlink them with structured LOD information sources. Following these links, users can access to relevant information that belong to the objects in their surroundings. Application developers can use these links to create (query, communication, *etc.*) mechanisms that conduct this automatically upon the demand of the user. For instance, the user may want to know more about an object during her visit and interact with it on her mobile device. The mobile application may automatically create specific queries, such as to retrieve more detailed 3D models of the building, an abstract about it, some information related to when and by whom it was built, *etc.* These are specific features (functionalities) of the application and have to be decided by application designers.

However, in order to explain how some basic mechanisms can be made operational, such as communicating information between the triple store, the 3D warehouse,

the LOD cloud and the mobile device *etc.*, we propose an architecture. After introducing this architecture, through a use case scenario, we analyse how our augmented reality approach can be used to tell stories related to the surrounding objects by extending our data model.

## 6.2 General Architecture and its Components

In order to facilitate the information search and discovery in our system, we propose a modular architecture which covers all the aspects mentioned in Chapter 3. We name our architecture ARCAMA-3D (Augmented Reality for Context Aware Mobile Applications with 3D). The components (modules) help application designers and developers to create their own AR experiences for different domains by being able to extend the data model, bind 3D models with other data sources on the LOD cloud, cover a selected part of LOD and feed their application only with these specific data sources.

The data sources in our architecture are LOD data sources (DBpedia and other information sources), 3D object data (that we eventually publish on the LOD cloud), and mobile sensor data. Figure 6.1 shows the general architecture of our approach. ARCAMA-3D provides the basic necessary communication channels between data sources in order to retrieve location aware information and manage knowledge, visualization and interaction mechanisms for the mobile augmented reality applications. Various applications can be developed using ARCAMA-3D. Data acquisition steps will be the same for all of them, but the content can change from one to another. Each application can provide different experiences, topics or concepts and the content available in LOD can be shaped around them. For instance, an application can focus on some domain specific concept by favoring (through querying) some parts of the triple store (*e.g.* specific types of building roles) and associating some multimedia documents with 3D objects in order to tell a *story* about the place or the structure. This way, the content (*i.e.* triple store, 3D objects, media items, LOD datasets, *etc.*) is re-used across different contexts and by doing so, its value increases.

### 6.2.1 Triple store

The triple store is the most important component of our architecture (see Figure 6.1). Data acquisition is performed using the triple store. It holds the triples that are generated while *publishing* the 3D models using our *data model* (details

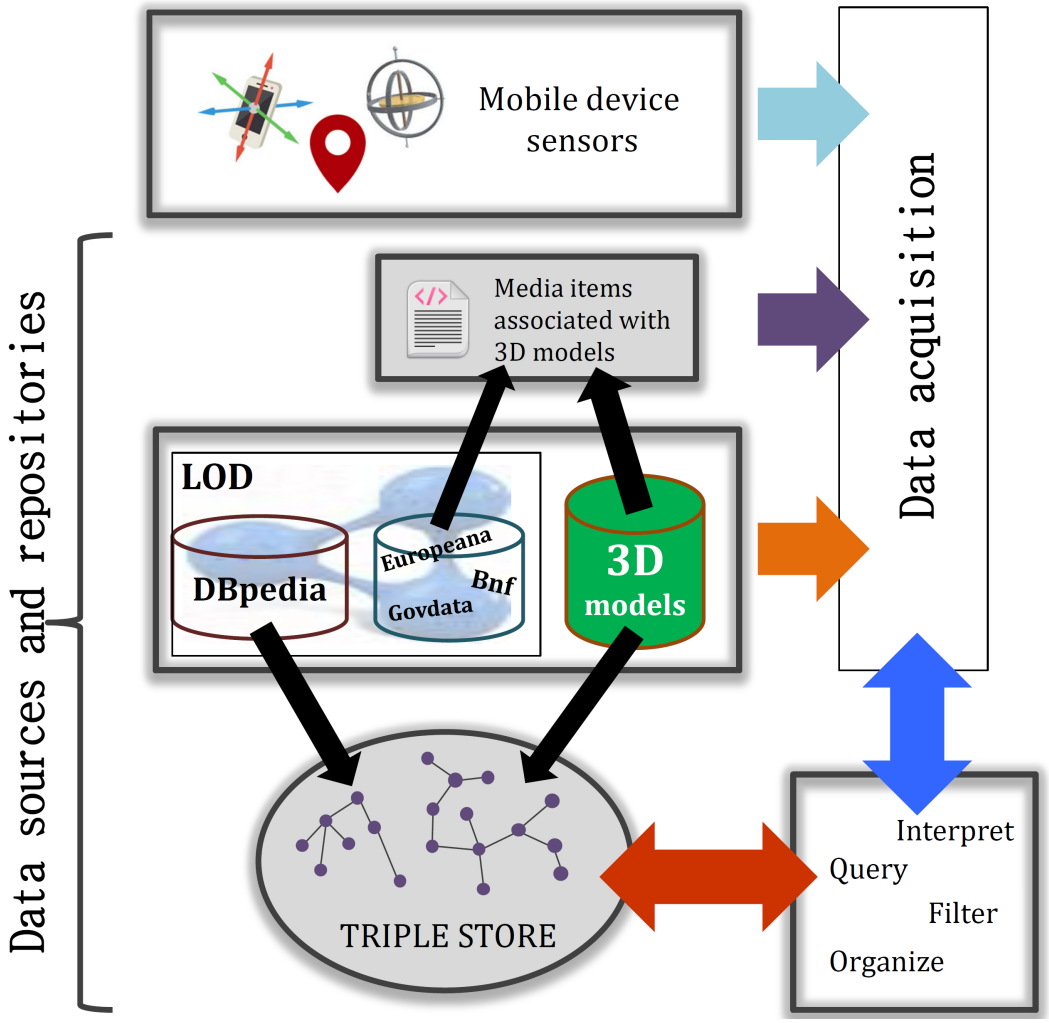



Figure 6.1 – Overview of the general architecture.

can be found in Section 4.4). It is where the application communicates and retrieves the temporal associations, roles, location information and 3D geometry related to the building structure. Domain-specific applications can be designed based on these data.

The triple store is queried through its SPARQL endpoint, as illustrated in the UML diagram in Figure 6.2:

(1) Initially, a location-based query is sent to the server from the mobile device.

(2) This query is executed to retrieve the 3D objects that are within a certain radius.

(3) These 3D objects are stored on the mobile device.

(4) At this step, the device has the URIs of the 3D geometries. Then, using

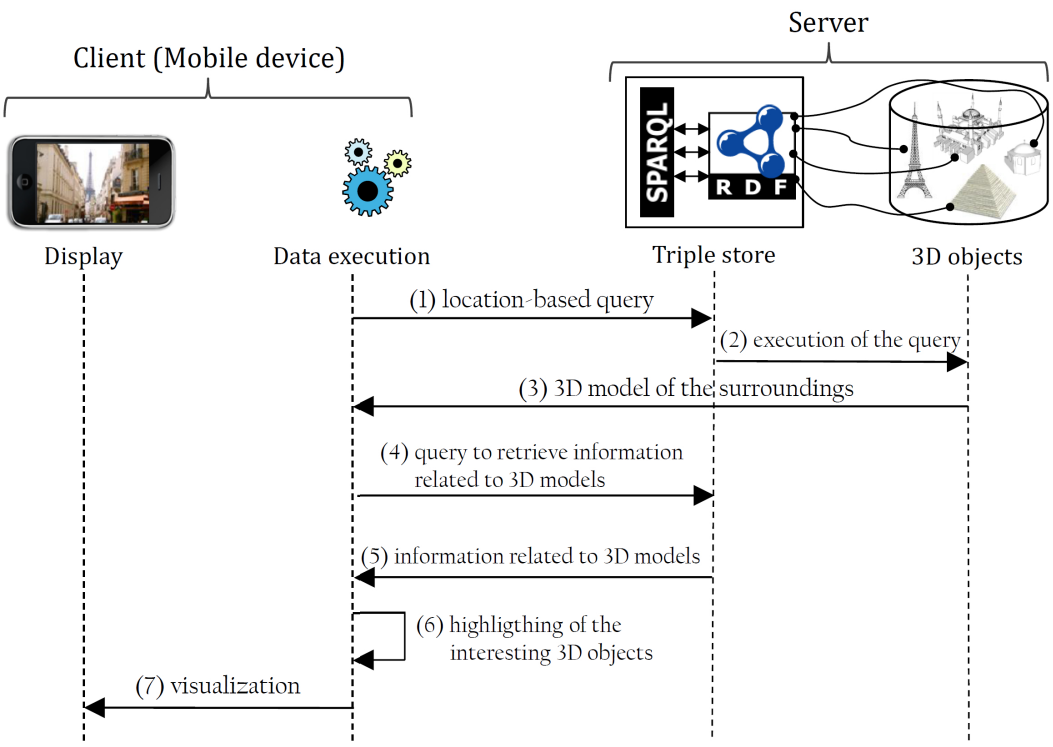



Figure 6.2 – The UML diagram of the information query-retrieval process between the mobile device and the triple store.

the data links that are on the triple store, it acquires information about these 3D objects, according to the user's selection. Which information to be searched for in the triple store depends on the application (role of the object, 3D geometry of the objects in different temporal intervals, *etc.*).

(5) The identifiers of objects that has some interesting information (*e.g.* an abstract from DBpedia about a specific building type, 3D models of a real-world object that existed during a certain time period, *etc.*) are stored on the mobile device as a list.

(6) The application changes some parameters of these 3D objects in order to render them differently.

(7) The AR view is superimposed on the captured scene with these interesting 3D objects highlighted and interactive. From now on, the user can interact with these objects to access the information that is retrieved from the LOD cloud.

### 6.2.2 Providing Further Information from the LOD Cloud Depending on Different Application Scenarios

As we have previously described, a 3D object may have multiple representations. In our data model, we name these representations as *temporal representations*, which consists in the unique combination of the spatial (3D), temporal and thematic information about an object. We think that these representations can also be used to bind other multimedia documents from LOD resources or other available sources with the purpose of telling *stories* related to these objects. '*Story*', in our approach, corresponds to providing digital media in order to assemble together a coherent narrative that allows user to learn or envision about that site or structure. The stories are generated by storing a logical sequence of some multimedia documents in a metadata file. For instance, let us imagine that a photo of a church is made available on the LOD cloud and interlinked with the corresponding DBpedia resource. Thus, once we retrieve this DBpedia resource, we can also query this photo from the LOD cloud following the links that bind them. Therefore, by doing this, we can access all the multimedia documents, if they are interlinked to that DBpedia resource. A representation of our approach can be seen in Figure 6.3. In this figure, we illustrate that domain-specific applications with stories related to themes such as "*Traveling in time*", "*Islamic architecture*", "*Construction stories*", "*16th century*", *etc.* can be told to the user by using metadata files which hold a sequence of multimedia documents and by realizing query-retrieval mechanisms to search data on the triple store.

In our approach, the user is interacting with the environment using a mobile device. Thus, she may not have the time to access these multimedia documents and understand what they are about. However, by displaying some multimedia documents in a logical order, a story can be told to the user which she cannot perceive by just looking at the representations of the object. Though, the order of consultation of these documents is important in telling a story about that object. Now, let us imagine that there are two photos available on the LOD cloud related to that church and they were taken 30 years ago, before the tower of the church was built. So these photos belong to a previous 3D geometry which has a temporal validity. Hence, they belong to another *temporal representation*. At this point, we propose to interlink these two documents with the church's corresponding temporal representation depicting the time interval when its tower was not yet built. We illustrate our approach in Figure 6.3. This way, when the user, for instance, looks

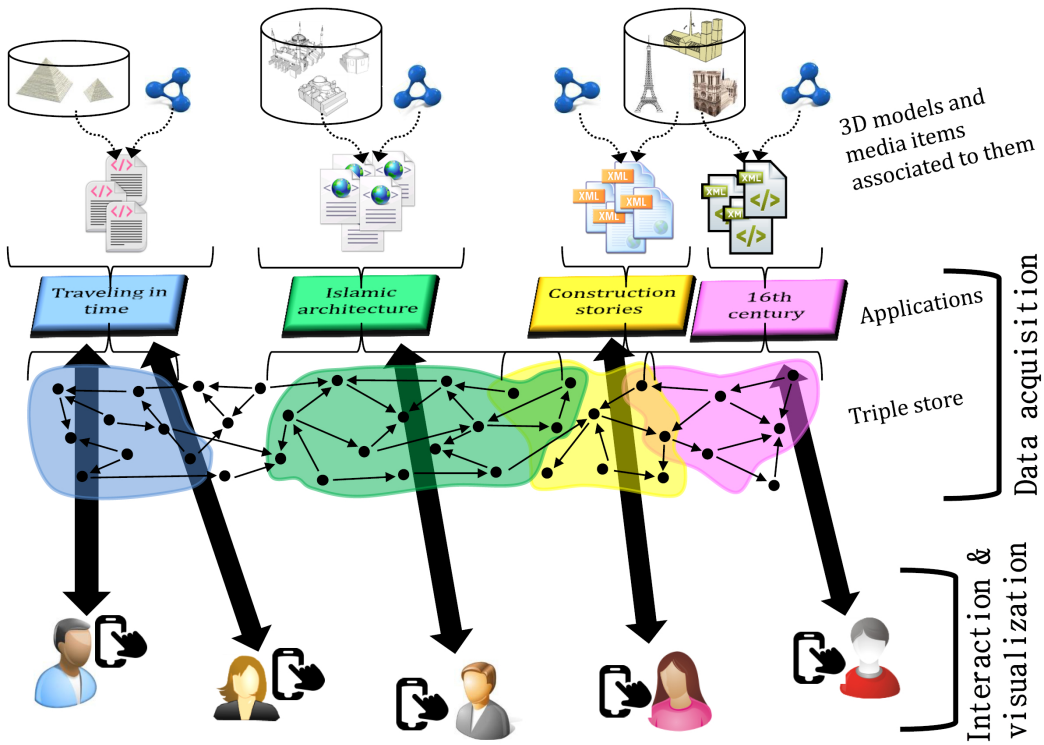



Figure 6.3 – Various applications can be developed using ARCAMA-3D.

at the previous 3D geometries of this church, she can also interact with them to access to these multimedia documents.

In Chapter 5, we mention that when the application starts, it uses the data model to access to the 3D geometries with temporal and thematic information. Then, it asks user which type of objects interests her during her visit. Then, for the object type that she selects, for instance religious buildings, the 3D objects of that type are highlighted and interactive. She can access different 3D representations of an object by selecting them on the screen. At this stage, while a 3D geometry from a past time is displayed, the application also highlights it (or its parts) to indicate that there are *stories* about this temporal representation of the object.

We can also imagine that, instead of selecting the object types that are interesting, the application may also provide the user with a temmporal visit by asking her to select a temporal interval at the beginning of the application. According to her selection, for instance 20th century, the triple store is queried and the 3D model identifiers and geometries are sent to the user's mobile device and displayed in an augmented reality view. Then, the objects for which some information belonging to this temporal interval is available are highlighted. The user walks around the loca-

tion and chooses one of the objects on the mobile device's screen while observing her environment at the same time. This generates another query using that *temporal representation* of that object and provides user, if any, with a story belonging to that building.

One of the benefits of this approach is that it provides a way to access various multimedia documents that are available on the cloud. The LOD cloud has many digital libraries, as mentioned in Chapter 5, such as, Europeana [1], CulturaItalia [2], the National Library of France (BnF) [3], DigitaltMuseum [4], *etc.* These libraries provide access to digitalized books, paintings, films, museum objects, *etc.* Recently, several initiatives related to open access to knowledge have been supported by governments, organizations and a global network of people. Some of these initiatives are within the cultural heritage domain (such as ICOMOS (International Council on Monuments and Sites) Open Archive [5], OpenGlam [6], Open Cultuur Data [7], Culture Grid [8], *etc.*). While more and more data are available through open formats and licences, researchers are also concerned by developing applications to exploit and put these information into use. Without proper interpretation, these data would be useless [Schultz *et al.* 2011], [Keßler & Kauppinen 2012]. As described in [Gurstein 2011], the necessary three-step process of Open Data is "access", "interpretation" and "use". This is the reason why we propose in the following an approach that suggests new opportunities exploiting the datasets of LOD and Open Data. It is based on creating content (meaning, stories, *etc.*) by effectively accessing, interpreting and using these data coupled with a data model for Augmented Reality. Making 3D models accessible in the LOD cloud using our data model is one of our contributions. But we also bring an insight in using these 3D models as an interaction layer in order to access site specific stories.

In order to achieve this storytelling-oriented approach, we propose using metadata files that are uploaded as an attachment to the *temporal representations* of a real-world object. These metadata files contain links to multimedia files that are related to that object, or a part of it (see Figure 6.3). They will be afterwards executed by the mobile device in order to tell a story about that real world object

1. http://labs.europeana.eu/api/linked-open-data/introduction/
2. http://www.culturaitalia.it/opencms/index.jsp?language=it
3. http://data.bnf.fr/semanticweb
4. http://www.digitaltmuseum.no/
5. http://openarchive.icomos.org/
6. http://openglam.org/
7. http://www.opencultuurdata.nl/english/
8. http://www.culturegrid.org.uk/

(such as the story of its construction, historic figures related to it, events, intangible cultural heritage elements, or other interesting stories that might interest the user).

## 6.3 Use Case Scenario: Royal Monastery of Brou

The Royal Monastery of Brou, located in Bourg-en-Bresse, France, is a monument with a flamboyant gothic architecture built as a monastery in the 16th century (between 1506 and 1532). The monastery serves principally as a museum nowadays. A project has been initiated in 2013 between the curators of the monastery and academics from IUT Lyon 1 [9] in the field of information technologies and from *Université-Pierre-Mendès-France* [10] in Grenoble, whose expertise is the art history of the middle ages.

Sometimes, visiting a place may not be enough to reveal all the stories behind their architectures. There are many untold stories related to the objects surrounding us which cannot be accessed through exhibitions or over the websites of these monuments, and most of them cannot be grasped by simply visiting the place. These stories may be related to the historic figures that had lived/worked there, related to construction or restoration of the monastery, sculpted details that are on its façades or figures that appear on its stained glass windows, *etc.* However, often most of these stories are kept in the archives as documents, photos, videos, *etc.* As mentioned earlier in this dissertation, mobile users have often limited time to read long documents over the internet or collect pieces of information which are stored in different locations (brochures, archives, internet, *etc.*) in different formats. Then, many narratives of history remain accessed by only a portion of people who are experts of specific fields (historians, writers, scholars, *etc.*) or by only few enthusiasts. However, if such knowledge is presented to users with a systematic and user-friendly approach, the general public can get the most out of it by interacting with the object on their mobile devices.

With this idea in mind, in this section, we explain how remarkable site-specific stories can be transmitted to mobile users by generating narratives from multimedia documents. We develop our ideas working on the case study of the Royal Monastery of Brou and create a use case scenario for a user visiting the monastery. Our aim is to allow visitors of this monument to access and explore these site-specific stories through an augmented reality interface. Using our architecture, we extend the spa-

9. http://iut.univ-lyon1.fr/
10. http://www.upmf-grenoble.fr/

tial, temporal and thematic representations of the 3D structure with these stories. To provide an interactive access, we associate these narratives with the temporal representations of the monastery as metadata files. Then, the application reads and interprets these files to provide the user with these stories. The interaction to access them can be made through various ways. A list of stories can be provided to the user and upon her selection, the parts of the 3D building can be highlighted while telling the story via multimedia files; *e.g.* an audio file can be played while different parts of the structure are highlighted, or a photo with a text can be displayed indicating which parts of the building they belong to, or a 3D animation can be shown, *etc.*

### Stories Related to the Monastery

The 3D evolution of the building through time (*i.e.* construction of the structure and further changes in its geometry) may be one of the stories that can be told to users using the ARCAMA-3D architecture. When a user is interested in getting more information about a specific real-world object (or her preferences indicate a specific theme such as religious buildings), the application can query the triple store to see if there is more than one 3D geometry depicting different temporal representations.

For instance, using an application that displays the 3D evolution of the monastery, the user may want to access all its 3D representations to analyse and interact with it further (by zooming, rotating, *etc.*). Thus, she interacts with the 3D view of the monastery on the screen. The displayed 3D object view is initially LOD1 and the most recent temporal representation of the structure. Interacting with that model creates a query as in the following which interrogates our triple store to access all other geometrical representations with their corresponding temporal interval information and level-of-detail:

```
PREFIX : <http://steamer.imag.fr/arcama/resource/RoyalMonasteryOfBrou#> .
PREFIX arcama-owl: <http://steamer.imag.fr/ontology/arcamaOnto#> .

SELECT ?geometryfile ?lod ?t
WHERE {
    :entity arcama-owl:hasRepresentation ?TR .
    ?TR arcama-owl:hasTimePeriod ?t ;
        arcama-owl:has3D ?CG .
    ?CG arcama-owl:hasComponent ?EG .
    ?EG arcama-owl:hasGeometryFile ?geometryfile .
    ?geometryFile arcama-owl:levelOfDetail ?lod .
 }
```

In this query above, we interrogate the entity (RoyalMonasteryOfBrou) using the information available to get all the temporal representations (represented with «TR» in the query). As described by our data model, these representations can have a unique combination of geometry, temporal interval and thematic information. This SPARQL query above is automatically generated on the mobile device when the user interacts with the object and sent to the SPARQL endpoint of the triple store. It uses the unique identifier of the entity that is queried from the triple store (see line 1 in the query) to access further information related to it. Accessing to this identifier is trivial since when the application is executed initially, it retrieves the nearby object URIs in order to superimpose them on the captured scene (as explained in Section 5.4.1). In order to display the temporal evolution of the entity, we need the time intervals, «t» (see line 4 in the query), and compound geometries «CG» (see line 7) to be retrieved from the triple store. These compound geometries are made up of several (or only one) elementary geometries, EG (line 7). We retrieve all the geometry files (line 10) and the level-of-detail of these geometries (line 11) in the query.

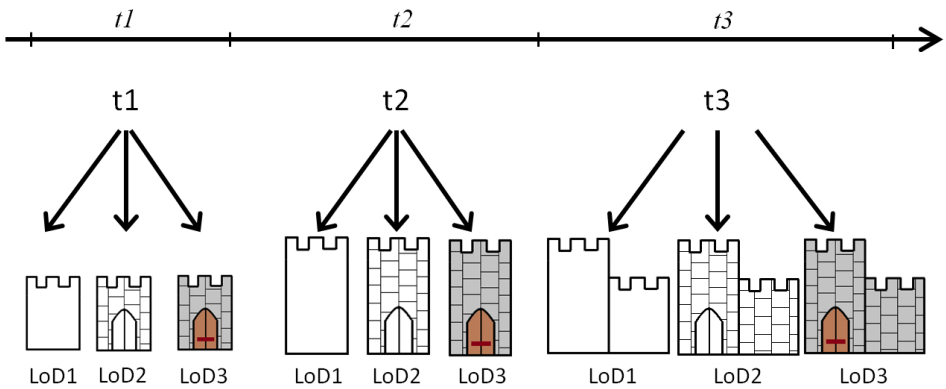



Figure 6.4 – The representation of the queried information to depict the 3D evolution of an object: *3D geometries* with their *levels-of-detail* and *temporal intervals*, ordered chronologically as *t1* is the oldest and *t3* is the most recent interval.

The geometry files, and temporal and level-of-detail information related to these geometries are then stored on the mobile device and interpreted by the application. To this point, the device has retrieved the 3D geometries, level-of-detail information regarding these geometries and time interval during which the geometries are valid. A representation of this information can be seen in Figure 6.4. This figure represents the query results, which is the evolution of a monument. In this example, the object is composed of three temporal representations which are valid during the temporal

intervals *t1*, *t2* and *t3*, in chronological order. Between each representation, there is a change in the geometry. Such as from interval *t1* to *t2*, the construction of the monument continues; or from *t2* to *t3* an additional structure is added to the monument. In each representation, the object is represented with three different levels-of-detail, LoD1, LoD2, LoD3 (as explained in Section 4.2.2).

As depicted in this figure, such information can be easily interpreted by the application. The 3D evolution of the building is displayed through a fixed LoD, such as the 3D objects with LoD1 of intervals *t1*, *t2* and *t3* in the figure. In the meantime, while geometries are displayed, the user can also increase/change the level-of-detail, navigating from LoD1 to LoD2 and LoD3, which will be proposed by the application to the user. This presentation of temporal evolution of 3D spatial information with different levels-of-detail creates a narrative, namely the *3D temporal evolution* of the object.

For the Royal Monastery of Brou, we want to display the geometrical evolution of the bell tower depicted in Figure 6.5. Therefore, the 3D elementary geometries depicting the bell tower in different time intervals (*t1*, *t2*, *t3*) are published in the triple store. Above in this figure are shown these representations of the tower. Below are shown the 3D representation (compound geometry) of the monastery with the bell tower (elementary geometry) displayed in another color (blue). This is achieved by the application using the identifiers of these elementary geometries that are retrieved from the triple store as explained in this section. If there is more than one elementary geometry file with the same level-of-detail depicting the bell tower, it means that this change is reflected as a different *temporal representation*. Using this change of representation in the retrieved data, the application highlights this geometrical part in order to inform the user that there is a story available behind it (the bell tower as depicted in this figure), and that she can interact with it to visualize this 3D evolution with the temporal information attached to it.

On the other hand, an object (or a part of it) may contain many other stories which can be transmitted to the user by using multimedia documents, as we have mentioned previously. In the case of the bell tower, the application may also present the user some images, play an audio record or a video to tell a story related to this object part. However, these multimedia documents should be displayed in a logical sequence in order to create a meaningful narrative.

For this purpose, first of all, the multimedia documents should be accessible by the application. Therefore, we upload them on the Web, or we re-use the ones that

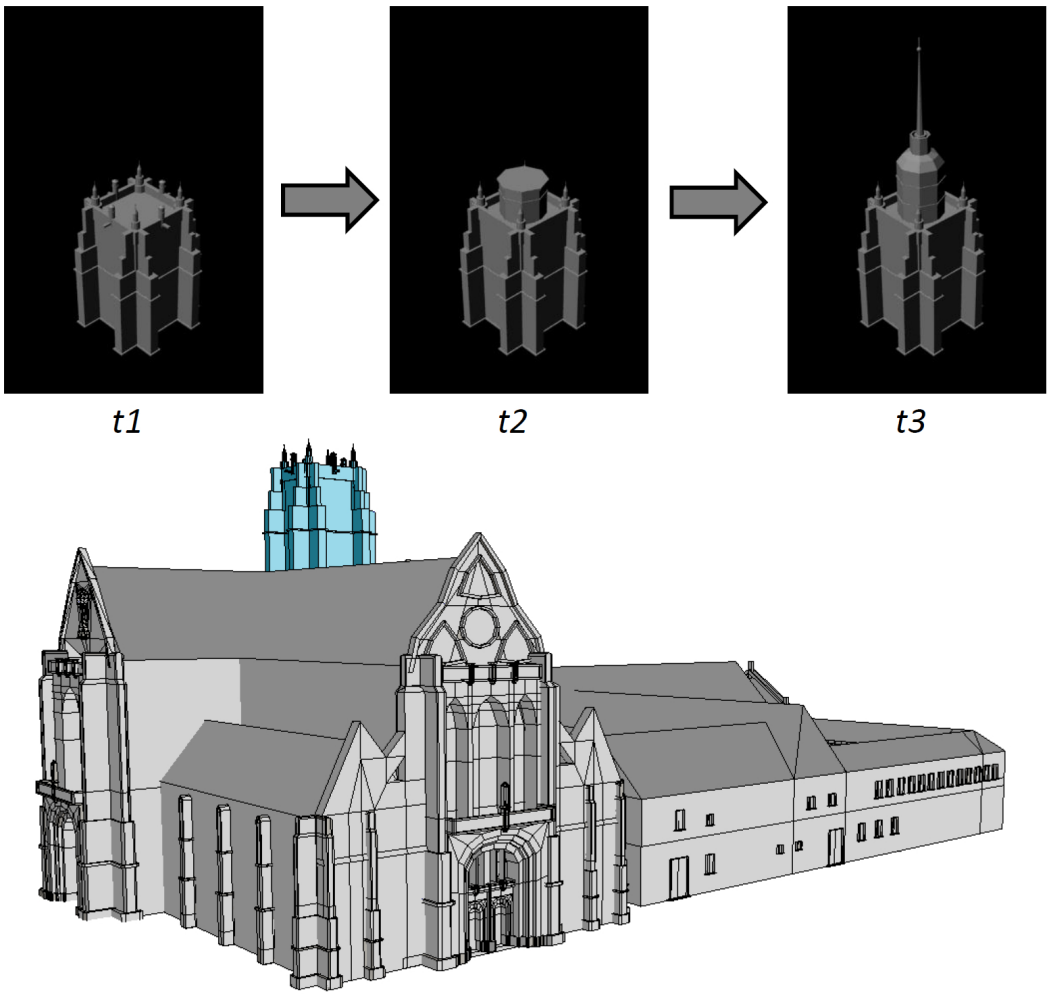



Figure 6.5 – The 3D temporal evolution of the monastery's bell tower is indicated using another color on that part of the geometry.

are already published on the Web. The verification of these documents, whether they correspond to the story that is intended to be told, is to be achieved by the application developer. As mentioned earlier (in Section 5.4.2), the Europeana portal holds cultural heritage data (texts, images, videos, audio files, *etc.*). These items are connected to some LOD cloud datasets. We intend to use the URLs of these documents and order them in a logical way (in order to create a narrative). To this end, we use SMIL (Synchronized Multimedia Integration Language), a W3C recommended XML-based language that provides timing control for multimedia presentations integrating audio and video with images, text or any other media type [11]. We attach these files to corresponding temporal representations. Therefore,

11. http://www.w3.org/AudioVideo/

we extend our data model (shown in Figure 4.12 in Chapter 4) as in Figure 6.6. The StoryFile class (blue box in the figure) is attached with the TemporalRepresentation class using arcama-owl:hasStory association. A temporal representation can have no stories or many of them interlinked to it.

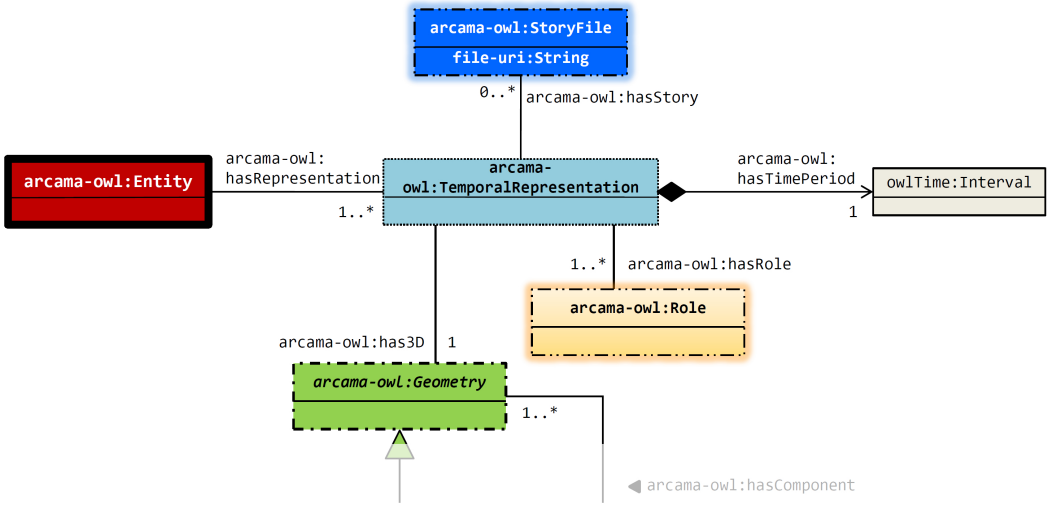



Figure 6.6 – The stories are attached to temporal representations of the data model.

With this approach, it is possible to attach a story to a 3D geometry that is valid within a time interval. In the example of the bell tower, which has three valid geometrical representations during different time periods, each of them can have a specific (SMIL) file bound to it, providing users with an access to a different story. These SMIL files can be uploaded to the server of the 3D warehouse while uploading the 3D objects (as explained in Section 5.3.1) or to another server which will be accessed by the application. However, in order to be able to query these stories (SMIL files) through the triple store, the triples binding this file with the temporal representations should be stored in the triple store. If not, it could then be accessible in another triple store on the Web.

One of the advantages of using multimedia contents is that the same content can be re-used to compose different stories. Similar multimedia files can be referred to many times in different XML files and used by different applications. Sequences of the multimedia content can be changed as desired by the application developer, by modifying the XML file.

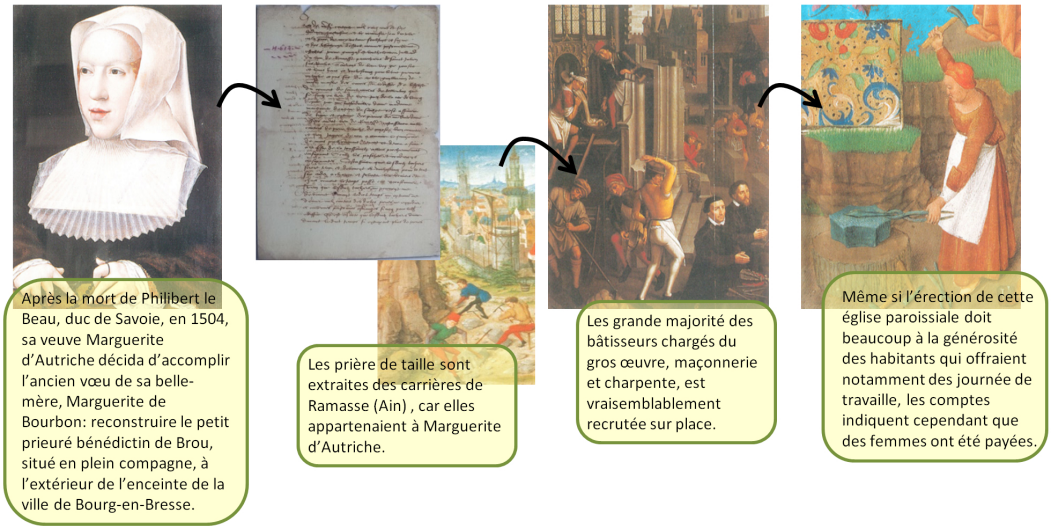



Figure 6.7 – The construction story of the monastery is told using visual transitions of photos and texts.

### SMIL: Synchronized Multimedia Integration Language

For the monastery, we intend to make available the story related to its construction. The construction of the monastery is an important event of its era and it took 26 years (1506-1532). There are many documents related to this event with texts and photos. However, not all the visitors stop and spend time to read them, while, if this story is told through a mobile augmented reality application, the users who visit the site can have access in short time to this historical events in a few clicks. The archived collections are therefore open for more people.

On the other hand, SMIL fits well for our purpose, since it defines markups for timing, media embedding and visual transitions of different media. Files from multiple web servers can be included into the markup.

For the story related to the construction of the monastery, we prepare a SMIL file which stores the URLs, timing related concepts and the sequence of multimedia files with the purpose to construct the narrative. The story that we would like to make accessible to users contains some texts and images and shown in Figure 6.7. As it can be seen in this figure, each step in the story consists of an image (or images) and a text related to that image. Thus, the SMIL document (*construction_brou.smil*) that we create for this purpose is as in the following:

```
1   <smil xmlns="http://www.w3.org/ns/SMIL" version="3.0"
2    baseProfile="UnifiedMobile">
3     <head>
```

```
4         <meta xml:id="meta-smil1.0-b" name="Date" content="2014-05-10" />
5            <meta xml:id="meta-smil1.0-c" name="Rights" content="Copyright
6            Royal Monastery of Brou" />
7        <textStyling>
8           <textStyle xml:id="HeadlineStyle" textFontFamily="serif"
9           textFontStyle="italic" textColor="black"
10           textBackgroundColor="white" />
11       </textStyling>
12       <layout>
13          <topLayout width="640px" height="360px"
14          open="whenActive" close="whenNotActive"/>
15          <region id="Image" left="0px" width="320px" height="100%"
16          fit="meet"/>
17          <region id="Text" right="0px" width="320px" height="100%"
18          fit="scroll" backgroundColor="yellow"/>
19       </layout>
20    </head>
21    <body>
22    <par>
23       <img src="http://steamer.imag.fr/arcama/multimedia/
24       Marguerite_d_Autriche_en_costume_de_veuve.jpeg" region="Image"
25       dur="14s"/>
26       <text src="http://steamer.imag.fr/arcama/multimedia/
27       Marguerite_la_veuve.txt" region="Text" dur="14s"/>
28    </par>
29    <par>
30       <seq>
31          <img src="http://steamer.imag.fr/arcama/multimedia/
32          le_prix_fait_l_extraction_de_la_pierre.jpeg" region="Image"
33          dur="4s"/>
34          <img src="http://steamer.imag.fr/arcama/multimedia/
35          la_pierre_transportee_sur_le_chantier.jpeg" region="Image"
36          dur="4s"/>
37       </seq>
38       <text src="http://steamer.imag.fr/arcama/multimedia/
39       l_extraction_de_la_pierre_de_Ramasse.txt"" region="Text"
40       dur="8s"/>
41    </par>
42    <par>
43       <img src="http://steamer.imag.fr/arcama/multimedia/
44       quatre_couronnés.jpeg" region="Image" dur="7s"/>
45       <text src="http://steamer.imag.fr/arcama/multimedia/
46       les_batisseurs.txt" region="Text" dur="7s"/>
47    </par>
48    <:par>
49       <img src="http://steamer.imag.fr/arcama/multimedia/
50       les_femmes_travaillaient.jpg" region="Image" dur="9s"/>
51       <text src="http://steamer.imag.fr/arcama/multimedia/
52       les_femmes_sur_le_chantier.txt" region="Text" dur="9s"/>
```

```
53    </par>
54    </body>
55  </smil>
```

In this SMIL document, <smil> element contains <head> and <body> attributes. <head> element (lines 3-20) holds meta information (*e.g.*, author/creator, creation date, a list of keywords, *etc.*) and layout information (elements and attributes that allow for the positioning of media elements on visual and audio rendering surfaces and controlling audio volume). For instance, in this example, *images* are scaled to "meet" (lines 15-16), left half of the screen, while *text* is allowed to "scroll" (lines 17-18) the other half of the screen [12]. The <body> element (lines 21-54) contains the timing semantics and linking of multimedia content. Also, the <par> container, short for "parallel" in this document, defines a simple timing group in which a photo and some text are displayed at the same time on the screen (*e.g.* lines 22-28). The <seq> container (*e.g.* lines 30-37) defines a sequence of elements in which elements play one after the other (two photos in this example). The dur attribute specifies the active duration of a multimedia element.

If an object has some stories attached to its temporal representations, the application proposes them to the user. With the interaction of the user, this SMIL file is downloaded on the mobile device and interpreted by the application in order to visualize the story related to that object. The SMIL file can be executed on mobile devices by SMIL players, such as RealPlayer [13].

## 6.4 Conclusion

In this chapter, we have proposed a way to embody the specific stories about one site into the AR applications using our architecture. These stories can be the 3D evolution of a building or a narrative that is constructed using multimedia documents regarding events, lives or characters in the history of that object. In order to achieve this, we have introduced a use case scenario based on a real-world monument: the Royal Monastery of Brou, located in France. This monastery holds significant historical or architectural stories (events, characters, architectural details, *etc.*) which are often ignored by the visitors as they are not easily accessible. Eventually, ARCAMA-3D architecture allows application developers to provide users with choices of stories specific to different domains. Walking in a city, a user might

12. These sizes are selected by taking into account an average smartphone screen size, but they can be adapted to different mobile device screens by the application.

13. http://real.com/

want to access stories related to historical monuments or at a distant location, a user might receive an alert while she is passing nearby a castle.

Considering the thematic, 3D and temporal information give possibilities to develop different applications based on these information. Our data model represents an object with these minimum three main information and regards them as orthogonal three separate dimensions where a point in this orthogonality gives a representation of this object with a particular 3D geometry and a thematic attribute with a specific time. The main contribution of representing an object with changes of spatial, temporal and thematic attributes is that some information or a multimedia document can be linked to a particular representation. For instance, an object at a specific time interval represented with a geometry and a thematic attribute can also be linked with photos, texts, video files, *etc.* Hence, the XML documents are linked to these representations of the object.

Our main concern in bringing a storytelling approach in our architecture is because these stories are in general not told to the visitors, and in other words, they are not discovered. Storytelling is one of the interesting aspects to keep the user informed about the place, as well as it makes information, which is hard to access, available to the user. Through the narratives created using multimedia documents, the application opens up ways for the user to interact with her environment and reveals the stories related to the surrounding objects.

# Conclusion

## Sommaire

Contributions . . . . . 185
Perspectives . . . . . 187

The context of this thesis is at the frontier of three domains: location-based information access, Augmented Reality (AR) and 3D mobile GIS. In this dissertation, we have introduced our proposition of a data model and an architecture that bring a different approach to the Auugmented Reality and enable developers to use LOD mechanisms in their application. The modular architecture has various components that makes the following features possible:

- reuse of the 3D models by several applications as they are stored in a warehouse and available in different levels-of-detail;
- access to and reuse of different representations of the real-world objects (as they are represented with spatial, temporal and thematic information using the data model);
- bind these representations with documents to build keep stories about these real-world objects.

We explain how the information retrieval is achieved using the architecture and how different components of the architecture communicate with each other to make this retrieval possible.

The key idea behind our work is to provide mobile users with information belonging to the surrounding objects in a systematic manner. For a person who visits a place, finding and providing *all* the information by using her geo-location as the unique criterion, may not be considered as satisfactory. Our point of view is that the context-aware mobile augmented reality platforms should also consider filtering the information to keep only the relevant data that fit the expectations or the interests of the user. To this end, the Semantic Web technologies, especially the LOD cloud, may be used to provide a meaningful exploration of the information available on the Web with general and/or specialized knowledge expressed in a formal and structured manner. The advantages of using LOD resources as a knowledge base for the AR applications is two fold:

- The AR developer can use the knowledge base as desired, in a systematic way, and generate a specific application which focuses (*i.e.* queries) only certain information on the cloud. For instance, while an application provides users with some types of buildings and architecture styles of these buildings, another may focus on types of buildings and their 3D structural evolution in time and/or some historical stories about these objects.
- The user can find further information by following the links between data. For instance, a user interested about historical objects may also demand information about their 3D evolution, the duration and period of construction, or the name of the architect, *etc.*

With this approach, the user is not overloaded with *all* the information available in the surroundings. Instead, by providing her with mechanisms to limit the data according to her interests, she may focus on her environment during the AR experience.

On the other hand, information published on the LOD cloud increases its usability since it is accessible by others. This is why our proposition is based on publishing 3D models on the LOD cloud by associating them with thematic and temporal information. These 3D geometries can be used by several other AR developers for various AR applications as desired. Therefore, publishing a dataset on the cloud helps not only to enrich it with the LOD knowledge sources, but also allows others to use this dataset as a knowledge base for their own purpose.

From an *interaction* point of view, our augmented reality approach consists in superimposing 3D light-weight objects as the initial AR view provided to the user. The idea behind this is to allow the user to observe her environment which is behind these transparent objects, instead of displaying all the information available on the screen, which may create a cognitive overload. Instead, we propose to display the information when she interacts with the corresponding 3D object by clicking on the mobile device's screen. Hence, according to the interests of the user (*i.e.* her preferences), the 3D AR view acts as an interaction layer in our approach. We propose using this layer to indicate the existence of information related to an object (or its parts) by highlighting them with different colors. From the developer's point of view, if we can access, for intance, the type of buildings (available in the triple store), then displaying the objects in a desired color is trivial. The information stored in the triple store (using data model) makes this kind of application design possible. For instance, different colors might be chosen to indicate certain type of information

(for instance structures with the same role can be displayed in same color). Using 3D models as an interaction layer also gives the possibility to interact with any of the surrounding objects by clicking on them on the screen. This interaction creates a query and using the object identifier the application can query the triple store and LOD cloud following the links and return existing and hopefully relevant information to the user.

## Contributions

1. **Architecture:** Our most specific contribution is the proposition of an architecture, ARCAMA-3D, which includes a generic data model that supports AR application development from various domains. By making the LOD cloud datasets, the main source of information, developers can structure (*i.e.* organize, group, *etc.*) the presented information and allow the user to select among these sources according to her interests. Therefore, we think that the architecture helps retrieving information in a systematic way, as the information sources are structured datasets. Another advantage of using LOD resources is that when new data are added to these datasets (such as when the page of a museum building is updated), such updated or new information will be queried and displayed to the user. Moreover, the developer can add additional structured data on the cloud, if it is not available.

2. **Interlinking with DBpedia entities:** An intermediary contribution is the mapping between a mediator ontology and DBpedia classes and categories in an effort to provide a mapping-based query with a larger coverage of DBpedia entities to AR applications. DBpedia is the dataset at the center of LOD cloud which guarantees some connectivity and easy entry points to find interesting resources over the LOD cloud. However, as it is an extraction of structured information from Wikipedia articles, it presents advantages as well as weaknesses. It may have some inconsistencies related to its categories or properties of classes since they are not part of a cohesive structure or a conceptual model, which may create significant redundancy. For instance, an entity may be associated with a related class, but it may not be associated with the related category (as we mention in Section 4.2.1 in detail). Thus, while it can be accessed using the DBpedia ontology, it may not be queried through DBpedia categories. On the other hand, we intend our architecture to be able to query *all* the DBpedia entities according to their architectural types (library, town

hall, museum, *etc.*), whether this information associated with related DBpedia classes or categories. In this regard, we increase the precision (*i.e.* correct thematic information) and provide coverage to all the DBpedia entities with *type* information.

3. **Interaction and access to information:** By simply pointing a mobile device towards a monument or a building around, the user can easily access and interact with the information through the AR application developed using our architecture. Our contribution with this regard is that we consider every surrounding object as an information source represented with a 3D geometry on the LOD cloud, such that they can be interlinked with extra information. If these objects are represented on the cloud by connecting them to a structured dataset (DBpedia), the user can select (*i.e.* the application can query) through the information links and discover further objects with similar information (real-world objects with same roles or more precise information such as all the objects constructed in 17th century, *etc.*). Therefore, by interacting with these 3D models, the user can access what is not seen in the real view, such as past or future constructions, hidden 3D elements, or images and photos related to the object.

4. **Storytelling:** Visiting a place, tourists may not have access to many stories that exist as documents or digital files (photos, videos, audio documents, *etc.*) nor they have time to discover them. For instance, when and by whom a library was built, who lived at that building, a significant event happened a century ago in that church, photos of old engravings depicting the area in the past, *etc.* Thus, most of the time, these stories are not discovered by the user. It is often not possible to access this kind of information without doing some research or talking with experts (historians, curators, a person who lived there, *etc.*). However, there are multimedia content including videos, audio files, images, podcasts, texts which can be used to tell these stories to users. Some of these documents are held in archives. Thus, one of our contributions is to help revealing these stories by making them available over the Web attaching to the related 3D objects (or parts). In this way, the media can tell the story itself and share the knowledge and experiences related to that site. Another point that should be emphasized is that since the 3D objects are represented with temporal and thematic attributes, these stories can also be attached to an object on the basis of temporal and thematic criteria.

## Perspectives

The contributions presented above bring up some perspectives for the continuation of our approach. The perspectives can be grouped into five topics and differ regarding to their time horizons as short-, medium- and long-term perspectives.

The short term perspectives are related to application design using our architecture, namely interaction and visualization:

- When developing an AR application, one of the most important elements is the correct superimposition of the digital object on the real-world view. In order to achieve this, we describe a method of manual correction of this superimposition in our proposition. A hybrid tracking method with visual-based tracking would provide a more accurate result, as sensor data may create a slightly oscillating 3D object view since it can be affected by other environmental factors (such as magnetic objects around or tall buildings which may block signals). However, perfecting the superimposition using visual tracking methods is not one of the main focuses of this thesis, we left it as a future improvement that would be realized for a better implementation of AR.
- For the visualization of information according to the preferences of the user, we have proposed that 3D objects with interesting information are displayed in different colors to indicate the existence and type of information to the user. In order to have a more user-friendly visualization, this would be elaborated by displaying a view of the surroundings from a 3D elevated view (*i.e.* bird's-eye-view/aerial view) to improve the perception of the user. This would be achieved by a common visualization technique where the user holds the mobile device parallel to the ground to observe an elevated view and vertical in order to switch to the AR view. This might be also helpful in finding the next interesting point in the surroundings.

The medium-term perspective is related to data sources and retrieval:

- As a structured information source on the LOD cloud, we use DBpedia and interlink our 3D models with it. Then, by querying these triples according to type, temporal interval, *etc.*, we can access to the 3D models and thus their corresponding DBpedia resources by following these links. We have, thus, more data (on DBpedia) related to these real-world objects. However, multiple data sources can also be interlinked to these objects, queried and presented to the user. An interesting future work would be using several data sources on the LOD cloud, improving the data reasoning process and enrich the acquired

information.

The long term perspectives have broader aspects such as predicting user preferences from previous interactions, user guidance and implementing our approach in new platforms:

- Our approach regarding user preferences is based on proposing the user a list with, for instance, object types, time intervals or stories related to the surroundings. This can be improved by automatizing this process. This subject may necessitate a deeper study, such as involving the user's social media profile where her favorites, her reading lists, her or her friends' interests would be gathered. This way AR application would update what she might be interested in or looking for at a location when running the discovery application. This list of acquired interests may also be modifiable by the user at any time. Another approach might be achieved with data mining. The choices of users with similar selections can be analyzed in order to extract user profiles and provide personalized recommendations to future users with similar interests.
- During the AR experience, the user may also be guided to the next closest object according to her preferences. This can be achieved visually (with arrows) or vocally (with audio comments) to tell her how to get to the next location. The user can also be guided to walk around a specific object in order to make her look at certain parts of that structure for the purpose of telling a story that necessitates her to interact with those parts. In this regard, the extraction of pathways according to the interesting 3D objects will be the challenging aspect of this future study.
- Mobile technologies have become essential parts of our lives today, as the number of people using a mobile device is rapidly growing. In the mean time, wearable technologies, such as smart glasses (*e.g.* Google Glass, Microsoft HoloLens, *etc.*), provide hands-free computing based on voice controls, using a touchpad or through some specific applications. These devices represent areas of innovation that mobile application development is exploring. As a future perspective, we would like to remark that our proposition with its visualization and interaction approach using 3D augmented reality objects may very well used with wearable technologies. As the user will be hands-free, the 3D objects will seamlessly integrate onto the captured view of the surroundings and she can interact with these objects with different interaction mechanisms.

# Bibliography


[Abowd *et al.* 1997] Gregory D. Abowd, Christopher G. Atkeson, Jason Hong, Sue Long, Rob Kooper et Mike Pinkerton. *Cyberguide: A Mobile Context-aware Tour Guide*. Wirel. Netw., vol. 3, no. 5, pages 421–433, Octobre 1997. (Cited at page 32.)

[ARToolkit 2013a] ARToolkit. *ARToolKit Home Page*. Date of Access: 13 November 2014, Available from http://www.hitl.washington.edu/artoolkit/, 2013. (Cited at page 38.)

[ARToolkit 2013b] ARToolkit. *Creating and training new ARToolKit markers*. Date of Access: 13 November 2014, Available from http://www.artoolworks.com/support/library/Creating_and_training_new_ARToolKit_markers, 2013. (Cited at page 39.)

[Ashikaga *et al.* 2011] Erika Ashikaga, Mayu Iwata, Daijiro Komaki, Takahiro Hara et Shojiro Nishio. *Exploring Map-based Interactions for Co-located Collaborative Work by Multiple Mobile Users*. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '11, pages 417–420, New York, NY, USA, 2011. ACM. (Cited at page 24.)

[Auer & Lehmann 2010] Sören Auer et Jens Lehmann. *Creating knowledge out of interlinked data*. Semantic Web, vol. 1, no. 1-2, pages 97–104, 2010. (Cited at page 99.)

[Auer *et al.* 2007] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak et Zachary G. Ives. *DBpedia: A Nucleus for a Web of Open Data*. In ISWC/ASWC, pages 722–735, 2007. (Cited at pages 62 et 68.)

[Aydin *et al.* 2012a] Betül Aydin, Jérôme Gensel, Sylvie Calabretto et Bruno Tellez. *ARCAMA-3D - A Context-Aware Augmented Reality Mobile Platform for Environmental Discovery*. In W2GIS, pages 17–26, 2012. (Cited at page 96.)

[Aydin *et al.* 2012b] Betül Aydin, Jérôme Gensel, Sylvie Calabretto et Bruno Tellez. *ARCAMA-3D: Une application mêlant réalité augmentée et 3D pour la présentation d'informations contextuelles*. In 8èmes journées francophones

Mobilité et Ubiquité (UBIMOB), Anglet, France, pages 34–41, 2012. (Cited at page 96.)

[Aydin *et al.* 2013a] Betül Aydin, Jérôme Gensel, Philippe Genoud, Sylvie Calabretto et Bruno Tellez. *An architecture for surroundings discovery by linking 3D models and LOD cloud.* In MobiGIS, pages 9–16, 2013. (Cited at page 97.)

[Aydin *et al.* 2013b] Betül Aydin, Jérôme Gensel, Philippe Genoud, Sylvie Calabretto et Bruno Tellez. *Extending Augmented Reality Mobile Application with Structured Knowledge from the LOD Cloud.* In IMMoA, pages 21–27, 2013. (Cited at page 97.)

[Aydin 2010] Betül Aydin. *Hybrid Tracking using Dempster-Shafer Data Fusion in Real-Time Augmented Reality Applications for Man-Made Environments*, 2010. (Cited at page 39.)

[Azuma *et al.* 1999] Ronald T. Azuma, Bruce R. Hoff, Howard E. Neely III, Ronald Sarfaty, Michael J. Daily, Gary Bishop, Leandra Vicci, Greg Welch, Ulrich Neumann, Suya You, Rich Nichols et Jim Cannon. *Making Augmented Reality Work Outdoors Requires Hybrid Tracking.* In Proceedings of the International Workshop on Augmented Reality : Placing Artificial Objects in Real Scenes: Placing Artificial Objects in Real Scenes, IWAR '98, pages 219–224, Natick, MA, USA, 1999. A. K. Peters, Ltd. (Cited at page 40.)

[Baldauf *et al.* 2007] Matthias Baldauf, Schahram Dustdar et Florian Rosenberg. *A Survey on Context Aware Systems.* Int. J. Ad Hoc Ubiquitous Comput., vol. 2, no. 4, pages 263–277, Juin 2007. (Cited at page 12.)

[Ballatore *et al.* 2010] Andrea Ballatore, Gavin McArdle, Caitriona Kelly et Michela Bertolotto. *RecoMap: An Interactive and Adaptive Map-based Recommender.* In Proceedings of the 2010 ACM Symposium on Applied Computing, SAC '10, pages 887–891, New York, NY, USA, 2010. ACM. (Cited at pages 23 et 24.)

[Baltrunas *et al.* 2011] Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Aykan Aydin, Karl-Heinz Lüke et Roland Schwaiger. *InCarMusic: Context-Aware Music Recommendations in a Car.* In Christian Huemer et Thomas Setzer, editeurs, E-Commerce and Web Technologies - 12th International Conference, EC-Web 2011, Toulouse, France, August 30 - September 1, 2011. Proceedings, Lecture Notes in Business Information Processing, pages 89–100, 2011. (Cited at page 16.)

[Baltrunas *et al.* 2012] Linas Baltrunas, Bernd Ludwig, Stefan Peer et Francesco Ricci. *Context relevance assessment and exploitation in mobile recommender systems.* Personal and Ubiquitous Computing, vol. 16, no. 5, pages 507–526, 2012. (Cited at pages ix, 20 et 21.)

[Battle & Kolas 2012] Robert Battle et Dave Kolas. *Enabling the geospatial Semantic Web with Parliament and GeoSPARQL.* Semantic Web, vol. 3, no. 4, pages 355–370, 2012. (Cited at pages 73 et 74.)

[Baus *et al.* 2002] Jörg Baus, Antonio Krüger et Wolfgang Wahlster. *A Resource-adaptive Mobile Navigation System.* In Proceedings of the 7th International Conference on Intelligent User Interfaces, IUI '02, pages 15–22, New York, NY, USA, 2002. ACM. (Cited at page 23.)

[Bechhofer *et al.* 2004] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider et Lynn Andrea Stein. *OWL Web Ontology Language Reference.* Date of Access: 09 August 2014, Available from `http://www.w3.org/TR/owl-ref/#Sublanguage-def`, 2004. (Cited at page 116.)

[Becker & Bizer 2008] Christian Becker et Chrisitan Bizer. *DBpedia Mobile: A Location-Aware Semantic Web Client.* In Proceedings of the Semantic Web Challenge at ISWC, 2008. (Cited at pages ix, 25 et 83.)

[Becker & Bizer 2009] Christian Becker et Christian Bizer. *Exploring the Geospatial Semantic Web with DBpedia Mobile.* Journal of Web Semantics, vol. 7, no. 4, pages 278–286, 2009. (Cited at page 83.)

[Behringer *et al.* 2002] Reinhold Behringer, Jun Park et Venkataraman Sundareswaran. *Model-Based Visual Tracking for Outdoor Augmented Reality Applications.* In Proceedings of the 1st International Symposium on Mixed and Augmented Reality, ISMAR '02, pages 277–, Washington, DC, USA, 2002. IEEE Computer Society. (Cited at page 39.)

[Belz *et al.* 2003] B. Schmidt Belz, H. Laamanen, S. Poslad et A. Zipf. *Location-based Mobile Tourist Services - First User Experiences.* In Proceedings of the International Conference on Information and Communication Technologies in Tourism. Springer-Verlag Wien, 2003. (Cited at pages 23 et 24.)

[Berners-Lee *et al.* 2001] Tim Berners-Lee, James Hendler et Ora Lassila. *The Semantic Web.* Scientific American, May 2001. (Cited at pages 47, 55 et 56.)

[Berners-Lee 1989] Tim Berners-Lee. *Information Management: A Proposal.* Date of Access: 14 October 2013, Available from http://www.w3.org/History/1989/proposal.html, 1989. (Cited at page 45.)

[Berners-Lee 1998] Tim Berners-Lee. *The World Wide Web: A very short personal history.* Date of Access: 24 October 2013, Available from http://www.w3.org/People/Berners-Lee/ShortHistory, 1998. (Cited at page 45.)

[Berners-Lee 2000] Tim Berners-Lee. *Semantic Web - XML2000.* Date of Access: 07 August 2014, Available from http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html, 2000. (Cited at pages ix et 48.)

[Bizer *et al.* 2007] Christian Bizer, Richard Cyganiak et Tom Heath. *How to publish linked data on the web.* Date of Access: 27 November 2013, Available from http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/, 2007. (Cited at page 62.)

[Bizer *et al.* 2009] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak et Sebastian Hellmann. *DBpedia - A Crystallization Point for the Web of Data.* Web Semantantics, vol. 7, no. 3, pages 154–165, Septembre 2009. (Cited at page 110.)

[Bizer 2012] Christian Bizer. *Expert Report On Data Linking and Publications.* Global Research Data Infrastructure of European Union - GRDI2020, October 2012. (Cited at pages 63 et 65.)

[Blanco *et al.* 2011] Roi Blanco, Peter Mika et Sebastiano Vigna. *Effective and Efficient Entity Search in RDF Data.* In International Semantic Web Conference (1), pages 83–97, 2011. (Cited at page 77.)

[Bogdahn & Coors 2009] Jurgen Bogdahn et Volker Coors. *Using 3D Urban Models for Pedestrian Navigation Support.* In Proceedings of ISPRS Joint Workshop Cityscapes at GeoWeb, 2009. (Cited at pages 4 et 31.)

[Breitman *et al.* 2007] Karin K. Breitman, Marco Antonio Casanova et Walter Truszkowski. Semantic web: Concepts, technologies and applications. NASA Monographs in Systems and Software Engineering. Springer, London, 2007. (Cited at pages 56 et 59.)

[Brin 1998] Sergey Brin. *Extracting Patterns and Relations from the World Wide Web.* In WebDB, pages 172–183, 1998. (Cited at page 44.)

[Brush *et al.* 2010] A.J. Bernheim Brush, John Krumm et James Scott. *Exploring End User Preferences for Location Obfuscation, Location-based Services, and*

*the Value of Location.* In Proceedings of the 12th ACM International Conference on Ubiquitous Computing, Ubicomp '10, pages 95–104, New York, NY, USA, 2010. ACM. (Cited at page 12.)

[Buján *et al.* 2013] David Buján, David Martín, Ortzi Torices, Diego López-de Ipiña, Carlos Lamsfus, Joseba Abaitua et Aurkene Alzua-Sorzabal. *Context Management Platform for Tourism Applications.* Sensors, vol. 13, no. 7, pages 8060–8078, Juin 2013. (Cited at page 13.)

[Burigat & Chittaro 2005] Stefano Burigat et Luca Chittaro. *Location-aware Visualization of VRML Models in GPS-based Mobile Guides.* In Proceedings of the Tenth International Conference on 3D Web Technology, Web3D '05, pages 57–64, New York, NY, USA, 2005. ACM. (Cited at pages 27 et 34.)

[Caarls *et al.* 2003] Jurjen Caarls, Pieter P. Jonker et Stelian Persa. *Sensor Fusion for Augmented Reality.* In Emile H. L. Aarts, René Collier, Evert van Loenen et Boris E. R. de Ruyter, editeurs, EUSAI, volume 2875 of *Lecture Notes in Computer Science*, pages 160–176. Springer, 2003. (Cited at page 40.)

[Celino *et al.* 2012] Irene Celino, Simone Contessa, Marta Corubolo, Daniele Dell'Aglio, Emanuele Della Valle, Stefano Fumeo et Thorsten Krüger. *UrbanMatch - linking and improving Smart Cities Data.* In WWW2012 Workshop on Linked Data on the Web (LDOW), Lyon, France, 16 April, 2012. (Cited at pages 13, 84 et 85.)

[Chen & Kotz 2000] Guanling Chen et David Kotz. *A Survey of Context-Aware Mobile Computing Research.* Rapport technique, Department of Computer Science, Dartmouth College, Hanover, NH, USA, 2000. (Cited at page 14.)

[Chen *et al.* 2009] Yunhao Chen, Wei Su, Jing Li et Zhongping Sun. *Hierarchical object oriented classification using very high resolution imagery and LIDAR data over urban areas.* In Advances in Space Research, volume v43 i7, pages 1101–1110, 2009. (Cited at page 28.)

[Cheverst *et al.* 1999] Keith Cheverst, Keith Mitchell et Nigel Davies. *Design of an object model for a context sensitive tourist GUIDE.* In Computers and Graphics, pages 883–891, 1999. (Cited at page 16.)

[Cheverst *et al.* 2000] Keith Cheverst, Nigel Davies, Keith Mitchell, Adrian Friday et Christos Efstratiou. *Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'00, pages 17–24, New York, NY, USA, 2000. ACM. (Cited at pages 18 et 19.)

[Chou *et al.* 2005] Shih-Chun Chou, Wen-Tai Hsieh, Fabien L. Gandon et Norman M. Sadeh. *Semantic Web Technologies for Context-Aware Museum Tour Guide Applications.* In AINA, pages 709–714. IEEE Computer Society, 2005. (Cited at page 16.)

[Christoph *et al.* 2010] Uta Christoph, Karl-Heinz Krempels, Janno von Stülpnagel et Christoph Terwelp. *Automatic Context Detection of a Mobile user.* In WINSYS 2010 - Proceedings of the International Conference on Wireless Information Networks and Systems, Athens, Greece, July 26 - 28, 2010, WINSYS is part of ICETE - The International Joint Conference on e-Business and Telecommunications, pages 189–194, 2010. (Cited at page 18.)

[Church & Smyth 2009] Karen Church et Barry Smyth. *Understanding the Intent Behind Mobile Information Needs.* In Proceedings of the 14th International Conference on Intelligent User Interfaces, IUI '09, pages 247–256, New York, NY, USA, 2009. ACM. (Cited at page 18.)

[Clear *et al.* 2010] Adrian K. Clear, Thomas Holland, Simon Dobson, Aaron J. Quigley, Ross Shannon et Paddy Nixon. *Situvis: A sensor data analysis and abstraction tool for pervasive computing systems.* Pervasive and Mobile Computing, vol. 6, no. 5, pages 575–589, 2010. (Cited at page 16.)

[Coors & Wiedmann 1998] V. Coors et B. Wiedmann. *Using Wearable GIS in outdoor applications.* In Proceedings of the Symposium on Interactive Applications for Mobile Computing - IMC'98, 1998. (Cited at pages 33 et 34.)

[Coors & Zipf 2007] Volker Coors et Er Zipf. *MoNa 3D – Mobile Navigation using 3D City Models.* In 4th International Symposium on LBS and Telecartography, 2007. (Cited at page 32.)

[Costabello *et al.* 2012] Luca Costabello, Serena Villata, Nicolas Delaforge et Fabien Gandon. *Linked Data Access Goes Mobile: Context-Aware Authorization for Graph Stores.* In WWW2012 Workshop on Linked Data on the Web, 2012. (Cited at page 13.)

[Costabello 2011] Luca Costabello. *DC Proposal: PRISSMA, Towards Mobile Adaptive Presentation of the Web of Data.* In Proceedings of the 10th International Conference on The Semantic Web - Volume Part II, ISWC'11, pages 269–276, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited at pages x et 84.)

[Damova *et al.* 2010] Mariana Damova, Atanas Kiryakov, Kiril Ivanov Simov et Svetoslav Petrov. *Mapping the central LOD ontologies to PROTON upper-*

*level ontology.* In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao et Isabel F. Cruz, editeurs, Proceedings of the 5th International Workshop on Ontology Matching (OM'10), volume 689 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010. (Cited at page 114.)

[de Bruijn *et al.* 2006] Jos de Bruijn, Marc Ehrig, Cristina Feier, Francisco Martíns-Recuerda, François Scharffe et Moritz Weiten. Ontology mediation, merging, and aligning. Juillet 2006. (Cited at page 114.)

[Dey & Abowd 1999] Anind K. Dey et Gregory D. Abowd. *Towards a better understanding of context and context-awareness.* In In HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pages 304–307. Springer-Verlag, 1999. (Cited at page 15.)

[Dey 2001] Anind K. Dey. *Understanding and Using Context.* Personal Ubiquitous Comput., vol. 5, no. 1, pages 4–7, Janvier 2001. (Cited at page 14.)

[Díez-Díaz *et al.* 2007] Fidel Díez-Díaz, Martín González-Rodríguez et Agueda Vidau. *An Accesible and Collaborative Tourist Guide Based on Augmented Reality and Mobile Devices.* In Proceedings of the 4th International Conference on Universal Access in Human-computer Interaction: Ambient Interaction, UAHCI'07, pages 353–362. Springer-Verlag, 2007. (Cited at page 35.)

[Ding *et al.* 2010] Li Ding, Dominic DiFranzo, Alvaro Graves, James Michaelis, Xian Li, Deborah L. McGuinness et Jim Hendler. *Data-gov Wiki: Towards Linking Government Data.* In AAAI Spring Symposium: Linked Data Meets Artificial Intelligence, 2010. (Cited at page 74.)

[DiNucci 1999] Darcy DiNucci. *Fragmented future.* Print, vol. 53, no. 4, page 32, 1999. (Cited at page 46.)

[Division 2000] United Nations Statistical Division. Series Studies in methods. Series F; no.79. United Nations Publication, New York, NY, USA, 2000. (Cited at page 22.)

[Döllner *et al.* 2006] Jürgen Döllner, Konstantin Baumann et Henrik Buchholz. *Virtual 3D City Models as Foundation of Complex Urban Information Spaces.* In Manfred Schrenk, editeur, 11th international conference on Urban Planning and Spatial Development in the Information Society (REAL CORP), pages 107–112. CORP – Competence Center of Urban and Regional Planning, 2006. (Cited at page 31.)

[Domingue *et al.* 2011] John Domingue, Dieter Fensel et James A. Hendler, editeurs. Handbook of semantic web technologies. Springer, Berlin, 2011. (Cited at pages 50, 54, 56 et 60.)

[DunedinProject 2011] DunedinProject. *DiscovAR Dunedin: Dunedin Augmented Reality for Heritage and Arts.* Date of Access: 22 October 2014, Available from `http://www.discovar.in/`, 2011. (Cited at page 35.)

[Dunham & Kumar 1998] Margaret H. Dunham et Vijay Kumar. *Location Dependent Data and Its Management in Mobile Databases.* In Proceedings of the 9th International Workshop on Database and Expert Systems Applications, DEXA '98, pages 414–, Washington, DC, USA, 1998. IEEE Computer Society. (Cited at page 12.)

[Duží *et al.* 2011] Marie Duží, Martina Číhalová et Marek Menšík. *Ontology as a Logic of Intensions.* In Proceedings of the 2011 conference on Information Modelling and Knowledge Bases XXII, pages 1–20, Amsterdam, The Netherlands, 2011. IOS Press. (Cited at page 50.)

[Espeter & Raubal 2009] Martin Espeter et Martin Raubal. *Location-based decision support for user groups.* J. Location Based Services, vol. 3, no. 3, pages 165–187, 2009. (Cited at page 16.)

[Farman 2012] Jason Farman. *Mapping and Representations of Space.* In Routledge, editeur, The Mobile Interface of Everyday Life: Embodied Space and Locative Media, pages 35–55, 2012. (Cited at page 5.)

[Feiner *et al.* 1993] Steven Feiner, Blair MacIntyre et Dorée D. Seligmann. *Knowledge-Based Augmented Reality.* Commun. ACM, vol. 36, no. 7, pages 53–62, 1993. (Cited at page 34.)

[Feiner *et al.* 1997] Steven Feiner, Blair MacIntyre, Tobias Hollerer et Anthony Webster. *A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment.* In Proceedings of the 1st IEEE International Symposium on Wearable Computers, ISWC '97, pages 74–81, Washington, DC, USA, 1997. IEEE Computer Society. (Cited at pages 32, 33 et 34.)

[Fensel 2003] Dieter Fensel. Ontologies: A silver bullet for knowledge management and electronic commerce. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 édition, 2003. (Cited at page 49.)

[Franz *et al.* 2009] Thomas Franz, Antje Schultz, Sergej Sizov et Steffen Staab. *TripleRank: Ranking Semantic Web Data by Tensor Decomposition.* In International Semantic Web Conference, pages 213–228, 2009. (Cited at pages 77 et 78.)

[Fröhlich *et al.* 2011] Peter Fröhlich, Antti Oulasvirta, Matthias Baldauf et Antti Nurminen. *On the Move, Wirelessly Connected to the World.* Commun. ACM, vol. 54, no. 1, pages 132–138, Janvier 2011. (Cited at pages 4, 27 et 30.)

[Früh & Zakhor 2003] Christian Früh et Avideh Zakhor. *Constructing 3D City Models by Merging Aerial and Ground Views.* IEEE Comput. Graph. Appl., vol. 23, no. 6, pages 52–61, Novembre 2003. (Cited at page 27.)

[Gadolou *et al.* 2010] Eleni Gadolou, Haroula Papadaki, Emmanuel Stefanakis, Georgios Kritikos, Yiwei Cao, Anna Hannemann, Ralf Klamma et Dejan Kovachev. *Storytelling, Spatial Standards and Cultural Heritage Management.* In the Proceedings of the 13th AGILE Conference, Guimaraes, Portugal, May 2010. (Cited at page 82.)

[Gamallo *et al.* 2001] Pablo Gamallo, Caroline Gasperin, Alexandre Agustini et José Gabriel Pereira Lopes. *Syntactic-Based Methods for Measuring Word Similarity.* In TSD, pages 116–125, 2001. (Cited at page 118.)

[Gamma *et al.* 1995] Erich Gamma, Richard Helm, Ralph Johnson et John Vlissides. Design patterns: elements of reusable object-oriented software. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995. (Cited at page 131.)

[Gandon *et al.* 2012] F. Gandon, O. Corby et C. Faron-Zucker. Le web sémantique: Comment lier les données et les schémas sur le web ? Management des systèmes d'information. Dunod, 2012. (Cited at pages ix, 60, 61 et 62.)

[García-Silva *et al.* 2011] Andrés García-Silva, Max Jakob, Pablo N. Mendes et Christian Bizer. *Multipedia: enriching DBpedia with multimedia information.* In K-CAP, pages 137–144, 2011. (Cited at page 78.)

[Genc *et al.* 2002] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar et N. Navab. *Marker-less Tracking for AR: A Learning-Based Approach.* In Proceedings of the 1st International Symposium on Mixed and Augmented Reality, ISMAR '02, pages 295–, Washington, DC, USA, 2002. IEEE Computer Society. (Cited at page 38.)

[Goetz & Zipf. 2012] Marcus Goetz et Alexander Zipf. *OpenStreetMap in 3D – Detailed Insights on the Current Situation in German.* In AGILE 2012, 2012. (Cited at pages ix, 28 et 29.)

[Gotow *et al.* 2010] J.Benjamin Gotow, Krzysztof Zienkiewicz, Jules White et DouglasC. Schmidt. *Addressing Challenges with Augmented Reality Applications on Smartphones.* In Ying Cai, Thomas Magedanz, Minglu Li, Jinchun Xia et Carlo Giannelli, editeurs, Mobile Wireless Middleware, Operating Systems, and Applications, volume 48 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 129–143. Springer Berlin Heidelberg, 2010. (Cited at page 39.)

[Grimm *et al.* 2007] Stephan Grimm, Pascal Hitzler et Andreas Abecker. *Knowledge Representation and Ontologies.* In Semantic Web Services: Concepts, Technology and Applications, pages 51–106, 2007. (Cited at pages ix, 51 et 53.)

[Gruber 1993] Thomas R. Gruber. *A translation approach to portable ontology specifications.* Knowl. Acquis., vol. 5, no. 2, pages 199–220, June 1993. (Cited at page 50.)

[Grussenmeyer *et al.* 2008] P. Grussenmeyer, T. Landes, T. Voegtle et K. Ringle. *Comparison methods of terrestrial laser scanning, photogrammetry and tacheometry data for recording of cultural heritage buildings.* In ISPRS Congress Proceedings, pages 213–218, 2008. (Cited at page 28.)

[Guarino 1997] Nicola Guarino. *Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration.* In SCIE, pages 139–170, 1997. (Cited at pages 51 et 52.)

[Gurstein 2011] Michael Gurstein. *Open data: Empowering the empowered or effective data use for everyone ?* First Monday, vol. 16, no. 2, 2011. (Cited at page 171.)

[Haav *et al.* 2009] Hele-Mai Haav, Aivi Kaljuvee, Martin Luts et Toivo Vajakas. *Ontology-Based Retrieval of Spatially Related Objects for Location Based Services.* In On the Move to Meaningful Internet Systems: OTM 2009, Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009, Vilamoura, Portugal, November 1-6, 2009, Proceedings, Part II, pages 1010–1024, 2009. (Cited at page 13.)

[Haslhofer & Isaac 2011] Bernhard Haslhofer et Antoine Isaac. *data.europeana.eu - The Europeana Linked Open Data Pilot.* In International Conference on

Dublin Core and Metadata Applications, DC-2011, The Hague, 2011. (Cited at page 158.)

[Heath & Bizer 2011] Tom Heath et Christian Bizer. Linked data: Evolving the web into a global data space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011. (Cited at pages 62, 63, 65 et 75.)

[Heim *et al.* 2010] Philipp Heim, Steffen Lohmann et Timo Stegemann. *Interactive Relationship Discovery via the Semantic Web.* In ESWC (1), pages 303–317, 2010. (Cited at page 77.)

[Henrysson *et al.* 2005] Anders Henrysson, Mark Billinghurst et Mark Ollila. *Face to Face Collaborative AR on Mobile Phones.* In Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '05, pages 80–89, Washington, DC, USA, 2005. IEEE Computer Society. (Cited at page 38.)

[Höfle 2012] B. Höfle. *Towards Digital Earth - 3D Spatial Data Infrastructures.* Rapport technique No. 13022012, Scientific Report of Hengstberger Symposium, Septembre 2012. (Cited at page 28.)

[Hile *et al.* 2010] Harlan Hile, Alan Liu, Gaetano Borriello, Radek Grzeszczuk, Ramakrishna Vedantham et Jana Kosecka. *Visual Navigation for Mobile Devices.* IEEE MultiMedia, vol. 17, pages 16–25, 2010. (Cited at page 34.)

[Höllerer *et al.* 1999] Tobias Höllerer, John V. Pavlik et Steven Feiner. *Situated Documentaries: Embedding Multimedia Presentations in the Real World.* In ISWC, pages 79–86. IEEE Computer Society, 1999. (Cited at pages 35 et 36.)

[Hol *et al.* ] Jeroen D. Hol, Thomas B. Schön, Fredrik Gustafsson et Per J. Slycke. *Sensor Fusion for Augmented Reality.* In Fusion, pages 1–6. IEEE. (Cited at page 40.)

[Hugues *et al.* 2011] Olivier Hugues, Jean-Marc Cieutat et Pascal Guitton. *GIS and Augmented Reality: State of the Art and Issues.* In Borko Furht, editeur, Handbook of Augmented Reality, pages 721–740. Springer, 2011. (Cited at pages 4 et 33.)

[Hull *et al.* 1997] Richard Hull, Philip Neaves et James Bedford-Roberts. *Towards Situated Computing.* In Proceedings of the 1st IEEE International Symposium on Wearable Computers, ISWC '97, pages 146–, Washington, DC, USA, 1997. IEEE Computer Society. (Cited at page 15.)

[Hyvönen *et al.* 2009a] Eero Hyvönen, Eetu Mäkelä, Tomi Kauppinen, Olli Alm, Jussi Kurki, Tuukka Ruotsalo, Katri Seppälä, Joeli Takala, Kimmo Puputti, Heini Kuittinen, Kim Viljanen, Jouni Tuominen, Tuomas Palonen, Matias Frosterus, Reetta Sinkkilä, Panu Paakkarinen, Joonas Laitio et Katariina Nyberg. *CultureSampo – Finnish Culture on the Semantic Web 2.0. Thematic Perspectives for the End-user.* In Proceedings, Museums and the Web 2009, Indianapolis, USA, April 15-18 2009. (Cited at pages ix et 82.)

[Hyvönen *et al.* 2009b] Eero Hyvönen, Eetu Mäkelä, Tomi Kauppinen, Olli Alm, Jussi Kurki, Tuukka Ruotsalo, Katri Seppälä, Joeli Takala, Kimmo Puputti, Heini Kuittinen, Kim Viljanen, Jouni Tuominen, Tuomas Palonen, Matias Frosterus, Reetta Sinkkilä, Panu Paakkarinen, Joonas Laitio et Katariina Nyberg. *CultureSampo: A National Publication System of Cultural Heritage on the Semantic Web 2.0.* In ESWC, pages 851–856, 2009. (Cited at page 81.)

[Ilarri *et al.* 2006] S. Ilarri, E. Mena et A. Illarramendi. *Location-Dependent Queries in Mobile Contexts: Distributed Processing Using Mobile Agents.* IEEE Transactions on Mobile Computing (TMC), vol. 5, no. 8, pages 1029–1043, August 2006. (Cited at page 20.)

[Ilarri *et al.* 2010] Sergio Ilarri, Eduardo Mena et Arantza Illarramendi. *Location-dependent Query Processing: Where We Are and Where We Are Heading.* ACM Comput. Surv., vol. 42, no. 3, pages 12:1–12:73, Mars 2010. (Cited at pages 12 et 19.)

[Ilarri *et al.* 2011] Sergio Ilarri, Arantza lllarramendi, Eduardo Mena et Amit Sheth. *Semantics in Location-Based Services – Guest Editors' Introduction for Special Issue.* IEEE Internet Computing, vol. 15, no. 6, pages 10–14, November/December 2011. (Cited at page 12.)

[Isaac & Summers 2009] Antoine Isaac et Ed Summers. *SKOS Simple Knowledge Organization System Primer.* Date of Access: 05 June 2014, Available from `http://www.w3.org/TR/skos-primer/#secskosowl`, 2009. (Cited at page 116.)

[Iwata *et al.* 2011] Mayu Iwata, Takahiro Hara, Kentaro Shimatani, Tomohiro Mashita, Kiyoshi Kiyokawa, Shojiro Nishio et Haruo Takemura. *A Location-based Content Search System Considering Situations of Mobile Users.* In Proceedings of the 2nd International Conference on Ambient Systems, Networks and Technologies (ANT 2011), the 8th International Conference on Mobile Web Information Systems (MobiWIS-2011), Niagara Falls, Ontario,

Canada, September 19-21, 2011, pages 426–433, 2011. (Cited at pages 17, 20 et 34.)

[Jazayeri *et al.* 2013] Ida Jazayeri, Abbas Rajabifard et Mohsen Kalantari. *3D Data Sourcing for Land and Property Information: A Geometric and Semantic Perspective.* In Proceedings of Global Geospatial Conference 2013, 2013. (Cited at page 28.)

[Joshua *et al.* 1998] David Franklin Joshua, David Franklin et Joshua Flachsbart. *All gadget and no representation makes Jack a dull environment.* In In AAAI Spring Symposium on Intelligent Environments. AAAI TR, pages 155–160, 1998. (Cited at page 15.)

[Junnila *et al.* 2006] Miikka Junnila, Eero Hyvonen et Mirva Salminen. *Describing and Linking Cultural Semantic Content by Using Situations and Actions.* In Proceedings of the 12th Finnish Artificial Intelligence Conference STeP 2006, 2006. (Cited at page 81.)

[Kaasinen 2003] Eija Kaasinen. *User Needs for Location-aware Mobile Services.* Personal Ubiquitous Comput., vol. 7, no. 1, pages 70–79, Mai 2003. (Cited at page 23.)

[Kalman 1960] R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems.* Transactions of the ASME - Journal of Basic Engineering, no. 82 (Series D), pages 35–45, 1960. (Cited at pages 18, 40 et 151.)

[Kamura *et al.* 2011] Tetsuro Kamura, Hideaki Takeda, Ikki Ohmukai, Fumihiro Kato, Toru Takahashi et Hiroshi Ueda. *Study Support and Integration of Cultural Information Resources with Linked Data.* In Culture and Computing, pages 177–178, 2011. (Cited at pages ix, 80 et 81.)

[Karmacharya *et al.* 2012] A. Karmacharya, Cruz C. et F. Boochs. *Spatialization of the Semantic Web.* In Muhammad Tanvir Afzal, editeur, Advances in Theories and Mathematical Models, chapitre 7, pages 152–192. InTech, 2012. (Cited at page 71.)

[Kato & Billinghurst 1999] Hirokazu Kato et Mark Billinghurst. *Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System.* In Proceedings of the 2Nd IEEE and ACM International Workshop on Augmented Reality, IWAR '99, pages 85–, Washington, DC, USA, 1999. IEEE Computer Society. (Cited at page 38.)

[Kauppinen *et al.* 2011] Tomi Kauppinen, Panu Paakkarinen, Eetu Mäkela, Heini Kuittinen, Jari Väätäinen et Eero Hyvönen. *Geospatio-Temporal Semantic Web for Cultural Heritage.* In Lily Diaz Miltiadis Lytras Ernesto Damiani et Patricia Ordonez De Pablos, editeurs, Digital Culture and E-Tourism: Technologies, Applications and Management Approaches, chapitre 4, pages 48–64. IGI Global, 2011. (Cited at page 82.)

[Keet 2004] C. Maria (Marijke) Keet. *Aspects of ontology integration*, 2004. (Cited at pages ix et 51.)

[Keßler & Kauppinen 2012] Carsten Keßler et Tomi Kauppinen. *Linked Open Data University of Muenster—Infrastructure and Applications.* In Demos the Extended Semantic Web Conference 2012 (ESWC2012), Heraklion, Crete, Greece, May 2012. (Cited at page 171.)

[Kim *et al.* 2005] Jong Woo Kim, Ju-Yeon Kim, Hyun-Suk Hwang, Sung-Seok Park, Chang Soo Kim et Sung-gi Park. *The Semantic Web Approach in Location Based Services.* In Computational Science and Its Applications - ICCSA 2005, International Conference, pages 127–136, Mai 2005. (Cited at page 13.)

[Klippel & Winter 2005] Alexander Klippel et Stephan Winter. *Structural Salience of Landmarks for Route Directions.* In Spatial Information Theory, International Conference, COSIT 2005, Ellicottville, NY, USA, pages 347–362, Septembre 2005. (Cited at page 32.)

[Kobilarov *et al.* 2009] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer et Robert Lee. *Media Meets Semantic Web - How the BBC Uses DBpedia and Linked Data to Make Connections.* In ESWC, pages 723–737, 2009. (Cited at page 79.)

[Kolbe *et al.* 2005] Thomas H. Kolbe, Gerhard Gröger et Lutz Plümer. *CityGML – Interoperable Access to 3D City Models.* In Proceedings of the first International Symposium on Geo-Information for Disaster Management, pages 21–23. Springer Verlag, 2005. (Cited at pages ix, 29 et 30.)

[Kray *et al.* 2003] Christian Kray, Christian Elting, Katri Laakso et Volker Coors. *Presenting Route Instructions on Mobile Devices.* In Proceedings of the 8th International Conference on Intelligent User Interfaces, IUI '03, pages 117–124, New York, NY, USA, 2003. ACM. (Cited at page 14.)

[Lane *et al.* 2010] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury et Andrew T. Campbell. *A Survey of Mobile Phone*

*Sensing*. Comm. Mag., vol. 48, no. 9, pages 140–150, Septembre 2010. (Cited at page 17.)

[Lee & Hollerer 2006] Taehee Lee et Tobias Hollerer. *Viewpoint Stabilization for Live Collaborative Video Augmentations*. In Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR '06, pages 241–242, Washington, DC, USA, 2006. IEEE Computer Society. (Cited at page 39.)

[Lee *et al.* 2009a] Ryong Lee, Yong-Jin Kwon et Kazutoshi Sumiya. *Layer-Based Media Integration for Mobile Mixed-Reality Applications*. In Third International Conference on Next Generation Mobile Applications, Services and Technologies, pages 58–63, Septembre 2009. (Cited at pages ix, 36 et 37.)

[Lee *et al.* 2009b] Tae-Hoon Lee, Jung-Hyun Kim, Hyeong-Joon Kwon et Kwang-Seok Hong. *Keyword-based Semantic Retrieval System using Location Information in a Mobile Environment*. In International Symposium on Web Information Systems and Applications, pages 434–438, Mai 2009. (Cited at pages 19 et 20.)

[Liagouris *et al.* 2011] John Liagouris, Spiros Athanasiou, Alexandros Efentakis, Stefan Pfennigschmidt, Dieter Pfoser, Eleni Tsigka et Agnès Voisard. *Mobile Task Computing: Beyond Location-based Services and Ebooks*. In Proceedings of the 10th International Conference on Web and Wireless Geographical Information Systems, W2GIS'11, pages 124–141, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited at pages ix, 21 et 22.)

[Lim & Dey 2011] Brian Y. Lim et Anind K. Dey. *Design of an intelligible mobile context-aware application*. In Proceedings of the 13th Conference on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2011, Stockholm, Sweden, August 30 - September 2, 2011, pages 157–166, 2011. (Cited at page 17.)

[Marimon & Adamek 2011] David Marimon et Tomasz Adamek. *Where is Reality In This Browser ? Or Why Visual Tracking is Key in Mobile AR*. In International AR Standards Meeting, Février 2011. (Cited at page 38.)

[Mate *et al.* 2014] Pramila Mate, Hariram Chavan et Vishwaji Gaikwad. *Android Based Continuous Query Processing in Location Based Services*. International Journal of Advanced Research in Computer Science and Software Engineering, vol. 5, pages 134–138, May 2014. (Cited at page 24.)

[Matsumura *et al.* 2012] Fuyuko Matsumura, Iwao Kobayashi, Fumihiro Kato, Tetsuro Kamura, Ikki Ohmukai et Hideaki Takeda. *Producing and Consuming Linked Open Data on Art with a Local Community.* In COLD, 2012. (Cited at pages ix, 80 et 81.)

[Mendes *et al.* 2012] Pablo N. Mendes, Max Jakob et Christian Bizer. *DBpedia: A Multilingual Cross-domain Knowledge Base.* In LREC, pages 1813–1817, 2012. (Cited at page 68.)

[Meng 2008] Liqiu Meng. *The State of the Art of Map-Based Mobile Services.* In Liqiu Meng, Alexander Zipf et Stephan Winter, editeurs, Map-based Mobile Services: Design, Interaction and Usability, Lecture Notes in Geoinformation and Cartography, pages 1–12. Springer Berlin Heidelberg, 2008. (Cited at page 32.)

[Milgram & Kishino 1994] Paul Milgram et Fumio Kishino. *A Taxonomy of Mixed Reality Visual Displays.* IEICE Transactions on Information Systems, vol. E77-D, no. 12, Décembre 1994. (Cited at pages ix, 33 et 35.)

[Milgram *et al.* 1994] Paul Milgram, Haruo Takemura, Akira Utsumi et Fumio Kishino. *Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum.* In Proceedings of the SPIE Conference on Telemanipulator and Telepresence Technologies, pages 282–292, 1994. (Cited at page 96.)

[Mirizzi *et al.* 2010] Roberto Mirizzi, Azzurra Ragone, Tommaso Di Noia et Eugenio Di Sciascio. *Ranking the Linked Data: The Case of DBpedia.* In ICWE, pages 337–354, 2010. (Cited at page 77.)

[Mirizzi *et al.* 2012] Roberto Mirizzi, Tommaso Di Noia, Azzurra Ragone, Vito Claudio Ostuni et Eugenio Di Sciascio. *Movie Recommendation with DBpedia.* In IIR, pages 101–112, 2012. (Cited at pages ix, 78 et 79.)

[Mokbel & Levandoski 2009] Mohamed F. Mokbel et Justin J. Levandoski. *Toward Context and Preference-aware Location-based Services.* In Proceedings of the Eighth ACM International Workshop on Data Engineering for Wireless and Mobile Access, MobiDE '09, pages 25–32, New York, NY, USA, 2009. ACM. (Cited at page 12.)

[Mulay & Kumar 2011] Kunal Mulay et P. Sreenivasa Kumar. *SPRING: Ranking the results of SPARQL queries on Linked Data.* In COMAD, pages 47–56, 2011. (Cited at page 77.)

[Naimark & Foxlin 2002] Leonid Naimark et Eric Foxlin. *Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker.* In ISMAR, pages 27–36. IEEE Computer Society, 2002. (Cited at page 38.)

[Nedkov 2012] Simeon Nedkov. *Knowledge-based optimisation of three-dimensional city models for car navigation devices*, 2012. (Cited at page 14.)

[Nelson 1965] T. H. Nelson. *A file structure for the complex, the changing and the indeterminate.* In Proceedings of the 1965 20th national conference, ACM '65, pages 84–100, New York, NY, USA, 1965. ACM. (Cited at page 45.)

[Neubauer *et al.* 2009] N. Neubauer, M. Over, Schilling A. et A. Zipf. *Virtual Cities 2.0: Generating web-based 3D city models and landscapes based on free and user generated data (OpenStreetMap).* In Contribution of Geovisualization to the concept of the Digital City. Workshop., GeoViz 2009, 2009. (Cited at page 28.)

[Neumann 2010] Andreas Neumann. *Spatial, Temporal and Thematic Navigation - Visualizing Biographies of European Artists Using a Task Oriented User Interface Design Approach.* PhD thesis, ETH Zurich, 2010. (Cited at page 3.)

[Niwattanakul *et al.* 2013] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn et Supachanun Wanapu. *Using of jaccard Coefficient for Keyword Similarity.* In Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS), 2013. (Cited at page 118.)

[Nixon *et al.* 2012] Lyndon J. B. Nixon, Jens Grubert, Gerhard Reitmayr et James Scicluna. *Semantics Enhancing Augmented Reality and Making Our Reality Smarter.* In OTM Conferences (2), pages 863–870, 2012. (Cited at page 85.)

[Noguera *et al.* 2012] José M. Noguera, Manuel J. Barranco, Rafael J. Segura et Luis MartíNez. *A Mobile 3D-GIS Hybrid Recommender System for Tourism.* Inf. Sci., vol. 215, pages 37–52, Décembre 2012. (Cited at page 14.)

[Noia *et al.* 2012] Tommaso Di Noia, Roberto Mirizzi, Vito Claudio Ostuni, Davide Romito et Markus Zanker. *Linked open data to support content-based recommender systems.* In I-SEMANTICS, pages 1–8, 2012. (Cited at pages 76 et 78.)

[Nurminen & Oulasvirta 2008] Antti Nurminen et Antti Oulasvirta. *Designing interactions for navigation in 3D mobile maps.* Lecture Notes in Geoinfor-

mation and Cartography, chapitre Mobile Maps, pages 198–227. Springer-Verlag, London, UK., 2008. (Cited at pages 27, 31 et 32.)

[Nurminen *et al.* 2011] Antti Nurminen, Ernst Kruijff et Eduardo Veas. *HYDROSYS: A Mixed Reality Platform for On-site Visualization of Environmental Data.* In Proceedings of the 10th International Conference on Web and Wireless Geographical Information Systems, W2GIS'11, pages 159–175, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited at page 36.)

[Nurminen 2006] Antti Nurminen. *m-LOMA - a Mobile 3D City Map.* In Proceedings of the Eleventh International Conference on 3D Web Technology, Web3D '06, pages 7–18, New York, NY, USA, 2006. ACM. (Cited at pages 26, 27 et 34.)

[Nurminen 2008] Antti Nurminen. *Mobile 3D City Maps.* IEEE Computer Graphics and Applications, vol. 28, no. 4, pages 20–31, 2008. (Cited at pages 14 et 31.)

[Ostuni *et al.* 2012] Vito Claudio Ostuni, Tommaso Di Noia, Roberto Mirizzi, Davide Romito et Eugenio Di Sciascio. *Cinemappy: a Context-aware Mobile App for Movie Recommendations boosted by DBpedia.* In SeRSy, pages 37–48, 2012. (Cited at page 85.)

[Oulasvirta *et al.* 2005] Antti Oulasvirta, Sakari Tamminen, Virpi Roto et Jaana Kuorelahti. *Interaction in 4-second Bursts: The Fragmented Nature of Attentional Resources in Mobile HCI.* In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI'05, pages 919–928, New York, NY, USA, 2005. ACM. (Cited at page 31.)

[Oulasvirta *et al.* 2009] Antti Oulasvirta, Sara Estlander et Antti Nurminen. *Embodied interaction with a 3D versus 2D mobile map.* Personal and Ubiquitous Computing, vol. 13, no. 4, pages 303–320, 2009. (Cited at pages 3 et 14.)

[Over *et al.* 2010] M. Over, Arne Schilling, S. Neubauer et Alexander Zipf. *Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany.* Computers, Environment and Urban Systems, vol. 34, no. 6, pages 496–507, 2010. (Cited at page 28.)

[Parent *et al.* 1997] Christine Parent, Stefano Spaccapietra, Esteban Zimányi, Pier Donini, Corinne Plazanet, Christelle Vangenot, Nadia Rognon et Pierre-André Crausaz. *MADS, modèle conceptuel spatio-temporel.* Revue Internationale de Géomatique, vol. 7, no. 3–4, pages 317–352, Décembre 1997. (Cited at page 107.)

[Pascoe 1998] Mr. Jason Pascoe. *Adding Generic Contextual Capabilities to Wearable Computers.* In Proceedings of the 2Nd IEEE International Symposium on Wearable Computers, ISWC '98, pages 92–, Washington, DC, USA, 1998. IEEE Computer Society. (Cited at pages 15 et 32.)

[Pattuelli & Rubinow 2013] Cristina Pattuelli et Sara Rubinow. *The knowledge organization of DBpedia: a case study.* In Journal of Documentation, pages 762–772, 2013. (Cited at pages 109, 110 et 111.)

[Paul 2012] Christiane Paul. *Contexts as Moving Targets: Locative Media Art and the Shifting Ground of Context-Awareness.* In Ulrik Ekman, editeur, Throughout: Art and Culture Emerging with Ubiquitous Computing, pages 399–418. MIT Press, 2012. (Cited at page 12.)

[Persa 2006] Stelian-Florin Persa. *Sensor Fusion in Head Pose Tracking for Augmented Reality.* PhD thesis, Delft University of Technology, 2006. (Cited at page 40.)

[Pospischil *et al.* 2002] Günther Pospischil, Martina Umlauft et Elke Michlmayr. *Designing LoL@, a Mobile Tourist Guide for UMTS.* In Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction, Mobile HCI '02, pages 140–154, London, UK, UK, 2002. Springer-Verlag. (Cited at page 23.)

[Prieto & Izkara 2012] Iñaki Prieto et Jose Luis Izkara. *Visualization of 3D City Models on Mobile Devices.* In Proceedings of the 17th International Conference on 3D Web Technology, Web3D '12, pages 101–104, New York, NY, USA, 2012. ACM. (Cited at pages 30 et 31.)

[Priyantha *et al.* 2011] Bodhi Priyantha, Dimitrios Lymberopoulos et Jie Liu. *LittleRock: Enabling Energy-Efficient Continuous Sensing on Mobile Phones.* IEEE Pervasive Computing, vol. 10, no. 2, pages 12–15, 2011. (Cited at page 20.)

[Prud'hommeaux & Seaborne 2008] Eric Prud'hommeaux et Andy Seaborne. *SPARQL Query Language for RDF.* W3C Recommendation, Date of Access: 18 Janvier 2014, Available from `http://www.w3.org/TR/rdf-sparql-query/`, January 2008. (Cited at page 71.)

[Rakkolainen *et al.* 2000] Ismo Rakkolainen, Jani Timmerheid et Teija Vainio. *A 3D City Info for Mobile Users.* Computers and Graphics, vol. 25, pages 619–625, 2000. (Cited at pages 26 et 31.)

[Reichenbacher 2001] Tumasch Reichenbacher. *The World In Your Pocket Towards A Mobile Cartography.* In In: Proc. of the 20th International Cartographic Conference (CDROM, pages 2514–2521, 2001. (Cited at page 23.)

[Reitmayr & Drummond 2006] Gerhard Reitmayr et Tom W. Drummond. *Going out: Robust modelbased tracking for outdoor augmented reality.* In Proceedings of 5th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR, pages 109–118, 2006. (Cited at page 40.)

[Reitmayr & Schmalstieg 2004] Gerhard Reitmayr et Dieter Schmalstieg. *Collaborative Augmented Reality for Outdoor Navigation and Information Browsing.* In Proceedings of the Second Symposium on Location Based Services and TeleCartography, pages 53–62. TU Wien, 2004. (Cited at pages 35 et 36.)

[Retscher 2005] Günther Retscher. *An Intelligent Multi-sensor System for Pedestrian Navigation.* In GNSS Hong Kong 2005, 2005. talk: GNSS Hong Kong 2005 Conference, Hong Kong, PR China; 2005-12-08 – 2005-12-10. (Cited at page 18.)

[Reynolds *et al.* 2010] Vinny Reynolds, Michael Hausenblas, Axel Polleres, Manfred Hauswirth et Vinod Hegde. *Exploiting Linked Open Data for Mobile Augmented Reality.* In W3C Workshop: Augmented Reality on the Web, 2010. (Cited at page 85.)

[Ribo *et al.* 2002] Miguel Ribo, Peter Lang, Harald Ganster, Markus Brandner, Christoph Stock et Axel Pinz. *Hybrid Tracking for Outdoor Augmented Reality Applications.* IEEE Comput. Graph. Appl., vol. 22, no. 6, pages 54–63, Novembre 2002. (Cited at page 40.)

[Rinner & Raubal 2004] Claus Rinner et Martin Raubal. *Personalized Multi-Criteria Decision Strategies in Location-Based Decision Support.* Annals of GIS, vol. 10, no. 2, pages 149–156, 2004. (Cited at page 16.)

[Roa-Valverde 2011] Antonio J. Roa-Valverde. *Multimedia information retrieval as a practical application for interlinking approaches.* In I-SEMANTICS, pages 230–233, 2011. (Cited at page 78.)

[Robusto 1957] C. C. Robusto. *The cosine-haversine formula.* The American Mathematical Monthly, vol. 64, no. 1, pages 38–40, 1957. (Cited at page 154.)

[Rodden *et al.* 1998] Tom Rodden, Keith Chervest, Nigel Davies et Alan Dix. *Exploiting Context in HCI Design for Mobile Systems.* In in Workshop on Hu-

man Computer Interaction with Mobile Devices, 1998. (Cited at pages 15, 17 et 18.)

[Sabou *et al.* 2012] Marta Sabou, Adrian M. P. Brasoveanu et Irem Arsal. *Supporting tourism decision making with linked data.* In I-SEMANTICS, pages 201–204, 2012. (Cited at page 79.)

[Saeedi *et al.* 2014] Sara Saeedi, Adel Moussa et Naser El-Sheimy. *Context-Aware Personal Navigation Using Embedded Sensor Fusion in Smartphones.* Sensors, vol. 14, no. 4, pages 5742–5767, 2014. (Cited at pages 16 et 18.)

[Salton *et al.* 1975] Gerard Salton, A. Wong et C. S. Yang. *A Vector Space Model for Automatic Indexing.* Commun. ACM, vol. 18, no. 11, pages 613–620, 1975. (Cited at page 77.)

[Samadzadegan & Saeedi 2005] Farhad Samadzadegan et Sara Saeedi. *Fusion of remotely sensed data in the context of sensors, features and decisions.* WSEAS Transactions on Environment and Development, vol. 1, no. 3, pages 363–370, 2005. (Cited at page 18.)

[Satyanarayanan 2001] Mahadev Satyanarayanan. *Pervasive computing: vision and challenges.* IEEE Personal Commun., vol. 8, no. 4, pages 10–17, 2001. (Cited at page 12.)

[Savage *et al.* 2011] Saiph Savage, Maciek Baranski, Norma Elva Chavez et Tobias Hollerer. *I'm Feeling LoCo: A Location Based Context Aware Recommendation System.* In Advances in Location-Based Services: 8th International Symposium on Location-Based Services, Vienna 2011, Lecture Notes in Geoinformation and Cartography. Springer, 2011. (Cited at page 16.)

[Schall *et al.* 2009] Gerhard Schall, Daniel Wagner, Gerhard Reitmayr, Elise Taichmann, Manfred Wieser, Dieter Schmalstieg et Bernhard Hofmann-Wellenhof. *Global pose estimation using multi-sensor fusion for outdoor Augmented Reality.* In Gudrun Klinker, Hideo Saito et Tobias Höllerer, editeurs, ISMAR, pages 153–162. IEEE Computer Society, 2009. (Cited at pages 18, 39 et 40.)

[Schall *et al.* 2013] Gerhard Schall, Stefanie Zollmann et Gerhard Reitmayr. *Smart Vidente: advances in mobile augmented reality for interactive visualization of underground infrastructure.* Personal and Ubiquitous Computing, vol. 17, no. 7, pages 1533–1549, 2013. (Cited at pages 4, 14 et 36.)

[Schilit & Theimer 1994] B. N. Schilit et M. M. Theimer. *Disseminating Active Map Information to Mobile Hosts.* Netwrk. Mag. of Global Internetwkg., vol. 8, no. 5, pages 22–32, Septembre 1994. (Cited at pages 14 et 20.)

[Schilling *et al.* 2003] Arne Schilling, Volker Coors, Martin Giersich et Rune Aasgaard. *Introducing 3D GIS for the Mobile Community - Technical aspects in the case of TellMaris.* In Proceedings of International Workshop on Mobile Computing: Assistance, Mobility, Applications (IMC 2003), pages 86– 92, Rostock, Juin 2003. Fraunhofer IRB Verlag. (Cited at pages 26 et 27.)

[Schilling *et al.* 2005] Arne Schilling, Volker Coors et Katri Laakso. *Dynamic 3D Maps for Mobile Tourism Applications.* In L. Meng, T. Reichenbacher et A. Zipf, editeurs, Map-based Mobile Services, pages 227–239. Springer Berlin Heidelberg, 2005. (Cited at page 29.)

[Schinke *et al.* 2010] Torben Schinke, Niels Henze et Susanne Boll. *Visualization of Off-screen Objects in Mobile Augmented Reality.* In Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI '10, pages 313–316, New York, NY, USA, 2010. ACM. (Cited at page 35.)

[Schmachtenberg *et al.* 2014] Max Schmachtenberg, Thorsten Strufe et Heiko Paulheim. *Enhancing a Location-based Recommendation System by Enrichment with Structured Data from the Web.* In Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), WIMS '14, pages 17:1–17:12, New York, NY, USA, 2014. ACM. (Cited at page 13.)

[Schmalstieg & Reitmayr 2007] Dieter Schmalstieg et Gerhard Reitmayr. *The World as a User Interface: Augmented Reality for Ubiquitous Computing.* In Georg Gartner, William Cartwright et MichaelP. Peterson, editeurs, Location Based Services and TeleCartography, Lecture Notes in Geoinformation and Cartography, pages 369–391. Springer Berlin Heidelberg, 2007. (Cited at page 36.)

[Schmeiß *et al.* 2010] Daniel Schmeiß, Ansgar Scherp et Steffen Staab. *Integrated mobile visualization and interaction of events and POIs.* In Alberto Del Bimbo, Shih-Fu Chang et Arnold W. M. Smeulders, editeurs, ACM Multimedia, pages 1567–1570. ACM, 2010. (Cited at pages 23 et 24.)

[Schmidt 2013] Albrecht Schmidt. *Context-Aware Computing: Context-Awareness, Context-Aware User Interfaces, and Implicit Interaction.* In Mads Soegaard

et Rikke Friis Dam, editeurs, Encyclopedia of Human-Computer Interaction, Aarhus, Denmark, 2013. The Interaction Design Foundation. (Cited at page 24.)

[Schultz *et al.* 2011] Andreas Schultz, Andrea Matteini, Robert Isele, Christian Bizer et Christian Becker. *LDIF - Linked Data Integration Framework.* In COLD, 2011. (Cited at page 171.)

[Segaran *et al.* 2009] Toby Segaran, Colin Evans et Jamie Taylor. Programming the semantic web - build flexible applications with graph data. O'Reilly, 2009. (Cited at pages 57, 58 et 61.)

[Shadbolt *et al.* 2006] Nigel Shadbolt, Tim Berners-Lee et Wendy Hall. *The Semantic Web Revisited.* IEEE Intelligent Systems, vol. 21, no. 3, pages 96–101, Mai 2006. (Cited at page 44.)

[Shannon 2006] Victoria Shannon. *A 'more revolutionary' Web.* Internationnal Herald Tribune, May 2006. (Cited at page 47.)

[Simon *et al.* 2008] Rainer Simon, Peter Fröhlich et Thomas Grechenig. *GeoPointing: evaluating the performance of orientation-aware location-based interaction under real-world conditions.* J. Location Based Services, vol. 2, no. 1, pages 24–40, 2008. (Cited at page 17.)

[Stadler & Kolbe 2007] Alexandra Stadler et Thomas H. Kolbe. *Spatio-semantic Coherence in the Integration of 3D City Models.* In Proceedings of the 5th International Symposium on Spatial Data Quality, 2007. (Cited at pages ix et 30.)

[Stadler 2010] Claus Stadler. *Community-Driven Engineering of the DBpedia Infobox Ontology and DBpedia Live Extraction.* PhD thesis, Universität Leipzig, 2010. (Cited at pages 72 et 76.)

[Stankovic *et al.* 2011] Milan Stankovic, Werner Breitfuss et Philippe Laublet. *Linked-data based suggestion of relevant topics.* In I-SEMANTICS, pages 49–55, 2011. (Cited at page 77.)

[Stevens *et al.* 2000] Robert Stevens, Carole A. Goble et Sean Bechhofer. *Ontology-based Knowledge Representation for Bioinformatics.* Briefings in Bioinformatics, vol. 1, no. 4, pages 398–414, 2000. (Cited at page 51.)

[Stoter & Zlatanova 2003] Jantien Stoter et Siyka Zlatanova. *3D GIS, where are we standing.* In Joint Workshop on Spatial, Temporal and Multi-Dimensional Data Modeling and Analysis, 2003. (Cited at page 14.)

[Suchanek *et al.* 2008] Fabian M. Suchanek, Gjergji Kasneci et Gerhard Weikum. *YAGO: A Large Ontology from Wikipedia and WordNet.* eb Semantics: Science, Services and Agents on the World Wide Web, vol. 6, no. 3, pages 203–217, Septembre 2008. (Cited at page 110.)

[Suomela 2006] Riku Suomela. *Constructing and Examining Location-Based Applications and Their User Interfaces by Applying Rapid Software Development and Structural Analysis.* PhD thesis, Tampere University of Technology, 2006. (Cited at pages 3, 4 et 37.)

[Tao *et al.* 2003] Yufei Tao, Nikos Mamoulis et Dimitris Papadias. *Validity Information Retrieval for Spatio-Temporal Queries: Theoretical Performance Bounds.* In Thanasis Hadzilacos, Yannis Manolopoulos, John F. Roddick et Yannis Theodoridis, editeurs, SSTD, volume 2750 of *Lecture Notes in Computer Science*, pages 159–178. Springer, 2003. (Cited at page 20.)

[Terzi *et al.* 2003] Evimaria Terzi, Athena Vakali et Mohand-Said Hacid. *Knowledge Representation, Ontologies, and the Semantic Web.* In APWeb, pages 382–387, 2003. (Cited at pages 54 et 55.)

[Thuraisingham 2008] Bhavani M. Thuraisingham. *Data Mining for Security Applications and Its Privacy Implications.* In PinKDD, pages 1–6, 2008. (Cited at page 49.)

[Toupikov *et al.* 2009] Nickolai Toupikov, Jürgen Umbrich, Renaud Delbru, Michael Hausenblas et Giovanni Tummarello. *DING ! Dataset Ranking using Formal Descriptions.* In LDOW, 2009. (Cited at page 77.)

[Trapp *et al.* 2011] Matthias Trapp, Lars Schneider, Christine Lehmann, Norman Holz et Jürgen Döllner. *Strategies for visualising 3D points-of-interest on mobile devices.* J. Location Based Services, vol. 5, no. 2, pages 79–99, 2011. (Cited at pages 14, 27 et 32.)

[Törnroos *et al.* 2013] Juha Törnroos, Eetu Mäkelä, Thea Lindquist et Eero Hyvönen. *World War 1 as Linked Open Data*, 2013. Submitted for review. (Cited at page 80.)

[Trotman & Zhang 2013] Andrew Trotman et Jinglan Zhang. *Future Web Growth and its Consequences for Web Search Architectures.* CoRR, 2013. (Cited at page 44.)

[van Aart *et al.* 2010] Chris J. van Aart, Bob J. Wielinga et Willem Robert van Hage. *Mobile Cultural Heritage Guide: Location-Aware Semantic Search.* In EKAW, pages 257–271, 2010. (Cited at pages 5, 13 et 85.)

[van Heijst *et al.* 1997] Gertjan van Heijst, A. Th. Schreiber et Bob J. Wielinga. *Using explicit ontologies in KBS development.* Int. J. Hum.-Comput. Stud., vol. 46, no. 2, pages 183–292, 1997. (Cited at pages ix et 52.)

[Verbert *et al.* 2012] Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic et Erik Duval. *Context-Aware Recommender Systems for Learning: A Survey and Future Challenges.* IEEE Trans. Learn. Technol., vol. 5, no. 4, pages 318–335, Janvier 2012. (Cited at pages 15 et 16.)

[Viktoratos *et al.* 2014] Iosif Viktoratos, Athanasios Tsadiras et Nick Bassiliades. *Providing a context-aware location based web service through semantics and user-defined rules.* In 4th International Conference on Web Intelligence, Mining and Semantics (WIMS 14), WIMS '14, Thessaloniki, Greece, June 2-4, 2014, page 9, 2014. (Cited at page 13.)

[Wagner & Schmalstieg 2003] Daniel Wagner et Dieter Schmalstieg. *ARToolKit on the PocketPC Platform*, 2003. (Cited at page 38.)

[Weiser *et al.* 1999] M. Weiser, R. Gold et J. S. Brown. *The Origins of Ubiquitous Computing Research at PARC in the Late 1980s.* IBM Syst. J., vol. 38, no. 4, pages 693–696, Décembre 1999. (Cited at page 12.)

[Weiser 1991] Mark Weiser. *The computer for the 21st century.* Scientific American, vol. 265, no. 3, pages 66–75, Septembre 1991. (Cited at page 11.)

[Welch & Bishop 1995] Greg Welch et Gary Bishop. *An introduction to the Kalman filter.* University of North Carolina at Chapel Hill, Chapel Hill, NC, pages 1–16, 1995. (Cited at page 18.)

[Willmes *et al.* 2010] C. Willmes, U. Baaser, K. Volland et Jun Bareth G. *Internet based distribution and visualization of a 3D model of the university of Cologne campus.* In 3rd ISDE Digital Earth, 2010. (Cited at page 30.)

[Wilson & Matthews 2004] Michael D. Wilson et Brian Matthews. *The Future of the World Wide Web ?* In BNCOD, pages 4–15, 2004. (Cited at page 49.)

[Wilson *et al.* 2005] Max Wilson, Alistair Russell, Daniel A. Smith, Alisdair Owens et M. C. Schraefel. *mSpace Mobile: A Mobile Application for the Semantic Web.* In User Semantic Web Workshop, ISWC2005, 2005. (Cited at page 21.)

[Wyssusek 2004] Boris Wyssusek. *Ontology and Ontologies in Information Systems Analysis and Design: A Critique.* In AMCIS, page 535, 2004. (Cited at page 50.)

[Yang 2009] Jun Yang. *Toward Physical Activity Diary: Motion Recognition Using Simple Acceleration Features with Mobile Phones.* In Proceedings of the 1st International Workshop on Interactive Multimedia for Consumer Electronics, IMCE '09, pages 1–10, New York, NY, USA, 2009. ACM. (Cited at page 16.)

[Ye *et al.* 2011] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin et Krzysztof Janowicz. *On the semantic annotation of places in location-based social networks.* In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011, pages 520–528, 2011. (Cited at page 13.)

[You *et al.* 1999] Suya You, Ulrich Neumann et Ronald Azuma. *Hybrid Inertial and Vision Tracking for Augmented Reality Registration.* In Proceedings of the IEEE Virtual Reality, VR '99, pages 260–, Washington, DC, USA, 1999. IEEE Computer Society. (Cited at pages 35 et 40.)

[Yu *et al.* 2005] S. Yu, L. Al-Jadir et S. Spaccapietra. *Matching User's Semantics with Data Semantics in Location-Based Services.* Semantics in Mobile Environments, Ayia Napa, Cyprus, 2005. (Cited at page 13.)

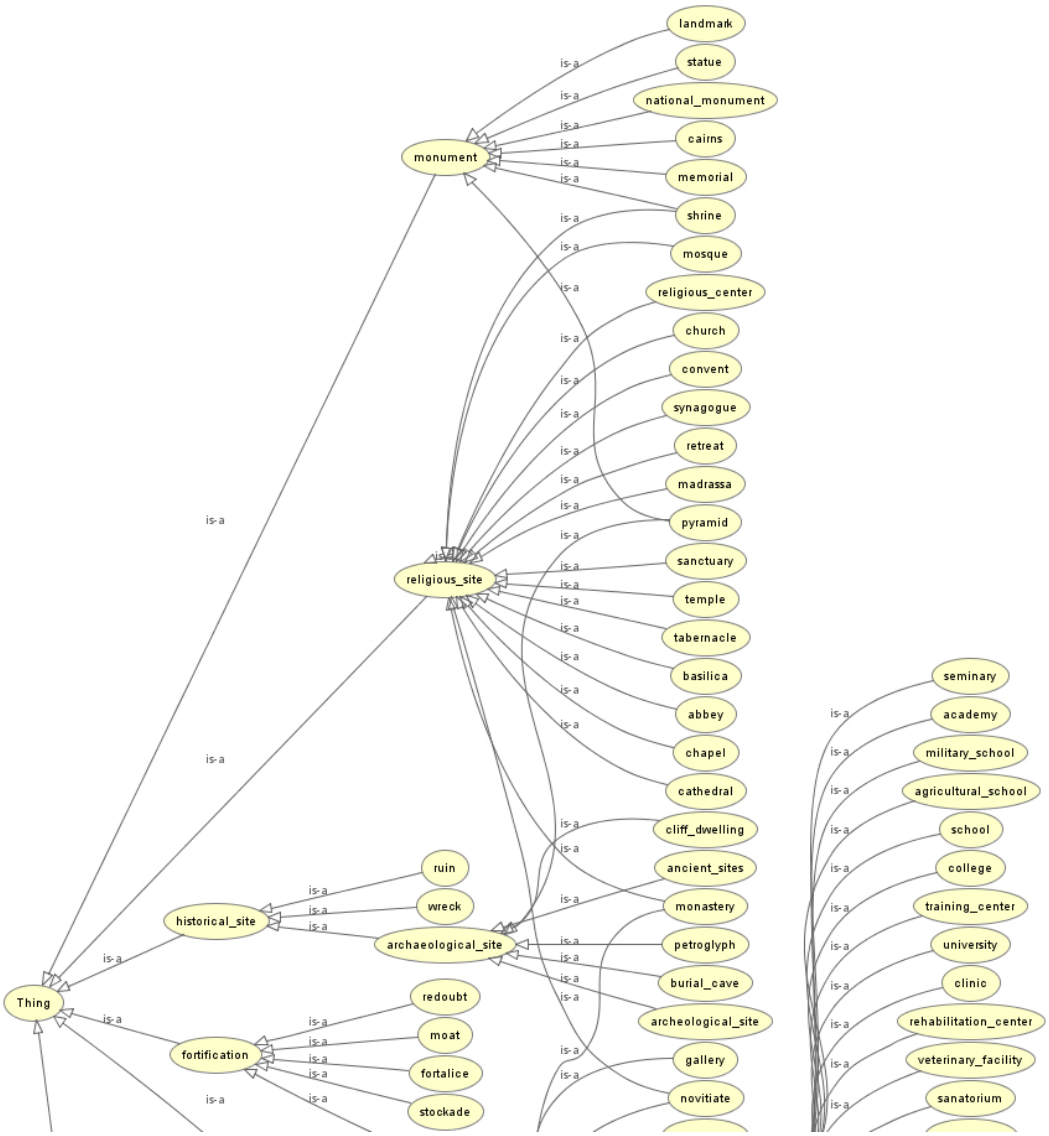
[Yu 2011] Liyang Yu. A developer's guide to the semantic web. Springer, 2011. (Cited at pages 69 et 72.)

[Zettsu *et al.* 2010] Koji Zettsu, Bernhard Thalheim, Yutaka Kidawara, Elina Karttunen et Hannu Jaakkola. *Future Directions of Knowledge Systems Environments for Web 3.0.* In EJC, pages 413–446, 2010. (Cited at page 47.)

[Zhou *et al.* 2008] Feng Zhou, Henry Been-Lirn Duh et Mark Billinghurst. *Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR.* In Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, ISMAR '08, pages 193–202, Washington, DC, USA, 2008. IEEE Computer Society. (Cited at pages 39 et 40.)

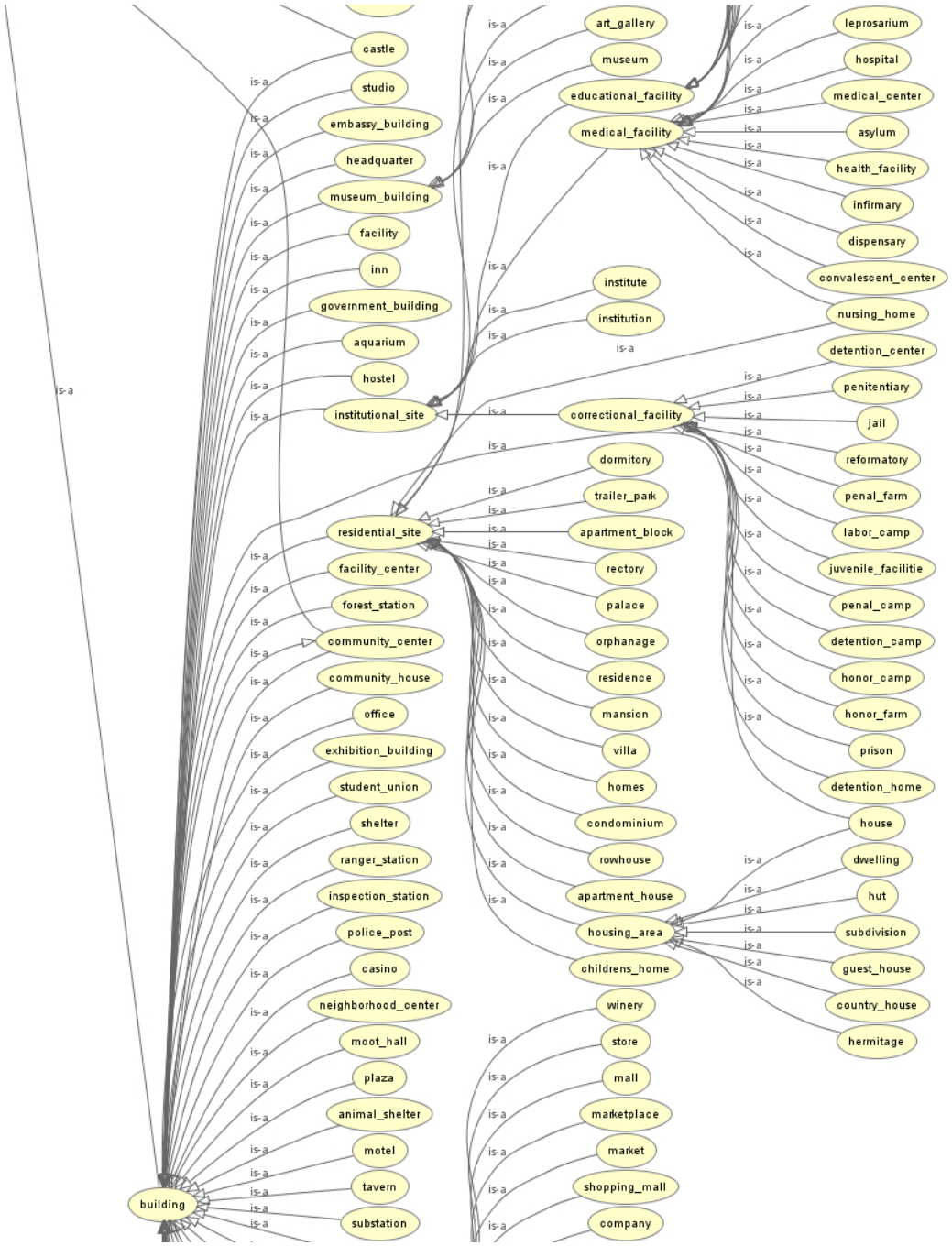
Appendix A

# MediOnto.owl



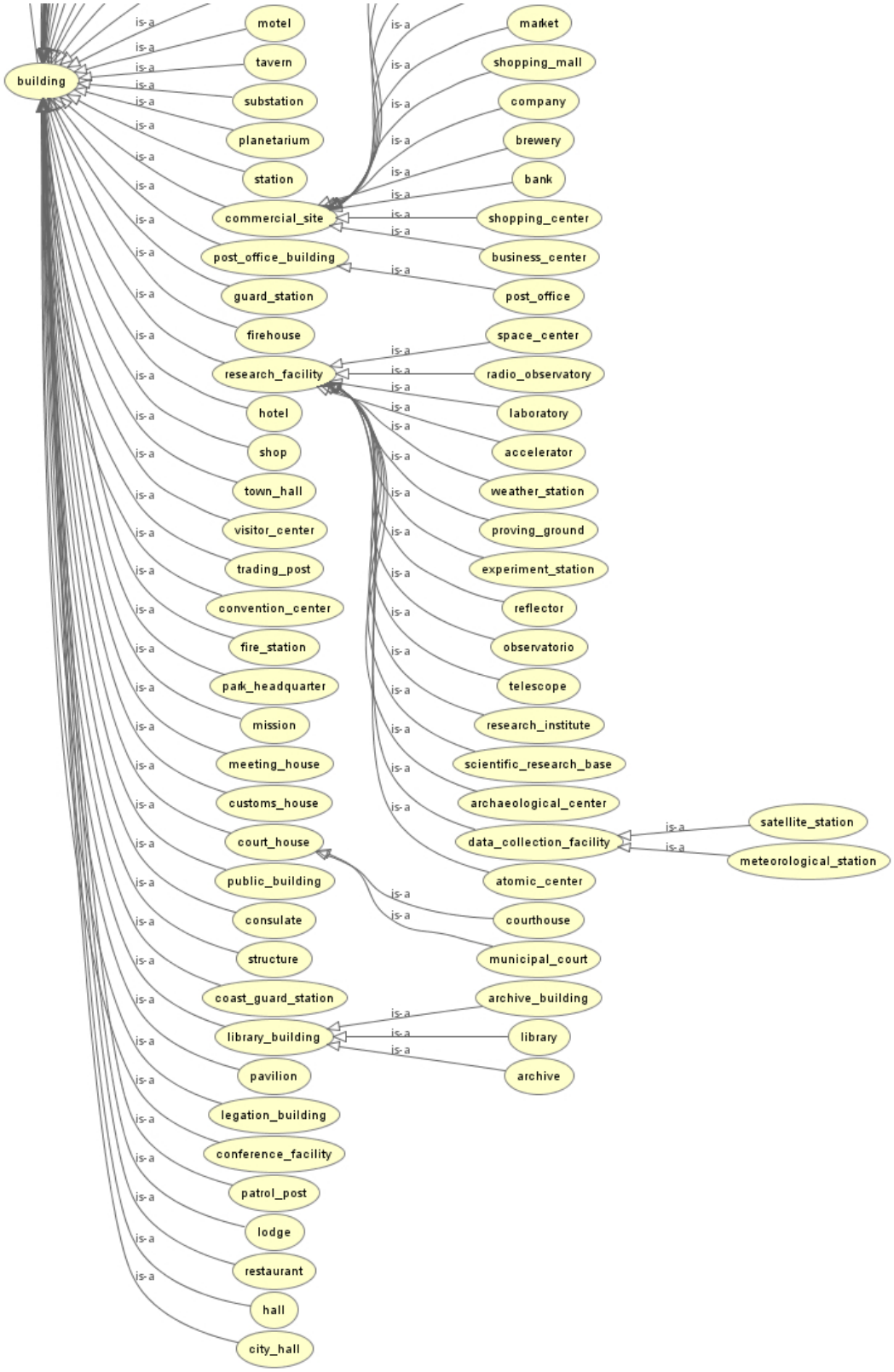

(a)

Figure A.1 – MediOnto.owl ontology (continued).




(b)

Figure A.1 – MediOnto.owl ontology (continued).




(c)

Figure A.1 – MediOnto.owl ontology.