State of the Art

Citisim

Smart City 3D Simulation and Monitoring Platform

ITEA3 – Project Citisim

Document Properties

Authors	Citisim Partners
Date	2018-08-29
Visiblity	Public
Status	Final

History of Changes

Release	Date	Author, Organization	Changes
0.0	26/12/2016	Emine Ferraro, ARGEDOR	SotA - Initial Document
0.1	21/01/2017	Emine Ferraro, ARGEDOR	SotA - GIS, 3D Modelling
0.2	01/02/2017	George Suciu, BEIA	SotA - Smart Cities– an overview
0.3	02/02/2017	Ignacio Brodin PRODEVELOP	SotA - Data Visualization Frameworks for IoT and for 2D and 3D modelling
0.4	10/02/2017	George Suciu, BEIA	SotA - State of the art in IoT
0.5	21/02/2017	George Suciu, BEIA	SotA - Relevant European research projects
0.6	26/02/2017	George Suciu, BEIA	SotA - State of the art in GIS Technologies
0.7	13/03/2017	Carmen Rusu, Luigi Mistodie, Marius Ivanov, ALTFACTOR	SotA - 3D City Modelling
0.8	03/04/2017	Emine Ferraro, ARGEDOR	SotA - Review and acronyms
0.9	10/04/2017	Jorge Luján, ABALIA	SotA - Review, Relevant European research projects
0.10	18/04/2017	Amelia del Rey, PRODEVELOP	SotA - Review as WP2 Leader
0.11	25/04/2017	Jesús Martínez, ANSWARE	SotA - Data sources and Augmented reality
0.12	27/04/2017	Carmen Rusu, Marius Ivanov, Luigi Mistodie, ALTFACTOR	SotA - Reviewed contribution of references
0.13	27/04/2017	George Suciu, BEIA	SotA - Review contributions, references list of abbreviations and acronyms, conclusions for 7.1 and 7.2
0.14	02/05/2017	Carmen Rusu, ALTFACTOR	SotA - Added abbreviations and acronyms
0.15	05/05/2017	Carmen Rusu, ALTFACTOR	SotA - Reviewed footnotes in 6.1-6.4 chapter
0.16	08/05/2017	Marcello Leida, TAIGER	SotA - Added domain ontologies review
1.0	01/08/2017	Pedro Ferrer, PRODEVELOP	Final Document

Release	Date	Author, Organization	Changes
2.0	29/08/2018	Carlos Jiménez, ABALIA	Final version, including other ITEA3 projects related to the challenge Smart cities

List of Figures

List of Tables

Table of Contents

1. Introduction

1.1. Project

The general purpose of CitiSim is devoted to the design and implementation of a new generation platform for the Smart City ecosystem.

This platform will provide a powerful monitoring and control infrastructure to enable planners to make critical management decisions on tactical and strategic levels based on the knowledge provided by the specific platform developed. For a natural interaction and better understanding of the events that happen in the city, 3D visualization techniques as augmented virtuality and augmented reality will be explored.

CitiSim will provide service developers with a set of services, standards and tools for the development of applications for the smart city. The major expected technical outcomes in CitiSim are:

- a platform defined by its services, protocols and tools for supporting advanced service development;
- a 3D visualization tool for smart city monitoring and control;
- a large-scale simulation framework for supporting the strategic and tactical decision process on smart cities.

1.2. Work package

This document has been produced as part of a deliverable within the WP2: Reference Architecture Framework. It corresponds to the deliverable D2.1 State of art & requirement analysis as result of the work done in the Task 2.1 Review of the state-of -the- art and requirement analysis that started in M1 and finished in M8.

1.3. Document overview

1.3.1. Document overview

The deliverable State of art & requirement analysis cover two different aspects that are directly related with T8.1 Requirements for reference architecture and T9.2 Ensuring market access and T9.4 Exploitation.

Thus, this document presents the state-of-the-art (SoTA) analysis of technologies and IoT projects, which will be relevant for the CitiSim project.

After an introduction to this document an overview of Smart Cities describes basic concepts and current trends. It is followed by relevant European IoT research projects and chapters dealing with IoT and global technologies for data visualization frameworks, GIS technologies, 3D City Modelling and Simulation and visualization 3D in Browsers. Finally, the document presents SoTA conclusions in each chapter.

1.3.2. Abbreviations and Acronyms

Abbreviation/Acronym	Definition
SoTA	State-of-the-art
FI-PPP	Future Internet Public-Private Partnership
IoT	Internet of Things
QoS	Quality of Service
RFID	Radio Frequency Identification
ARM	Architectural Reference Model
M2M	Machine to machine
HTTP	Hypertext transfer protocol
MQTT	Message Queue telemetry transport
NB-IoT	Narrowband- Internet of Things
GSM	Global Systems for mobile Communication
WCDMA	Wideband Code Division Multiple Access
LTE	Long-Term Evolution.
RRC	Radio Resource Control
3GPP	3rd Generation Partnership Project
GIS	Geographical Information System
SDI	Spatial Data Infrastructure
LBS	Location-Based Service (LBS)

Abbreviation/Acronym	Definition
LIDAR	Light Detection And Ranging
HMD	Head Mounted Display
GPS	Global Positioning System
DTM	Digital Terrain Model
SfM	Structure from Motion
DAP	Digital Aerials Photos
RFM	Rational Function Model
SAT-PP	Satellite Image Precision Processing
DEM	Digital Elevation Models
INS	Inertial Navigation System
RANSAC	Random sample consensus
VRML	Virtual Reality Modeling Language
TLS	Three-Line-Scanner / Terrestrial Lasers Scanning
CCD	Charge-Coupled Device
UAV	Unmanned Aerial Vehicle
GNSS	Global Navigation Satellite System
UAS	Unmanned Aerial Systems
IMU	Inertial Measurement Unit
MMS	Mobile Mapping System

Abbreviation/Acronym	Definition
VISAT	Video-Inertial-Satellite
NBV	Next Best View
LMU	Land Mark Update
ANN	Artificial Neural Networks
CAD	Computer Aided Design
CAAD	Computer Aided Architectural Design
MLP	Multi-Layer Perceptron
ADE	Application Domain Extensions
LOD	Level of Details
3D	Third Dimension
CityGML	City Geography Mark Up Language
LOD	Level of Detail
GIS	Geographic Information System
URL	Unified Resource Locator
XML	Extensible Markup Language
OGC	Open Geospatial Consortium
WP	Work Package
ICTs	Information and Communication Technologies
UTC	Urban Traffic Control

Abbreviation/Acronym	Definition
TMS	Traffic Management Systems
ITS	Intelligent Transportation Systems
IPv6	Internet Protocol Version 6
KPI	Key Performance Indicators
BI	Business Intelligence
BIRT	Eclipse Business Intelligence and Reporting Tools Project
ELK	ElasticSearch + Logstash* + Kibana
ETL	Extract, Transformation, and Load
XMLHttpRequest	XHR
API	Application Programming Interface
GPU	Graphics Processor Unit
GML3	Geography Markup Language
OGC	Open Geospatial Consortium
LOD	Levels of detail
AEC	Architecture, Engineering and Construction
OSM	OpenStreetMap
(ODbL)	Open Data Commons Open Database License
(OSMF)	OpenStreetMap Foundation

2. Smart Cities– an overview

2.1. What is Smart City?

2.1.1. Definition and key characteristics

In order to improve the management of urban processes and inhabitants' requirements, several administrations all over the world have presented significant number of future city's models in which technology, connectivity, sustainability, comfort, safety and attractiveness shape the crucial objectives to achieve on the one hand and let cities aware of the concept of "Smart Cities" on the other hand.

In this regard, and for the large part of the 20th century, the city's smartness was a media's science fiction. But thanks to telematics development and the devices' intelligence, the Smart City "is fast becoming a reality". Furthermore, the use of Information and Communication Technologies (ICTs) is important to cause a higher systems' automation with enabling individuals to monitor, understand, analyze and plan the city. Thus, Smart City is rooted in intelligent infrastructures' creation and ICTs-Human connection, where the city growth must respect these three axes: sustainability; by improving the city/environment relationship and using green economy. Smartness; context aware economy and governance. Inclusiveness, by fostering a high-employment, economy delivering social and territorial cohesion.

Smart City can be defined as the aim to reach all of those objectives using ICT. Furthermore, when we talk about using ICT to improve the functionality of smart objects, it means that we add two features to its normal functioning: The Sensing and the Automation. Devices like wireless sensor, camera, road sensor and GPS are components able to sense and gather information like temperature, location and pollution. Arduino, Raspberry and other embedded systems are the automation components which had proved effectiveness on improving the functioning of an object. The appearance of the "Smart City" collocation was an opportunity for individuals to ask the following questions: "What actually is a Smart City? What is the vision of a Smart City?" in order to find a concept's definition. In addition, regarding to not only the technological and industrial revolutions, but sociological development as well, all cities have looked for a unique definition of this concept, which was almost impossible because of the various interpretations. As a result, they finally end up in front of a considerable number of definitions.

We gathered some definitions from different sources to highlight the non-unified character of the concept:

"A city can be defined as "smart" when investments in human and social capital and modern transport and communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance." 1

"Smart Growth is so many different things. It's not just transportation. It's a mindset toward creating a more holistic community. We've talked about quality of life. And what has been more fundamental to quality of life than physical health?" 2

"A city well performing in a forward-looking way in economy, people, governance, mobility, environment, and living, built on the smart combination of endowments and activities of self-decisive, independent and aware citizens." 3

[1] T. Nam, T.A. Pardo, Conceptualizing Smart City with Dimensions of Technology, People, and Institutions, Proceedings of the 12th Annual Digital Government Research Conference, 2011, pp. 282-291

[2] A. Caragliu, C. D. Bo, K. Kourtit and P. Nijkamp, Performane of the Smart Cities in the North sea basin,http://www.smartcities.info/files/13n%20-n%20Petern%20Nijkampn%20n%20Performancen%20ofn%20Smartn%20Cities.pdf, May 1, 2013

[3] Geller, A. L., Smart growth: a prescription for livable cities, American Journal of Public Health, 93(9), 1410e1415, 2003

Barbara McCann claims that focusing on the quality of life is one of essential Smart City mainstays to guarantee the orientation towards the better lifestyle. The smartness meaning precision varies from one author to another. Giffinger et al considers "Smart" as a mean of a prospective performing considering the development aware, flexible, transformable, synergistic, individual, self-decisive and strategic aspects for smartness achieving.

To make things easier to understand, we can consider that the city is just a system that gathers many systems. Thus, we can define the city as a "System of Systems". To get to Smart City, we just bring the intelligence to systems. The significance of the Smart City is the city that receives many contribution technologies from lots of companies by giving innovated products for the city markets to realize this intelligence, without forgetting innovations developing history evoked from traffic flows to the wireless technologies in order to precise and control parameters and, consequently, the individual choices to build smartness in various sectors.

We can seek for lots of aware, flexible, transformative, synergistic, individual, self-decisive and strategic aspects for smartness achieving. Since city is a group of systems, several researches identified some characteristics that ensure the real city smartness and thus, made from Smart City a more specific term. The EU project rested on a hierarchic structure in intention to present the relation between analyse levels. Thus, each characteristic is defined by some factors as it is highlighted in Figure 1: Characteristics of a Smart City.

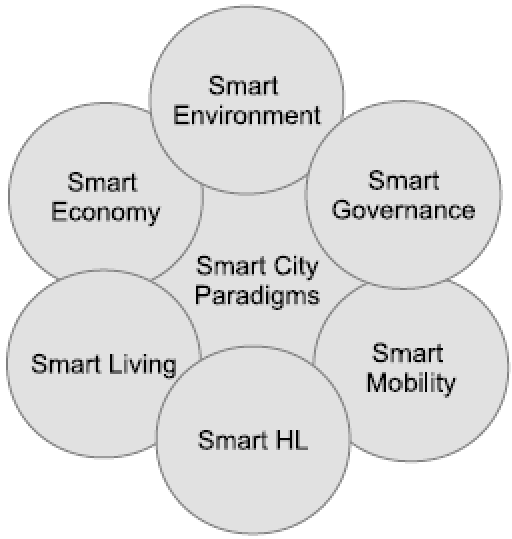


Figure 1: Characteristics of a Smart City

2.1.2. Smart Economy

Considerable researches have revealed the absence of the universal definition of the Smart Economy and described the concept in different ways. What's more, it includes smart companies that produce innovative ideas and improve the price-quality ratio based on the resource optimisation concept. However, this definition did not show all the specific sides of the Smart Economy. For this reason, researchers continue in developing more definitions. According to the researchers, here are common characteristics of the Smart Economy: Innovative: ideas that increase the productivity and reduce cost; Digital: widespread use of ICTs in the economy; Competitive: be open, employ knowledge and innovation to obtain good quality of higher profits, productive resources and efficient costs; Green: focus on sustainable fundamentals, use natural energy resources and recover clean areas; Socially responsible: seek to promote the welfare of individuals.

2.1.3. Smart HL

Cities can’t achieve smartness without creativity, education, knowledge and learning. Consequently, when we talk about Smart City, people dimension is hardly present and consequently the four keys of people axis must be highlighted.

3. Relevant European research projects

3.1. MobiWay Project

Traffic congestions are realities of urban environments, partly because road infrastructure capacities cannot cope anymore with the increase rate in the number of cars. This coupled with traffic incidents, work zones, weather conditions make traffic congestions a major concern for municipalities. Advanced traffic control technologies may lead to more efficient use of existing road network systems, resulting in reduced traffic congestion, delays, local pollutant emissions, energy consumption and improved safety. The communication capabilities provided by modern mobile devices offer opportunities for the development of traffic control applications, where cars and devices within the road infrastructure collaborate to solve traffic problems that directly influence the quality of life. Intelligent Transportation Systems (ITS), such as advanced traveller information systems, travel advisories, and others, attempt to relieve congestion and decrease travel time by assisting drivers on making informed decisions while driving, by selecting better routes, departure times, and even various modes of travel. The main drawback of current ITS platforms is their focused or limited set of solutions, and the inadequacy to support collaborative features. Due to such features it is often difficult or even impossible to introduce a new service from the scratch, since the required quantity of data hinders the quality of the service itself. We already experienced similar issues developing at UPB the Traffic Collector application. This is an experimental ITS application designed to support advanced ITS congestion and pollution control features. Unfortunately, the amount of data required to construct accurate traffic models acted as a barrier to a prototype implementation of the concept on city-level scales.

Hence, together with the BEIA's experience as an ITS integrator, MobiWay proposes the development of a collaborative platform designed to support ITS applications by acting as a middleware connection hub, offering an optimal support to different ITS partners and municipalities through and a data sharing and ITS support service integration platform.

The goal of the Mobility Beyond Individualism: An Integrated Platform for Intelligent Transportation Systems of Tomorrow (MobiWay) project is to provide a platform of ecosystems, which allows engaging communities in exploiting the shared value of mobility and collaborative cultures that can be leveraged beyond the classical views of social networks, respectively the current trends of service creation. The project sets out a new vision for using mobile computing capacities for provisioning and support of ubiquitous connectivity and real-time applications and services for smart cities' needs. Data procured from highly distributed, heterogeneous, real and virtual devices (sensors, actuators, smart devices), in the range of millions, will be automatically managed, analysed and controlled by distributed services. The consortium will develop and demonstrate automatic procurement and management of ITS big data as a real-time cloud-based service in a scalable distributed cloud platform. A demonstration testbed will validate smart city scenarios for (a) business context and (b) societal context.

The project will generate new possibilities of extra value and improvement for already existing ITS solution providers by allowing them to share information and services with each other, besides their standard user base. As an example, daily mobility is better supported by choosing suitable traveling routes, given the data shared between multiple ITS platforms. Also, it brings support for novel service ideas and initial development efforts by boosting up collaboration with mobility users and fostering participation in the ecosystem of services. Informing the user on energy and environmental impacts, and on eventual cost/price impacts to the user and to communities, may also lead to more sustainable decisions regarding the transport mode choice in the direction of a low carbon society.

3.2. FIWARE

Internet is one of history's great success stories and one of which more has influenced the habits of people. Today's Internet was designed, however, in 1970s with the purpose of internetworking. Nowadays, the today's scenarios and the future trends drive towards a new era in the evolution of Internet that is known as Future Internet. The industrialization of IT in form of cloud computing and open service delivery platforms with on-demand pay-per use models of provisioning, new wireless networking technologies (as 4G, 5G, and Fiber to the Home), and the IoT supporting a mass of devices connected everywhere, every time and every way, are just three motivating examples. With the purpose of facing technological, economic, and social challenges of the future Internet, the European Union through its FP7 launched in 2011 an initiative called Future Internet Public-Private Partnership (FI-PPP) to promote an Internet able to sustain the modern societies and address their challenges. FI-PPP program is being approached in three phases that take as technological common core the FIWARE project4. FIWARE aims at providing the foundations for the Future Internet by enabling an innovative infrastructure for cost-effective creation and delivery of versatile digital services, providing high QoS and security guarantees. In the context of IoT FIWARE is concerned with the definition of architecture, implementation of generic enablers associated to things in order to become available, searchable, accessible, and usable resources, and with the definition of IoT use-case scenarios that exploit such architecture.

3.3. SOFIA

SOFIA (Smart Objects for Intelligent Applications) is a project involving eighteen European partners into the search of making information from the physical world available for smart services. One of the key aspects of this project is the search of interoperability among many heterogeneous devices and embedded systems; thus, SOFIA provides a platform to support the interaction among entities and to create innovative services adapted to the user's situation and profile. To this end, SOFIA provides, and Application Development Kit based on Eclipse for easily developing smart applications on top of the SOFIA architecture. 7 cross-domain pilots were implemented in 4 European Countries, 5 of them related to smart cities. After the success of SOFIA, the Spanish company Indra (part of SOFIA consortium) started to develop SOFIA2, which adapts the concepts and the well-practices learned focusing on standardization, innovation aspects, and extending the scope in a driven-enterprise approach (e.g. smart automotive, smart finances, smart insurance). New capabilities of SOFIA2 include the usage of Big Data techniques and the user's interaction through social networks. Another novelty is that SOFIA2 enables to any citizen, company, or developer the access to the data collected to develop their own applications in a basic, freeware modality. Recently, the Cyber-Physical Systems Engineering Labs project announced the election of SOFIA2 as base platform. The objective of the project is to increase the competitiveness in Europe by means of accelerating the knowledge transfer to industry.

3.4. IOT-A

IoT-A 5 deals with the integration and interoperability among the different ways of devices in IoT such as RFID, sensors and actuators (located on the bottom of the architecture) and the different ways of smart applications in IoT. IoT-A proposes the creation of an architectural reference model (ARM) and the definition of an initial set of key building blocks. IoT-A employed a Model-Driven Engineering approach as well as other good practices from software engineering to describe the ARM6. It will be validated in the health, home, and logistics domain.

4 FIWARE. https://www.fiware.org/

5 The Internet of Things Architecture. http://www.iot-a.eu/public

6 Architectural Reference Model for the IoT. http://www.iot-a.eu/arm/d1.2/view

3.5. CASAGRAS

CASAGRAS7 is a project conceived to assist the European Commission in developing a strategy to realize IoT. In CASAGRAS-2 two classes of things are distinguished: smart things with capabilities of processing and communication (e.g. sensor networks) and things that achieve connectivity by means of data carrier technology. According to that classification, CASAGRAS-2 addresses the definition of the interfaces and protocols for these classes of Things as well as proposing identification structures and their coding in heterogeneous data carriers in real-time.

3.6. SUS - Smart Urban Spaces

An ITEA 2 Call 3 project to introduce interoperable e-city services based on the latest mobile technologies and ubiquitous mobile computing techniques.

3.7. NEMO_CODED

An ITEA 2 Call 3 project for Networked Monitoring & Control, Diagnostic for Electrical Distribution.

A simple and powerful idea: using natural resources much more productively - efficiently - is both profitable and better for the environment. Energy efficiency is the quickest, cheapest, cleanest way.

3.8. IMPONET

An ITEA 2 Call 4 project for Intelligent Monitoring of Power Networks. The main purpose of this project was the research, definition, design and development of new generation IT platforms for energy management and network optimization.

3.9. ENERFICIENCY

An ITEA 2 Call 5 project for User Led Energy Efficiency Management. The goal of this project was the creation of a new open platform for industrial plants, large scale buildings and citizens for the optimization of energy consumption and the definition and monitoring of business models for energy savings.

3.10. SEAS

ITEA 2 Call 7 project focused on Smart Energy Aware Systems.

Environmental, economic and sustainability challenges of continuously increasing energy consumption are present all over the world. Meeting the challenges requires cross industry cooperation and the means for consumers to influence their energy consumption in terms of the quantity and type of energy consumed. The SEAS project addressed the problem of inefficient and

7 Coordination and Support Action for Global RFID-related Activities and Standardization, 2012. http://www.iot-casagras.org

unsustainable energy consumption, which is due to a lack of sufficient means to control, monitor, estimate and adapt the energy use of systems versus the dynamic use situations and circumstances influencing the energy use. The objective of the SEAS project was to enable energy, ICT and automation systems to collaborate at consumption sites, and to introduce dynamic and refined ICT-based solutions to control, monitor and estimate energy consumption. An additional aim was to explore business models and solutions that would enable energy market participants to incorporate micro-grid environments and active customers.

3.11. WATER-M

ITEA 2 Call 8 project focused on Unified Intelligent WATER Management.

The scope of the Water-M project enabled the creation of new products and services to build a unified water business model that would benefit European Union water stakeholders. The Water-M project combined real-time monitoring and operational control, service-oriented approaches and event driven mechanisms in the water management domain.

3.12. M2MGrids

ITEA 2 Call 8 project focused on Smart M2M Grids – M2M Internet for dynamic M2M Information Business ecosystem.

The Smart M2M grids project was focused on creating enablers for a dynamic cyber-physical information business ecosystem connecting the physical world with the business processes of companies in real-time. The first goal was to connect physical world sensors, actuators and various embedded devices and machines (physical M2M objects) with IT systems automatically/semi-automatically by applying and extending horizontal open standards based M2M infrastructures for communication and services. The second goal was to enable information management for embedded and distributed application for smart interaction with physical M2M objects and IT back-office systems. The third goal was enabling smart information exchange between selected business cases related to energy, buildings, transportation and consumer M2M products and services to make the future world smart, smooth and secure for consumers/prosumers. The resulting system was aimed at boosting transfer towards a more sustainable society and a novel real-time service economy within selected industrial business cases.

3.13. FUSE-IT

ITEA 2 Call 8 project focused on a Future Unified System for Energy and Information Technology.

Fuse-IT was intended to address the need for sustainable, reliable, user-friendly, efficient, safe and secure Building Management System (BMS) in the context of smart critical sites. A main purpose was to solve the dilemma between efficiency and security in intelligent & strategic buildings. The result of FUSE-IT would be a smart secured building system, incorporating secured share sensors, effectors and devices strongly interconnected through trusted federated energy & information networks, a core building data processing & analysis module, a smart unified building management interface and a full security dashboard. Remote multisite monitoring were implemented, taking advantage of big data analytics.

3.14. C³PO

ITEA 2 Call 8 project for the creation of a Collaborative City Co-design PlatfOrm.

C³PO aimed at providing a Cloud collaborative and semantic platform for city co-design. The C³PO platform was unique in that it covered the whole urban project development process where cities empower, encourage and guide different stakeholders (citizens, decision makers, architects, etc.) to develop an urban project together. C³PO did not intend to replace or modify the existing applications offering unique but partial solutions of city co-design (simulation tool, open API, 3D modelling and visualisation, gaming tool, etc.) but could be seen as an open and generic intermediary that enables the interaction between existing applications through a unique multi-dimensional semantic repository (covering the different types of information in city codesign like GIS, BIM, electricity grids, traffic, etc.). As such, C³PO would enable the capitalization of existing applications and data sources by enabling their integration as services, or by enabling them to exploit the C³PO Open API.

3.15. PS-CRIMSON

ITEA 3 Call 2 project focused on a Public Safety and Crisis Management Service Orchestration.

A key challenge faced by city operators, municipalities and political decision makers is the fragmentation of information into silo oriented closed systems and organization models. This project aims to deliver an integrated 3D digital model and information platform that facilitates information collection, sharing, management, analysis and dissemination from diverse public and private urban infrastructures and resources. The platform supports public authorities to improve quality and efficiency of municipal services. Furthermore, adequate security and authentication methods allow selected urban data sources to be exposed to the full smart city ecosystem, enabling new innovative data-driven applications and services.

3.16. MOS2S

ITEA 3 Call 2 project about Media Orchestration - Sensor to Screen.

The MOS2S project aims to develop and test audiovisual Smart City technologies addressing the needs of its inhabitants and embed these solutions in a dedicated Smart City Playground. This playground provides a venue platform as stepping stone towards a full Smart City Operating System, and the support of proof-of-concepts and trials. As such, the playground has the unique potential to accelerate the creation and market introduction of new unique Smart City applications, based on a range of sensors and datasets, to improve profitability, sustainability, safety and customer experience.

3.17. SPEAR

ITEA 3 Call 3 project focused on Smart Prognosis of Energy with Allocation of Resources.

SPEAR aims to develop a flexible optimization platform that helps to improve a broad spectrum of industrial production processes in terms of energy-related aspects. Hence, a focus within the project is the energy optimization of plants' production processes, production lines and (industrial) buildings. The platform will be used to optimize the energy consumption of existing and new production plants, and the method will be applicable to both virtual commissioning as well as running production systems.

3.18. ProSe

ITEA 3 Call 3 project aimed at the creation of a Proximity Services Framework.

The goal of the ProSe project is the design of a software-intensive system to support the development, deployment and execution of proximity services, which are applications that allow users to intuitively interact with the surrounding IoT enabled environment. Proximity services are automatically deployed when the user is at a specific location. Services are deployed and executed on a generic proximity app without the need for an additional explicit download. Proximity services adapt their behaviour to the surrounding context and interact with the IoT devices in proximity.

3.19. BIMy

ITEA 3 Call 3 project focused on BIM in the City.

Building Information Modelling (BIM) is a digital representation of a construction project that is increasingly used by the Architect, Engineering and Construction industry. The BIMy project aims at providing an open collaborative platform for sharing, storing and filtering BIM among different BIM owners/ users and integrating and visualising them in their built and natural environment. BIMy can be seen as an open, generic and secure intermediary vehicle that enables interactions between existing and new applications through a standardised open API platform.

3.20. POLDER

ITEA 3 Call 4 project focused on Urban Data Policy Lab: POLicy & Data Exploitation & Re-use.

The POLDER project aims to design, develop and deploy a software tool-suite to support government, city councils and related organisations in the elicitation, design, application and validation of policymaking. POLDER proposes a hybrid policymaking model, where policy is made: · Data-driven · Model-driven · Society-driven

3.21. CityStory

ITEA 3 Call 4 focused on Citizen Storytelling.

The project CityStory wants to innovate through a creative, intelligent, safe and social storytelling development environment. Do-it-yourself and do-it-with-others, around media and make it accessible for everyone. The project aims to stimulate collaboration with a co-creation and design platform to share ideas and get opinions heard. Through new modes of interactive storytelling, city touchpoints, interactive screens, innovative media recognition and data analysis, tools that assist while filming and intelligent and deep learning tools, the project will enable ideas to be turned into a story and valuable media output.

4. State of the art in IoT

4.1. Sensors, IoT and Big Data

Urban planning and development smart city applications have major impact area on the life of citizens. This includes the effect on the citizen in terms of health and safety, disaster management, pollution control, and so on so forth. The Big Data generated by the various IoT systems is used to analyze different aspects of smart city.

A smart city can create an efficient and smart services delivery platform for public and municipal workers by installing sensors in the city and to create platforms that allow the share of information and give it for proper use to the public, city managers, businesses and professionals. The platform can have common data warehouse where different sensor system store their information.

In smart cities, a network of sensors, cameras, wireless devices, data centres form the key infrastructure, which allows civic authorities to provide essential services in a faster and more efficient manner. Smart cities are also far more environmentally friendly as they use sustainable materials for building facilities and reduce energy consumption. Efficient use of technology helps create an efficient transport management system, improve healthcare facilities and develop a robust communication network to connect all businesses, people and beyond the relationships between central and sub-national levels of governments[8].

4.1.1. Sensors

One of the aspects of Smart Cities is the optimal use of available resources. Sensors can help make optimal use of resources with connectivity to tell us when and where to save. These sensors can control, detect and manage the unnecessary use and make certain adjustments as per the need[9].

4.1.1.1. Water Management

At present, the major cities waste up to 50% of water due to pipe leakages. With sensors fitted on each pipe, water leaks can be easily detected and corrected before any heavy loss. Besides this, the irrigation systems in public parks can automatically turn off whenever rain is detected to save water.

Some examples of sensors types used in the water management are described below.

- Water level sensors:
  - Floats. Floats work on the simple principle of placing a buoyant object with a specific gravity intermediate between those of the process fluid and the headspace vapour into the water, then attaching a mechanical device to read out its position. Early float level transmitters provided a simulated analog or discrete level measurement using a network of resistors and multiple reed switches, meaning that the transmitter's output changes in discrete steps. Unlike continuous level-measuring devices, they cannot discriminate level values between steps.
  - Hydrostatic Devices. Displacers, bubblers, and differential-pressure transmitters are all hydrostatic measurement devices. Any change in temperature will therefore

[8] Jayavardhana Gubbi, Rajkumar Buyyab, Slaven Marusic, Marimuthu Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions, Future Generation Computer Systems 29 (2013) 1645–1660.

[9] D. Tang, Event detection in sensor networks, School of Engineering and Applied Sciences, The George Washington University, 2009.

cause a shift in the liquid's specific gravity, as will changes in pressure that affect the specific gravity of the vapour over the liquid.

  - Displacers work on Archimedes' principle. The displacer's density is always greater than that of the process fluid and it must extend from the lowest level required to at least the highest level to be measured.
  - A bubbler-type level sensor is used in vessels that operate under atmospheric pressure. A dip tube having its open end near the vessel bottom carries a purge gas (typically air, although an inert gas such as dry nitrogen may be used when there is danger of contamination of or an oxidative reaction with the process fluid) into the water.
  - A differential pressure (DP) level sensor essential measurement is the difference between total pressure at the bottom of the water and the static or head pressure in the vessel.

- Load Cells. A load cell or strain gauge device is essentially a mechanical support member or bracket equipped with one or more sensors that detect small distortions in the support member. To measure level, the load cell must be incorporated into the vessel's support structure. As process fluid fills the vessel, the force on the load cell increases. Knowing the vessel's geometry and the fluid's specific gravity, it is a simple matter to convert the load cell's known output into the fluid level.
- Magnetostrictive Level Transmitters. Instead of mechanical links, magnetostrictive transmitters use the speed of a torsional wave along a wire to find the float and report its position.
- Ultrasonic Level Transmitters. Ultrasonic level sensors measure the distance between the transducer and the surface using the time required for an ultrasound pulse to travel from a transducer to the fluid surface and back (TOF). These sensors use frequencies in the tens of kilohertz range, transit times are ~6 ms/m.
- Laser Level Transmitters. Designed for bulk solids, slurries, and opaque liquids such as dirty sumps, milk, and liquid styrene, lasers operate on a principle very similar to that of ultrasonic level sensors. Instead of using the speed of sound to find the level, however, they use the speed of light.
- Probes thermometers for testing the temperature of water liquids – These tend to be fully waterproof, and are great for dipping in a pond, or measuring water as it flows from a tap.
- Pond Thermometers:
  - Digital pond thermometers can float in the pond and display a digital readout either on the float or some will transmit this via wireless transmission to a base station inside your home. In some cases, an alarm can be set to warn you of predefined temperature is reached.
  - Analogue pond thermometers are designed to be placed into a pond and are easy to read whether in stick or dial format.
- Standard rain gauge consists of a funnel emptying into a graduated cylinder, 2 cm in diameter, which fits inside a larger container which is 20 cm in diameter and 50 cm tall. When measurements are taken, the height of the water in the small graduated cylinder is measured, and the excess overflow in the large container is carefully poured into another graduated cylinder and measured to give the total rainfall.

- Pluviometer of intensities consists of a rotating drum that rotates at constant speed, this drum drags a graduate sheet of cardboard, which has the time at the abscissa while the y axis indicates the height of rainfall in mm of rain.
- Weighing precipitation gauge consists of a storage bin, which is weighed to record the mass. Certain models measure the mass using a pen on a rotating drum, or by using a vibrating wire attached to a data logger.
- Tipping bucket rain gauge consists of a funnel that collects and channels the precipitation into a small seesaw-like container. After a pre-set amount of precipitation falls, the lever tips, dumping the collected water and sending an electrical signal.
- Acoustic rain gauge it is able to sense the sound signatures for each drop size as rain strikes a water surface within the gauge. Since each sound signature is unique, it is possible to invert the underwater sound field to estimate the drop-size distribution within the rain.
- Electronic inclinometer - This enables very precise readings of angles. This type of inclinometer uses an internal gyroscope to measure the direction of gravity's pull. The gyroscope remains in one position, no matter the orientation. A solid object is placed along the gyroscope, and the angle between the gyroscope and the object is determined via the inclinometer and displayed on an electronic readout.
- Mercury inclinometer - Similar in operation to an electronic inclinometer, but instead of the gyroscope, mercury liquid is used.
- Manual Inclinometers - This is an older model of inclinometer also called the gas bubble type. The instrument contains a glass tube with liquid and an air bubble inside. As the instrument moves, the bubble stays level. Its position indicates the incline angle on a scale.
- Gravity inclinometer - consists of a rotating scale and fixed pointers. The scale is driven by an internal pendulum weight. The weight is fastened to the back and suspended from a bearing, which is supported in the front by a diagonal cross-piece. There are two buttons on the instrument that can be pressed so as to lock the scale firmly in place to enable easy reading.

- Pressure transducers:
  - Standard pressure transducer - The simplest form of an electronic pressure measurement system is the pressure sensor. It is the pressure sensor which changes the physical variable “pressure” into a quantity that can be processed electronically.
  - Top Mount transducer - is internal, but the signal amplifier is mounted on top to lower the height of the unit.
  - Side Mount traducer - is internal and is the same as the Standard transducer, but the signal amplifier is mounted to the side of the cap instead of on top to lower the height of the unit.
  - High Temperature traducer – is similar to the High Accuracy (external) pressure transducer but can be heated up to 200o C. The transducer amplifier, in the cable near the pump, cannot be heated to that temperature.
- Humidity Sensors:
  - Sensors based on capacitive effect - Humidity sensors relying on this principle consists of a hygroscopic dielectric material sandwiched between a pair of electrodes forming a small capacitor. Most capacitive sensors use a plastic or polymer as the dielectric material, with a typical dielectric constant ranging from 2

to 15. In absence of moisture, the dielectric constant of the hygroscopic dielectric material and the sensor geometry determine the value of capacitance.
  - Sensors based on Resistive effect - Resistive type humidity sensors pick up changes in the resistance value of the sensor element in response to the change in the humidity. Thick film conductor of precious metals like gold, ruthenium oxide is printed and calcinated in the shape of the comb to form an electrode. Then a polymeric film is applied on the electrode; the film acts as a humidity sensing film due to the existence of movable ions. Change in impedance occurs due to the change in the number of movable ions.
- Multi-Parameter water quality sensors:
  - The Professional Plus multi-parameter water quality meter brings laboratory accuracy and the ease of push-button operation to water quality measurement in the field. The multi-parameter water quality meter can measure thirteen parameters: pH, ORP, temperature, dissolved oxygen, conductivity, total dissolved solids (TDS), salinity, resistivity, specific conductance, ammonium, nitrate, chloride, and barometric pressure.
  - Multi-parameter Water Quality Meter Display Case - A rubber overmolded case and an adjustable right or left-handed strap provide the multi-parameter water quality meter with superior grip and handling. The cable connection between the multi-parameter water quality meter's display case and sensor head is made from Mil-spec connectors providing you with the most reliable sensor connection possible. In-house testing procedures also include 1-meter drop tests from all angles to ensure expected field-durability for the multi-parameter water quality meter.
  - Multi-parameter Water Quality Meter Sensor - meter's rugged four port Quatro cable and sensor housing allows you to choose up to three sensors from the eight available, giving you access to recording thirteen different parameter measurements. Sensors available for the four port Quatro cable assembly include Galvanic DO, Polarographic DO, and any two ISEs among pH, ORP, pH/ORP combination, ammonium, chloride, or nitrate creating an extremely powerful handheld multi-parameter water quality meter.

4.1.1.2. Energy Management

Sensors have also enabled the concept of “Advanced Metering Infrastructure (AMI)” underpinning energy management in cities. Cities are considering use of “Smart Meters” embedded with Phase Measurement Unit (PMU) sensors and communication module which facilitates a two-way communication between the consumer and the supplier. For utility service providers, it helps check meter status prior to sending a repair crew in response to a customer call. These checks prevent needless field crew dispatch to customer sites. For consumers, it can provide the real-time energy usage detail in a way which a user can understand quite easily. Based upon this data, users can change the preferences and make more informed decisions about their usage without waiting for their energy bill at the end of month.

4.1.1.3. Smart Streetlights

In cities street lights remain ON even when there’s no activity in the area (sometime in daylight too!). Additionally, it becomes very difficult for authorities to detect any fault and theft of street lights. With sensors, lights can go dim when they aren’t needed, and authorities can get a text message almost instantly whenever there is a fault or tampering in street lights.

For the smart street lights can be used the sensors: presence sensor, the light sensor and motion sensor.

The presence sensor or PIR sensor has the task of detecting the passage of a vehicle or pedestrian causing the switching on and off of the lamps. This feature permits to activate the lamps when necessary and avoids waste of energy. Light sensor will measure the external light intensity and provides assurance to a minimum level of illumination of the road, as needed by regulations. The sensor should have high sensitivity which is within the visible spectrum. This provides a photocurrent which is high enough for low-light luminance levels.

The motion sensor, the communication device and the controller. It sends out the message to other units under the condition that motion is detected. This unit is placed to many locations, at electric poles, at house gates, at house fence and inside or outside of the door, to ensure that every street light turn on before pedestrians or vehicles notice the lights. As for power supply, the solar battery can be a good option.

4.1.1.4. Waste Management

With sensors fitted in the garbage bins, the municipal authorities can be notified when they are close to being full. The Netherlands became the first ever to produce “Intelligent Bins” that report to the officials via text messages whenever the bins are either full or if there is any damage.

4.1.1.5. Transport Management (Smart Parking)

Traffic can be reduced with sensors that detect where the nearest available parking slot is. Motorists get timely information via text messages, so they can locate a free parking slot quickly, saving time and fuel. A similar project is being carried out at San Francisco called SFPark - where parking spaces have been installed in 8200 on-street places. This concept would be replicated in several other states in coming days.

4.1.1.6. Real-time Pollution Management

Sensors mounted on poles can monitor the Ambient Air Quality (AAQ) of cities. Citizens can monitor the pollution concentration in each street of the city or they can get automatic alarms when the pollution level rises beyond a certain level.

4.1.2. IoT

IoT is now the state of art technology used for many multimedia applications, smart transport systems and smart city design and deployment issues. The smart transport system can be a part of the smart city projection for days to come. This may be because of the nature of the contents involved in applying and developing IoT applications.[10]

The Internet of Things provides interaction among the real/physical and the digital/virtual worlds. The physical entities have digital counterparts and virtual representation and things become context aware and they can sense, communicate, interact, exchange data, information and knowledge. Through the use of intelligent decision-making algorithms in software applications, appropriate rapid responses can be given to physical entity based on the very latest information collected about physical entities and consideration of patterns in the historical data, either for the same entity or for similar entities. These paves new dimension of IoT concept in the domains such as supply chain management, transportation and logistics, aerospace, and automotive, smart environments (homes, buildings, infrastructure), energy, defence, agriculture, retail and more.[11]

IoT has created revolution in smart vehicle technology by introducing wireless standard communication devices. In smart transport system if the vehicle is moving at a speed greater than

---

[10] San Murugesan, The Internet of Things (IoT): Opportunities Abound, IEEE India Info Vol. 8 No. 9 September 2013.

[11] H. El-Sayed, A. Mellouk, L. George, S. Zeadally, Quality of service models for heterogeneous networks: overview and challenges, Annals of Telecommunications 63 (2008) 639–668

20kmlhr. The GPS communication will not be a right solution because the bandwidth of GPRS/GSM is in the range of 64 to 128 kilobytes per second. A good solution strategy to be used is a Wi-Fi communication mode. The key attributes of Wi-Fi namely the speed and flexibility play an important role.

The IoT play a vital role to improve the smartness of cities includes many applications to monitoring of parking spaces availability in the city, monitoring of vibrations and material conditions in buildings and bridges, sound monitoring in sensitive areas of cities, monitoring of vehicles and pedestrian levels, intelligent and weather adaptive lighting in street lights, detection of waste containers levels and trash collections, smart roads, intelligent highways with warning messages and diversions according to climate conditions and unexpected events like accidents or traffic jams. Some of IoT smart cities applications are smart parking, structural health, noise urban maps, traffic congestion, smart lightning, waste management, intelligent transportation systems and smart building. These smart cities IoT applications use RFID, Wireless Sensor Network and Single sensors as IoT elements and the bandwidth of these applications ranges from small to large.

The already developed IoT applications reported on the literature are:

- Effective perception: Any object which wants to be a "thing" in the "Internet of Things" should have effective and efficient sensors network which should be able to comprehensively sense the data where is it placed to do the required task.
- Reliable Transmission: There are variety of available networks like Radio networks, telecommunication networks and internet. The selected network should be so efficient that data sensed by sensor network will be available anytime, so this data has be transmitted when it is available.
- Communication here may be wired or wireless. Smart processing: When the data has been collected into the cloud by using appropriate transmission medium, data has to compute in the cloud. So that the system can take further actions for which the whole system is designed for.

Although the IoT enabling technologies have tremendously increased in the past decade, there are many issues to be open and addressed. Consequently, this paves the new way or dimension for researchers involved in IoT. The issues and challenges of IoT include architecture, privacy and security, data intelligence, Quality of Service, communication protocols, GIS based visualization, etc.

The different architectures proposed already in the literature roughly based on which application domain the IoT used. Most of the works relating to IoT have been classified in to four types of architectures. The wireless sensor networks perspective, European Union projects of SENSEI, Internet of Things Architecture (IoT-A) and cloud architecture. However, these may not be the best option for every application domain particularly for defense where human intelligence is relied upon. The selection of architecture of IoT itself is the big challenge and this paves the way to develop new architecture and modify the existing architecture.[12]

Security will be a major concern, wherever network consists of many devices or things are connected. There are many ways the system could be attacked; disabling the network availability, pushing erroneous data into the network, and accessing personal information. It is impossible to impose proper privacy and security mechanism with current already existing techniques. Therefore, privacy becomes a major concern and need to incorporate appropriate security measures.

There are huge volumes of data will be collected from connected from network of devices. According to a rough estimate, more than 2.5 trillion bytes of new data every day will be logged by these

[12] Demirkol, C. Ersoy, F. Alagoz, MAC protocols for wireless sensor networks: a survey, IEEE Communications Magazine 44 (2006) 115–121.

systems. Analysis of data and its context will play a key role and poses significant challenges. The data collected through IoT devices to be stored and used intelligently for smart IoT applications. These leads to develop artificial intelligence algorithms, and machine learning methods based on evolutionary algorithms, genetic algorithms, neural networks, and other artificial intelligence techniques.

The QoS of IoT applications is measured from the primary factors such as throughput and bandwidth. It is easy to provide QoS guarantees in wireless sensor networks due to resource allocation and management ability constraints in shared wireless media. Quality of Service in Cloud computing is another major research area which will require more and more attention as the data and tools become available on clouds. This leads to develop a controlled, optimal approach to serve different network traffics and better resource allocation and management.

The protocols for communication of things or devices will play a key role in complete realization of IoT applications. The protocols form the backbone for the data tunnel between sensors and the outer world. Many MAC protocols have been proposed for various domains with TDMA, CSMA and FDMA for collision free, low traffic efficiency and collision free but require additional circuitry in nodes respectively. Internet Protocol Version 6 (IPv6) is the latest protocol which vastly increases the number of internet addresses, and the ability to process and analyze huge volumes of data. This IPv6 would be able to communicate with devices attached to virtually all human-made objects because of the extremely large address space (128 bit). Major goals of the transport layer are to guarantee end-to-end reliability and to perform end-to-end congestion control. In this aspect, many protocols may fail to co-operate proper end- to –end reliability.

Iot and M2M Protocols

The IoT and M2M protocols are:

- **Bluetooth:** It is a short-range communication technology, playing a very important role in connecting to the IoT through a smartphone in most of the cases. The Bluetooth Low-Energy (BLE) or the Bluetooth smart is an important protocol for IoT applications. Equipment and gadgets that operate Bluetooth Smart features integrate the Bluetooth Core Specification version 4.0 with a joined basic-data-rate and low-energy core configuration for a RF transceiver. The version 4.2 through its internet protocol support will grant Bluetooth smart sensors to access the internet directly through 6LoWPAN connectivity. The Bluetooth has the frequency of 2.4 GHz, range 50-150m and Data Rates 1Mbps.
- **Z-Wave:** It is one of the most leading wireless home control technology in the market today. Z-Wave is a low-power RF communications technology designed for home automation. It operates under 1GHz band immune to interference from Wi-Fi and other wireless technologies in the 2.4 GHz range. It supports full mesh network not requiring node and the enabling controls up to 232 devices. Z-wave uses a protocol which enables faster and simpler development. It standardizes alliance ZAD12837 / ITU-T G.9959, with the frequency of 900MHz, range 30m and data rates 9.6/40/100Kbit/s.
- **Thread**: Launched in 2014 by the Thread group, is a protocol based on different standards. Thread supports IEEE802.15.4, IPv6 and 6LoWPAN, offering a resilient IP-based solution for the IoT. It is essentially designed to correlate to the Wi-Fi. Its standard is based on IEEE802.15.4 and 6LowPAN, the frequency is 2.4GHz, it does not support range and Data Rates.
- **Wi-Fi:** is everywhere these days, from people's homes to airports, hotels, libraries and every other place where people use their computers or wireless devices (laptops, smartphones and iPads/tablets). The most common Wi-Fi standard which is used in homes and in business is 802.11 in which is hundreds of megabit per second in range but unfortunately too much power-consuming for IoT applications. Its data rates are 600Mbps maximum, normally it is between 150 – 200Mbps which is totally depending on the channel frequency

and the number of antennas, with the range of approximately 50m, frequency 2.4 and 5 GHz bands.

- **Cellular:** Any application of IoT which needs to be operated for a wider distance with the help of GSM/3G/4G cellular communication. The advantage of cellular is that it can send large quantity of data for the 4G, but the consumption of power will be too high for the applications. It supports GSM/GPRS/EDGE (2G), UMTS/HSPA (3G), LTE (4G), with the frequency of 900/1800/1900/2100MHz. Range of 35Km max for the GSM, 200Km max for HSPA. Data range of 35-170kps for the GPRS, 120-384Kbps for the EDGE, 384Kbps-2Mbps for the UMTS, 600Kbps-10Mbps for HSPA and 3-10Mbps for LTE.
- **Sigfox:** is a cellular style system that has been set up to provide low power low data rate, and low cost communications for remote connected devices. It uses the ISM bands, which are free to use with no license required for transferring data over a narrow spectrum to and from the connected object. The basic concept of Sigfox is that for many M2M applications which are running on the small power battery and only requires low level of data transfer, the Wi-Fi range is too short and therefore the cellular is too expensive and consumes a great amount of power. The Sigfox system uses silicon such as the EZRadioPro wireless transceivers from silicon labs, which provides with wireless performance, with the frequency of 900MHz, range of 30-50Km in the rural areas and 3-10km in the urban areas, and the data rates with 10-1000bps.
- **LoRaWAN:** is a low power wide area network, with low cost, mobile, and secure bi-directional communication for internet of things, machine-to-machine, and lastly industrial applications. The LoRaWAN specification describes the LoRaWAN network protocol including MAC layer commands, frame content, security, flexible network frequency management, device EIRP and TX dwell time, power control and relay protection. It is various in frequency, ranging from 2 to 5 km in the urban areas and 15 km in the suburban areas, and consisting of data rate from 0.3 to 50 Kbps.
- **NFC:** NFC stands for Near Field Communication, it permits a short dimension communication between the suitable devices, specifically applications for the smartphones which helps the customers to pay transactions. The process requires at least one transmitting device and another one to receive the signal. It is standardized with ISO/IEC 18000-3, frequency of 13.56MHz (ISM), range of 10cm and Data Rates from 100 to 420 Kbps.
- **Neul:** it plays an important role in contribution to the air-interface, weightless, designed specifically for the IoT/M2M market, and the open-standard on which the technology is based. The advantage of Neul is that it is highly scalable, high coverage, low power and low-cost wireless networks. The main system is dependent on Iceni chip, which communicate through the white space radio to access the high-quality UHF spectrum. Weightless is a communication technology achieving coverage, battery life, module cost and efficiency goals that covers GPRS, 3G, CDMA and LTE WAN solutions. It has the frequency of 900MHz (ISM), 458MHz (UK), 470-790MHz (White Space), with the range of 10Km and data Rates starting from Few bps up to 100kbps.
- **6LoWPAN:** 6LoWPAN is the name of an achieved working group in the internet area of the IETF. The basic idea of this is that the internet protocol can be and should be applied to the smallest devices also and the low power devices with the limited processing capabilities should be able to take part in the IoT. 6LoWPAN is a wireless mesh network with low-power, where every node has its own Ipv6 address, which allows it to connect directly with the internet using open standards. The standard has the ability of frequency band and physical layer and can also be used across multiple communications platforms, such as Ethernet, Wi-Fi, 802.15.4 and sub – 1GHz ISM. The standard is RFC6282, with the frequency

(adapted and used over a variety of other networking media including Bluetooth Smart (2.4GHz) or ZigBee or Low- power RF (sub-1GHz), not supporting any range and Data rate.

- **Zigbee:** it manages to provide high data in the applications where the duty cycle is low and low power consumption is an important consideration, the low-power allows devices to last for years on a single battery, with the Green Power features, you don't even need any batteries. ZigBee is reliable, it harnesses the power of the mesh to connect every product to every other product. So basically, if one of the products fails the other will continue to communicate without the interruption. ZigBee consists of a large mean of operation. The latest version of ZigBee is 3.0, which is necessarily the transformation of the various ZigBee wireless standards into a single standard. ZigBee standardizes with 3.0 based on IEEE802.15.4, with the frequency of 2.4GHz, range from 10 to 100m and data rates up to 250kbps.
- **MQTT:** stands for Message Queuing Telemetry Transport MQTT is a machine-to-machine IoT connectivity protocol it is a simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency and inaccurate networks. It is basically used in remote locations with the small code footprint or the network bandwidth. MQTT allows devices to send information about a liable topic to a server that behaves as an MQTT message broker. MQTT is a good option for the wireless networks that involve levels of latency due to infrequently bandwidth constraints or inaccurate connections. The MQTT session is distributed into four different stages connection, authentication, communication and termination. A client starts by creating a TCP/IP connection to the broker by using a standard port or a custom port from the broker's operators. The ports are set to standard which are 1883 for the non-encrypted communication and 8883 for encrypted communication using SSL/TLS.
- **XMPP:** Extensible Messaging and Presence Protocol is a set of open technologies for instant messaging, presence, multi-party chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data. XMPP is a set of classified that permits systems to communicate to each other. XMPP is used universally across the web but is usually underplayed. The presence indicator of the XMPP informs the server about being online, offline or busy. Technically speaking the presence resolves the chances of an XMPP entity. The messaging part of XMPP, the instant messages sent between the clients. The XMPP is constructed in such a way that all the messages are send in real-time using a very efficient push mechanism. XMPP is extensible open standard for using the system in the development and application. XMPP applications includes network management, content syndication, collaboration tools, file sharing, gaming, remote systems monitoring, web services, lightweight middleware, cloud computing, and many more. An immense number of companies and open source projects use XMPP to make the real-time applications and the services.
- **CoAP:** The Constrained Application Protocol is a specified Protocol; its function is to web transfer with the constrained nodes and networks in the IoT. The CoAP Protocol is specially designed for machine-to-machine applications, for example smart energy and building automation. CoAP consists of mechanism such as SMS on mobile communication networks. The main function of CoAP is that it easily translates to HTTP for simplified integration with the help of the web, along with the facility of multicast support, low overhead and simplicity. Current web technologies do not consider memory, energy and computation constraints of embedded devices. CoAP implements request/response interaction model as in WWW-HTTP. CoAP is based on UDP, it supports asynchronous messages, low overheads, URL and content type support and provides simple proxy and caching possibilities. TCP complexities are reduced by the use of UDP. The request methods include GET, POST,

PUT and DELETE. Its features include confirmable message type, non-confirmable, acknowledgement and reset. It supports unicast and multicast requests.

- **NB-IoT:** targets on indoor coverage, which is low cost and long battery lasting, which helps to operate a large number of devices. NB-IoT technologies are making its place globally as in the industry of wireless in the connections. NB-IoT is the pathway for the low data rate devices which could be connected to the mobile networks easily. NB-IoT has been architecture with a number of distributions for the GSM, WCDMA, or LTE spectrum. With a carrier bandwidth of 200 kHz which is equal to a GSM carrier, an NB-IoT carrier can be distributed between LTE carrier guard bands. This concerns on the use of such bands in LTE, to deal in a guard band by not creating any interaction. The NB-IoT layers are designed according to the requirements of in-LTE-guardband, where the design of NB-IoT has accepted LTE, using 15Hz subcarriers in the up and downlink. The NB-IoT carrier network substance uses a single block for the traffic. NB-IoT uses two protocol called RRC_idle and RRC_connected. In the RRC_idle protocol, the equipment saves power so the resources that are used to send the required measurements reports the uplink reference signals to be free. For RRC_connected protocol, it can send and receive direct data. For RRC_connected the application of DRX reduces the number of measurement reports equipment send the downlink control channels are monitored, by saving the battery timing. In different places, NB-IoT devices are limited by signal strength other than transmission bandwidth. This equipment can control the transmission with the narrow bandwidth without any loss of energy. To allow such narrow bandwidth the NB-IoT uses tons or subcarrier the NB-IoT is made with multiplexing data rates which can be adapted so it is able to meet the required capacity. NB-IOT is the GPP radio-access technology developed to meet the connection.
- **OneM2M**: a global action to prevent the most productive deployment of M2M communications systems and for IoT. It provides functions that M2M applications across different industry segments commonly need. The main purpose of oneM2M is to improve the technical specifications for the services to be embedded taking in consideration about the hardware and software to help to connect the devices in any part of the world with the M2M application server. The original design of data in request and response messages basically depends on the protocol binding. OneM2M entries are located and are recognized by using M2M identifiers, though the M2M identifier is a protocol independent. When the oneM2M item is communicating to another oneM2M item, the address seen on the oneM2M primitive from the parameter. Binding oneM2M protocol elementary types to HTTP method, binding oneM2M response status codes if successful or unsuccessful to HTTP response codes. To succeed the communication through the MQTT protocol, a request has to be sent where the client is able to send the serialized request, and after that, the client will recover the serialized response. Therefore, the information regarding the verification, the operation, the response status and the rest of the boundaries will be placed in the serialized request or the response. OneM2M provides applications logic for the end-to-end M2M solutions. OneM2M condition gives a framework to help applications and services such as the smart grid, home automation, connected car, public safety, and health. OneM2M movement inspires industry associations and forums with specific application conditions to participate in oneM2M.

4.1.3. Big Data

The term ‘Big Data’ in itself is a poor definition of its representation it often only conveys the idea of a large volume of data too large to be handled by current processing power of computers. However, Big Data does concern the large volume of data but also includes the capacity to search, process, analyze and present valuable information coming from large, diverse and rapidly changing datasets. This leads to Big Data being defined by volume, variety and velocity. Veracity is another characteristic of Big Data which is growing in popularity and discusses the rising issue of certainty involved with using data.

Data quality dimensions are the ways to express the notion of data quality. The major investigation concerning dimension for this paper concerns whether the same dimensions which have been applied for traditional data quality strategies such as completeness, relevancy, consistency amongst others, could also be relevant for Big Data quality strategies? Or could there be new dimensions only applicable in the context of Big Data?

In order to quantify dimensions so as to bring some notion of measurability and comparison to the dimensions, metrics need to be applied. Thus, there is the need to investigate whether the same metrics which have been traditionally applied to measure data quality dimensions could still be applicable for dimensions determined applicable to Big Data, and whether existing metrics and their formulas would need to be upgraded.

The execution part of a data quality strategy would consist of several activities whose common goal is to improve the quality of data based upon the dimensions identified in a particular context. Some of the most cited data quality activities are:

- Data profiling: examination of data sources to generate information about the data and the datasets.
- Data cleansing: consists of techniques whose main purpose is to enhance or transform data based upon certain business rules.
- Construction of data quality rules (DQRs): the rules which would be used to determine whether there are issues with a given datasets and which would be used for the cleansing activity.

The high volume and velocity properties of Big Data entails that segregating correct and erroneous data for further data analysis is more important. Also, due to data coming from multiple sources, there is a need of higher method of data integration to harmonize the semantics of the data being used. However, the importance of improving data quality for Big Data might not be so important as the amount of incorrect data is deemed to be negligible to affect the outcome after data have been analysed. Consequently, which of those two completely contrasting schools of thought is relevant seems to depend upon the amount and impact of the erroneous or ‘dirty’ data as part of a big dataset. This increases the importance of understanding which dimensions are more relevant for Big Data.

Profiling of data is basically about examining data available in a given data source and collecting/producing statistics and information such as metadata, relationships, dependencies, patterns and cardinalities. The results out of a data profiling job usually proves to be very useful as they contribute towards creation of constraints and rules which could be applied during data cleansing. The traditional use cases of profiling include query optimization, data cleansing, data integration, scientific data management and data analytics; out of those, data cleansing and data analytics seem to be the more relevant when we consider data quality for Big Data.

Big Data is reported to raise three main impacts/challenges when it comes to data profiling:

- Profiling could be very useful in assessing usefulness of known and unknown data; this could help deployment of future Big Data use cases.
- High variety and volume of data challenges existing data profiling tools and techniques in terms of computational complexity and memory requirements amongst others.
- New data management and architectures are involved with Big Data, such key-value and document-based stores, high usage of parallel and distributed systems and so on. Thus, there is the need to re-think how data profiling needs to be carried out in the context of Big Data.

The following chart maps the upcoming research challenges associable between Big Data and data profiling.

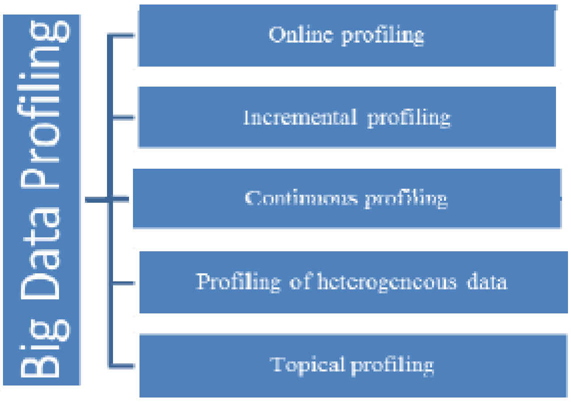


Figure 2: Research challenges between Big Data and profiling

Online profiling is triggered by the fact that users might need to wait a substantial amount of time before viewing the results if profiling is handled by traditional data profiling methods; hence, the idea of displaying intermediate results with proven level of confidence might help the user to take a decision whether they would want to continue to work for a specific use case. Given the huge amount for Big Data, this would certainly be an appreciable feature for potential Big Data users such as data analysts and data scientists. Furthermore, it is imperative to understand that profiling would be an activity to be performed both before and after data cleansing as after the cleansing process, the information about the data and the data source would be updated.

Incremental profiling is linked with the velocity characteristic of Big Data. As data is expected to be updated quite often, there should means to perform profiling upon changing data with periodic timespans. Re-using past profiling results to improve the computational response times of profiling might be an avenue to explore!

Continuous profiling is almost the same idea as incremental profiling with the exception that we are now expecting profiling to be performed upon data while it is being created or updated. One of the aim to profile Big Data would be to determine the common properties or level of heterogeneity of different datasets. Thus, two types of heterogeneity is of particular importance when profiling for Big Data, namely syntactic heterogeneity, which is mainly about finding inconsistent formatting between data and structural heterogeneity, which is about unmatched schemata. Consequently, the information generated by profiling would be very useful in the integration aspect of different sources of a Big Data use case. This is an open area of research as existing solutions simply appears to be ineffective for Big Data.

Recognizing the domain of yet unknown data is the aim of topical profiling. With Big Data use cases, there could be the use of unknown datasets such as social media data. There should be some thorough research upon how topical profiling could be implemented efficiently for Big Data.

Issues and Challenges in Big Data Management are:

- **Security can be of a major concern**, as Big Data being a new technology; it may not be understood well by all companies. Since most of the data in the data sets are important, securing the data from security breaches is on the top priority. Consider, for example, data have gleaned from a location-based service. This service requires a user to share his/her current location with the service provider. If a security breach is encountered, it may cause serious privacy concerns as the user’s location is compromised. Hiding the user’s identity is not the only concern, as the user’s identity can be determined by his/her several trails of locations. Employing techniques like encryption and logging is necessary to make big data secure.

- **Heterogeneity and Incompleteness**. Unlike human beings, machines understand only homogeneous data. A machine is not able to comprehend nuances like a human being. Hence, carefully structured data is imperative for efficient and accurate data analysis. Incomplete data can also lead to incorrect data analysis. Consider a health record database that records birth date, occupation and blood group for each patient. Many a time's, patients do not provide all the information and hence the value of such data is set to Null. An analysis that classifies patients as per occupation must take into account the patients, for whom the information is not known. The unknown dataset cannot be ignored. It may be assumed that the unknown values are statistically similar with the known values. However, this will lead to a skewed result.
- **Scale.** As data volumes are scaling faster than compute resources and CPU speeds are static due to the power constraints, innovative method for big data handling is required. First, over the last five years the processor clock speed has largely stalled, and processors are being built with increasing number of cores. In the past, large data processing systems had to worry about parallelism across nodes in a cluster; whereas one has to deal with parallelism within a single node. Parallel data processing techniques applied for inter-node parallelism are different than that is required for intra-node parallelism, since the architecture looks very different. These unprecedented changes require us to rethink how we design, build and operate data processing components. Second, the move towards cloud computing, which now aggregates multiple disparate workloads with varying performance goals into very large clusters. This level of sharing of resources on expensive and large clusters requires new ways of determining how to run and execute data processing jobs so that we can meet the goals of each workload, cost-effectively, and to deal with system failures, which occur more frequently as we operate on larger and larger clusters. Third, is the change of the traditional I/O subsystem. Hard Disk drives (HDDs) are being replaced by Solid State Drives and other technologies such as Phase Change Memory. The implications of this changing storage subsystem potentially touch every aspect of data processing, including query processing algorithms, query scheduling, database design, concurrency control methods and recovery methods.
- **Timeliness refers to data being available much faster in real time**. The digitization of virtually "everything" now creates new types of large and real-time data across a broad range of industries, so huge amounts of data need to be processed and it takes longer time to analyze. We need a system design that will take care of all these parameters. In this fast moving world, we need result of the analysis in a fraction of seconds. For example, if a fraudulent credit card transaction is suspected, it should ideally be flagged before the transaction is completed, potentially preventing the transaction from taking place at all. Full analysis of a user's purchase history is not likely to be feasible in real-time. Rather, we need to develop partial results in advance so that a small amount of incremental computation with new data can be used to arrive at a quick determination.

4.2. Data Visualization Frameworks

4.2.1. Introduction

This section describes the SoTA in the representation and display of critical information. Information and data represented in the best and fastest way to capture the message. The visualization of data is the process of searching, interpreting, contrasting and comparing data that allows an in-depth knowledge and detail of them in such a way that they become information comprehensible for the user.

The right way to display these views is one that makes up a collection of assembled resources to create a single unified visual display (dashboard).

4.2.2. Data Visualization Selection

In the large majority of dashboard projects, we've found that Key Performance Indicators (KPIs) fall into a handful of core categories, which have specific implications for visual analytics. As such, we've mapped visualization guidelines and tips to the different varieties of KPIs, along with examples.

While this section starts out with the basics, a further analysis should be done into each one as each view is considered valid for the panel in question.

4.2.2.1. Quantities

One may think of these as counts or absolute measures, such as the number of units sold, time taken to complete an examination or tons of freight carried by a train car.

There are a wide variety of options for visualizing basic quantities. Bar or column charts are a good place to start, where each bar represents a quantity of something (like units sold) for a specific grouping (like product lines). If there are a limited number of groupings, these types of charts provide a visually intuitive way to compare quantities, as represented by length or height.

Line graphs are also a good choice if the quantities are tracked over time.

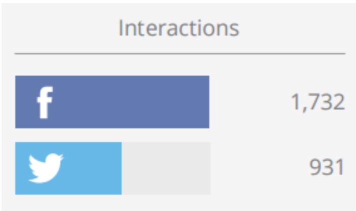


Figure 3: Simple bar charts are great if the user needs to rapidly compare different quantities.

Otherwise, the interest is focused in the direct relationship between two different quantities and some type of scatter plot visualization would be probably needed. When a substantial correlation between the variables exists, a linear pattern will form on the plot. If identifying extreme data points or outliers that don't conform to a normal relationship is important, this could also be a useful way to visualize data for the users.

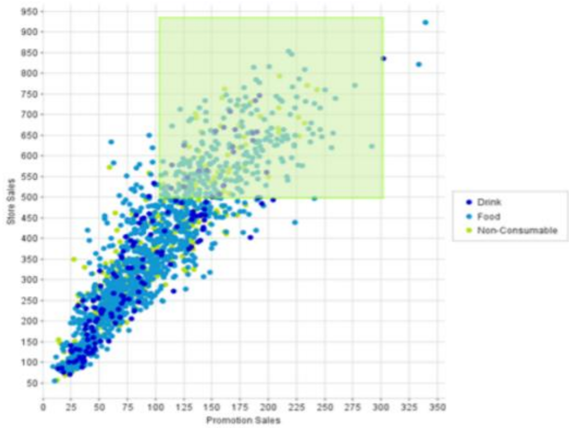


Figure 4: Scatter plots help show correlation between variables – this example shows the relationship between total store sales (vertical) and sales from a specific promotion (horizontal). The colours represent different product types

4.2.2.2. Trends & Changes over time

Time series data is a deep analytics topic, but the crucial point for most visuals it that it adds a new dimension to the analysis, normally where time is displayed on the horizontal axis of your chart. The more lines or series you add to a line graph, the more difficult it becomes to understand and interpret at a glance.

Figure 5: Basic time series graph plotting sales and gross profit over a month.

These types of charts give users an intuitive feel for changes over time - i.e. growth rates and decline rates. When looking at rates of change for a specific measure over time, it can be useful to simply indicate the rate with clear text near the appropriate series on the chart. Showing a directional arrow icon (up, down, or flat) may also provide the user with a clear sense of the change rate.

Figure 6: Displaying a growth rate and icon next to a KPI - in this case entrants in a race

4.2.2.3. Shares and proportions

Shares and proportions display a relationship between the parts and the whole, rather than differences over time. Examples here are shares of an investment portfolio allocated to stocks vs. bonds as well as website conversion rates - i.e. the portion of people who opened an email, clicked it, or didn't open it. If the primary interest is to display the composition of an aggregate quantity at one point in time - say the breakdown of the investment portfolio by asset class today - then a pie chart would make sense, with slices representing different portions that add to 100%.

Figure 7: Basic pie chart showing proportions of medical patients who spent time in different types of hospital rooms

Pie charts on their own, however, do not convey a tremendous amount of information. If we wanted to show proportions over time, we would use a stacked bar or area chart that can be configured to display total quantities over time broken down by categories in either absolute or percentage terms.

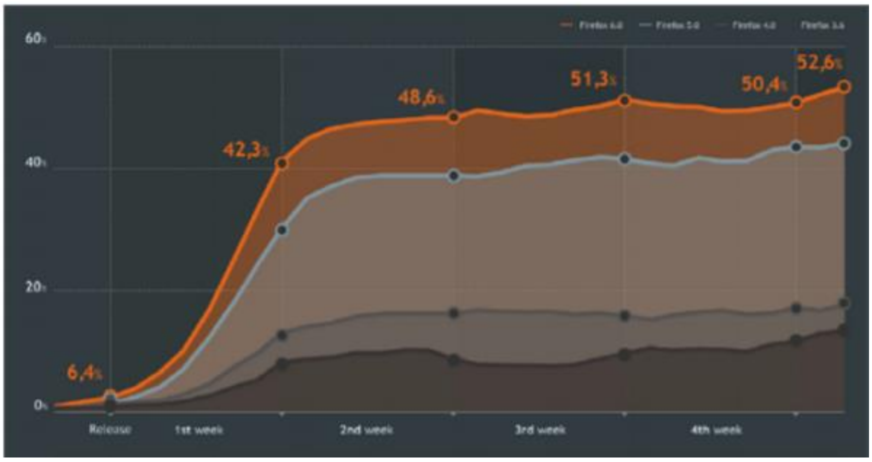


Figure 8: Area chart comparing the uptake over time for different software versions following their initial release

A unique alternative for showing proportions could also be a radar chart (also known as a spider or web chart). In this visual, the proximity of the data point to a corner indicates the relative importance of that corner, while several 'webs' can be overlaid to illustrate relative results for different periods or other groupings. Indeed each 'web' creates a different shape representing a different scenario to the viewer.

Figure 9: Radar chart captures the relative importance of different customer churn drivers for a telecom company both in 2013 and in 2012 in a quickly consumable format.

4.2.2.4. Ranked lists

Presenting a ranked list of the highest or lowest items from a data set is a common dashboard approach. Although is not really data visualization, this is a good way to provide the 'need to know' information, assuming there is a way for the user to access greater detail. For instance, a sales management dashboard may include quota attainment for the 5 best and worst performing sales representatives on the team, with the ability to click through to a view that shows all sales representatives with a wider array of metrics for each representative.

RANKING	TRACK	NR STREAMS
1	Hurry Up We're Dreaming	5,236
2	Small Things	4,563
3	Dandelion	4,022
4	Float	3,785
5	The Last	3,102
6	We're Dreaming	4,563
7	We Stay Together	2,956
8	One	2,854
9	Fennesz01	2,820
10	Passed Me By	2,310

Figure 10: Ranked list of top 10 music tracks by number of streams over a time period.

4.2.2.5. Geography & location

As you might expect, in this type of visual you are looking at data on a map or physical representation of a real location, which gives the user a richer information experience based on known spatial and physical relationships, a picture is worth a thousand words, as the saying goes.

These locations and their spatial relationships lead to a deeper understanding of behavior and influences. Since a high percentage of data already have geographic information attached to it, knowledge about these relationships is easily available.

Location intelligence now allows to incorporate external data from a variety of sources that can be dynamically combined and updated. Examples of maps with this type of information are heat maps, density maps, thematic or categorized maps and maps with charts (all of them located in one location).

Figure 11: Map of bubbles, each one is in a city, and the volume represents a measure.

4.2.2.6. Dealing with multiple measurements at once

Usually some of the richest visualizations display more than one type of quantity and one type of grouping in the same space. However, as the variety of information you are trying to communicate increases, so does the risk that your users will miss the point. Two types of multi-dimensional visualizations will be described: bubble-scatter charts and heat grids.

In a bubble-scatter visualization, the scatter plot discussed earlier is augmented with one or more additional dimensions. First would be the size of the data point (hence the ‘bubble’) – where the area of the bubble represents a numeric value associated with that point. This would be followed by color of the data point, to represent another factor.

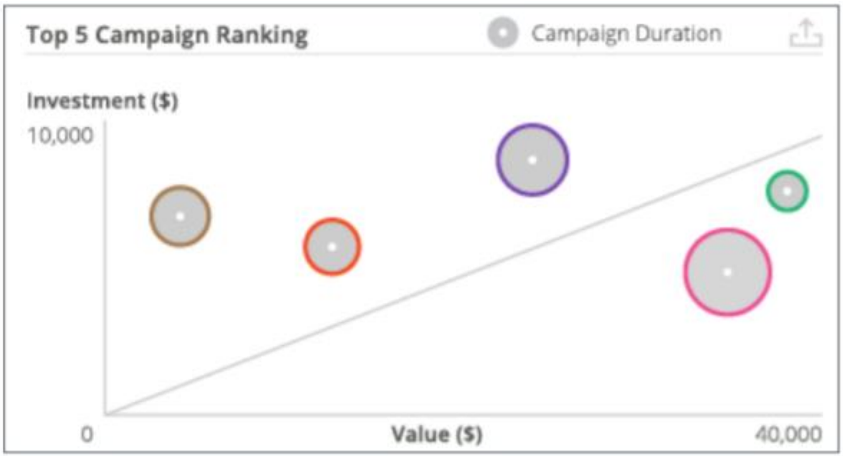


Figure 12: Bubble-scatter chart showing marketing campaigns by value generated (horizontal) and invested cost (vertical), as well as duration the campaign ran for (bubble size).

In a heat grid, we can leverage color to convey values or 'temperatures' of data points on a horizontal and vertical access. This type of visual is more appropriate where we expect multiple Y axis values for every X axis value, and thus makes sense for representing values sliced and diced by different general or neutral categories. Put another way, a heat grid can help where a stacked bar chart might fall short - rather than trying to display values by size stacked on one another, the size of the data points can be left constant with the colors representing the value scale. That said, the size of a data point could be added in as another dimension in some situations.

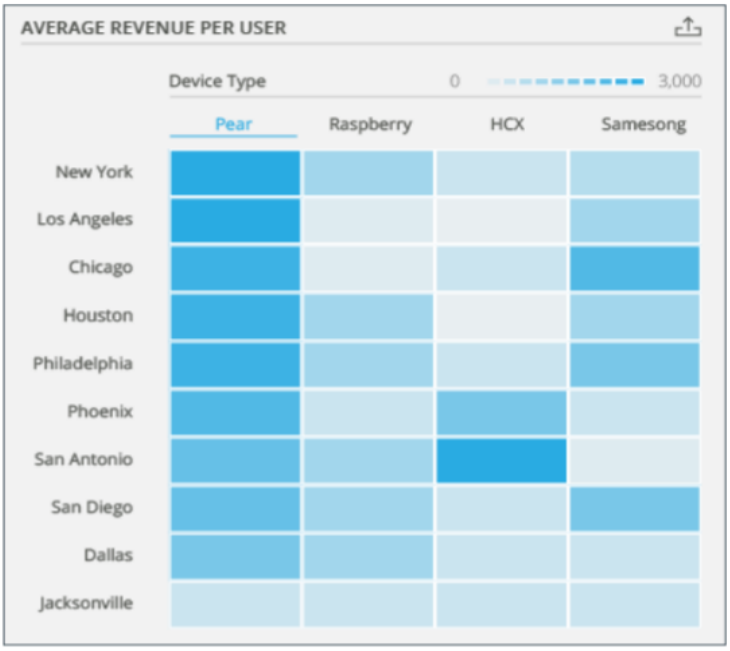


Figure 13: Heat grid displaying average revenue per user (colour scheme) across customers, by both geographic market (vertical) and type of mobile device used (horizontal).

4.2.3. Existing Main Frameworks

This work list describes some of the existing main frameworks for building dashboards and data visualization. Main criteria to choose among them are:

- The quality of the solution, which includes usability, learning and dedication time to be able to master it, the options offered, number of graphics, possibilities and the result of visualization.
- Support, community and number of users.
- Priority to free and open source solutions.
- Regarding the purposes, it should be oriented to the web and have attractive, dynamic and interactive graphics.

Therefore, for each solution the more it fits the criteria, the more it will be described. So, some of them, as the pure graphical JavaScript libraries, will only be briefly described, while BI subscribe solutions or SaaS solutions and others will be widely described and compared to each other.

4.2.3.1. JavaScript Libraries to Create Charts

The following libraries are benchmark when graphing drawings and SVG objects (scalable vector graphics), but they are only going to be mentioned and summarized because most dashboard building frames already use these libraries for graphics. Therefore, they solve much of the work that would have to be done if "native" graphics library would be used. Among the best known and used are:

RShiny

Designed to create analysis output graphs made with R. It is very useful if you have used this language to perform the process of analysis and extraction of data information.

D3.js

D3js.org is an open source JavaScript library for data visualization. It allows you to link arbitrary data to a data document model, and then apply data-based transformations to the document. Its emphasis on web standards offers all the capabilities of modern browsers without tying itself to a proprietary framework, while combining powerful visualization components and a DOM-driven data manipulation approach (Document Object Model). It supports large sets of data and dynamic behaviors of interaction and animation.

This framework follows the imperative paradigm. It offers application help methods that you can use to write code that visualizes your data step by step.

For example, to create a bar chart with D3.js you need to initialize the canvas, calculate where to draw the axis, draw the axis, calculate where to draw the columns, draw columns, legend, point data, add events, etc.

D3 has become a standard for data visualization. There are also libraries built on D3 (dimples, NVD3, xCharts, Rickshaw, C3.js ...) for people looking for top-level APIs.

Chart.js

Chart.js is a small open source library that supports only six types of graphics: line, bar, radar, polar, pie and donut. But the reason why it is good is because it is sometimes all you need for a project. If the application is large and complex, then libraries like Google Charts and Fusion Charts makes sense, while for small hobby projects Chart.js is the perfect solution.

Chart.js is perfect for small projects - when you need flat, clean, elegant, fast JavaScript elements. It is a small open source library taking only 11kb when it is minified and compressed. This includes

6 main types of graphics (line, bar, radar, polar, pie and donut), each in its own module, so you can only load only what the project needs, making its size even smaller.

It uses the HTML5 canvas element to represent graphics and polyfills support to work on IE7 / 8. All graphics are responsive.

4.2.3.2. Frameworks to Create Dashboards

Freeboard

It is a free and open source software for dashboard construction. It is not a complete BI solution, but it is easy to integrate with multiple data sources prepared in production, and with an elegant design. You can hire a hosting subscription or install it in your own local environment. The weather data display of the Open Weather Map is already loaded, and it has graphs ready to be fed by JSON data provided by URLs. It is configured to retrieve data from IoT platforms such as Citrix, BuildingLink, Orion, PubNub and Xively. For each data source parameters and data types are specified, as well as update rates. It solves the CORS problem (Cross-origin resource sharing / Cross-domain shared resources).

Additional subscription services: Paid Freeboard subscriptions include free subscription features and provide protected dashboards with private password. You can get 5 different private boards for $12 a month, 10 for $22 / month, 20 for $42, and 50 for $100 a month. The price of the business subscription must be requested.

- Advantages:
  - It has an intuitive, orderly, well designed and highly flexible web box.
  - Very easy to use and practically no programming skills are needed.
  - "Drag and drop" widgets.
- Disadvantages
  - The documentation and error reporting for Freeboard is practically non-existent, so things like incompatibility problems of a data source will not be marked and it will be difficult to find the error and debug a data source that it does not work.

3DS Netvibes

Dassault Systèmes [13] has advanced an integrated solution that expands the growth and helps increase your margin, making sure you develop a successful product that the consumer would love. The 3Dexperience comes as a multi-disciplinary social association baseline. Dassault Systèmes solution experiences help the technical departments, personnel for better communicate and monitor and to observe issues early in the development process to optimize product quality. It can be prolonged with disciplinary kit helping educational, research, and organizational processes in designing and engineering, system engineering, in manufacturing the products. The 3D experience supports an environment to expand digital labs involving originally behaving virtual 3D devices directionally coupled with other remote devices. With the help of IoT within the diffuse learners, educators, devices, and content, it is a platform for new horizons for innovations educational practices, for instance, distant education and flipped learning. Netvibes Dashboard Intelligence[14] enables enterprises to listen, learn and act on all the information that matters to their business. Furthermore, it aggregates content from across the social web alongside enterprise data, analyze business metrics in their social context and automate alerts and actions to drive faster decision-making, 24/7/365.

[13] www.3ds.com
[14] www.netvibes.com

4.2.3.3. Complete BI SaaS Solutions

Such a solution takes care of both, all regarding preparing the data and data visualization tools. Preparing the data includes data extraction from different sources like files, operational data bases, services, etc.., prepare and clean it and finally construct the processes for defining the metrics, KPI's and visualizations. In this section are described the three more popular and used BI software, two non-open source solutions, QlikView and Tableau, both allow you to try their product for free during a trial. If you find one you like and decide to use them for a commercial use, it is expected to pay the big bucks. And one open source platform focused in location intelligence, CARTO.

Tableau

Tableau is one of the fastest evolving business intelligence and data visualization tool. It is very fast to deploy, easy to learn and very intuitive to use Tableau has a nice user interface and a clean dashboard, which—after you've mastered the learning curve—makes for a good user experience. The simple data drag-and-drop visualization system is the highlight of this software, and many users claim it as the best data visualization tool on the market. These advanced visualization tools allow you to see dots on a map or interesting graphs instead of basic tables. These tools are primarily used in mapping and viewing current trends. Other advantages of Tableau are:

- Can be integrated with R.
- Based on training videos, blog/forum posts, and twitter buzzes Tableau certainly leads the community building effort.

Drawbacks: In order to pull the data, you need out of your system, you'll have to Extract, Transform and Load (ETL). This process takes time or puts you in line behind coworkers who have requested other reports to be built. Also, you can still integrate with many data sources, but the list is shorter than that others.

A private license on Tableau ranges from $999 to $1,999 per user, and gets more expensive depending on server and data access. There is a free desktop version, but keep in mind that the data you use will be made public.

Good reasons to look at Tableau if:

- Enticed by really easy, user friendly drag-and-drop visualization formats.
- Searching for a scalable software solution to use within your organization.
- In need of software that provides excellent data visualization.

Qlik

Qlik is a business intelligence data discovery product that is used to create guided analytics applications as well as dashboards designed for business challenges. You can use Qlik Associative Data Indexing Engine to uncover data insights and relationships across various sources.

Qlik exposes data that is not revealed with query-based tools. The tool also offers guided exploration and discovery and collaborative analytics for sharing insights. Plus, the program enables users to create and deploy analytic apps without requiring technical skills. This helps drive quicker response to changing business requirements, shorter time to value, and more insight across a company.

Qlik is more of an all-encompassing dashboard application, while Tableau is more focused on visualization and analytics. It focuses on a variety of business intelligence tools, like Qlik Expressor (a quick and smart metadata intelligence solution) and NPrinting (an application for report generation, scheduling, and distribution). These tools have helped brand Qlik as a powerful reporting solution. Because of its many facets, it is a good solution for enterprise-level companies who can utilize different features across different departments. Other positive features include good third-party integration, advanced data filtering options, and data manipulation.

Prizing
- Qlik Personal Edition – offered free with unlimited access
- Each private user on Qlik is $1,350 and concurrent users are $15,000. A server license is $35,000. Other services are available at an additional cost.

The advantages of Qlik are:
- Part of a large organization that can use its many tools in different departments.
- Looking for a variety of business intelligence tools––OEM versions, for example––other than visualization.
- Dealing huge sets of data and need powerful, multifaceted software.

CARTO

CARTO is a software as a service (SaaS) platform that serves geo-mapping tools to help businesses and designers create web visualizations focused on location. It provides visualizations and dashboards for a comprehensive data analysis tool, an easy-to-use solution that leverages open data and filters, enhancements and analysis to gain deeper insight into BI analysis.

Regarding to the Smart City applications, it is necessary to consider that much of the data and information to be visualized will refer to geographical points and geographic regions, such as neighborhoods, districts, streets, point locations, etc. In this field, we can conclude that the CARTO platform should be considered a highly adoptable solution to this case.

CARTO Builder is the component that is used to create dashboards and visualizations. It is a web-based "drag and drop" analysis tool that comes with a variety of public platform datasheets. It is different from its previous versions because of two features: widgets and prediction functions. None of these two versions require programming knowledge to be used.

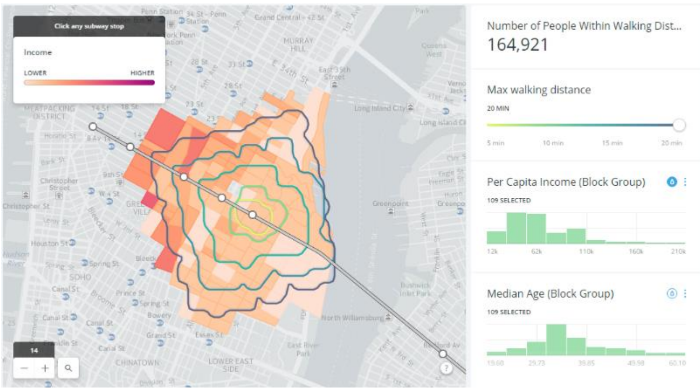


Figure 14: Dashboard built with CARTO

Examples of indicators and variables with localization monitored in a city are:
- Traffic, public transport and mobility in general.
- Contamination and environmental parameters.
- Demography.
- Tourism.
- Others...

Every metric and index can be aggregated in space, like districts and neighborhoods as well as representable on a map.

- Features:
  - Open code.

- Flexible interface, very simple and at the same time advanced, when needed.
- It offers spatial analysis tools such as the ones available in the standard GIS software: overlays, clusters calculations, calculation of service areas by time, proximity, etc.
- No programming experience is needed to create very complete maps or to understand any of the many manuals available.
- For advanced users with programming experience, it offers powerful capabilities through SQL queries and possibility to apply styles to the dashboard using Turbo-Carto, an open-source CartoCSS preprocessor (pseudo language of cascading web styles), which allows the evaluation of functions in asynchronous mode. Its potential for creativity and data integration is very great.
- In addition to importing numerous file formats, it includes connectors to files hosted in the cloud on platforms like Google Drive and Dropbox. You can also import data from different databases using connectors through its import API.
- It is allowed to synchronize the CARTO tables with the data, either automatically with a maximum frequency of 5 minutes for premium users and 15 for the rest, or manually with ogr2ogr commands. CARTO checks changes in the data, and if so, it proceeds to truncate its table and fill it again completely.

- Disadvantages:
  - CARTO platform offers limited services for the non-commercial version and does not offer privacy for the data in this case. The functionalities can be improved by switching to premium version ($149 per month) and business personal licenses.

4.2.3.4. Open BI Solutions

ELK (ElasticSearch + Logstash* + Kibana)

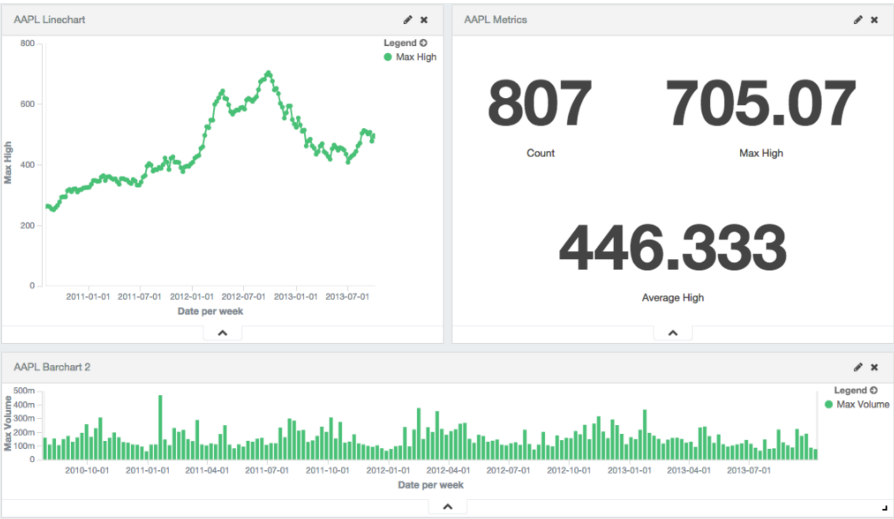


Figure 15: Appearance of Dashboard made with Kibana

ELK is a solution that is being used to store server logs and obtain metrics and analysis of any type: resource monitoring, log searches, more repeated SQL queries, slower SQL queries, geolocation of users, users that access more times to an application, application usage by users, etc. Or any other metrics about the logs of an application

ELK was designed for centralized monitoring of servers and production applications. It is deployed on a server or cluster in the cloud, and nodes to be monitored and their applications send their logs, which are pre-filtered with Logstash, stored in ElasticSearch and visualized with Kibana to extract metrics and usage patterns in order to improve the performance. Although this is its initial task, in fact ELK is much more, since it is a solution for data management that allows its transportation between machines, processing and query in real time. Currently there are products based on ELK for business intelligence, which are actually an extension of ELK.

Elasticsearch. Advanced search engine extremely fast. With Elasticsearch, all kinds of data can be searched and filtered through a simple API. The API is RESTful, so it can be used for data analysis, as well for web-based applications production.

Logstash. Tool designed to organize and search log files. But it can also be used to clean and transmit large data from all kinds of sources in a database. Logstash also has an adapter for Elasticsearch, so these two components interact very well together.

*Although ELK appeared as a joint monitoring solution, the Beats project has recently been incorporated into the group, whereby it is betting that it will be responsible for feeding Elasticsearch, either together or in parallel to Logtash.

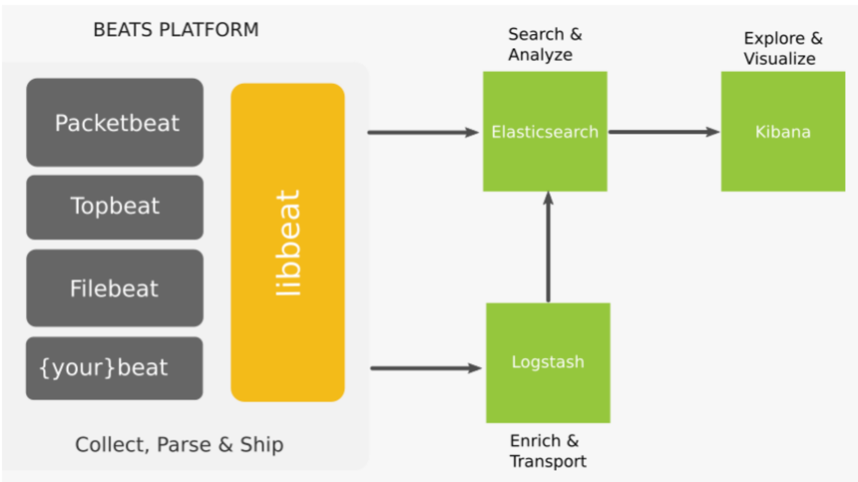


Figure 16: Beats components architecture

Beats is a single-purpose data loader platform, which is installed as lightweight agents on its servers to send different types of operating data to libbeat, the Go library (language) containing the common parts of all Beats to handle tasks Such as inserting data into Elasticsearch, sending events securely to Logstash, balancing the load of events to multiple Logstash and Elasticsearch nodes, and sending events in synchronous and asynchronous mode. The libbeat platform also includes mechanisms to detect when servers are overloading or the intermediate network is being congested, so it can reduce the sending rate.

The beats platform has four official beats: Filebeat, Packetbeat, Metricbeat and Winlogbeat, in addition to these three the development community has created many more, can be consulted from https://www.elastic.co/guide/en/beats/libbeat /current/community-beats.html.

Kibana is a visual interface for ELK that works in the browser. It is an open source visualization and analysis platform that allows to explore, visualize and represent the data content of an Elasticsearch cluster using the graphics, tables and maps provided by the powerful graphics library d3.

It is good enough to display the data stored in Elasticsearch and does not require programming knowledge, since the visualizations are completely configured through the interface.

Kibana customization

- There are two mechanisms for creating custom dashboards. The first is to create a custom control panel from the user interface. The second is to create a custom control panel from a file. Custom control panels can be created from a JSON document template based on a specific schema, or create a custom template as a Javascript file. By default, it brings examples of dashboards in JSON.
- There is also a way to create your own visualizations out of the list that Kibana offers, anyway, the most current are covered by those.

BIRT

The Eclipse Business Intelligence and Reporting Tools Project (BIRT) is an Eclipse top level project that is focused on tools and capabilities that extract data from data sources, process that data, and present the processed information to the end users. First released in 2004, BIRT is a mature open source reporting platform with millions of active users.

Actuate, by far the largest code contributor to the project was acquired by OpenText, Canada's largest software company, in January of 2015. At the time of this review, OpenText and Actuate have not indicated any changes to the BIRT Project.

BIRT consists of two main components:

- **BIRT Report Designer**. A graphical report designer that is designed to be easy to use. It is a fat-client tool, installed to the developer’s desktop, either within the Eclipse IDE or as a standalone Windows application (the “RCP Designer” package). Report developers use its graphical user interface to create report layouts, hook up data sources, and produce XML-based report designs.
- **BIRT Runtime**. Also known as the “BIRT Report Engine,” is a set of Java classes and APIs that takes the XML-based report designs (created by the BIRT Report Designer), “runs” the report designs by querying their data sources, merges the query data into the report layouts, and then produces report output in HTML, PDF, Excel, or other formats. You can embed the BIRT Runtime inside your Java applications. The BIRT Runtime can be deployed either as a set of OSGi jar files, or as a drop-in collection of jar files.

BIRT also includes four additional open-source components that can be used separately from the main components:

- **BIRT Chart Engine**. It is basically just a piece of the BIRT Runtime. With the Charting Engine API (CE API), you can incorporate charting into your Java application without the overhead of the entire BIRT Runtime.
- **BIRT Chart Designer**. OSGi based report designer that can be deployed and used within applications that support the OSGi framework that would like to design reports.
- **BIRT Viewer**. This is a sample “viewer” that is an Eclipse plug-in. It lets you preview reports, populated with the real query data, when you are working within Eclipse. It is also available as a standalone Java EE application. By deploying the BIRT Viewer to a Java application server, you can provide end users with a web-based user interface to view reports, navigate through paginated reports, and render reports into a variety of formats including HTML, PDF, and Microsoft Excel/PowerPoint/Word. The BIRT WebViewer also supports JSP Tag Library extensions for many common report interaction tasks.
- **Eclipse Data Tools Project (DTP)**. BIRT's access to data is designed to flow through the DTP, a stand-alone project. Initially co-sponsored by Actuate and Sybase, the development on DTP has slowed considerably.

The BIRT Report Designer is a robust, powerful tool for creating reports where the output is primarily presented over the web. The BIRT Report Designer is fairly easy to use (although occasionally overly complex) and is able to create reports ranging from the simplistic to the very complex with minimal custom code. BIRT has excellent support for essential report features such as charts, crosstabs, using multiple data sources within the same report, re-using queries within reports, adding custom code, and more.

One significant distinction when comparing BIRT to Pentaho and JasperReports is that BIRT lacks an open-source report server. For many applications, a report server provides functionality such as user management, report management, security, notifications, and more. Using a report server often lets report developers concentrate on report creation and the ideal presentation of data, instead of having to code up these separate capabilities. If report server capabilities are important to your application, and you are determined to go the open source route, then you should consider either Pentaho or Jasper.

A second significant area where BIRT does not match up to its open source competitors is its incompatibility with OLAP data as describe by either Mondrian or XML for Analysis (XMLA).

Jaspersoft community

JasperReports is one of the most popular and used open source projects for reporting and graphic creation. It has tools both to ETL (Extract, Transform, Load) and to create reports (visualizations).

Jasper server has two API, HTML and REST to interact with it and to ask for its resources. It has a JavaScript library to create dashboards, with the JavaScript API the report is created and embedded in a ready HTML element, without programming, therefore dashboard layout is 100% configurable, reports are added to the website without further action. The HTML API is used to obtain reports or visualizations, while the REST API is used to list, create, update and delete reports and resources of a Jasper server repository.

JasperReports Server is a standalone and integrative reporting server that allows the delivery of real-time or scheduled information to the web, the printer or a variety of file formats. The iReport and Jaspersoft Studio tools are used to design the reports that will be built and returned in JasperReports Server. Reports can be run or exported to a desired output, or scheduled to run at a later date.

- Main features:
  - Parameters can be sent to the report.
  - One report can have more than a graphic, list, etc.
  - Jaspersoft is 100% pure Java, so should work in any Java > 1.6 environment.
  - It consists of various components: JasperReports server, Jasper reports library, Jaspersoft ETL, visuality.js, iReport and Jaspersoft Studio. (iReport is no longer supported, since it has been replaced by Jaspersoft Studio).
  - Application servers: Apache Tomcat, JBoss, IBM WebSphere, Oracle WebLogic, GlassFish and SpringSource tc Server.
- Data Sources:
  - Any RDBMS compliant with JDBC 2.1 and SQL-92 (includes most relational databases).
  - Big Data & NoSQL: MongoDB, Hadoop, Cassandra, others.
  - Other data sources: Bean, JNDI, XML/A, Custom (e.g. Hibernate, XML, etc.).
  - In order to use CSV or Excel files, Jaspersoft ETL or Jaspersoft studio are required for using these files in the Jasper server.
- Other features
  - It has AWS, professional and enterprise editions.
  - Besides support and training, premium services offer extra features.

- Metadata layer. A virtual view of a data source that simplifies presentation in commercial terms while providing data-level security.
- Data virtualization. Allows integration of multiple data sources into a single metadata view to enable analysis and reporting across sources without the need for ETL or a Data Warehouse.

JasperSoft Studio is a tool somewhat less usable and more complicated than its counterparts in Pentaho and BIRT. Common operations like creating aggregates is complicated in some cases. This is because JasperReports was born with the main dedication of providing printed reports and later replicate this work for use on the web, so the workflows to create the reports seems a bit more complex compared to the other solutions. One more thing it is the fact the reports must be compiled to be used.

Pentaho

Pentaho is a prominent open source reporting project with a great vision of a single, comprehensive BI suite that covers the full spectrum of business intelligence (BI) life cycle including ETL, reporting, analytics, visualization, and data mining. Reporting and visualization are only a portion of Pentaho's overall vision of providing a complete open source BI solution. The Pentaho BI Suite Community Edition (BI) encompasses the following open source projects: Pentaho BI Platform & Server, Pentaho Reporting, Pentaho Analysis Services (Mondrian), Pentaho Data Integration (Kettle), and Pentaho Data Mining (Weka).

The open-source Pentaho BI Suite Community Edition (CE) includes the following components for reporting:

- **Pentaho Report Designer**. A WYSIWIG tool that lets you create reports using a graphical user interface, as opposed to creating reports programmatically or by directly creating and manipulating XML. These reports can then be run by the Pentaho Reporting Engine Classic or the Pentaho BI Server. The Pentaho Report Designer is a stand-alone, desktop-installed client tool, and is not available as an Eclipse or NetBeans plug-in.
- **Pentaho Reporting Engine Classic**. Formerly known as "JFreeReport", is a collection of Java classes and APIs that execute Pentaho's XML-based reports. The Pentaho Reporting Engine runs report designs against data sources, and renders report output in HTML, PDF, Excel, and other output formats. You can embed the Pentaho Reporting Engine inside your Java applications. You don't need the Pentaho Reporting Engine if you use the BI Server.
- **Pentaho Data Integration (Kettle)**. Kettle is a graphical data integration tool that allows developers to build jobs and transactions that can be used to Extract, Transform, and Load (ETL) data from a wide variety of sources.
- **Pentaho Reporting SDK**. The SDK is the Pentaho Reporting Engine Class, plus the documentation and supporting libraries that developers need to embed the Pentaho Engine Classic in their applications.
- **Pentaho BI Server**. The BI Server is a J2EE application that provides an infrastructure for multiple users to run reports and OLAP cubes through a web-based user interface. At the core of the BI Server are the Classic Engine and the Mondrian ROLAP Engine (which run the reports and OLAP cubes respectively), plus a host of server capabilities including authentication, user management, logging, email notification, server APIs, and report scheduling. The BI Server also provides the infrastructure for reports and analytic cubes to access data and metadata via the Pentaho Data Integration's ETL functionality (Kettle).
- **Pentaho User Console**. End-users can login, browse reports, run them, view report results in HTML or PDF, and download report results in other formats. Users can also, for the time being, create basic ad-hoc reports, and conduct some OLAP analysis, however this functionality is likely to be removed in future releases.

- **Pentaho Administrator Console**. Administrators and developers can deploy reports, manage users, set up security access privileges, and deploy workflows.

Pentaho has the editor that is the easiest to learn for creating basic listing reports and for grouped listing reports with aggregations. Its UI was not overly cluttered with sophisticated, less-commonly used functionality and the tool was easy to learn and performed well. In short, the User Interface is attractive and functionality is sensibly laid out.

Pentaho's stated goal is to provide a comprehensive solution for Data Integration and Business Analytics. This includes solutions for Extract, Transformation, and Load (ETL), basic reporting, data analytics, data exploration, data visualization and data mining. This broad company vision might explain why the reporting functionality is not as deep as BIRT or Jaspersoft. Functionality that report developers take for granted in BIRT and JasperReports - such as side-by-side report components, cross-tabs, and robust charting - are not as fully developed in Pentaho. Unfortunately, this means when using only Pentaho reporting, it can be more difficult to create complex reports.

A prominent feature is the metadata injection. The ETL Metadata Injection step is able to inject metadata using a transformation template. So, instead of introducing ETL metadata statically into a dialog box step by step, it is executed at runtime. It is possible to resolve repetitive ETL workloads such as text file upload, data migration and so on.

4.2.4. Suplements

To know firsthand about the possibilities of the ELK framework, a different approach of traditional BI software some tests have been done with some of the most interesting. For them it has started from data of tables of a relational database, data about vessel calls in some ports. The trials done were to create a simple dashboard with one KPI, a top 10 list and a time series and the raw data showed in a table and being synchronized with the visualizations.

1. For this test, it has been exported data from one relational database view to a csv file.

2. Then data will be loaded to Elasticsearch using the Logstash component, to do that there must be a configuration file for the load with the following information and structure:

```
input {
  file {
#Here it is specified, the starting point and sincedb file (this is for posloads or load interruptions, logstash uses it to know until the point is already loaded)
    path => "/vagrant/*.csv"
    start_position => "beginning"
   sincedb_path => "/dev/null"
  }
}
#file type and column names
filter {
  csv {
      separator => ","
     columns => ["COLUMN1", "COLUMN2",... ]
  }
#Here some data is converted
  mutate {convert => ["COLUMN1", "string"]}
  mutate {convert => ["COLUMN2", "integer"]}
  mutate {remove_field => [ "message" ]}
}
# specifying timestamp field/s
  date {
         match => ["FECATRAQUESOL", "dd/MM/YY"]
         target => "FECATRAQUESOL"
        }

#logstash output to elasticsearch
output {
```

```
            elasticsearch {
              hosts => "http://localhost:9200"
              index => "vgpestadia"
            }
          stdout {}
          }
```

Example 1: Logstash input configuration file example

Then we tell Logstash to do the load with the command:

/opt/logstash/bin/logstash -f /vagrant/confilename.conf

3. After that, opening Kibana (localhost: 5601) in a browser, we see that there is a new index of name con filename. From here we can build visualizations and the dashboard.

Figure 17: Pie chart, time series and top ten list visualization build with Kibana

Experience

The dashboard can be embedded in a html webpage as an iframe element, also every visualization can be embedded independently too. It is still missing to test how to build visualizations that receive parameters and how to communicate with Kibana.

The visualizations are very easy to build, just choose the right one, and with the graphic wizard it is arranged. More complex graphics cannot be built with the wizard, Kibana has other way to build them that is define the visualization with a JSON specification. Of course, deep knowledge about how it works is required.

Interactions and visualization synchronization. After the dashboard is created, one can click in some part of the graphic and this triggers to all the other graphics to be synchronized and then, they are filtered by the value of the field clicked. A tooltip shows up some interesting information about the part of the graphic the mouse hovers on it. All of these features are configurable, and can be enabled or disabled.

Original data can be present and available every time. You can create a search where all data is included, and the filter is updated when interacting with the graphs, so we see the composition of this search, in addition the table can be sorted by fields and is paginated (no need to handle huge amount of records do not reduce Kibana performance), so at all times have the complete dataset that make up the visualizations.

5. State of the art in GIS Technologies

5.1. Web GIS Services

5.1.1. Spatial Data Infrastructure

In the last decade, the use of geospatial information has experienced a large increase due to advances in software development in Geographic Information Systems (GIS), progress in data standardization and the implementation of Spatial Data Infrastructures (SDI) through Geoportals (web mapping applications).

A Spatial Data Infrastructures (SDI) is a basic set of technologies, polices and institutional arrangements to facilitate the availability and access to geographic information (maps, orthophotos, satellite images, etc.) via the Internet. From the technological point of view, a SDI can be defined as a decentralized network of servers that provide spatial data, metadata, search and visualization methods and standard interfaces to access to the geographic data.

The goals of a SDI are:

- To facilitate the access and the integration of the spatial information allowing the knowledge diffusion and the optimization of the decision making.
- To promote the generation of metadata for the spatial information by using standards; avoiding the duplicity of efforts and cutting off costs.
- To encourage the cooperation between agents, promoting the exchange of information

To get these goals it is essential that a SDI is composed of the following components:

- An institutional framework that encourages its creation and maintenance (INSPIRE)
- A data police that promotes the generation and the access to reference data
- Technology needed for system operation (Web GIS Technologies)
- Standards to improve the interoperability of spatial data allowing the exchange of information between different agents (OGC)

5.2. Institutional framework: INSPIRE Directive

Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community (INSPIRE) was published in the official Journal on the 25th April 2007[15].

The INSPIRE Directive is based on the infrastructures for spatial information established and operated by the 27 Member States of the European Union. The Directive addresses 34 spatial data themes needed for environmental applications, with key components specified through technical implementing rules. This makes INSPIRE a unique example of a legislative “regional” approach.

The INSPIRE directive came into force on 15 May 2007 and will be implemented in various stages, with full implementation required by 2019.

The INSPIRE directive aims to create a European Union (EU) spatial data infrastructure. This will enable the sharing of environmental spatial information among public sector organizations and better facilitate public access to spatial information across Europe.

[15] INSPIRE > WELCOME TO INSPIRE.” [Online]. Available: http://inspire.ec.europa.eu/

A European Spatial Data Infrastructure will assist in policy-making across boundaries. Therefore, the spatial information considered under the directive is extensive and includes a great variety of topical and technical themes.

INSPIRE is based on a number of common principles:

- Data should be collected only once and kept where it can be maintained most effectively.
- It should be possible to combine seamless spatial information from different sources across Europe and share it with many users and applications.
- It should be possible for information collected at one level/scale to be shared with all levels/scales; detailed for thorough investigations, general for strategic purposes.
- Geographic information needed for good governance at all levels should be readily and transparently available.
- Easy to find what geographic information is available, how it can be used to meet a particular need, and under which conditions it can be acquired and used.

5.3. Web GIS Technologies

Nowadays, there is a wide range of private and open source technologies for the implementation of Spatial Data Infrastructures (SDI) through Geoportals (web mapping applications).

Recently, the technology and architecture that could service and deployment models offer are a key area of research and development for GIS technology. The GIS technologies in the cloud are presented as SaaS (Software as a Service) that provide cloud-based clients and applications that solve complex business problems using GIS tools.

The software and the data are hosted on the servers of the company that supplies these services that are usually accessed through a web interface.

Some examples of GIS products (privative and open source software) are the following ones:

- **GIS Cloud[16]:** It is a service that allows the store, edit, publish, share geographic information in the cloud through a web interface. It allows different formats of spatial data importation, the connection with the data base PostGIS, with WMS severs, with tiles servers and the publication of maps from ArcMap Software. It offers two APIs, one REST and other JavaScript, for the creation of client applications. But due to these applications do not support standards, the integration of these services and data in other applications is complicate.
- **OpenGeo Suite[17]:** There is a version of the OpenGeo Suite for the cloud. The suite is composed of the main GIS open source products such as: PostGIS (database), Geoserver and Geowebcache (maps and tiles servers), OpenLayers (map web client) and Tomcat (application's server). Therefore, it offers applications to edit, print, style and share information allocated in PostGIS and published by Geoserver.
- **CartoDB[18]:** CartoDB is a product of the Spanish company Vizzuality that allows importing, designing and publishing maps. Besides it offers spatial analyze, advanced style through CartoCSS and APIs for the development of applications based on Leaflet and Google Maps. This products are mainly focused to the store of vector date in the database PostGIS.

[16] "GIS Cloud : It's about the Apps, not the Maps!" [Online]. Available: http://www.giscloud.com/.
[17] "OpenGeo Suite Cloud Edition - Boundless." [Online]. Available: http://boundlessgeo.com/2010/08/opengeo-suite-cloud-edition/.
[18] "Create amazing maps with your data — CartoDB." [Online]. Available: http://cartodb.com/.

- **ArcGIS Server[19]:** ESRI offers GIS services in the cloud of its product ArcGIS through Amazon EC2 and its own infrastructure.
- **IkiMap[20]**: ikiMap is a service created by the Spanish company Sixtema that allows the generation, publishing and especially the sharing of maps in the cloud. The main feature of the service is that it is intended to non-professional users through a friendly web interface that can be used by non-expert in GIS issues. IkiMap infrastructure uses some open source GIS products such as PostGIS, OpenLayers and gvSIG Mini.
- **MapBox[21]:** It offers publish and maps design service in the cloud, mainly oriented to the tiled services. It offers tools for the advanced design of maps such as TileMill, for layers composition, and options for traffic analyze in the tile server, etc.
- **QGIS – Formerly Quantum GIS:** Automate map production, process geospatial data, and generate drool-worthy cartographic figures. QGIS Plugins boost this mapping software into a state of epicness. If the tool doesn't exist, search for plugin developed by the QGIS community.
- **gvSIG:** gvSIG really outperforms QGIS for 3D. It really is the best 3D visualization available in open source GIS. The NavTable is agile in that it allows the user to see records one-by-one vertically. The CAD tools are impressive on gvSIG. Thanks to the OpenCAD Tools, the user can trace geometries, edit vertices, snap and split lines and polygons. gvSIG Mobile brings GIS to the mobile phone. This extension is perfect for field work because of its interface and GPS tools.
- **Whitebox GAT:** Whitebox GAT was intended to have a broader focus than its predecessor, positioning it as an open-source desktop GIS and remote sensing software package for general applications of geospatial analysis and data visualization. Whitebox GAT is intended to provide a platform for advanced geospatial data analysis with applications in both environmental research and the geomatics industry more broadly. It was envisioned from the outset as providing an ideal platform for experimenting with novel geospatial analysis methods. Equally important is the project's goal of providing a tool that can be used for geomatics-based education. An important characteristic of Whitebox GAT is the unique open-access development philosophy adopted by the project, which lends itself to experimenting with geospatial algorithm development.
- **SAGA GIS:** It started out primarily for terrain analysis such as hill shading, watershed extraction and visibility analysis. Now, SAGA GIS is a powerhouse because it delivers a fast grow set of geoscientific methods to the geoscientific community. Enable multiple windows to lay out all your analysis (map, histograms, scatter plots, attributes, etc.). It provides both a user-friendly GUI and API. It's not particularly useful in cartography but it's a lifesaver in terrain analysis. Closing gaps in raster data sets is easy. The morphometry tools are unique including the SAGA topographic wetness index and topographic position classification.
- **Diva GIS:** free GIS software package for mapping and analyzing data. This one specializes in mapping biological richness and diversity distribution including DNA data. It's possible to extract climate data for all locations on the land. From here, there are statistical analysis and modeling techniques to work with.
- **OrbisGIS:** Its goal is to be a cross-platform open source GIS software package designed by and for research. It provides some GIS techniques to manage and share spatial data.

[19] http://www.esri.es/es/productos/arcgis/arcgis/en-la-nube--cloud-gis/
[20] http://wikipedia.org/wiki/IkiMap
[21] https://www.mapbox.com/tour/..

OrbisGIS is able to process vector and raster data models. It can execute processes like noise maps or hydrology process without any add-ons.

- **uDig**: uDIG is an acronym to help get a better understanding what this Free GIS software is all about: u stands for user-friendly interface, D stands for desktop (Windows, Mac or Linux), I stand for internet oriented consuming standard (WMS, WFS or WPS) and G stands for GIS-ready for complex analytical capabilities. uDig's Mapnik lets you import basemaps with the same tune as ArcGIS.
- **GeoDa**: is a free GIS software program primarily used to introduce new users into spatial data analysis. It's main functionality is geostatistics. Perform autocorrelation, descriptive and regression statistics with GeoDa. It's an exciting analytical tool which includes lab users from Harvard, MIT and Cornell universities. This free GIS sifted serves as a gentle introduction to spatial analysis for non-GIS users.

The following frameworks are used for the development of Web GIS applications:

- **OpenLayers**[22]: It is an open-source library for the creation of Web GIS applications. Currently is the most robust and mature framework for the generation of web map applications based on the standards implementation with the support of the most formats, services and GIS protocols. It has an optimized and reduced version for mobile devices and the version 3 of the framework is being developed. This new version will offer an easier API and WebGL support.
- **LeafletJS**[23]: It is an open-source JavaScript library based on HTML5. Its main feature is that is very light and its optimization for mobile devices. Its API has less features and it is simpler than OpenLayers, but is very useful for applications requiring only the visualization of maps.
- **Google Maps**[24]: Google offers a JavaScript API for the development of web applications that access to its Google Maps services (tiling, routing, geocodification).
- **PolyMaps**[25]: It is a JavaScript library to render vector and raster maps by using the standard SVG.
- **ModestMaps**[26]: It is the open-source JavaScript library of MapBox. As LeafletJS and the PolyMaps frameworks, it is oriented to the visualization and interaction with easy maps. Its main feature is its lightness (10 KB compressed) and its optimization for mobile devices.
- **MapQuery**[27]: MapQuery is the open-source JavaScript framework that joins jQuery with OpenLayers. It is oriented to the development of interface user components. Its main drawback is that there are no many active developers.
- **JQuery Geo**[28]**:** It is an open-source JavaScript framework similar to MapQuery.
- **OpenGeo Suite**[29]**:** It is the open-source JavaScript framework that joins ExtJS and GeoExt with OpenLayers. It allows the quick configuration to web GIS applications through a set of plugins and widgets.
- **GeoExt**[30]**:** It is an open-source JavaScript framework that joins ExtJS with OpenLayers, allowing the generation of interface user components.

---

[22] http://www.openlayers.org/..
[23] http://leafletjs.com/..
[24] https://developers.google.com/maps/documentation/javascript/..
[25] http://polymaps.org/..
[26] http://geoext.org/..
[27] http://mapquery.org/..
[28] http://jquerygeo.com/..
[29] H.-K. P. J. J. S. H. S. H. H.-G. S. Seongsu Jeonga, "Productive high-complexity 3D city modeling with point clouds collected from terrestrial LiDAR." in Computers, Environment and Urban Systems, 2013, pp. 26-38.
[30] "JavaScript Toolkit for Rich Web Mapping Applications — GeoExt v1.1." [Online]. Available: http://geoext.org/..

- **GeoExt Mobile**: It is an attempt of port de GeoExt for using Sencha, the ExtJS version for mobile devices.

Regarding to the development of the Web GIS applications there are two types of components. Firstly, the JavaScript frameworks that allow to locate a map on the web, add layers, apply styles, interact with the maps (navigation, object selection, draw geometries, etc.), etc. and on the other hand, the developments frameworks, which allow to configure Web Mapping applications, add components of user interface, buttons to active controls, display alphanumeric information associated to the map, modify the characteristics of the display (connect to new servers, add new layers, advanced editing , queries, etc.).

Among the JavaScript libraries, the most optimal and robust library to develop Web GIS application, taking into account only the API) is OpenLayers. The problem of this library is that most of its features do not work on mobile devices, mainly because it was not designed for this type of devices and therefore the version for mobile devices is simply a reducer version.

Apart from OpenLayers, over the last years it has been appeared other open-source libraries, based on HTML5, optimized for mobile devices, both in library size and optimized for mobile, both in library size, and quantity of KB transferred, memory consumption, etc. However, these libraries have a lack of advanced features such as: editing, theme support multiple formats, protocols, etc.

Regarding to the frameworks, all the aforementioned are based on OpenLayers. This fact is due OpenLayers is the most used and the unique that has been adapted for mobile devices is GeoExt Mobile, although it offers only some services (navigation, selection, etc.)

5.4. WEB GIS Standards (OGC)

The Open Geospatial Consortium (OGC) is an organization composed of companies, universities and public administrations that has developed over recent years a number of protocols and standards that provide the technology framework for interoperability in GIS and Spatial Data Infrastructure[31].

The OGC Standards are developed within the OGC Standards Program. Here the OGC Technical Committee and OGC Planning Committee work in a formal consensus process to arrive at approved (or "adopted") OGC® standards.

The most common approved standards used in Web mapping applications for publishing geographic data, using the criteria of OGC, are:

5.4.1. WMS y WMTS (Web Map Service and Web Map Tile Service)

The Web Map Service Interface Standard (WMS) provides a simple HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. A WMS request defines the geographic layer(s) and area of interest to be processed. The response to the request is one or more geo-registered map images (returned as JPEG, PNG, etc.) that can be displayed in a browser application. The interface also supports the ability to specify whether the returned images should be transparent so that layers from multiple servers can be combined or not.

A WMTS enabled server application can serve map tiles of spatially referenced data using tile images with predefined content, extent, and resolution.

5.4.2. WFS (Web Feature Service)

The Web Feature Service (WFS) represents a change in the way geographic information is created, modified and exchanged on the Internet. Rather than sharing geographic information at the file level

[31] "Open Geospatial Consortium | OGC(R)." [Online]. Available: http://www.opengeospatial.org/..

using File Transfer Protocol (FTP), for example, the WFS offers direct fine-grained access to geographic information at the feature and feature property level. Web feature services allow clients to only retrieve or modify the data they are seeking, rather than retrieving a file that contains the data they are seeking and possibly much more. That data can then be used for a wide variety of purposes, including purposes other than their producers' intended ones.

In the taxonomy of services defined in ISO 19119, the WFS is primarily a feature access service but also includes elements of a feature type service, a coordinate conversion/transformation service and geographic format conversion service.

5.4.3. WCS and WCTS (Web Coverage Service and Web Coverage Tile Service)

The OGC Web Coverage Service (WCS) supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space/time-varying phenomena. A WCS provides access to coverage data in forms that are useful for client-side rendering, as input into scientific models, and for other clients. The WCS may be compared to the OGC Web Feature Service (WFS) and the Web Map Service (WMS). As WMS and WFS service instances, a WCS allows clients to choose portions of a server's information holdings based on spatial constraints and other query criteria.

A WCTS enabled server application can serve coverage tiles of spatially referenced data using tile images with predefined content, extent, and resolution making faster the access and the updating of geographic information.

5.4.4. Catalogue Web Services (CSW)

The OGC Catalogue Web Services allows searching for and retrieve any type of information object (data, service instance, service type, schema, style description, etc.) based on its properties described in “metadata.”

5.4.5. WPS (Web Processing Service)

The Web Processing Service (WPS) Interface Standard provides rules for standardizing how inputs and outputs (requests and responses) for geospatial processing services, such as polygon overlay. The standard also defines how a client can request the execution of a process and how the output from the process is handled. It defines an interface that facilitates the publishing of geospatial processes and clients’ discovery of and binding to those processes. The data required by the WPS can be delivered across a network or they can be available at the server.

5.4.6. NetCDF

The OGC NetCDF encoding supports electronic encoding of geospatial data, specifically digital geospatial information representing space and time-varying phenomena. NetCDF is a data model for array-oriented scientific data. A freely distributed collection of access libraries implementing support for that data model, and a machine-independent format are available. Together, the interfaces, libraries, and format support the creation, access, and sharing of multi-dimensional scientific data.

5.4.7. WFS Gazetteer

WFS Gazetteer is a Gazetteer Service profile of the OGC Web Feature Service Standard. The OGC Gazetteer Service allows a client to search and retrieve elements of a georeferenced vocabulary of well-known place-names.

5.4.8. SWE, Sensor Web Enablement

The SWE standards standardize the use of sensors in Web. SWE allows the discovery, exchange and processing of sensor observations. Thus, its aim is the same as the standards HTTP and HTML that enable the exchange of different types of information

5.5. Data sources

OpenStreetMap (OSM) is a young and dynamic project that creates and distributes free geographic data for the world. It is a free, editable map of the whole world that is being built by volunteers largely from scratch and released with an open-content license.

OpenStreetMap is open data, licensed under the Open Data Commons Open Database License32 (ODbL) by the OpenStreetMap Foundation33 (OSMF). This license offers users to copy, distribute, transmit and adapt the data, as long as the user credit OpenStreetMap and its contributors. It is possible to distribute the result of the data only under the same license. The cartography in OpenStreetMap map tiles and the documentation, are licensed under the Creative Commons Attribution-ShareAlike 2.0 license34 (CC BY-SA).

OpenStreetMap wants to be for geodata what Wikipedia is for encyclopedic knowledge. Its focus is mainly on transport infrastructure (streets, paths, railways, rivers), but OpenStreetMap also collects a multitude of points of interest, buildings, natural features and land use information, as well as coastlines and administrative boundaries. OpenStreetMap relies mostly on data collected by project members using their GPS devices and entered into the central database with specialized editors. For some areas, third party data has been imported.

OpenStreetMap data quality and coverage differ between regions. Many European cities are covered to a good level of detail. Often, OpenStreetMap is the first to have a new housing development or a new motorway exit mapped. But in some, mostly rural areas there might be nothing in the database except some primary roads. Interestingly, this is a community project in which everyone can participate. Prior geography, cartography or GIS knowledge is not required.

Geofabrik GmbH was incorporated in late 2007 by Jochen Topf and Frederik Ramm, out of the conviction that free geodata created by projects like OpenStreetMap will become increasingly attractive for commercial uses. Geofabrik bridges the gap between free project and professional users with custom data offerings, with consulting, support, training, and software development. The Geofabrik's download server has free and current geodata from OpenStreetMap in various formats. Most of these files are updated every day – any change it is uploaded to OpenStreetMap should be on their download server the next day. Data on the download server is organised by regions. The */openstreetmap* directory contains files that have a whole continent's data in them, and for some continents there are subdirectories in which you find individual files for various countries. Some countries again have their own subdirectories with data for administrative subdivisions. Geofabrik offers this information in two formats:

- Raw data (OSM source data): The OpenStreetMap raw data files are compressed and ends in ".osm.bz2" are bzip2. The XML based data format is described in the OSM Protocol Version 0.6 article on the OpenStreetMap wiki. The XML DTD is available in the OSM Protocol Version 0.6/DTD article. Processing OSM raw data requires software.
- Shape Files: Files ending in ".shp.zip" are shape files that can be processed with almost any GIS software. In converting OSM data to shapefiles, Geofabrik has made a simple default selection of layers where the most important features get exported (road and railway network, forests, water areas, some points of interest).

[32] http://opendatacommons.org/licenses/odbl/
[33] http://wiki.osmfoundation.org/wiki/Main_Page
[34] https://creativecommons.org/licenses/by-sa/2.0/?

5.5.1. Data Source Specifications

Short Name	Source	Protocol Exchange Channel	Data Exchange Format	Geo-localization	Size	Privacy	Additional Information
Point of Interests Critical Infrastructures	Geofabrik	HTTP	Compressed Shape Files with Metadata	Yes	Variable *(Depends on the country)*	PU	http://geofabrik.de

Table 1: Specifications for OpenStreet Data.

5.5.2. Example data

POIs are mapped at OpenStreetMap either as a node or a way (possibly a closed polygon) or a multi-polygon. These objects have tags that describe the feature they represent.

For example, it is possible to have a node with latitude 51.398, longitude -1.082, and tags amenity=place_of_worship, name=St Mary's Church. This indicates a church named St Mary's Church and its location. Geofabrik offers the information organized in regions. For example:

- The SHP file for the Spanish region: http://download.geofabrik.de/europe/spain-latest-free.shp.zip
- Its raw data link is: http://download.geofabrik.de/europe/spain-latest.osm.pbf

In OSM there are no layers in the traditional GIS sense. All features are in one big coherent geodatabase. Geofabrik has extracted the features stored in the OSM database into different layers depending on their type. The layer that is going to be used in this project is called “pois”.

These shape files have the following columns/attributes:

Attribute	Description
id	Id of this feature. Unique in this layer.
osm_id	OSM Id taken from the Id of this feature (node_id, way_id, or relation_id) in the OSM database.
lastchange	Last change of this feature. Comes from the OSM last_changed attribute. Reflects changes in the attributes of a feature;

Attribute	Description
code	4 digit code (between 1000 and 9999) defining the feature class. The first one or two digits define the layer, the last two or three digits the class inside a layer
fclass	Class name of the feature. This does not add any information that is not already in the “code” field but it is better readable. Examples: police (OSM tag: amenity=police), fire_station (OSM tag: amenity=fire_station), post_office (OSM tag:amenity=post_office), embassy(OSM tag:amenity=embassy) ...
name	Name of this feature, like a street or place name.

Table 2: Example of OpenStreetMap shape files

6. State of the art in 3D City Modelling and Simulation

6.1. 3D City Modelling

3D city models are digital models of urban areas that represent terrain surfaces, sites, buildings, vegetation, infrastructure and landscape elements as well as related objects (e.g., city furniture) belonging to urban areas. Their components are described and represented by corresponding two-dimensional and three-dimensional spatial data and geo-referenced data. 3D city models support presentation, exploration, analysis, and management tasks in a large number of different application domains. In particular, 3D city models allow "for visually integrating heterogeneous geoinformation within a single framework and, therefore, create and manage complex urban information spaces." (35). It is an emerging field and applied in a variety of applications. Since Google Earth and Microsoft Virtual Earth (now Bing Maps) were introduced in 2005, 3D city models have been gaining popularity among the public. A large number of commercial automotive navigation systems have adopted 3D city models, though models differ in level of detail and quality. The launch of Microsoft's Photosynth in 2009 allowed users to build and explore 3D city models using their own digital photographs. Furthermore, with availability of geographic data open to public, 3D city models are created from the Open Street Map in Germany. The advent of more advanced graphics technologies along with more powerful computing resources makes it possible to generate 3D city models everywhere as a replica of the real world. The intense competition in global 3D mapping between Google and Microsoft confirms the central place of 3D mapping products, including 3D city models, in the mainstream IT industry. Of more recently, even Apple and Nokia entered into 3D mapping world with acquisition of C3 and Earthmine. Furthermore, 3D city models are now evolving to platforms, more than just important contents. This trend is more accelerated with the growing popularity of smart phones equipped with positioning technology. Location-Based Service (LBS) is shifting into a higher gear, while mobile services and applications related to location devices are becoming more and more popular. LBS in fact, with its capacity to tailor and target advertisement based on individual smart phone users' locations, particularly in urban area, is opening up a new revenue stream for the mobile service industry. Such trend is supposed to be extended to high-complexity 3D urban space for differentiating smart city service, which is supported by recent industry movements. Into the quest to fully realize the mobile advertisement market and a future ubiquitous world, high-complexity 3D city modelling is the essential requirement.

Nowadays, 3D City modeling is an important issue in all over the world for Geomatics researchers. Geomatics techniques are playing a key role in creating virtual 3D City models. The geomatics field is an umbrella for the mapping technologies. Main Geomatics techniques are Photogrammetry, Remote sensing, Geographical Information System, Global Positioning System, Lasergrammetry, Radargrammetry etc. Photogrammetry and Laser techniques play a major role in modeling virtual 3D Cities. Singh S.P., Jain K., and V. Mandla R. V. (36) make a review related to applications of 3D City models used by various researchers for various applications. The researchers present the most representative Geomatics techniques for 3D City modeling and the related other researchers work for each category and applied methods.

6.1.1. Geomatics Techniques and Methods for 3D City Modeling

The 3D modeling methods are mainly categorized using the following approaches:

- Based on Automation
  - Automatic
  - Semi-automatic

[35] Seongsu Jeonga, Hyo-Keun Parka, Jaehoon Jung, Soohee Han, Sungchul Hong, Hong-Gyoo Sohn, Productive high-complexity 3D city modeling with point clouds collected from terrestrial LiDAR. Joon Heoa, 2013, Computers, Environment and Urban Systems, pp. 26-38

[36] Singh S.P., Jain K., and V. Mandla R. V. Virtual 3D city modeling: techniques and applications, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-2/W2, ISPRS 8th 3DGeoInfo Conference & WG II/2 Workshop, 27 – 29 November 2013, Istanbul, Turkey

- Manual
- Based on Data input techniques
  - Photogrammetry based methods
  - Laser Scanning based methods
- Photogrammetry based methods for 3D City Model generation:
  - Aerial Photogrammetry based model
  - Satellite Photogrammetry based model
  - Close Range Photogrammetry based model
- LASER scanning based model-
  - Aerial Laser based model
  - Terrestrial laser based model

HYBRID METHODS:

Combination of these methods is also a method to create virtual 3D City model.

6.1.2. Photogrammetry based 3D City Models

6.1.2.1. Aerial Photogrammetry based model

In present days, mostly airborne data is used for the collection of 3D city models. Aerial photos are the most commonly using as a raw data. Stereo pair images are useful to create the 3D point cloud. A semi-automatic method for acquiring 3D topologically structured data from 2D aerial stereo images has been presented in (37). In the paper, the researchers used a digital photogrammetric workstation (Traster T10), Microstation CAD package, and Consob, (in-house developed software); they used Digital aerial images of scale 1:2200 for Enschede (Netherlands). Data Acquisition, Data Processing, Superimposition, Database Updating, and Visualisation are the main steps for this work. Figure 18 presents a reconstructed 3D objects and buildings developed in the paper 37.

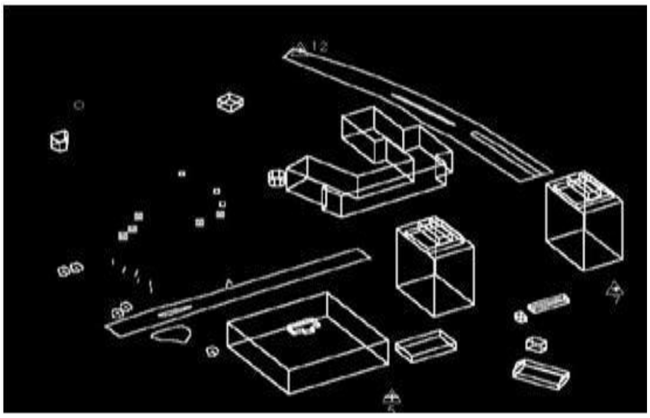


Figure 18: Reconstructed 3D objects and Buildings

They classified the objects (Roofs) as various classes such as No Longitudinal breaks (flat roof), two facades roof, three facade roof, Roof composite etc. This was very basic and primitive model for 3D City.

Kobayashi (38) studied the relation between Photogrammetry and 3D City modeling. He recommends a methodology to create 3D City model by using Photogrammetric processing. He

[37] Zlatanova S., Paintsil J., Tempfli K., 1998,"3D Object reconstruction from aerial stereo images", In Proceeding of the 6th international conference in Central Europe on computer graphics and visualization, Vol. 98, 472-478
[38] Kobayashi Y., 2006, "Photogrammetry and 3D City modelling", Digital Arch, School of Architecture and landscape Architecture, Arizona State University, USA

also used aerial images to create 3D city model with Photogrammetry techniques and discusses the efficiency and effectiveness of the model in terms of time, labor, and reusability. In the work, the author makes a 3D city model of Phoenix, USA.

Shashi and Jain (39) explore the use of Photogrammetry for 3D modeling and scene visualization. They suggested and give an approach in creating 3D model of any building by normal digital camera and close range photogrammetric processing for any project with good accuracy. The main advantages of this work were the use of digital cameras easily available in market, at low cost. They concluded that Close range Photogrammetry gives the best solution for 3D modeling.

Leberl et al. (40) investigates the difference between Point cloud generated from Images and Point cloud generated from Laser. They discuss that the photogrammetric accuracy is good with compare to the LIDAR-method, and also the density of surface points is much higher from images. They also found some additional advantages of the photogrammetric approach.

Portales C. et al. (41) researched about augmented reality and Photogrammetry a way to visualize physical and virtual city environments. The synergy of AR and photogrammetry opens up new possibilities in the field of 3D data visualization, navigation and interaction far beyond the traditional static navigation and interaction in front of a computer screen. Low-cost outdoor mobile AR applications, that integrate buildings of different urban spaces, are presented. High-accuracy 3D photo-models derived from close-range photogrammetry are integrated in real urban worlds. The augmented environment presented required for visualization a see-through video head mounted display (HMD), whereas user's movement navigation is achieved in the real world with the help of an inertial navigation sensor. After introducing the basics of AR technology, the paper presented a real-time orientation and tracking in combined physical and virtual city environments, merging close-range photogrammetry and AR.

Amat et al. (42) investigates a methodology to create virtual 3D City model by using the combination of Aerial Photogrammetry and Close range Photogrammetry. In this method, they suggest that, small 3D buildings, window, door are not visible in aerial images, so CRP used to create photo-realistic virtual model of small building and large buildings with roof structure created with stereo-images from aerial data. Certainly,with the help of combination of these Close range photogrammetric and aerial Photogrammetry techniques, Photo-realistic Virtual 3D city model can be created.

Hammoudi and Dornaika (43) also give an approach for reconstructing 3D polyhedral building models from aerial images. Geometric and Photometric properties used with perspective projection of planar structures are presented. The advantage of this method is in its featurelessness and in its use of direct optimization based on image raw brightness. They avoid feature extraction and matching. They estimated 3D polyhedral model directly by optimizing an objective function that combines an image-based dissimilarity measure and a gradient score over several aerial images. The Differential Evolution algorithm used for the optimization process is elaborate. In this approach, they provide more accurate 3D reconstruction than feature-based approaches. Fast updating and fast 3D model rectification are the main advantage for this approach. They also tested this method for various images.

[39] Shashi M. and Jain Kamal, 2007, Use of Photogrammetry in 3D Modeling and visualization of buildings, Asian Research Publishing Network (ARPN) - Journal of Engineering and Applied Sciences. Vol. 2, No. 2, April, ISSN 1819-6608

[40] F. Leberl, A. Irschara, T. Pock, P. Meixner, M. Gruber, S. Scholz, and A. Wiechert, Point Clouds: Lidar versus 3D Vision, Photogrammetric Engineering & Remote Sensing Vol. 76, No. 10, October 2010, pp. 1123–1134

[41] Cristina Portalés, José Luis Lerma, Santiago Navarro, 2010, Augmented reality and Photogrammetry: A synergy to visualize physical and virtual city environments", ISPRS Journal of Photogrammetry and Remote Sensing 65, 134- 142

[42] Amat Nor' Ainah, Setan Halim and Majid Zulkepli, 2010, Integration of aerial and close range Photogrammetric methods for 3D City modeling generation", Geoinformation Science Journal, Vol. 10, No. 1, 49-60

[43] Hammoudi Karim and Dornaika Fadi, 2011, "A Featureless Approach to 3D Polyhedral Building Modeling from Aerial Images", Sensors,11, 228-259

6.1.2.2. 3D City model by Aerial Images and Cadastral Map

Flamanc et al. (44) create buildings reconstruction framework for 3D city models production by using aerial images and cadastral maps. They tested model driven and the data driven approaches.

Figure 19 presents a 3D City model created using Aerial Images and Cadastral Map.

Figure 19: 3D City model by Aerial Images and Cadastral Map

6.1.2.3. 3D City model by Computer Vision Techniques

Lang and Forstner (45) describe a semi-automatic system for acquiring the 3D shape of buildings as topographic objects. Figure 20 shows a 3D Building model from one eye stereo camera.

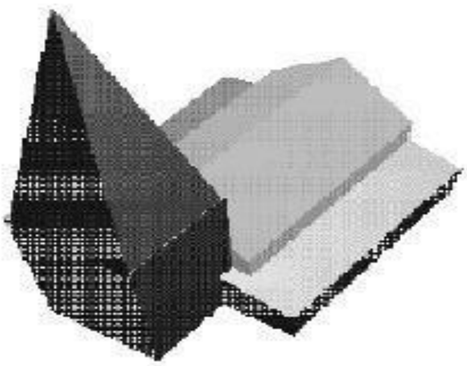


Figure 20: 3D Building model from one eye stereo camera

Pollefeys et al. (46) gives a methodology to create automated reconstruction of good textured 3D model from images sequences. The main advantage of this work is that, without camera parameters information, 3D model generated. This system is based on computer vision algorithms. In this method, the accuracy is not very good for metrology application, but due to photo-realistic texturing, visual quality is very convincing. So this system is useful for various applications in Archeology. They tested this approach for the Roman site of Sagalassos southwest Turkey.

---

[44] Flamanc D., Maillet G., Jibrini H., "3D City models: an operational approach using aerial images and cadastral maps", ISPRS Archives, Vol. XXXIV, Part 3/W8, Munich, 17.-19 Sept 2003

[45] Lang Felicitas, Forstner Wolfgang, 1996, "3D City modeling with a digital one-eye stereo system", International Archives of Photogrammetry and Remote Sensing, 261–266

[46] Pollefeys M., Koch R., Vergauwen M., Van Gool L., 2000, "Automated reconstruction of 3D scenes from sequences of images", ISPRS Journal of Photogrammetry & Remote Sensing 55, 251–267

Jang and Jung (47) create 3D City Model from Ground Images. Digital camera is used for ground images of an area. This camera is mounted with GPS and digital compass. The advantage of this method is that it can handle a huge number of images. They used an algorithm based on Structure from Motion (SFM) to correct their poses; they also used a method of global pose estimation that can register 3D isolated building models in a global coordinate system. This system is based on SFM and presented for large-scale 3D City modeling.

Jürgen Döllner et al. (48) create Virtual 3D City model of Berlin. In this project, they developed a system for integrating, managing, presenting, and distributing complex urban geoinformation. Virtual 3D city models, therefore, constitute a major concept in 3D geoinformation systems (3D GIS). In this projects Digital Aerials Photos, DTM, Geo-referenced thematic data, 3D Geodata, Digital Architecture models, and Cadastral data used as input and 3D Geodata base System and Virtual 3D City model comes as output products.

Cornelis et al. in (49) investigates about 3D Urban Scene Modeling Integrating Recognition and Reconstruction. In this proposed system, the input data are two video streams, recorded by a calibrated stereo rig mounted on top of a survey vehicle with GPS/INS measurements. The researchers used stereo camera pairs and by Real time Structure from Motion (SfM) concept and also used object detection techniques. The beauty of this work was is that they create Real time 3D reconstruction for a 3D City model.

Snavely Noah et al. (50) proposed a new method for modeling the world from internet photo collections. They introduced a concept of Photo-Tourism. They proposed an approach to create 3D model of any building or site by using unordered collection of photographs downloaded from internet. In this project they used the concept of structure-from-motion and image-based rendering algorithms. They tested this approach for Google image search for “Notre Dame” (Paris), Mount Rushmore, a set of photos of Mount Rushmore National Monument, South Dakota, Trafalgar Square (London), Half dome in Yosemite National park, Trevi Fountain (Rome), Sphinx (Giza), St. Basil’s Cathedral (Moscow), Colosseum in Rome and also Great Wall of China.

Jianxiong Xiao et al. (51) proposed an automatic approach to generate street-side 3D photo-realistic models from images captured along the streets at ground level. They propose an inverse patch-based orthographic composition and structure analysis method for facade modeling that efficiently regularizes the noisy and missing reconstructed 3D data. The main drawback of this method was that the upper parts of large buildings are not modeled due to the limited viewing field of a ground-based camera.

6.1.3. 3D City models by GIS

Gruen A. and Xinhua W. (52) developed a powerful system CyberCity Modeler (CC-Modeler). CyberCity-Modeler (CC-Modeler) is a methodology and software for the automatic generation of the topology of an unstructured 3D point cloud. It has been developed in order to generate structured data for city modeling from photogrammetrically measured points. It is specially designed for the handling of 3D city data and the integration of raster images and vector data in terms of a hybrid GIS.

[47] Jang Kyung Ho and Jung Soon Ki, 2006, “3D City Model Generation from Ground Images”, Eds.: H.P. Seidel, T. Nishita, and Q. Peng, CGI 2006, Springer-Verlag Berlin Heidelberg. LNCS 4035, 630 – 638
[48] Jürgen DÖLLNER, Konstantin BAUMANN, Henrik BUCHHOLZ, 2006, “Virtual 3D City Models as Foundation of Complex Urban Information Spaces”, CORP 2006 & Geomultimedia06, Vienna
[49] Cornelis Nico, Leibe Bastian, Cornelis Kurt, Van Gool Luc, 2007, “3D Urban Scene Modeling Integrating Recognition and Reconstruction”, International Journal of Computer Vision
[50] Snavely Noah, Seitz Steven M., and Szeliski Richard, 2008, “Modeling the World from Internet Photo Collections”, International Journal of Computer Vision, 80,189–210
[51] Jianxiong Xiao, Tian Fang, Peng Zhao, Maxime Lhuillier, Long Quan, 2009, “Image-based Street-side City Modeling”, SIGGRAPH ASIA
[52] Gruen, A., Wang, X., 1998, “CC Modeler: A Topology Generator for 3D City Models”, ISPRS Commission IV Symposium on "GIS - Between vision and application", Stuttgart/Germany, IAPRS. Vol. 32, Part 4, 188-196

Nedal Al-Hanbali et al. (53) Worked for a 3D GIS model of Jerash City and Artemis temple. They use the Photogrammetric principles and GIS method to create this model. The main aim of this work was to build a 3D virtual reality model of the Artemis Temple and to construct a GIS model of the Jerash City. These are the 3D GIS model and useful for very accurate measurements and detailed texture, which can allow visualization, preservation and reconstruction of the temple with City. This work is planned to be carried out from macro level, i.e. the modeling of the modem and ancient city via 3D GIS, to micro level, i.e. building 3D virtual models as well as 3D GIS databases for each monument.

Nedal Al-Hanbali et al. (54) makes a three dimensional Model for Yarmouk University by using GIS and Photogrammetry techniques.

Figure 21: 3D Model for Yarmouk University

Malumpong C., and Chen X. (55) used the interoperable 3D GIS City Modeling with GIS and 3D Modeling software. Google SketchUp Pro is used for reconstruction of buildings. This work aims to describe the integration of geo-informatics techniques with 3D modeling software to develop 3D GIS database which includes reconstruction of buildings, terrain, and other features that are relevant to a city model.

Zhou Li et al. (56) studied the 3D GIS based techniques and Digital Photogrammetry. The work discussed about the object-oriented 3D modeling technique used in existing digital photogrammetry software platform for secondary development.

Razzak A. et al. (57) developed 3D Virtual Maps for Mosul City by using GIS Techniques. The researchers followed the Virtual GIS methodology in their work. The final 3D digital map contained all necessary information for users and designers to reach to the best decision.

---

[53] Al-Hanbali Nedal, Al Bayari Omar, Saleh Bassam, Almasri Husam and Baltsavias Emmanuel, 2006, "Macro to Micro Archaeological Documentation: Building a 3D GIS Model for Jerash City and the Artemis Temple", Innovations in 3- D Geo-information systems, 447-468

[54] Al-Hanbali Nedal, Fedda Iyad, Awamleh Bashaar, and Dergham Mohannad, 2006, "Building 3D GIS Model of a University Campus for Planning Purposes: Methodology and Implementation aspects", Map Middle East, March 26-29

[55] Chayakrit Malumpong, Xiaoyong Chen, Interoperable Three-Dimensional GIS City Modeling with Geo-Informatics Techniques and 3D Modeling Software, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing 2008, pp. 975-980

[56] Zhou Li, Sun Jia-long, Li Wei-xiao, Bai He, and Chen Wei-wei, 2008, "The study on the techniques of the 3-D GIS modeling based on the digital Photogrammetry", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII, Part B, Beijing

[57] Razzak Abdul, Ziboon T., and Mohsin Amjed N., 2009, 3D Virtual Maps Production for Mosul City by using GIS Techniques, Eng. & Tech. Journal, Vol. 27, No.9

Thompson and Horne (58) worked for Virtual NewcastleGateshead (VNG) project and focus mainly on issues relating to data exchange, CityGML, data accessibility and interoperability in piloting Autodesk LandXplorer.

Figure 22: VNG Model extents

6.1.4. 3D City from Satellite Photogrammetry

Tao and Yong Hu (59) evaluated a concept of Rational Function Model (RFM) for 3D Reconstruction. Due to RFM parameters the Ortho-rectification, 3D reconstruction, and DEM generation is possible without the physical sensor model. In their study, the authors investigated two methods for RFM-based 3D reconstruction, the inverse RFM and the forward RFM method. They concluded that the forward RFM can achieve better reconstruction accuracy. Finally, real Ikonos stereo pairs were used to verify the applicability and the performance of the 3D reconstruction method.

Fraser et al. (60) investigated with Ikonos imagery for sub metre 3D positioning and building extraction. A quantitative and qualitative assessment, using stereo Ikonos imagery for generating building models, for the campus of the University of Melbourne was achieved.

Kocaman et al.61 tested a methodology for 3D City modeling from high resolution satellite images with the help of software called SAT-PP, (Satellite Image Precision Processing) and CyberCity Modeler software. They tested the potential of SAT-PP and CyberCity Modeler to extract DSMs and buildings, and to produce 3D city models with IKONOS and Quickbird stereo images.

[58]Thompson Emine Mine and Horne Margaret, 2009, "3D g28GIS Integration for Virtual NewcastleGateshead, School of Built Environment, University of Northumbria, England
[59] Tao C. Vincent and Hu Yong, 2002, "3D Reconstruction methods based on Rational Function Model", Photogrammetric Engineering and Remote Sensing Vol. 68, No. 7, July, 705-714
[60] Fraser C.S., Baltsavias E., Gruen A., 2002, Processing of Ikonos imagery for sub metre 3D positioning and building extraction, ISPRS Journal of Photogrammetry & Remote Sensing 56, 177– 194.
[61] Kocaman S., Zhang L., Gruen A., Poli D., 2006, "3-D City modeling from high resolution satellite images", Proc. of the ISPRS Conference Topographic Mapping from Space (With Special Emphasis on Small Satellites)

Tack et al. (62) developed and tested a methodology for semi-automatic city model extraction from tri-stereoscopic very high resolution satellite imagery. They studied the IKONOS triplet data for city Istanbul, Turkey with photogrammetric software platform, called SAT-PP, (Satellite Image Precision Processing).]

6.1.5. 3D City model from Single Satellite Image

Huang et al. (63) proposed a method for objects reconstruction from single high-resolution satellite image using Monoplotting technique. The authors used Rational Polynomial Coefficients (RPC) and high resolution of Digital Elevation Models (DEM) to achieve the high precision geometry of the 3D reconstructed object. The ray of the satellite determines from RPC and shadow of the object on the ground, determine from the Azimuth and elevation angles of the sun. The vertices of the polygon, manually extracted from top of the surface of an object are used for the prediction of the base and shadow positions on the 2D satellite image based on the height of the object. The height of the object is determined from predicted base and shadow positions match their position in the image by observation. They tested this method on IKONOS image and also developed software for real-time extracting, editing, reconstructing and visualizing of the 3D objects from single IKONOS image.

Izadi, and Saeedi (64) investigated a method for Three-Dimensional Polygonal Building Model using Single Satellite Images. The researchers developed a methodology and a system for the automatic detection and height estimation of buildings with polygonal shape roofs in single satellite images. The proposed system includes two main parts: 2-D Rooftop Detection and 3D Building Estimation. This system is capable of detecting multiple flat polygonal buildings with no angular constraints or shape priors. In this project, they verify the effectiveness of the presented system with overall mean shape accuracy of 94% and mean height error of 0.53 m on QuickBird satellite (0.6 m/pixel) imageries.

6.1.6. 3D City from Panorama Photogrammetry

Luhmann, and Tecklenburg (65) discussed about 3D Object reconstruction from multiple-station panorama imagery. They describe about image acquisition, panorama generation by frame imagery and by rotating line-scanner imagery, calibration, tie point and control point measurement, panorama bundle adjustment to the final image compilation of 3D objects. The authors suggested that, due to the stable geometry of the cylindrical panorama model the bundle adjustment can be performed with very few object points. Once each panorama is oriented with respect to global coordinate system, then photogrammetric object reconstruction procedures such as space intersection or a moving floating mark can be applied and finally 3D model created. In this study, they tested this work for entrance hall of the university as 3D wireframe model of the interior and Great Hall of the Oldenburg castle. They created four panoramas and processed to produce a high-quality color panorama.

In Fangi G.66 highlighted the importance of Spherical Photogrammetry for 3D cultural heritage. They tested and suggested that, Spherical-Panorama techniques are also useful to create virtual

---

[62] Tack F., Goossens R., Buyuksalih G., "Semi automatic city model extraction from tri-streoscopic VHR satellite imagery", In: Stilla U, Rottensteiner F, Paparoditis N (Eds) CMRT09. IAPRS, Vol. XXXVIII, Part 3/W4, Paris, France, 3-4 September 2009

[63] Huang Xiaojing, Kwoh Leong Keong, 2008, "Monoplotting- A semi-automated approach for 3-D reconstruction from single satellite image", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII, Part B3b. Beijing

[64] Izadi Mohammad, and Saeedi Parvaneh, 2012, "Three- Dimensional Polygonal Building Model Estimation From Single Satellite Images", IEEE Transactions on Geoscience and Remote sensing Vol. 50, NO. 6, June, 2254-2272

[65] Luhmann Thomas, and Tecklenburg Werner, 2004, "3-D Object reconstruction from multiple-station panorama imagery", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 34, No. 5/W16: 8

[66] Fangi G., 2009, "Further developments of the spherical Photogrammetry for culture heritage", 22nd CIPA Symposium, October 11-15, Kyoto, Japan

3D model. Koseck and Micusık (67) provided a methodology for Piecewise Planar City 3D Modeling from Street View Panoramic Sequences. In figure 23 a detailed view from a city model reconstruction from car image sequences is shown.

Figure 23: City model reconstruction from car image sequences

6.1.7. 3D City Models from Video

For 3D city modeling, the video is an easy method to obtain low cost data. From this point of view, many researchers have increased their interest in videogrammetry, which is a measurement technique mainly based on the principles of photogrammetry (68). Videogrammetry refers to video images taken using a camera recorder or movie function on digital still camera. Video movie consists of sequences of images (or frames). If video speed is 25 fps (frame per second) and taken for 1 minute (i.e. 60 seconds), there are 25 frame per second or overall 1500 image.

Clip et al. (69) designed a Mobile 3D City Reconstruction system as an efficient flexible capture and reconstruction system for the automatic reconstruction of large scale urban scenes. This system is both backpack and vehicle mounted allowing the capture of interior or less accessible areas as well as large outdoor scenes. In the work, the authors propose an efficient capture system to capture the 3D-geometry of existing cities through computer vision techniques. The system is able to deliver 3D reconstructions of large urban scenes with near real time, and is modular and man portable, able to record both from a backpack mounting for interior areas and from an automobile for exterior

[67] J. Kosecka, B. Micusik, "Piecewise planar city 3D modeling from street view panoramic sequences", 2013 IEEE Conference on Computer Vision and Pattern Recognition, vol. 00, no. , pp. 2906-2912, 2009, doi:10.1109/CVPRW.2009.5206535

[68] Gruen A., Fundamentals of videogrammetry — A review, Human Movement Science 16(2-3):155-187, April 199, DOI: 10.1016/S0167-9457(96)00048-6

[69] Clipp Brian, Raguram Rahul, Frahm Jan-Michael, Welch Gregory, Pollefeys Marc, 2008, “A Mobile 3D City Reconstruction System”, UNC Chapel Hill, ETH, Zurich

recording. GPS and INS also used in the product. Figure 6.7 shows a Video only 3D reconstruction with the backpack system.

Zhao et al. (70) gives a concept of Alignment of Continuous Video onto 3D Point Clouds. They work on video data and LiDAR data and propose a general framework for aligning continuous (oblique) video onto 3D sensor data. They align a point cloud computed from the video onto the point cloud directly obtained from a 3D sensor. The capability to align video before a 3D model is built from the 3D sensor data offers new practical opportunities for 3D modeling. They introduce a novel modeling, through registration approach, that fuses 3D information from both the 3D sensor and the video.

Figure 24: Video only 3D reconstruction with the backpack system

Fulton and Fraser (71) explained a method for automatic reconstruction of building by using a handheld video camera. In this method, a video recording takes of a building of interest. Video sequence transferred into computer and saved as individual JPEG Frames. Blurred frames remove, and Non-blurred key frames selected and registered these Non-blurred key frames by using Phase correlation method and after this feature extraction done.

Colleu et al. (72) gives a method for Automatic Initialization for the registration of GIS and Video data. The main aim of this work is to automatically compute the initial registration of a GIS model and a video sequence. In this method a coarse registration is obtained using GPS data and the theory of epipolar geometry. Then, a simultaneous pose and correspondence determination is done and RANSAC algorithm applied on line features. They also made a system which generates a 3D virtual map automatically with a VRML form.

Zhang et al. (73) gives a concept for Consistent Depth Maps Recovery from a Video Sequence. Video Image sequence frame used and created Depth maps from these frames. In this method, they used the Structure From Motion (SFM) to recover the camera parameters, Disparity Initialization, Bundle optimization, and Space-Time fusion techniques to create depth maps. These depth maps are very useful to create virtual 3D model of an area or object. Figure 25 shows video depth results of the “Stair” and “Great Wall” sequences. The “Stair” sequence is taken by a vertically moving camera and the “Great Wall” sequence, the camera moves surrounding the beacon on the

[70] Zhao W., Nister D., and Hsu S., 2005, “Alignment of Continuous Video onto 3D Point Clouds”, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 Aug., 1305-1318
[71] Fulton J.R. and Fraser C.S., 2009, “Automated Reconstruction of Buildings using a Hand Held Video Camera”, S. Jones, K. Reinke (eds.), Innovations in Remote Sensing and Photogrammetry, Lecture Notes in Geoinformation and Cartography, Springer-Verlag Berlin Heidelberg, 393-404
[72] Colleu Thomas, Sourimant Gaël, and Morin Luce, 2008, “Automatic Initialization for the Registration of GIS and Video Data”, 3DTV, May, Istambul, Turkey
[73] Zhang Guofeng, Jia Jiaya, Wong Tien-Tsin, and Bao Hujun, 2009, “Consistent Depth Maps Recovery from a Video Sequence, IEEE Transactions on pattern analysis and machine intelligence, June, Vol. 31, No. 6, 974-988

mountain. Similar to all other examples (73), both of these sequences contain video noise and complex occlusions. The recovered dense depth maps demonstrate the robustness of the method.

Figure 25: Video depth results of the (left) "Stair" and (right) "Great Wall" sequences

Figure 26: Reconstruction buildings

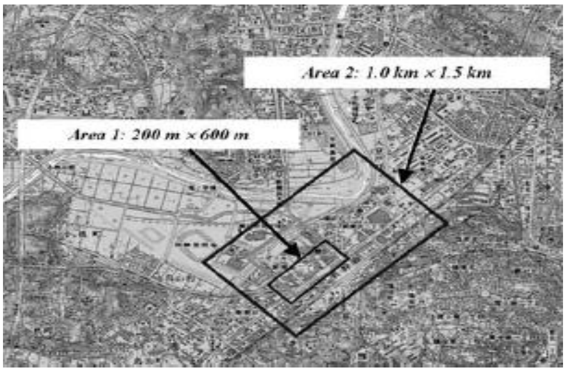
Pollefeys et al. (74) created a detailed real time urban reconstruction from video. The system collects video streams, as well as GPS and inertia measurements in order to place the reconstructed models in geo-registered coordinates. It is designed using current state of the art real-time modules for all processing steps. It employs commodity graphics hardware and standard CPU's to achieve real-time performance. The authors presented the main considerations in designing the system and the steps of the processing pipeline. The proposed system extends existing algorithms to meet the robustness and variability necessary to operate out of the laboratory. Some screenshots of 3D models resulted are shown in figure 6.9 and 6.10. All models are produced by more than one camera using GPS/INS data, except the one at the top left which is a vision-only reconstruction from a single camera.

[74] Pollefeys M., Nistér D., Frahm J.M., Akbarzadeh A., Mordohai P., Clipp B., Engels C., Gallup D., Kim S.J., Merrell P., Salmi C., Sinha S., Talton B., Wang L., Yang Q., Stewénius H., Yang R., Welch G., Towles H., 2008, "Detailed Real-Time Urban 3D Reconstruction from Video", International Journal of Computer Vision, 78: 143–167

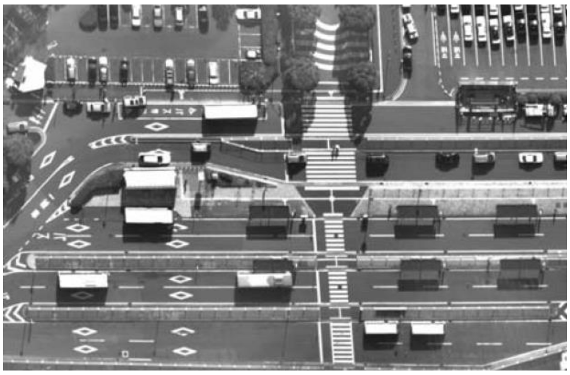
Figure 27: View from above and details of large scale reconstructions

6.1.8. 3D City Modeling with TLS data

The TLS (Three-Line-Scanner) system is an aerial multi-spectral digital sensor system, developed by STARLABO Corporation, Tokyo. It utilizes the Three-Line-Scanner principle to capture digital image triplets in a long-strip mode. It has panchromatic sensors of forward, nadir and backward direction, and also has multi-spectral sensors of RGB. The imaging system contains three times three parallel one-dimensional CCD focal plane arrays, with 10 200 pixels of 7μm each. The TLS system produces seamless high-resolution images with usually 5 - 10 cm footprint on the ground with three viewing directions (forward, nadir and backward). Gruen, et al. (75) tested the TLS system data to create a virtual 3D City model. The authors interfaced TLS data with CyberCity Modeler functionality and have produced two data sets over the city of Yokohama. The combination of two modern technologies from sensing and processing opens interesting perspectives for future applications in 3D virtual environment generation.

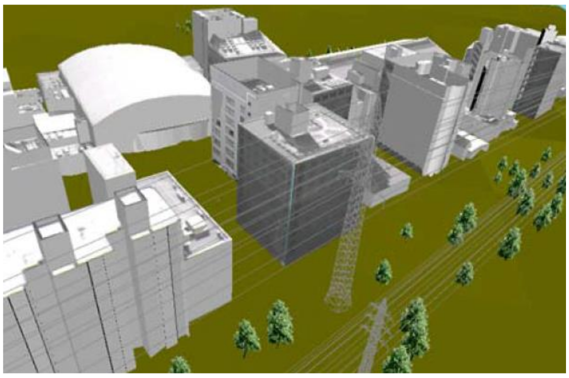


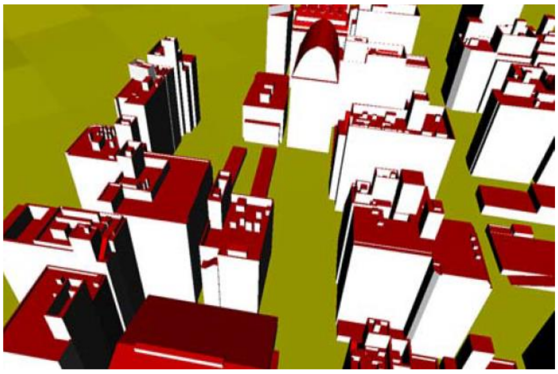
a)



b)

[75] Gruen Armin, Li Zhang, Wang Xinhua, 2003, "3D City Modeling with TLS (Three-Line Scanner) Data", International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXIV- 5/W10, 24-27





c) d)

Figure 28. - 3D model of Yokohama City

The first project includes an experimental area in downtown Yokohama, Japan, presented in figure 28 a) and a detailed view in figure 28 b). After the stereoscopic measurement module "TLS-SMS" as the measurement platform to measure the point clouds, followed the regulation of CC-Modeler the integration of CC-Modeler and TLS-SMS was crucial for the 3D model generation. After measuring all the objects, CC-modeler is employed to construct the 3D models. Figure 28 c) and 28 d) shows the views of the reconstruction results. The level of detail in reconstruction can be checked by the roof structures.

6.1.9. 3D City by Unmanned Aerial Vehicle (UAV)

Puschel et al. (76) created a high-resolution textured 3D-Model of the Landenberg castle combining terrestrial and UAV Photogrammetry. For the generation of a highly accurate 3D model, a combined photogrammetric processing of the UAV images was required. The conducted workflow consisted of: project planning, flight planning, terrestrial and aerial image acquisition, camera calibration, control point measurement using GNSS (Global Navigation Satellite System), measurement of control and tie points in the images using the photogrammetric close-range processing software Photomodeler 6, modelling and texturing of the object and visualization of the finalized model. The researchers' results consisted of a textured 3D model of Castle Landenberg in Blender shown in figure 29.

Figure 29: Textured 3D model of Castle Landenberg

[76] Püschel Hannes, Sauerbier Martin, Eisenbeiss Henri, A 3D model of Castle Landenberg from combined photogrammetric processing of terrestrial and UAV-based images, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B6b. Beijing 2008, pp. 93-98

Figure 30: Residuals at image plan and visual overlay accuracy direct vs. integrated geo-referencing solution

Eugster and Nebiker (77) studied about the real time Geo-registration of video stream from mini or micro UAS using Digital 3D City models. The researchers presented and compared in the work two different fully automated video imagery geo-registration approaches. The first geo-registration approach is based exclusively on the flight control states provided by the low-quality IMU/GNSS sensors and the second approach, additional 3D city model data – provided by a virtual globe technology – was used to improve the geo-registration accuracy. In figure 30, the remaining residual vectors of check points are shown for two arbitrarily chosen epochs. Additionally, the geo-registration accuracy with the overlaid 3D city model is visualized for the same two epochs.

6.1.10. LASER Based 3D City Models

Vosselman and Dijkman (78) recommend a methodology and algorithm for 3D building model reconstruction from point clouds and ground plan. They used the well-known Hough transform for the extraction of planar faces from the irregularly distributed point clouds. They explored two different strategies to reconstruct building models from the detected planar faces and segmented ground plans. The first strategy relied on the detection of intersection lines and height jump edges between planar faces and the second strategy adopted coarse initial models that were refined on the bases of fitting models to point clouds that did not correspond to the initial models. In figure 31 a reconstructed model and a photograph of the building from the same perspective is shown.

---

[77] Eugster H. and Nebiker S., “Real time Geo- registration of video stream from mini or micro UAS using Digital 3-D City models”, 6th International Symposium on Mobile Mapping Technology, Presidente Prudente, São Paulo, Brazil, 21-24 July, 2009

[78] Vosselman George and Dijkman Sander, 2001, “3D building model reconstruction from point clouds and ground plan”, International Archives of Photogrammetry and Remote Sensing, Volume XXXIV-3/W4 Annapolis, MD, 22-24 Oct

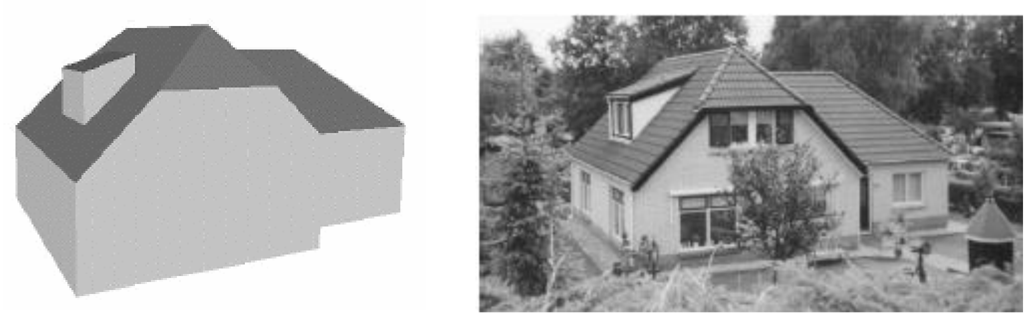


Figure 31: Reconstructed model and a photograph of the building from the same perspective

Dorninger and Pfeifer (79) proposed a comprehensive approach for automated determination of 3D city models from airborne acquired point cloud data. They used a comprehensive automated 3D approach for building extraction, reconstruction, and regularization from airborne laser scanning point clouds. This work assumes that individual buildings can be modeled properly by a composition of a set of planar faces. So, this work is based on a reliable 3D segmentation algorithm, detecting planar faces in a point cloud. Figure 32 right shows the final model as a rendered 3D city model. The texture maps have been acquired by airborne photogrammetry (roofs) and from terrestrial viewpoints (facades). They were mapped onto a 3D triangulation of the city model.

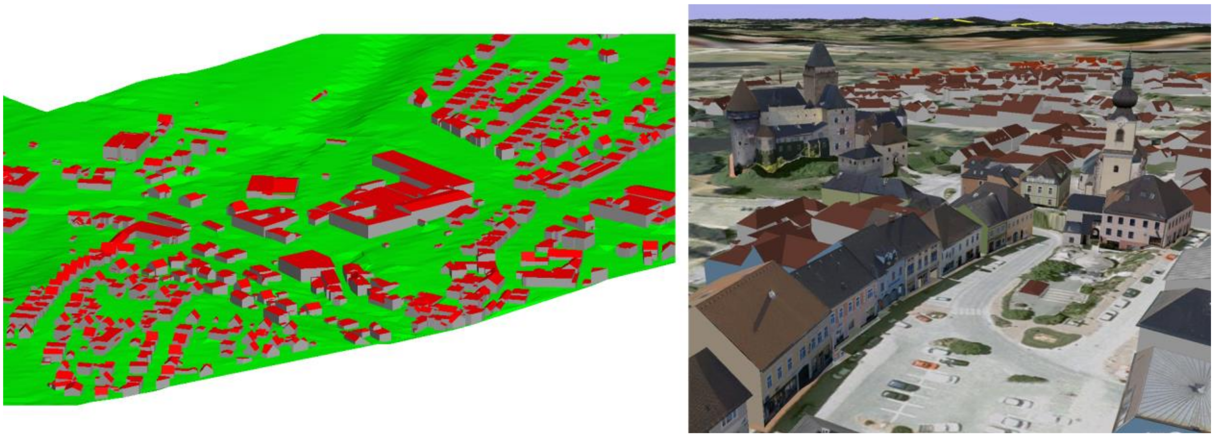


Figure 32: Rendered city model correctly intersected with a DTM triangulation (left); 3D city model with manually applied textures from terrestrial and airborne images (right)

Ming et al. (80) investigated and proposed the methodology and algorithms for automatic Generation of Three Dimensional Photo-Realistic Models from LiDAR and Image Data. They implemented automatic 3D point cloud registration, automatic target recognition that is used for geo-referencing and automatic plane detection algorithm that is used for surface modeling, and texture mapping. They proposed an approach that is useful to create the accurate and geo-referenced 3D photo-realistic models from point clouds and digital imagery. Figure 33 shows textured point cloud model with laser and Images.

[79] Dorninger Peter and Pfeifer Norbert, 2008, "A Comprehensive Automated 3D Approach for Building Extraction, Reconstruction, and Regularization from Airborne Laser Scanning Point Clouds", Published in Sensors, 8, 7323-7343
[80] Ming J. Li-Chee, Gumerov D., Ciobanu T., Armenakis C., 2009, "Generation of Three Dimensional Photo-Realistic Models from LiDAR and Image Data", IEEE-TIC-STH

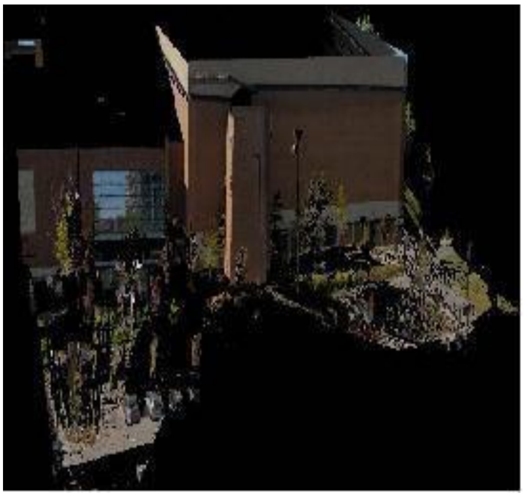


Figure 33: Textured point cloud model with laser and Images

6.1.11. 3D City modelling by using Mobile Mapping System (MMS)

The idea of mobile mapping has been around for at least as long as photogrammetry has been practiced. The early development of mobile mapping systems (MMS) was, however restricted to applications that permitted the determination of the elements of exterior orientation from existing ground control. In 1989, the Center for Mapping of the Ohio State University established a major research program that focuses on the development of Mobile Mapping Systems (MMS). These devices capture a comprehensive set of land-related data from airplanes, cars, or trains. Spatial positions and attributes of objects are extracted automatically on the mobile platform or during post-processing and are immediately transferred to a multi-media geographic database (81). The University of Calgary and GEOFIT INC., a high- tech company in Laval, Quebec, Canada have jointly developed the same system, The system named VISAT, stands for Video-Inertial-SATellite and integrates inertial and GPS technology with a cluster of CCD cameras (82).

Blaer and Allen (83) designed and developed a system for automatic view planning (Vue-Plan). They used Mobile robot and laser scanner for data capture to create a voxel based Next Best View (NBV) and create a final accurate and complete 3D model. They also developed a simulator tool to test view planning algorithm on simulated sites. They successfully tested this work on to construct precise 3D models of real- world sites located in New York City: Uris Hall on the campus of Columbia University and Fort Jay on Governors Island.

Pretto et al. (84) designed a scalable dense 3D reconstruction and navigation system suitable for real- time operation. They successfully tested in a challenging urban scenario along a large loop using an Omni-directional camera mounted on the roof of a car.

[81] Novak, K., 1993, "Mobile mapping systems: new tools for the fast collection of GIS information". In Optical Engineering and Photonics in Aerospace Sensing, International Society for Optics and Photonics, October, 188-198
[82] El-Sheimy, N., 1996, " A mobile Multi-Sensor System for GIS Applications in Urban Centers.", International Archives of Photogrammetry and Remote Sensing, Vol. XXXI, Part B2, Vienna, July 9-19
[83] Blaer Paul S. and Allen Peter K., 2009, "View Planning and Automated Data Acquisition for Three-Dimensional Modelling of Complex Sites", Journal of Field Robotics 26 (11–12), 865–891
[84] Pretto Alberto, Soatto Stefano, and Menegatti Emanuele, 2010, "Scalable Dense Large-Scale Mapping and Navigation", Proceedings of the 2nd. Workshop on Omni- directional Robot Vision, A workshop of the IEEE, International Conference on Robotics and Automation (ICRA2010), Anchorage, Alaska, USA, May 7, 49-56

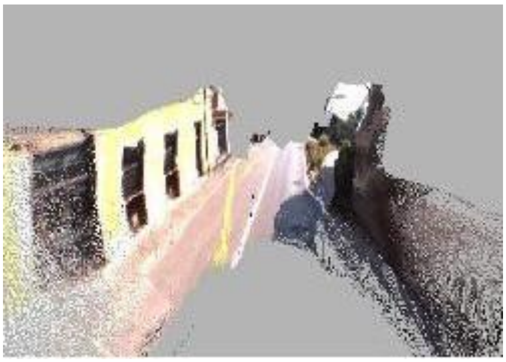
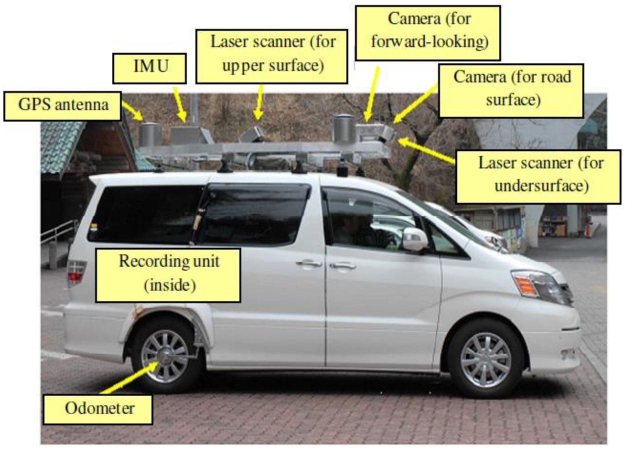



Figure 34: Vehicle with camera (left); Dense 3D reconstruction (right)

Imanishi et al. (85) in Japan, they suggested a method for accuracy of MMS data at GPS invisible area. This Land Mark Update (LMU) technique, corrects MMS vehicle position using control points is employed to maintain accuracy. It is useful for MMS measurement data at GPS invisible area. This method is very useful and valid for tunnel mapping and/or very busy main road mapping where re-observation is difficult and it is usable to create road management map quickly and precisely.



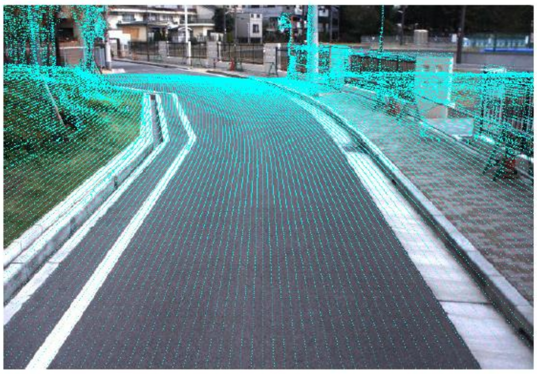


Figure 35: System configuration of MMS (TYPE-S) (left); MMS measurement data (right)

6.1.11.1. Hybrid Methods

3D City model by Aerial Laser and Terrestrial Laser

Böhm and Haala (86) developed a methodology for efficient integration of aerial and terrestrial laser data for virtual city modelling using LaserMaps. They used Leica HDS 3000 Terrestrial Laser scanner and TopScan"s OPTECH ALTM 1225 Airborne LiDAR. Terrestrial lasers scanning (TLS) used to collect building facades at a large amount of geometric detail, so the LaserMaps created from this methodology. This approach, which is frequently used for visualization applications, requires only a suitable interpolation and mapping of the terrestrial data against the planar facades of the coarse building model in order to present the geometric details as they are available from TLS. Figure 36 shows the super-imposition of the terrestrial laser scanner data, which was directly

[85] Imanishi Akihisa, Tachibana Kikuo and Tsukahara Koichi, 2011, "The Development of Accuracy Maintenance Method for Mobile Mapping System (MMS) Data at GPS Invisible Area", FIG Working Week, Bridging the Gap between Cultures Marrakech, Morocco, 18-22, May

[86] Böhm Jan, Haala Norbert, 2005, "Efficient integration of aerial and terrestrial laser data for virtual city modeling using LaserMaps", ISPRS WG III/3, III/4, V/3 Workshop "Laser scanning 2005", Enschede, the Netherlands, 12-14 September

geo-referenced by the low cost components to the DEM from airborne LIDAR and the available 3D building model, respectively.

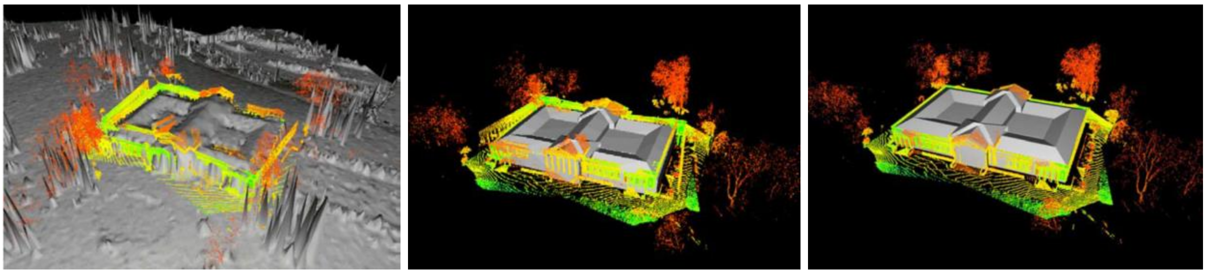


Figure 36: Super-imposition of directly georeferenced terrestrial laser scanner data and aerial LIDAR (left) and virtual city model (centre) with result after alignment (right)

Visintini et al. (87) recommended an approach to create Gorizia old downtown (Italy) 3D City model of the area by matching historical maps with aerial and terrestrial surveying techniques. The model production was obtained by a suitable integration of laser and imaging data, both acquired either by means of a helicopter and a terrestrial surveying. The virtual reconstruction can be compared with different perspective maps of the city to underline its transformation. An aerial high-density (15 points/m^2) laser surveying was achieved with an Optech ALTM 3033 system for an area of almost one squared kilometre. High-resolution Rollei DB44 digital images were also acquired for the same area. The filtering, classification and 3D building modeling have been successfully carried out by the well-known TerraScan software (Terrasolid); moreover, the high-resolution true orthophoto (0,2 m/pixel) has been accomplished by means of TerraPhoto (Terrasolid). To fulfill a photo-realistic 3D city model, the aerial orthophoto has been mapped onto the DTM and the building roofs. Nevertheless, to truthfully represent also the building façades of some streets and squares, even a terrestrial survey was carried out. Laser scanning was done by Riegl LMS Z360i system, and a simultaneous acquisition of digital images by "on system" Nikon D100 was carried out. Thanks to the reflecting targets set on the facades, topographically measured and automatically detected, the different terrestrial models have been registered and geo-referenced in the same datum (Gauss-Boaga Italian cartographic frame) of the aerial laser surveying. Finally, the paper shows the historical documents (maps, photos etc.) combined with the aerial and terrestrial orthophotos wrapped on the different surfaces and the extremely faithful 3D visualization of the city reached. Figure 37 shows the façades enhanced thanks to the terrestrial survey: first adding the laser scans, later wrapping the orthophotos.

Figure 37: Terrestrial laser scans of the building facades (left) and terrestrial ortho-photos wrapped on the facades (right)

[87] Visintini D., Guerra F., Adami A., Vernie P., 2007, "A 3D Virtual model of the Gorizia downtown (Italy) by matching historical maps with aerial and terrestrial surveying techniques", e-Perimetron, Vol. 2, No. 3, 117-133

Combination of Laser and Photogrammetry based 3D City Models

3D City model by Airborne images and LiDAR Data

Habib et al. (88) proposed a methodology to create a realistic 3D city by integration of LiDAR and Airborne. The authors present a framework for integrating photogrammetric and LiDAR data for realistic visualization of 3D urban environments. We will start by illustrating the benefits of integrating photogrammetric and LiDAR data. A methodology for the registration of LiDAR and photogrammetric data has been introduced in order to have a common reference frame. More specifically, conjugate features will be used to geo-reference the imagery relative to the LiDAR reference frame.

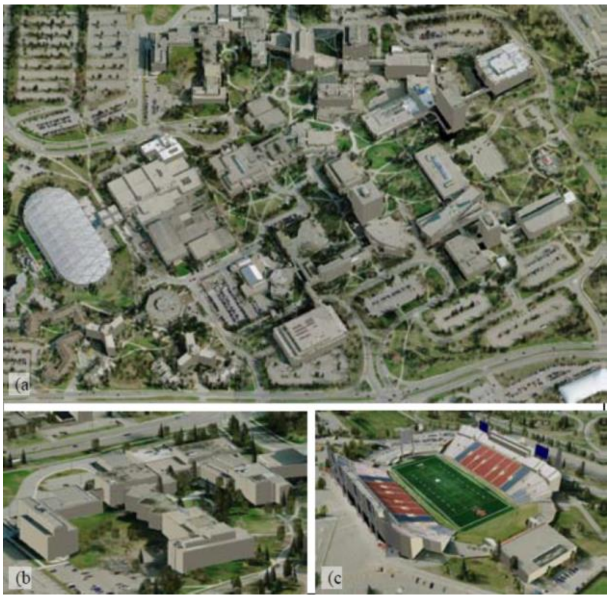


Figure 38: Realistic Views of Main Campusf

True orthophoto generation techniques will be incorporated to link the spectral and positional information in the imagery and LiDAR data while circumventing problems encountered when dealing with large scale geo-spatial data over urban environments. Finally, generated orthophotos will be draped over the LiDAR surface model for realistic visualization of an urban environment. Several techniques for refining building boundaries in the LiDAR data and in the orthoimage have been presented for the purpose of improving the quality of the derived visualization.

3D City model by Aerial Image and Ground Laser

Fruh and Zakhor described the techniques for 3D textured model construction of 3D City using camera and laser scanner. They used a truck with one camera and two laser scanner travels on City Street under normal traffic condition. One horizontal and one vertical laser scanner used. They developed the methodologies which are based on correlation techniques and Markov Carlo Localization. They used digital road maps and aerial photographs with laser scans and created a fairly accurate textured 3D Model of the area. In 2003, they also developed a textured 3D city model of Berkeley campus. Früh and Zakhor (89) developed and tested a methodology to construct 3D City models by using aerial and ground view. They used an Aerial laser and aerial images with

[88] Habib A. F., Kersting J., McCaffrey T. M., Jarvis A. M. Y., 2008, "Integration of LiDAR and airborne imagery for realistic visualization of 3D urban environments", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2, Beijing

[89] Früh, C. and Zakhor, A. (2003). Constructing 3D City Models by Merging Aerial and Ground Views. IEEE Computer Graphics and Applications, Vol. 23 No. 6, pp. 52-61

ground laser and ground images. They create airborne model and façade model and give a final fused textured model of a city.

Figure 39: Texture mapped model of the Berkeley Campus

3-D City model by Aerial Laser and Ground Laser with Aerial images and Ground Images:

Yoshihiro Kobayashi (90) used ANN for 3D scene reconstruction for the Arizona State University. He developed 3D City Model Generator by using Neuro-Fuzzy concept in CAD. He introduces a computer-aided design (CAD) system in which a neuro-fuzzy system is integrated as a main engine for learning. Specifically, a computer system that generates 3D city models from satellite images is formulated, implemented, and tested. Techniques from neural networks, fuzzy systems, image processing, pattern recognition, and machine learning constitute the methodological foundation of the system. The usability and flexibility of the system are evaluated in case studies. The main purpose of this research is to develop a system that can automatically generate the 3D models from satellite images.

Denipote Juliana and Assaf Rodrigo (91) give the concept of elements classification in a 3D urban virtual environment using artificial neural networks. In this work, with a good set of samples, the Multi-Layer Perceptron (MLP) network is a good solution for objects classification for creating a 3D virtual city.

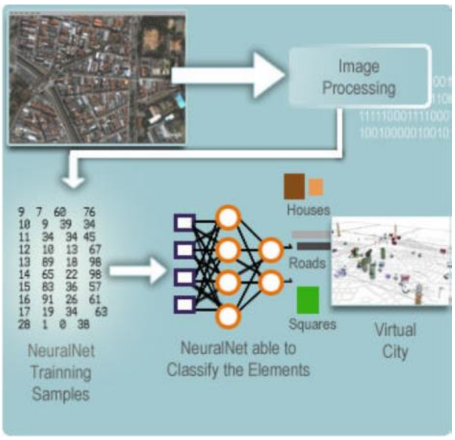


Figure 40: Solution for creating a virtual 3D City model generation using ANN

---

[90] Yoshihiro Kobayashi, 2002, "3D City Model Generator: The Application of Neuro-Fuzzy Systems in CAD", Arizona State University, USA

[91] Denipote Juliana and Assaf Rodrigo, Classification of Elements in an 3D Urban Virtual Environment Using Artificial Neural Nets

For a best performance of the MLP training, the image processing must reach a satisfactory accuracy during the segmentation stage in order to result on high quality samples for the neural network. With a good set of samples, the MLP network is a good solution for objects classification for creating a 3D virtual city. The chosen neural net architecture and segmentation method were good solutions for the problem of the automatic VE creation with low error rates.

Yoshihiro Kobayashi and Kostas Terzidis (92) also work with AA, ANN, and DIP. The main goal is to create a CAAD system that detects buildings from satellite images and produces computer city models allowing the system's users to manipulate the models utilizing machine learning technology. Soft computing technologies mainly neural networks and fuzzy systems are applied and tested as the system's methodology.

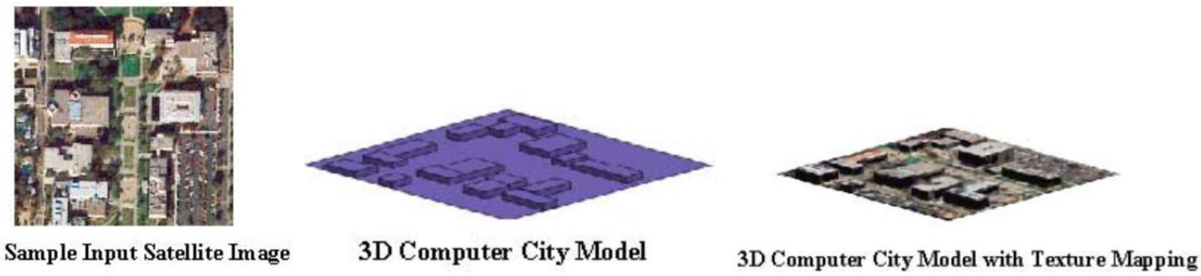


Figure 41: Output Image of 3D Computer City

6.2. CityGML

CityGML is a common information model and XML-based encoding for the representation, storage, and exchange of digital 3D city and landscape models. It defines the classes and relations for the most relevant topographic objects in cities and regional models with respect to their geometrical, topological, semantical, and appearance properties. "City" is broadly defined to comprise not just built structures, but also elevation, vegetation, water bodies, "city furniture", and more. Included are generalization hierarchies between thematic classes, aggregations, relations between objects, and spatial properties. This thematic information goes beyond graphic exchange formats and allow to employ virtual 3D city models for sophisticated analysis tasks in different application domains like simulations, urban data mining, facility management, and thematic inquiries. For specific domain areas, CityGML also provides an extension mechanism to enrich the data with identifiable features under preservation of semantic interoperability. The rich and general-purpose information model provided by CityGML is especially important with respect to the cost-effective sustainable maintenance of 3D city models, allowing for the possibility of selling the same data to customers from different application fields.

CityGML is applicable for large areas and small regions and can represent the terrain and 3D objects in different levels of detail simultaneously. Since either simple, single-scale models without topology and few semantics or very complex multi-scale models with full topology and fine-grained semantical differentiations can be represented, CityGML enables lossless information exchange between different GI systems and users.

CityGML is implemented as an XML application schema for the Geography Markup Language version 3.1.1 (GML3). GML3 is the extensible international standard for spatial data exchange issued by the Open Geospatial Consortium (OGC) and the ISO TC211. CityGML has been adopted as official international standard by the OGC (as an approved GML3 application schema), enabling

[92] Kobayashi Yoshihiro and Kostas Terzidis, Extracting the Geometry of Buildings from Satellite Images using Fuzzy Multiple Layer Perceptrons, Proceedings of the 4th Iberoamerican Congress of Digital Graphics, 236-239. SIGraDi. Rio de Janeiro, Brazil, 2000

easy and free access to all the international community (93). CityGML provides a standard model and mechanism for describing 3D objects with respect to their geometry, topology, semantics and appearance, and defines five different levels of details. Included are also generalisation hierarchies between thematic classes, aggregations, relations between objects, and spatial properties. CityGML is highly scalable (extensible with respect to a theme) through CityGML Application Domain Extensions (ADE), and datasets can include different urban entities supporting the general trend toward modelling not only individual buildings but also wider sites, districts, cities, regions, and countries. The CityGML standard (Open Geospatial Consortium, 2012) defines five LoDs, and its excerpt on this topic is listed below. An example of a house represented in CityGML with different LoDs is shown in Fig. 42 (94).

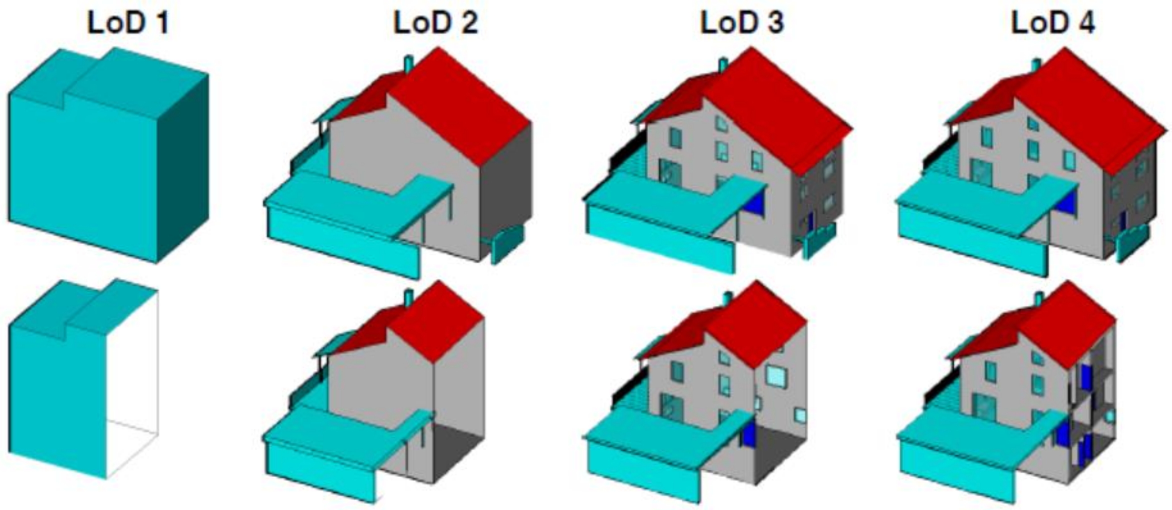


Figure 42: LoDs of CityGML

CityGML supports different Levels of Detail (LOD). LODs are required to reflect independent data collection processes with differing application requirements. Further, LODs facilitate efficient visualisation and data analysis. In a CityGML dataset, the same object may be represented in different LOD simultaneously, enabling the analysis and visualisation of the same object with regard to different degrees of resolution. Furthermore, two CityGML data sets containing the same object in different LOD may be combined and integrated. However, it will be within the responsibility of the user or application to make sure objects in different LODs refer to the same real-world object. Further the standard defines:

- The coarsest level LOD0 is essentially a 2.5D Digital Terrain Model (DTM) over which an aerial image or a map may be draped. Buildings may be represented in LOD0 by footprint and/or roof edge polygons (2D shape in 3D space). If a building is represented by both its footprint and the roof edge polygon, the polygons are stored separately, which means that models in LOD0 do not contain volume and are not 3D objects.

LOD0 is used for regional and landscape applications.

- LOD1 is the well-known blocks model comprising prismatic buildings with flat roof structures. This level is used for city and region coverage.
- In contrast, a building in LOD2 has differentiated roof structures and thematically differentiated boundary surfaces. LOD2 is most suitable for city districts and projects.

[93] http://www.citygmlwiki.org/index.php?title=Citygml_Wiki.
[94] Biljecki F., The concept of level of detail in 3D city models, GISt Report No. 62, Feb 2013

- LOD3 denotes architectural models with detailed wall and roof structures potentially including doors and windows. It is mostly used for landmarks.
- LOD4 completes a LOD3 model by adding interior structures for buildings. For example, buildings in LOD4 are composed of rooms, interior doors, stairs, and furniture.
- In all LoDs textures can be mapped onto the structures.

Therefore, practically the relation between LODs could be seen in the following manner:

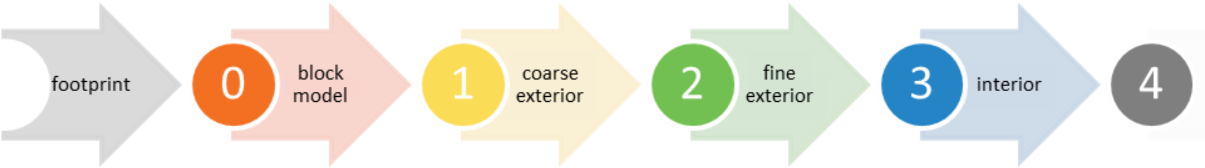


Figure 43: Relation between LoDs

A few observations can be derived from the above definitions, the standard, and practical usage of CityGML data:

- It can be argued that the LoD0 cannot be considered as a 3D city model since it is a boundary representation in 2D with a height as an attribute.
- Neglecting minor improvements in the exterior, LoD3 is basically in most of the cases a LoD2 with openings (e. g. windows and doors).
- LoD4 is a LoD3 upgraded with interior. The external geometry and the semantics remain equal.
- CityGML enables the simultaneous use of multiple representations in different LoDs, meaning that the concept of LoD is object-related, and not scene directed.
- There is not an explicit link between corresponding LoDs instances of different levels. For instance, in LoD3 when adding windows to a wall, the definition of the wall cannot be re-used from the LoD2. A single CityGML file can contain multiple LoDs, but their information is often duplicated. The relationship between LoDs is weak.
- The abstraction levels for features other than buildings are not clearly defined.

Furthermore, as other researchers observe:

- CityGML does not indicate methods for the automatic derivation of the different LoDs, and relationships between different LoDs are not maintained (95).

6.2.1. Features of CityGML

Geospatial information model (ontology) for urban landscapes based on the ISO 191xx family.

GML3 representation of 3D geometries, based on the ISO 19107 model.

Representation of object surface characteristics (textures, materials).

Taxonomies and aggregations

- Digital Terrain Models as a combination of (including nested) triangulated irregular networks (TINs), regular raster's, break and skeleton lines, mass points

[95] Fan, H., & Meng, L. (2012). A three-step approach of simplifying 3D buildings modeled by CityGML. International Journal of Geographical Information Science, 26(6), 1091–1107

- Sites (currently buildings; bridges and tunnels in the future)
- Vegetation (areas, volumes and solitary objects with vegetation classification)
- Water bodies (volumes, surfaces)
- Transportation facilities (both graph structures and 3D surface data)
- City furniture
- Generic city objects and attributes
- User-definable (recursive) grouping

Multiscale model with 5 well-defined consecutive Levels of Detail (LOD)

- LOD 0 – regional, landscape
- LOD 1 – city, region
- LOD 2 – city districts, projects
- LOD 3 – architectural models (outside), landmarks
- LOD 4 – architectural models (interior)

Multiple representations in different LODs simultaneously; generalisation relations between objects in different LODs.

Optional topological connections between features (sub) geometries.

Application Domain Extensions (ADE): Specific “hooks” in the CityGML schema allow to define application specific extensions, for example for noise pollution simulation, or to augment CityGML by properties of the new National Building Information Model Standard (NBIMS) in the U.S.

6.3. BLOM

BLOM is a Geomatics company from Norway. BLOM3D™ is the product name of its archive 3D models, which detail more than 20 million buildings in 340 urban models. The BLOM3D™ models have four different LoDs, ranging from simple wire frames to fully textured models. The product and LoD are described in BLOM’s white paper (96). The descriptions of LoDs are listed for an overview:

1. Block Model (BlomLOD1™) contains 3D buildings each represented as parallelogram blocks, without the information on roofs or additional structures. The model includes a single colour for each block based on the predominant colour of the original building taken from the aerial imagery. (Fig. 44a)
2. RoofTop (BlomLOD2™) contains 3D buildings represented as parallelogram blocks, with added roof structure and other constructions present on the building. As the previous LoD, this model includes a single colour in the blocks. (Fig. 44b.)
3. Library Texture Model (BlomLOD3™) is the RoofTop model to which library textures have been added to blocks. These textures are an approximation of reality and they are selected based on a similar colour and configuration of the real textures. (Fig. 44c.)
4. LOD 4 (BlomLOD4™) is the RoofTop model in which the building textures are extracted from oblique aerial images. (Fig. 44d.)

[96] Blom ASA (2011). Blom3D™ Whitepaper for Blom partners, clients and developers. Tech. Rep. 1.0, BLOM ASA, Oslo, Norway

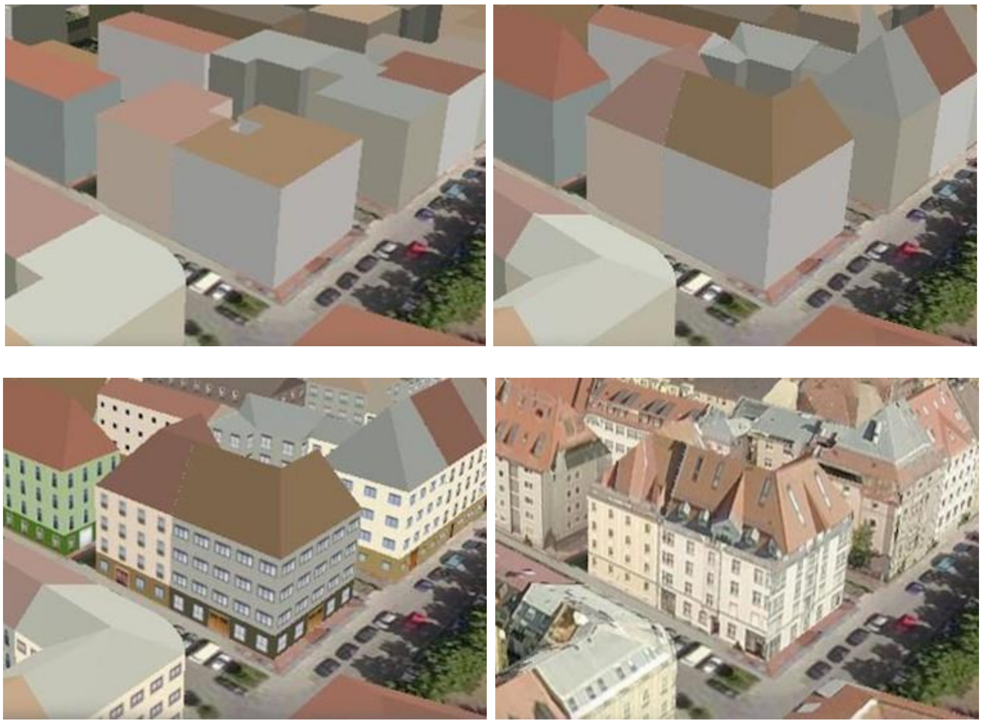


Figure 44: Example of different LoDs in the BLOM3D™ [78]. BlomLOD1™ (upper left), BlomLOD™ (upper right), BlomLOD3™ (bottom left), BlomLOD4™ (bottom right)

Thus, the geometry of the building has actually two LoDs (LoD1, while LoD2, LoD3, and LoD4 share the same geometry). The difference between the last three LoDs is in the texture: no texture, texture from a library, and (actual) photorealistic texture.

6.4. NAVTEQ

NAVTEQ is an American company that provides geo-information data for navigational products, and since they are suited for a specific application they are different from the datasets and standards presented so far. Their products relevant to this research are (97):

- 3D city models that enable applications to highlight specific buildings from search results and enable consumers to more easily orient themselves in unfamiliar or complex situations. They include a textured polygons outlining the footprint of a building, with basic rooftop shape and a height attribute defining the height of the building. What is unknown are the vertical polygons generated on-the-fly from the height of the building, or are included in the model as well.
- Enhanced 3D City Models, whose improvement over the above product is that they contain texture, and it has a realistic representation of roads, and textured land use. Enhanced 3D City Models improve consumer guidance experience by making relevant buildings more realistic.
- 3D Landmarks, are photorealistic models of prominent landmarks such as a monument, a building or other structure. 3D Landmarks can improve user orientation and sense of place in unfamiliar or complex situations.

The first two products have one LoD, however, loosely, they can be seen as one product with two LoDs distinguished only by texture. Landmarks have a finer geometry, and photorealistic textures. Each structure is delivered in two LoDs, which are distinguished by their varying polygon count.

[97] NAVTEQ (2011). 3D Landmarks - A product guide for developers. NAVTEQ

- Light resolution: up to 500 polygons (faces). The average is 200.
- Standard resolution: up to 1000 polygons. The average is 500.

There is not a clear boundary between these levels, since one object in light resolution can contain more polygons than another object in a standard resolution. Additionally, each structure is delivered as an icon available for smaller applications in 64 x 64 pixels, but this cannot be considered as a LoD in the frame of our research. The Figure 6.27 shows an example of a combination of the products Enhanced 3D City Model, and 3D Landmarks, in Munich. This is essentially a city model with a different combination of LoDs: the buildings have textures from a generic library, while landmarks such as churches have a finer geometry and a texture derived from photographs of the actual object.

Summing up the three products listed above, we can conclude that NAVTEQ offers four levels of detail for their 3D city modelling products. The difference to the previous paradigms is that not every LoD is applicable to every structure in the model. Landmarks are never represented in a low LoD, while buildings and other regular objects are never represented by photorealistic textures or fine geometry as landmarks.

Figure 45: Mixed-LoD 3D city model of Munich by NAVTEQ as a combination of the Enhanced 3D City Model, and 3D Landmarks.

The mutual motivation for having a series of LoDs in 3D city modelling is:

- Removing clutter in visualisation
- Computational reasons
- Using no more details than it is actually required
- Economical aspects

Table 3 shows an overview of the presented views. LoDs in the presented examples are all driven by the exterior details. There are two more factors to denote: texture and interior. CityGML supports textures however, this is not LoD dependent.

Example	No. of LoD	Exterior-driven	Texture-driven	Interior-driven
CityGML	1+4	Yes	No	**Yes**
BLOM	4	Yes	Yes	**No**
NAVTEQ	**4**	**Yes**	**Yes**	**No**

Table 3: Comparison between different levels of detail paradigms

6.5. Visualization 3D in Browsers [PRO]

The main goal of this section is to provide an overview of the current state and a brief description of 3D visualization on the web and specifically in the context of cities, with the elements that compose them. It is also important to comment on the shortcomings regarding the most common formats / approaches and to discuss some of the upcoming standards and trends that may help with the current implementation.

6.5.1. Introduction

3D visualization is used in many fields and it has been extensively developed in recent years. Widespread use of WebGL along with the development of modern and powerful hardware available on small devices, the browser is becoming a common place for 3D rendering. Computer animation can be described as the rendering of objects on the screen, which can change shape and properties over time. There are many approaches to rendering animation on the web, but none provide a consistent approach in terms of transmission, compression and manipulation of animation data on the client side (browser). Although computer animation has become more accessible over the Internet, these challenges must be addressed in the same "minimalist" approach as any other multimedia content that has previously been addressed on the web.

Delivery and visualization of 3D content through the Web has come a long way behind other digital media, mainly due to the higher requirements of 3D graphics in terms of computing power. There are some web solutions, which calculate the 3D representation on the server side and then transmit only 2D images to the client. However, we focus on this state-of-the-art on the client side, that is, solutions for transmitting the full 3D rendering data. A lot of work on remote viewing web-based 3D data has been already done.

Before presenting this analysis on the SOtA, it is worth nothing some of the points to justify the use of web technology over other types of technologies.

The use of a web application has many advantages over the installable desktop applications. Even though we show some disadvantages, we still consider more convenient the use of the web applications. Among all the advantages offered by the web, the following ones can be highlighted:

- Immediately access. Web-based applications do not need to be downloaded, installed, and configured. Users access their online account and they are ready to work regardless of their configuration or hardware. (Cross-Platform Compatibility).
- Web applications are simple and cheap to update. The cost of desktop software maintenance is high, usually the software administrator has to go to the work computer to update software and it generally implies expensive plans. In the case of web applications, the application only has to be updated on the server and all users work with the new version.

- Web applications make easier data centralization. When access to data from different places is required, either companies that have several facilities or individuals from different sites, with this type of applications there is no need to synchronize or copy data from one place to another, the application always access to the same data.

6.5.2. Main Problems and Challenges in 3D Rendering on the Web

6.5.2.1. Data transfer

3D animation of an object or scene requires the transmission of additional data, beyond what is necessary to make the scene statically. Depending on the context, this additional data can be compressed before the original static data is transferred. Data transmission over the Internet uses basic and well-established protocols such as TCP and UDP. At the lowest level, the basic problems associated with these net protocols lie in latency (TCP), precision (UDP), and bandwidth (both).

3D data transmission is generally a two-step method. First, the structure of the object is sent to the client. Second, the handling and processing of data in the end customer. The 3D data format for the web client can be categorized as: text based or binary format. Formats encoded as text are human readable, therefore, their file size is bulkier. But native GZIP HTTP compression does an excellent job in this regard. A typical scene or 3D object structure contains references to all external resources such as meshes, images, textures, and animation data. The XML Http Request (XHR) API requests are used to request each resource separately and are transmitted using JSON, XML, XHR binary or any of the specific domain formats. Another proposed data transmission format is the use of 2D images to encode matrices representing coordinates and other attributes. This would allow us to use various techniques for encoding and compression of audio and video, and on the client side the data can be processed as texture in WebGL to the GPU, above all the calculation.

6.5.2.2. Compression and pre-processing

Due to the large amount of data to be transmitted for an animation, a certain level of compression and preprocessing benefits this process, in order to reduce the time needed to transfer the data. However, the balance between compression and speed is delicate, a complex compression can take time to decompress within the context of a browser, which requires the use of interpreted JavaScript to do this work (and therefore is slower). The balance between compression and bandwidth has been shown to be the most effective. Each mesh and its animation data is defined by: geometry (positions of the vertices), connectivity (triangle graph) and texture / color of the data.

6.5.2.3. Rendering

In this section, it is briefly examined the rendering context to direct web-based animation. WebGL has been used for real-time 3D visualization over the Internet, which is an imperative approach to 3D graphics on the web. The element that is used is the HTML5 canvas and JavaScript to display low-level graphics API that is based on OpenGL ES 2.0 (a restricted version of the OpenGL API, originally designed for embedded systems but capable of compiling vertices and fragment shaders). During the initialization of the WebGL context, the code is compiled and copied into the memory of the video card to be executed on the GPU. 3D data is captured over the Internet using any of the data exchange formats. This captured data is stored in an ArrayBuffer, with the recent TypedArray specification, which allows low level manipulation in JavaScript and, in addition, they are useful for a parallel decoding in the GPU, since the data can be copied without loss between the web browser and the main thread.

6.5.3. 3D Formats and Standards for Web Clients

The fact of being able to represent 3D content in a web environment is not something new, since it was already possible in the past with APIs and formats such as: Flash, O3D and VRML. The common feature of all these formats is the need to install third-party plugins with the usual problem of incompatibility between applications, browsers and such plugins. However, the appearance and gradual integration of the WebGL API in current browsers has simplified and improved in every aspect the visualization of 3D content to an unimaginable point a few years ago.

It is important to note other 3D formats and standards in use nowadays, such as:

CityGML

CityGML data model is a standard data model for the representation, storage and exchange of 3D models of urban objects (OGC, 2008). It is an application scheme of the Geography Markup Language (GML3) that allows the exchange of spatial data. Both are standards approved by the Open Geospatial Consortium (OGC). The goal of CityGML is to define in common the basic entities, relationships and relationships that establish a 3D city model, regardless of its field of application. In addition, it does not only represent the geometry, topology and appearance of objects in a coherent and homogeneous way, but also the semantic and thematic properties, taxonomies and aggregations. The thematic model is divided into different areas: digital terrain models, buildings, vegetation, transport systems, street furniture, etc.

CityGML allows to represent graphic information at different levels of detail (LOD), reusing the semantic information. The different levels of detail allow the visualization and analysis of the data at different resolution, depending on the requirements of each application area. There are up to 5 LOD levels in CityGML.

LOD0 represents only the digital model of the terrain in two and a half dimensions; LOD1 includes a block model where buildings are represented by prisms with flat roofs; LOD2 differentiates roof structures; LOD3 allows to represent walls, roofs, balconies and objects of transport; LOD4 adds interior structures like rooms, doors, windows, stairs and furniture. Thanks to the 5 levels of detail CityGML allows to represent from territory information up to the same building, reducing the gap between the models in Architecture, Engineering and Construction (AEC) environments and spatial data models.

KML/COLLADA

KML is a format also defined in XML, designed to store and represent 3D geographic data. Defined by and for the Web, it supports the representation of images, regions, polygons or POIs among other things; and in conjunction with COLLADA, also 3D models as buildings.

X3D

X3D has evolved from VRML format to a more defined and used format. X3D, ratified as an ISO standard, provides a system for storing, retrieving and then reproducing real-time 3D graphics in a wide range of applications. Defined in XML, it has four trunk profiles and several additional, which define the characteristics supported by any X3D compatible system. All of them together not only support the necessary geospatial component, but also the definition of the corresponding geometric primitives and their associated levels of detail among other things. X3D also defines a particular DOM, a series of predefined events, scripting capability, audio, video and external sensors support, shaders, animations or physical calculations. For these reasons, it is one of the formats to take more into account in the construction of virtual worlds when choosing from the currently existing ones.

GLTF

Since 2013 the group Khronos (Cesio developers) began to design a 3D models exchange format that optimizes the webGL model time and processing. Traditional 3D modeling formats, such as Collada, are designed to exchange 3D files within workflows, but they are not optimized for downloading or loading in terms of efficiency. This is the main advantage of these files, which are easy and efficient to render in WebGL.

A glTF file consists of a JSON describing the file; binary .bin files containing geometries, animations and skins; text files containing glsl shaders; and image files for textures. Binaries, GLSL, and image files can also be embedded in the JSON.

GlTF uses JSON, as it is cross-platform, compact and easy to read, it allows validation and it is compressed as well.

In this case, binary data are little endian. This type of binary allows the efficient creation of GL buffers and textures since they do not require additional analysis, except maybe decompression. A file can have any number of binary files for flexibility for a wide range of applications.

This new 3D models exchange format created by the group of Khronos, is already supported by the main Javascript 3D visualization libraries like three, babylon and Cesium itself, in addition to conversion tools between Collada and other 3D design proprietary software formats as Autodesk.

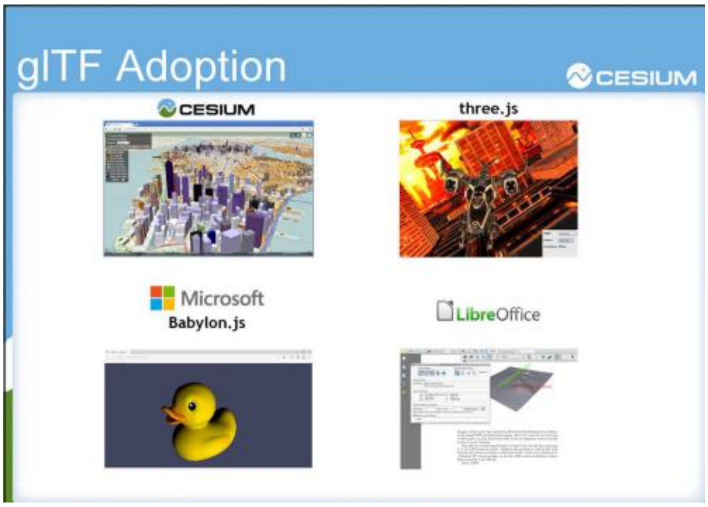


Figure 46: GlTF Adoption in web frameworks

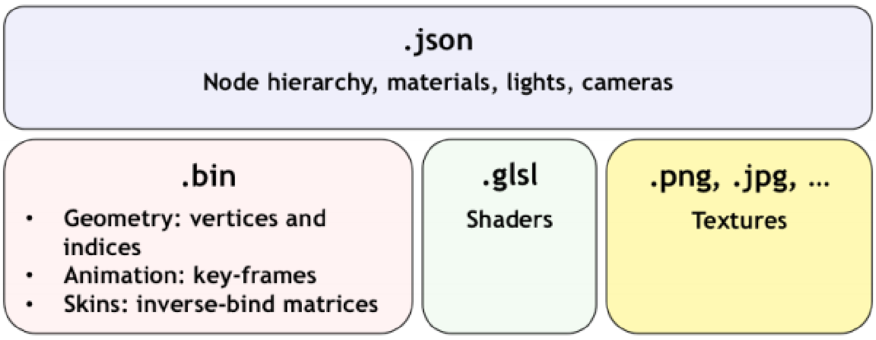


Figure 47: GlTF format specification

3DCityDB

3D City Database is a free geo database to store, represent, and manage virtual 3D city models on top of a standard spatial relational database. The database schema implements the CityGML standard with semantically rich and multi-scale urban objects facilitating complex analysis tasks, far beyond visualization. 3DCityDB is in productive and commercial use for more than 10 years in many places around the world. It is also employed in numerous research projects related to 3D city models.

The 3D City Database comes with tools for easy data exchange and coupling with cloud services. The 3D City Database content can be directly exported in KML, COLLADA, and glTF formats for the visualization in a broad range of applications like Google Earth, ArcGIS, and the WebGL-based Cesium Virtual Globe.

Apart to be a geodatabase, it provides a solution to ingest, maintain and query the models in other geodatabases like PostGIS and Oracle. With the newest (3.3) stable version users can choose to export content to glTF, which can be embedded into Cesium for web-based rendering using WebGL.

Cases of success

- Standardized by OGC.
- Many cities around the world and country-wide initiatives in Poland, the Netherlands (3D Pilot NL, AHN-2), Germany (AdV), UK and Ireland are using 3DCityDB to their city modeling. Ex. Berlin has 550000 buildings, reconstructed from 2D cadastre and LIDAR data.

3D Tiles by Cesium

With huge number of elements, Cesium has developed an open specification98 for streaming massive heterogeneous 3D geospatial datasets. To expand on Cesium's terrain and imagery streaming, 3D Tiles will be used to stream 3D content, including buildings, trees, point clouds, and vector data.

3D Tiles are not a 3D data format itself, they define a spatial data structure and a set of tile formats designed for 3D and optimized for streaming and rendering. Tiles for 3D models use glTF (99), the WebGL runtime asset format.

According to their description of the specifications, 3D tiles are designed for true 3D with a free-roaming camera; they are not limited to cases such as 2.5D perspective.

Instead of relying on 2D constructs such as zoom levels, 3D Tiles are based on geometric error for Level-Of-Detail (LOD) selection and a tunable pixel error. This allows performance/visual-quality tuning and is built for multiple "zoom levels" in the same view.

In 3D Tiles, bounding volumes are 3D, not 2D cartographic extents. In 2D, the tiling scheme is often based on the Web Mercator projection. Web Mercator is not ideal for 3D because the poles project to infinity and also because the NGA does not recommend Web Mercator (100) for DoD application. In contrast, in 3D Tiles the tiling scheme is adaptable in all three dimensions, depending on the models in the dataset and their distribution.

Traditional geospatial features, such as polygons and polylines, can be extruded or drawn above the surface. But 3D Tiles go beyond points, polylines, and polygons, to account for full 3D models with meshes, materials, and a node hierarchy.

3D Tiles specification it is still evolving, so there must be taken into account, if it is ok with things changing, then it is a good idea to jump in. Go to AnalyticalGraphicsInc/3d-tiles repository in Github to check out the current specification.

So far, it is only supported by cesium library. So, rather of thinking on this as a drawback, it is more a reason for choosing Cesium as framework for such a situation, having to handle huge amounts

[98] https://github.com/AnalyticalGraphicsInc/3d-tiles

[99] https://www.khronos.org/gltf

[100] http://earth-info.nga.mil/GandG/wgs84/web_mercator/

of 3D elements to visualize. To build 3D Tiles or b3dm files from other data formats at this time there is a tool, but not open source, that generates them from a CityGML database.

Thoughts when choosing the format in the current project scope:

Interoperability between different applications and software tools that use information related to real elements in space, such as infrastructures, is a problem that can currently be reduced thanks to the use of international standards such as CityGML, or KML/COLLADA, which are supported both by GIS editors and 3D editors, and are standards for Geographic Information Systems, extensible to attach additional thematic information such as heritage. A priori both formats, CityGML and KML would meet the requirements to be able to use technologies and tools for the visualization of 3D elements and scenes. Overlapping of semantic information in the data model facilitates the development of the intelligence of the final applications based on this model. At the same time, the geometric information of the model can be visualized on 3D rendering environments on the Web thanks to the interoperability between these formats, other open formats and the OGC Web Services. The use of standards has proven to be the most effective solution for integrating various multipurpose tools and reusing 3D digital documentation of infrastructures and buildings that have been captured and modeled using laser-based and photogrammetric techniques. The positive side of using CityGML is mainly that you can use relational geospatial database connectors and that it has tools to convert to formats such as KML and glTF, as well as offering WFS services.

Regarding glTF, it seems that it will be the reference 3D exchange format and standard default in the near future, since its 1.0 specification was presented during the year 2015, and seems to have been (not proven in this work) the most efficient when it is transferred and processed in the client part. Also, as explained previously, it already has tools to convert Autodesk obj files, Collada and glTF, being therefore an option to be considered as well. The new option of 3D Tiles Cesium, which uses glTF, seems a future option to be considered, especially if large amounts of 3D data manipulation is a requirement, as may be a Smart City.

- Cesium Language (CZML)

CZML is a non-standard format for model rendering used by Cesium

- JSON scheme to describe dynamic scenarios in virtual globes and maps.
- Describe property values that change over time.
- Allows streaming. It is not necessary the loading of the whole document in the client to be able to visualize the scene.
- Open and extendable to other data (static to create tables e.g.)

6.5.3.1. WebGL

WebGL is based on OpenGL. In fact, it can be considered an OpenGL ES API for JavaScript. OpenGL is a very popular graphical library among 3D graphics developers, which means that there are a large number of potential developers of WebGL.

The fact of being able to represent 3D content in a Web environment is not something new, since it was already possible to do it in the past with APIs and formats such as: Flash, O3D and VRML. Also, others like Unity, Virtools and Stage3D have multiplatform web players. Unity and Virtools for Windows and Mac, and Stage 3D for Linux using the Flash Player. All of them require the use of plug-ins. WebGL, on the other hand, does not require plug-ins to play. Most browsers support it natively. Therefore, the appearance and gradual integration of the WebGL API in current browsers has simplified and improved in every issue the visualization of 3D content to an unimaginable point a few years ago.

WebGL is a multiplatform JavaScript API that allows you to display real-time hardware accelerated 3D graphics (GPU) in current browsers without the need for any plugins. Internally it is based on the native implementation of OpenGL ES 2.0 with slight changes in a small set of instructions, being linked to the canvas element present in the DOM of HTML5. This not only achieves complete compatibility between platforms and applications, but also improves its graphical performance. Its main advantages are:

- **JavaScript**: JavaScript is a language that is natural to web developers and web browsers. Working with JavaScript allows you to access all parts of the DOM and also allows you to easily communicate between items rather than communicating with an applet. Because WebGL is programmed in JavaScript, this makes it easier to integrate WebGL applications with other JavaScript libraries, such as jQuery and other HTML5 technologies.
- **Automatic memory management**: Unlike its OpenGL cousin and other technologies where there are specific operations to allocate and deallocate memory manually, WebGL does not have this requirement. It follows the rules for determining the variable scope in JavaScript and the memory is automatically de-assigned when it is no longer needed. This greatly simplifies programming, reducing the code that is needed and making it clearer and easier to understand.
- **Performance**: WebGL applications performance is comparable to equivalent standalone applications (with some exceptions). This happens thanks to WebGL's ability to access local graphics hardware. Until now, many 3D rendering technologies use software-based rendering.
- **Zero compilation**: Given that WebGL is written in JavaScript, there is no need to compile the code before running it in the web browser. This gives you the power to make changes on the fly and see how these changes affect the 3D web application.

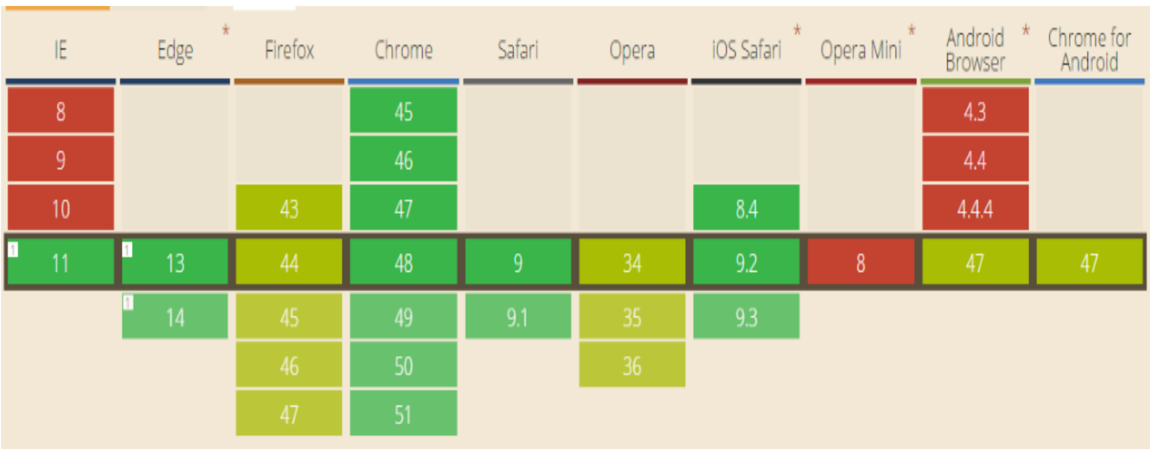


Figure 48: Browser support of WebGL

Also, one of the main issues when choosing a specification and/or standard to display 3D data is the acceptance between different types of browsers in their most widespread versions. In this sense WebGL standard, although not a specification of HTML5, is supported by most browsers, including Chrome, Firefox, Safari and IE since version 11 Figure 48

After the presentation of the main lines of development in this matter, it is clear that WebGL as a platform, and graphical web browsers as common representation window, present perspectives to visualize graphs through the Internet. With the new draft work of these standards, some or all of the barriers to real-time representation are rapidly diminishing. But it is also clear that there is a need for a common structured, coded and "in stream" standard for animated content on the web. Most of today's techniques are built upon fundamental works that have tackled the problem in various ways. For example, a general approach attempts to lessen mesh data based on connectivity, geometry, and the use of a spatial data structure for the path. While other approaches are more oriented to efficient provision on the client side. On the other hand, there is little work done in bringing the animation in real time on the web, most of it has focused on the transmission of static 3D mesh objects. However, there is a lot of research done recently regarding WebGL and generating objects on the web. WebGL 2.0 promises to bring many more features, and there are several interesting routes that should be explored, such as the use of GPU to parallelize the calculation process (in the browser), intellectual protection of data after the transmission, and efficient playback of animation data. Fortunately, the landscape on the web will soon catch up on installed applications and promises to bring exciting new features on the horizon.

6.5.3.2. Libraries to represent 3D Scenarios and animations on the Web

This is the anatomy for rendering in webGL (Figure 49):

1. Create a <canvas> element.
2. Get a drawing context.
3. Initialize the graphics windows.
4. Create one or more buffers.
5. Create one or more arrays.
6. Create one or more shaders.
7. Initialize the shaders.
8. Draw one or more primitives.

Figure 49: Browser support of WebGL

WebGL alone can be somewhat complicated and unstructured for using in complex projects. In this sense, it provides very little in terms of scene organization, transformations, 3D content help, shading and advanced effects, animation, physics, etc.

This is why we found several frameworks and libraries written in Javascript built on the WebGL API, in the following link you can see some characteristics of these and their types of licenses.

Next a brief summary will be offered, of what is considered to be the most interesting for this work, about visualization and modeling of 3D geographic information and web-based applications that offer the possibility of building three-dimensional scenarios on virtual globes representing our planet.

Cesium

Cesium is a Javascript library for creating 3D globes and 2D maps in a web browser without the need for a plugin. It uses WebGL for hardware-accelerated graphics, and it is cross-platform, cross-browser, and tuned to display dynamic data. Cesium is open source under the Apache 2.0 license. It is free for commercial and non-commercial use.

Cesium is an open source library reviewed by experts, tested with more than 90% code coverage, analyzed and documented. It supports an API and three views: 3D globe, 2D map, and Columbus view (2.5D) with the same API.

Cesium draws CZM dynamic scenes (terrestrial visualization of the world from multiple sources). It has an image layer from multiple sources, including WMS, TMS, OpenStreetMap, Bing Maps, ArcGIS MapServer, WMTS Tiles and Mapbox vector, and standard image files. Each layer can be alpha-type combined with the layers below it, and its brightness, contrast, gamma, hue and saturation can be changed dynamically.

Cesium draws KML vector data, ESRI shape files, and WebGL Globe JSO. It draws the following vector data:

- Polylines.
- Polygons and polygons with holes.
- Circles and ellipses.
- Extensions.
- Billboards.
- Labels.
- Ellipsoids.
- Sensors.
- In addition to all these formats, in recent years it has also added supports to add cartography of online map platforms such as cartoDB and you can embed the Cesium globe in a map built with the open layers library.

Cesium structure is basically organized and composed of four layers. In general, each layer adds functionality, raises the level of abstraction, and depends on the layers below it. The layers are:

- Core: calculation of numbers such as linear algebra, intersection tests, and interpolation.
- Processor: a fine abstraction on WebGL.
- Environment: world map and constructions such as image layers, polylines, labels and cameras.
- Dynamic Scene: Dynamic-time display constructions, including XML representation.

Each layer corresponds to a directory in the source tree. All layers are available for applications created with Cesium. All applications use Core. As shown below, a small subset of applications uses the renderer directly, many applications use the "Scene" directly, and perhaps like most apps, like the own viewer of Cesium.

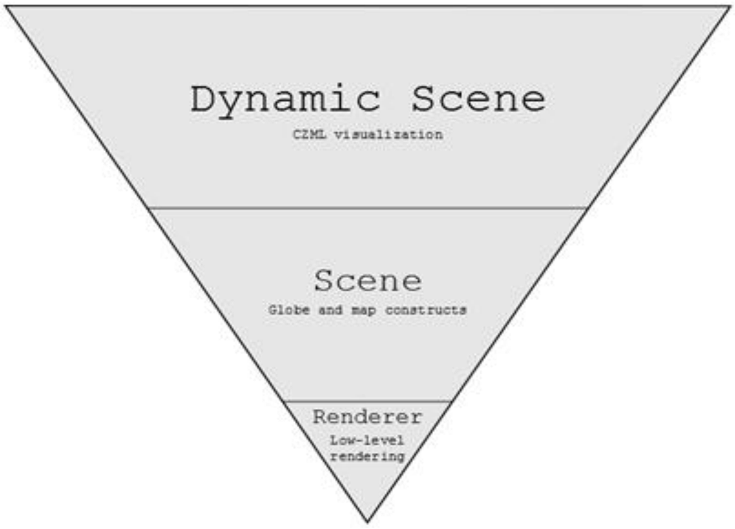


Figure 50: Cesium library layers

The main reasons for using the Cesium globe, compared to other globes or APIs are:

- An API, three views: 3D Globe, 2D maps and 2.5D Columbus View.
- Provides real-time visualization of geospatial data.
- It is built for good precision and performance, in terms of reference systems, projections and geometries.
- It has several applications and widgets:
  - Cesium viewer application for viewing CZML files.
  - Application for live code editing.
  - Calendar widget for view the evolution over time.
  - Cesium and time line for use with Dojo.

Moreover, in the year 2014 WebGL Earth made its last update and decided to be part of the Cesium project, in the Core part of the project, in order not to develop two equivalent globes. Therefore, this makes Cesium a very valid choice when it comes to rendering on a virtual globe.

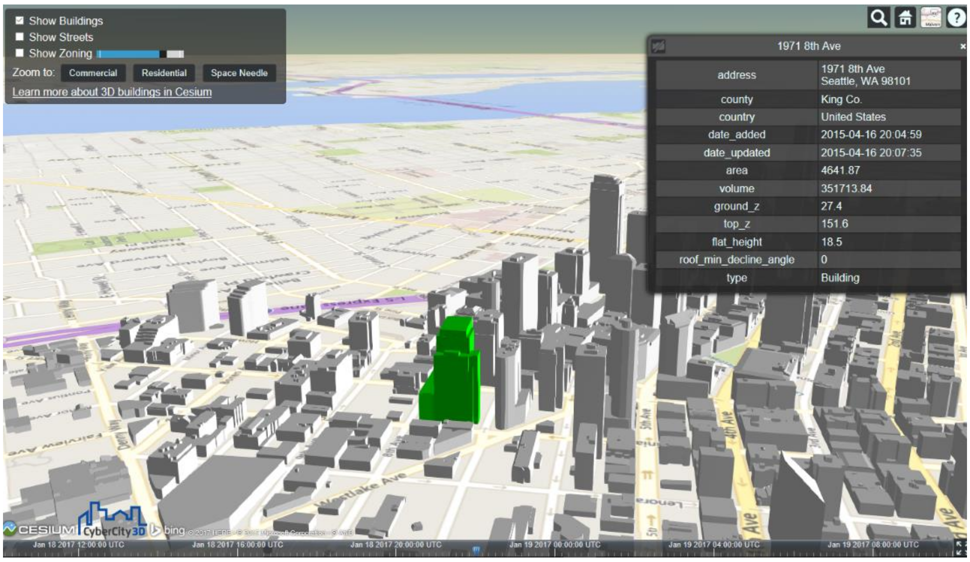


Figure 51: Cesium Globe with New York City building model using 3D Tiles

Three.js

One of the most important projects right now in WebGL is three.js - which actually also has renderings for canvas and SVG, but the WebGL processor is its strong point, because it reduces the input bar for WebGL. Three.js provides APIs for renderers, shaders, cameras, lighting, scenarios, effects and more. There is also a large community that exposes demonstrations and tutorials to start using this library.

Three.js supports the animation by ingestion of a JSON, which consist of parameters like the position, the rotation, the scale, etc. The library itself offers several scripts for exporting and / or converting most popular 3D formats (discussed above) in a JSON compatible three.js. Currently, the library only handles mesh models, other data formats of 3D objects are not compatible. For the animation of key frames, the weights associated with vertices must be indicated by the correct representation, and it is compatible with multiple animations for a single object.

This library is widely used because of its ease of interpretation and its extensive documentation and use in the creation of three-dimensional scenarios. Three.js has also support for 3D objects in GeoJSON format (it is a format for encoding a variety of geographic data structures) in planes and spheres.

Here is more in detail the most important elements that three uses and develops:

The Renderer

The render is the main object of WebGL, is where the GPU of the computer will locate, draw and paint all the elements that are wanted. To simplify the explanation, it could be seen as the "canvas" of the application. A terrain, or three-dimensional world can be huge, infinite, but users will only see everything that is seen from the "camera" and their perspective on the screen. This one we see, the 3D perspective, is known as Renderer. To create a new render, you must define at least three attributes: its width, its height and its initial color. This color could be seen as the color of the "space" surrounding the scene we want to draw on the canvas. An example can be seen in the following code.

```
var renderer;
var canvasWidth = 500;
var canvasHeight = 500;

renderer = new THREE.WebGLRenderer();
renderer.setClearColorHex(0x000000, 1);
renderer.setSize(canvasWidth, canvasHeight);
```

The Scene

In Three.js, the scene is the one that is in charge of containing all the elements and objects that would be drawn at the moment of running the application. Within a scene of Three.js, it can be found different types of objects, among which, the figures and the cameras are the most used. In every application of Three.js, you must have at least one scene, where all the objects will be found, and will be visualized depending on the focus of the camera in question. This does not mean that only one scene has to be used, but all the ones that are required can be used, to be able to give all the possible reality to a 3D environment. Creating a scene in this library is very simple, you just need to write the code presented below:

```
var scene;
scene = new THREE.Scene();
```

To then add objects to the created scene, you need the name of the scene followed by .add(), and sending the object as a parameter, as detailed:

```
Scene.add(objet);
```

The Camera

Cameras play a key role within any 3D world created with Three.js, or any other 3D rendering framework. This is because the perspective or location that gets the camera used at any given time will show a part of the 3D world, not the whole. In Three.js, a scene can have the number we want, but at the time of rendering the scene, only one of them can be used. It is possible to advance, rewind, rotate, elevate, and perform all kinds of movements with the camera, allowing change the focus of the 3D world. Although there are several types of camera, PerspectiveCamera is the most used.

```
var camera = new THREE.PerspectieCamera(45, width/height, 1, 1000);
```

The constructor of the PerspectiveCamera has 4 parameters to be sent for its correct instantiation, which are detailed as follows:

FOV: Angle of field of view degrees.

Aspect: Appearance ratio, WIDTH / HEIGHT ratio of the canvas.

Near: Minimum drawing distance

Far: Maximum draw distance once the camera is created, it is possible to include it in the scene

Once the camera is created, it is possible to include it in the scene:

```
scene.add(camera);
```

To show in the renderer the capture of the camera is executed:

```
renderer.render(scene);
```

Updating the camera view

This is the main advance feature of this framework. To update the view of the camera, it is not enough to call the renderer() method once, since in this way the camera will show only part of the movement given by a fraction of time. To keep the view continuously updated, you must call the render() method continuously for a fixed period of time, so that any movement that occurs in the camera will be displayed on the screen. For this it is necessary to implement a method, which could be called for example animate(), in charge of managing the updates of the application view. Javascript provides a native function that allows you to execute asynchronously each set time period a function or section of code called setInterval().

However, Three.js provides a new function called requestAnimationFrame() that works most optimally. This new function was developed because setInterval() performs the sending of requests to execute loops each determined number of milliseconds without considering whether the system is busy or not, causing the latter to store the request in a queue. setInterval() also issues requests, even if the system is not in focus, I.e, even if the browser is minimized, the function continues to send requests. This number of requests in memory made a totally unnecessary expense of processing and memory. The new requestAnimationFrame() function expects to receive a single parameter: The function that wants to execute, and unlike setInterval(), once the loop is finished it is necessary to call back the requestAnimationFrame() function by passing it back the function to execute. This new feature optimizes CPU usage, automatically updating as the CPU allows, at 60 frames per second in Pcs and notebooks (60 FPS), thus improving the ability to handle information in animations and consuming fewer resources. To use the resquestAnimationFrame() function, it is necessary to call it as the first line of a function, sending as a parameter the function that needs to be executed.

requestAnimationFrame() contains smooth animations because it uses a consistent image frequency. The CPU instead of being overloaded with rendering tasks, allows the processor to be able to handle other tasks at the same time as the animation sent. Another advantage of using this feature is that it works with a refresh rate of the screen, which defines how quickly the screen can display a new image. Finally, if the current tab of the browser where the animation occurs loses focus, requestAnimationFrame() will stop the animation, which produces significant energy savings and increases CPU performance.

OSMBuildings

OSM Buildings is a JavaScript library for visualizing OpenStreetMap building geometry on 2D and 3D maps. It was first created as a pseudo 3D library that could be used on top of Leaflet or Openlayers. The new GL version now comes with its own map engine called MapGL. It's a very lightweight library that doesn't use THREE.js or other webGL helpers. You can use standard tile servers for the basemap and it is easy to use GeoJSON as an additional input to the OpenStreetMap data. So luckily the city you are trying to display in 3D already has many buildings that are returned by the geojson osmbuildings tile server.

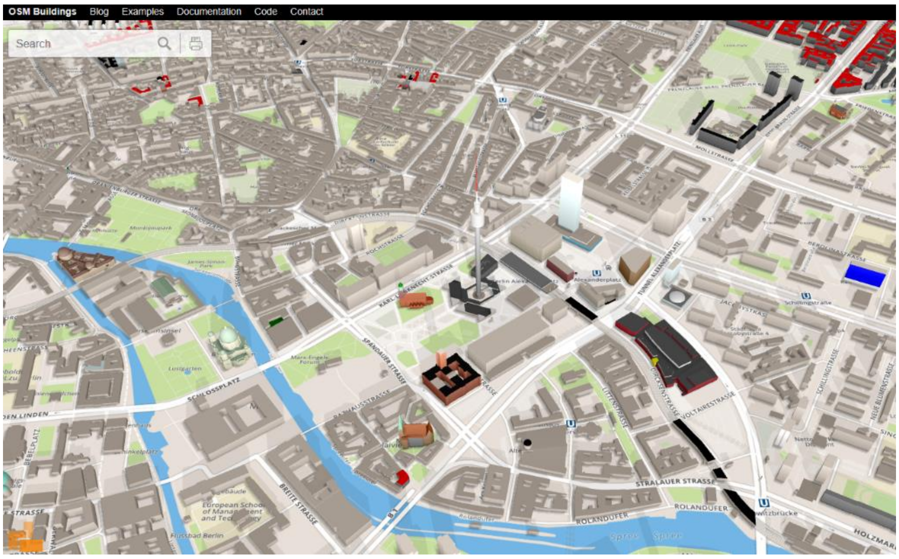


Figure 52: View of Berlin city model with OSMBuildings.

Source: https://github.com/kekscom/osmbuildings

The two OSMBuilding versions:
- Classic 2.5D
  - Best for:
    - great device compatibility
    - good performance on older hardware
    - shadow simulation
  - full integration with Leaflet or OpenLayers 2
- Modern 3D
  - Best for:
    - great performance on modern graphics hardware
    - huge amounts of objects
    - combining various data sources
  - This version uses GLMap for any events and layers logic.

Compatibility

Runs likely on earlier versions than listed below, but this is the baseline for tests.
- Win7: latest Chrome (IE11 not running, to be announced in Firefox)
- Win10: Edge, Firefox and Chorme
- OSX: latest Crome, Safari, Firefox
- Linux: TBD
- Android 5.0

- iOS 9.3

The main features are:

- It's very lightweight. That means that you have a smaller codebase and you can fix/add/change certain features easier.
- Support to add own .geojson (geojson tiles as well, very good for large 3d scenarios) and .obj data to the map.
- Many buildings available
- OSM Buildings server:
  - The tile generator script is not open sourced yet but they are working on it. However, there is a free but not public service, it can be used at: http://{s}.data.osmbuildings.org/0.2/{k}/tile/{z}/{x}/{y}.json It is indicated to contact them for heavy loads of data.
- Support animations:
  - If you want to add some animated flights to your map, you need to include an animation library. In our project, we used Tween.js. With Tween.js you can interpolate from a value to another and call functions with the current values while updating them.
- The CRS is constrained to the EPSG 4326

OSMBuilding usage:

Using OSMBuilding with the default 3D building from OSM are easy like the following Javascript script.

```
#Create the OSM buildings object, the map that handles 3D
 var osmb = new OSMBuildings({    position: { latitude:52.52000, longitude:13.41000 },    zoom: 16,    minZoom: 15,    maxZoom: 22  });  osmb.appendTo('map');
#Adding the base map
 osmb.addMapTiles(            'https://{s}.tiles.mapbox.com/v3/osmbuildings.kbpalbpk/{z}/{x}/{y}.png',            { attribution: '© Data <a href="http://openstreetmap.org/copyright/">
OpenStreetMap</a> · © Map <a href="http://mapbox.com">Mapbox</a>'});
 #Adding the buidings as geojson tiles
 osmb.addGeoJSONTiles('http://{s}.data.osmbuildings.org/0.2/anonymous/tile/{z}/{x  }/{y}.json');
```

The integration with openlayers and leaflet is like:

```
new OSMBuildings(map).load(); //Where map is a Leaflet/Openlayers map object
```

Babylon.js

Babylon.js is a complete framework and JavaScript engine graphics for creating 3D applications with HTML5 and WebGL. It has a scene chart complete with lights, cameras, materials and meshes. It has a collision engine and a physics engine using Cannon.js. It features crop scene, antialiasing, animations, particle systems, sprites, and 2D layers. For engine optimization, it is compatible with the use of truncated cone trimming, sub-scale trimming, hardware scaling, selection octrees, offline mode (files are stored locally to avoid reloading them), and progressive loading.

A standard material is used which is a per pixel material that supports diffuse illumination, ambient illumination and texture, specular illumination, texture opacity, reflection (spherical, flat, cubic, and projection) texture, mirror texture, emissive texture, specular texture, texture crashes, custom materials, skybox, apex color, four bones per apex, and up to 4 lights (dot, directional, spot, and hemispherical). It supports special effects such as fog, alpha blending, alpha test, bill boarding, full screen mode, shadow maps and variance, rendering layers, post-processing (blur, refraction, black and white, FXAA , customs, and more) lens flares, and multiple points of view. It has support for textures including render target text, textures, DDS textures, compressed video and 2D. It has arc camera support, camera rotation, free camera, touch camera, virtual joystick camera, and oculus rift camera. It supports meshes including mesh cloning, dynamic meshes, height maps, bones and constructive geometries. It supports import and export of different formats, including OBJ, FBX, MXB, Blender export, Cheetah3D export, and drag and drop support.

Other Libraries and Frameworks

In addition to these recently exposed frameworks, there are others that, although not described in this work, will also be listed because they are also used in 3D visualization on the web. The reason is that these frameworks are more focused on developments of video games and other areas outside the scope of this project, although they could still be valid for this work.

D3.js

D3js.org is an open source Javascript library for data visualization. Binds arbitrary data to a data driven document model, and applies data-based transformations to the document. It makes emphasis on web, while combining powerful visualization components and a DOM-driven data manipulation approach (Document Object Model).

It supports large sets of data and dynamic behaviors of interaction and animation.

The graphics built with D3 are visually striking, but in addition to this, these "smart" graphics are leaking into the domain of spatial geography.

The latest versions of D3, add a wide range of projective geometry to d3.geo. Geometries, projections and transformations combine to make a rich HTML 5 rendering library capable of animation effects not very common in the GIS environment on the web.

D3 uses a geoJSON-derived format that encodes the topology, this is the topoJSON. TopoJSON is still more or less a compressed format between OGC Simple Features in a client-side server and SVG database, but it appears that they are already developing conversions for PostGIS data or other spatial BBDDs to take advantage of the large possibilities of low latency.

Finally, other interesting libraries and frameworks for visualizing 3D graphics are listed, but not described, these are: MapboxGL, turbulenz, SceneJS, spiderGL and goo engine.

6.6. Augmented Reality Frameworks

6.6.1. KUDAN SDK

Kudan SDK is a Software Development Kit for Android, iOS and Unity to add augmented reality functionalities in an easy way. It allows marker and marker-free recognition to allow extended augmented reality. Simultaneous Localisation and Mapping (SLAM) is the main functionality used to allow a marker-free recognition based on device sensors.

Figure 53: Kudan Product snapshot

Website: http://kudan.eu/

6.6.1.1. Strengths and Weaknesses

Strengths: First SDK that has released a full SLAM functionality offering a marker-less free technology to show augmented reality in spaces just with a first load of 3D coordinates. Easy IoT integration.

Weaknesses: It's a paid solution, 1000€ per year per app if product needs own bundle ID (required to publish on app stores). Not supported by a big company.

6.6.1.2. Features

Indicate on a table like Table 4: Kudan Features checklist, below, a comparison of this product's features and capabilities.

Feature		Feature	
QR Code Reader	√	Digital overlay atop real world objects	√
Barcode Reader	√	Mobile devices integration	√
Current design overlay	√	Object recognition	√
Visual reminders	√	3D visualisation of as built and as planned on site	√
Object Tagging	√		

Table 4: Kudan Features checklist

6.6.2. WIKITUDE SDK

6.6.2.1. General Overview

Wikitude[101] is one of the oldest and more stable Augmented Reality Sofware development kits of the moment. It is including this year some new beta functionalities like 3D tracking and SLAM. It has libraries and packages for a huge number of platforms.

Table 5: Wikitude Product snapshot

6.6.2.2. Strengths and Weaknesses

Strengths: Stable SDK with a big community using it and ready to give feedback. Offer cloud platform service to store markers. Many languages and platforms supported.

Weakness: It's a paid solution when the moment to release the app arrives, 1980 € / year if product needs own bundle ID (required to publish on app stores). Not supported by a big company. Many of the new functionalities are beta at the moment and require a special key that must be requested.

6.6.2.3. Features

Feature		Feature	
QR Code Reader	√	Digital overlay atop real world objects	√
Barcode Reader	√	Mobile devices integration	√
Current design overlay	√	Object recognition	√

101 http://wikitude.com/

Visual reminders	√	3D visualisation of as built and as planned on site	√
Object Tagging	√		

Table 6: Wikitude Features Checklist

6.6.3. ARToolKit SDK

6.6.3.1. General Overview

A stable free and open source framework to create augmented reality experience in different platforms and devices. Daqri bought its parent company ARToolworks and released every pro feature as free.

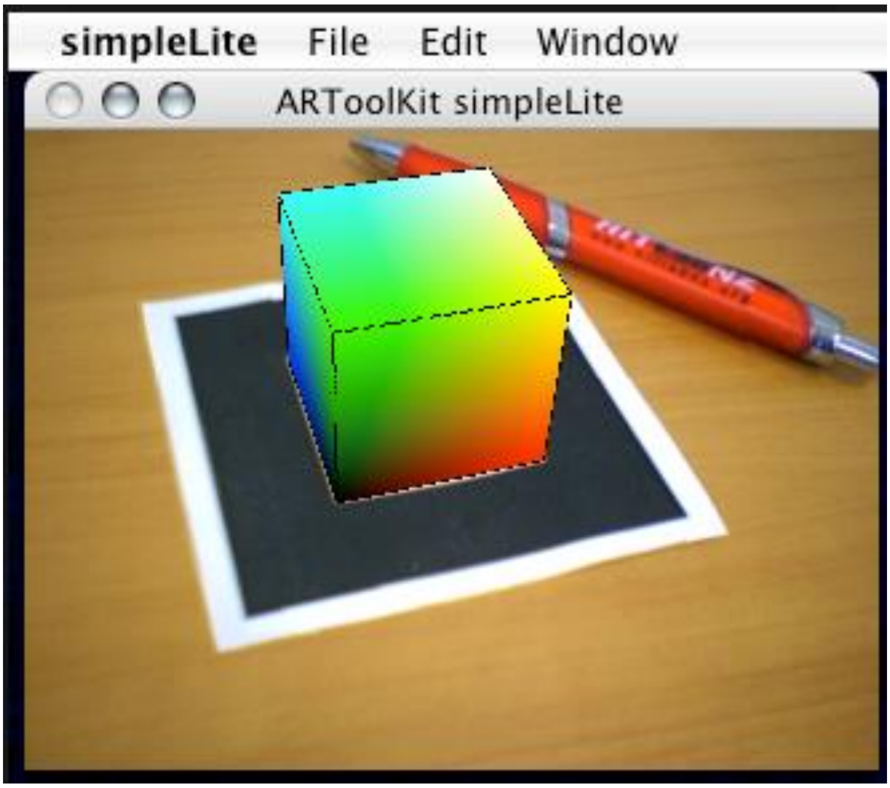


Table 7: ARToolKit[102] Product snapshot

6.6.3.2. Strengths and Weaknesses

Strengths: Free, open source, cross platform and compatible with Unity.

Weaknesses: It has no cloud platform, SLAM or basic 3D object tracking technologies or IMU sensors compatibility.

102 http://artoolkit.org/

6.6.3.3. Features

Feature		Feature	
QR Code Reader	√	Digital overlay atop real world objects	√
Barcode Reader	√	Mobile devices integration	√
Current design overlay	√	Object recognition	√
Visual reminders	√	3D visualisation of as built and as planned on site	√

Table 8: ARToolKit Features Checklist

6.6.4. Qualcomm Vuforia

6.6.4.1. General Overview

Vuforia is a platform that Qualcomm has developed for app developers to build vision based augmented reality applications on top of. It focuses on using images as the "targets" to launch an AR experience, rather than requiring consumers to scan QR codes or other glyphs. AS such, instead of scanning a barcode, you just scan a specific picture to start the AR experience on your mobile phone, tablet or even smart glasses (it could launch a video, or a 3D model, etc.) This system allows developers to build their own augmented reality apps for both tablets and smartphones and to use both local app storage of image targets as well as implement programmatic, API-based access using a cloud database system. Figure 4 below shows an overview diagram of the application development process with the platform.

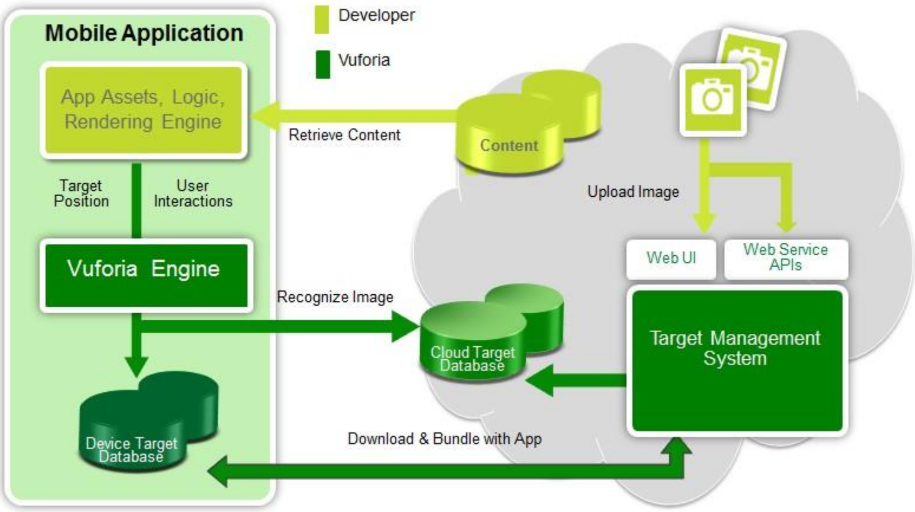


Table 9: Vuforia Overview Diagram

Website: www.qualcomm.com/products/vuforia

6.6.4.2. Strengths and Weaknesses

The following list summarizes some of the most key features of this product:

- Easy to use
- Offline mode for faster local detection of targets
- Cloud based image storage
- User defined targets that can be created at runtime from camera frames selected by the end-user
- Text recognition - recognition and tracking of printed text
- Recognize and track a broad number of objects, including cylindrical surfaces, in order to bring digital features to consumer products
- Images with sufficient detail including product packaging can be recognized
- Extended tracking - a capability that delivers a continuous experience even when the target is out of view
- Despite the impressive quality of the augmented reality features it offers, the Vuforia SDK also has some limitations, which are listed below:
- On-device image recognition and tracking is currently limited to 100 images

The device has to be connected to the internet in order to use the "Cloud Recognition" that offers image recognition on larger amount of images

6.6.4.3. Features

CitiSim could use the Qualcomm's Vuforia platform in order to generate vision-based content for the CitiSim AR application and enable object recognition capabilities. The following Table, shows some of the functionalities which could be implemented based on this platform.

Feature		Feature	
QR code reader	√	Digital overlay atop real world objects	√
Barcode reader	√	Wireless connectivity	√
Current design overlay	√	Mobile devices integration	√
Visual reminders	√	Object recognition	√
BIM metadata assignment	√	Taking photos	√

Table 10: Vuforia functionalities

7. State of the art in Semantic Modelling

Semantic data models are used to represent and annotate with enriched information various aspects of the CitiSim project data models. These aspects spread from sensor networks, city models, indoor building models, evacuation route representation and planning and some of the available application that are relevant to the CitiSim project.

7.1. Introduction to ontologies

An ontology is a semantic data representation model that explicitly defines the terminology and the inference rules of a domain of interest. Typically, ontologies are based on W3C standards, such as OWL, which are widely adopted for semantic knowledge representation. The term “ontology” derives from philosophy where this term is used to identify a branch of metaphysics dealing with the study and universal classification of being, their properties and relations between them. In the area of information technology, the term has been introduced together with the Semantic Web vision presented by Tim Berners-Lee 103 as a knowledge representation model for the web of data.

Ontologies are often represented as graphs, and they are typically made of the following components:

Concepts: concepts are abstract elements of an ontology that are used to refer to sets of instances. The concepts of “fire extinguisher”, an “escape route”, a “person” are ontological concepts as they do not refer to specific instances, but to their general categories. Concepts are categories that are used to classify proper instances.

Instances: these are tangible entities that can belong to one or more concepts. As an example “John White” is an instance of a person, “extinguisher AH123” refers to a real extinguisher positioned in a certain location, with a specific expiring date, escape route from A to B is a specific escape route from point A to point B.

Properties: properties are used to provide information about an instance in a structured manner. A property can be the birthdate of a person, its name, the expiring date of a fire extinguisher, its code reference. Properties relate atomic values (numbers, dates, text) with an instance of an ontology.

Relations: relations are used to connect elements of an ontology. While concepts define sets of instances, relations define mappings between them. As an example concept Fire Extinguisher and person can be connected by the relation revised by, indicating that a fire extinguisher is revised by a person and instance “extinguisher AH123” and “John White” can be connected with the same relation “revised by” indicating that “John White” is the specific person that revised the specific “Extinguisher AH123”. The domain and range of properties is often made explicit, for example specifying that the relations “revised by” connects entities of type “Fire Extinguisher” with entities of type “Person”.

Graphically, an ontology can be represented as in Figure 54, the concepts are represented in blue, instances in yellow and property values in Green. Relations are used to connect elements of the ontologies.

103 T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American Magazine, Retrieved March 26, 2008

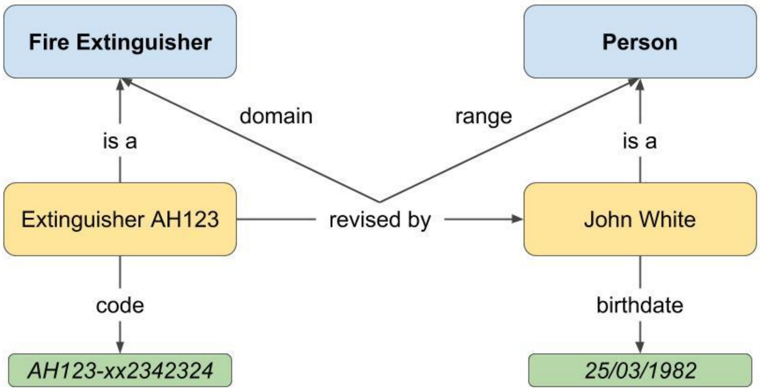


Figure 54: Example of graphical representation of an ontology.

A graph representing an ontology is defined as a set of triples in the form of "subject-predicate-object". The graph represented in Figure 54 is defined by the following set of triples:

Subject	Predicate	Object
http://www.citisim.com/ontology#revised_by	http://www.w3.org/2000/01/rdf-schema#domain	http://www.citisim.com/ontology#Person
http://www.citisim.com/ontology#revised_by	http://www.w3.org/2000/01/rdf-schema#range	http://www.citisim.com/ontology#Fire_Extinguisher
http://www.citisim.com/ontology#John_White	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.citisim.com/ontology#Person
http://www.citisim.com/ontology#Extinguisher_AH123	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.citisim.com/ontology#Fire_Extinguisher
http://www.citisim.com/ontology#Extinguisher_AH123	http://www.citisim.com/ontology#revised_by	http://www.citisim.com/ontology#John_White
http://www.citisim.com/ontology#John_White	http://www.citisim.com/ontology#birthdate	"25/03/1982"
http://www.citisim.com/ontology#Extinguisher_AH123	http://www.citisim.com/ontology#code	"AH123-xx2342324"

This type of graph defined as subject-predicate-object triples is called an RDF (Resource Description Framework) graph104. RDF is a data model defined by the W3C in the framework of the Semantic Web. It is a graph-oriented data model composed, as mentioned previously, as a set of triples <s,p,o> (subject, predicate, object). The elements composing the triples belong the two sets U and L: a set of URLs and a set of values (or literals), such as a birthdate or the code of the fire extinguisher. The elements of U are elements of the Semantic dictionary that are used to

104 Resource Description Framework (RDF) https://www.w3.org/2001/sw/wiki/RDF

generate the ontology graph. In most RDF graphs, U contains at least the elements of the RDF[105] and RDFS[106] vocabularies. The RDF vocabulary contains many useful concepts and relations, such as the relation that is used to define instances of a particular concept (<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>). URIs identify elements in U univocally, the same way as is possible to reference to a webpage, an online video or a document in the World Wide Web.

In order to be shared between different applications, users or domains, an RDF graph needs to be serialized under a compatible format. There exist several serialization formats that can be used for this task: in the following table the standard W3C serialization formats for RDF graph are reported together with their mime type and W3C specification reference page.

Serialization	Mime type	W3C Standard specification
RDF/XML	application/rdf+xml	https://www.w3.org/TR/rdf-syntax-grammar/
Turtle	text/turtle	https://www.w3.org/TR/turtle/
JSON-LD	application/ld+json	https://www.w3.org/TR/json-ld/
Notation3 (N3)	text/n3;charset=utf-8	https://www.w3.org/TeamSubmission/n3/
N-Triples	application/n-triples	https://www.w3.org/TR/n-triples/

Table 11: W3C serialization formats

RDF and RDFS vocabularies defined by the W3C have a well-defined semantics: these vocabularies can be understood and processed automatically by machines, so that an RDF graph defined with these vocabularies can be processed correctly by different applications without the need of sharing additional information. Systems capable of processing RDF graphs can be developed to understand the semantics of the RDF and RDFS. In the following table some of the most common elements of the RDF and RDFS vocabularies are provided together with their semantic interpretation in natural language. To improve readability, the following table uses the RDF and RDFS namespaces which provide a more compact representation of the URIs.

Vocabulary	URL	Semantics
RDF	rdf:type	Defines that the element in subject is a type of the element in object. As an example John White rdf:type Person.

105 Resource Description Framework (RDF) https://www.w3.org/2001/sw/wiki/RDF

106 RDF Schema https://www.w3.org/TR/rdf-schema/

Vocabulary	URL	Semantics
RDFS	rdfs:subClassOf	Defines that the element in subject is a specialization of the element in object.
	rdfs:label	Defines that the element in subject has a label represented by the text in object. As an example John White rdfs:label "J. White"

Table 12: RDF and RDFS namespaces

Additionally to the RDF and RDFS vocabularies, many other vocabularies exist to describe more specific domains of interest. Examples of vocabularies widely used in the Semantic Web community are:

Vocabulary	Definition	Domain
Friend-of-a-Friend (FOAF)	http://xmlns.com/foaf/spec/	It is used to describe concepts such as people, documents and the relationships between them. It is also used to provide general information about the concepts of persons and documents, for example the name, date of birth, email, ...
Simple Knowledge Organization System (SKOS)	https://www.w3.org/TR/skos-reference/	This vocabulary is used to provide a mapping between traditional knowledge organization systems and RDF graphs.
Dublin Core Metadata Element Set (DCMI)	http://purl.org/dc/elements/1.1/	Set of concepts and properties that can be used to describe resources on the Web (images, videos, audios, documents)
GEO	http://www.w3.org/2003/01/geo/wgs84_pos#	This vocabulary provides semantic definitions of concepts and properties that are used to define elements in a geographic space. For example, it provides definition of properties to describe the length and latitude of an element in an RDF graph.

Table 13: Examples of vocabularies widely used in the Semantic Web community

Linked Data

Another very important concept related to the Semantic Web and the definition of data in RDF is the data publication method called Linked Open Data107, defined by Tim Berners-Lee as a way to use the web as a large interlinked Database. The basic idea of this method is to provide a system of publishing data on the web in a structured and interconnected way. It is based on standard web technologies such as the HTTP communication protocol, URIs and the RDF semantic data model to provide web resources that can be understood by machines. The purpose of Linked Open Data is to allow structured data sharing on the Web as easily as documents are currently shared. A core idea behind Linked Open Data is that the value and utility of data increases the more they are interrelated with other data.

An extension of Linked Data is Linked Open Data (LOD), which combines the Linked Data publish in method with the Open Data principles, namely the removal of legal barriers that prevent data to be openly shared and reused. Data published following the LOD methodology needs to comply with a quality standard called “5-stars data”. The five stars of this quality ranking system are awarded according to the following criteria:

★ Data needs to be available on the web (any format) but with an open license so they can be consumed without any licensing issues.

★★ Data needs to be available as machine-readable structured data (e.g. in Excel format instead of the image of a data table)

★★★ Data needs to be available in non-proprietary data format (e.g. CSV instead of Excel)

★★★★ Data needs to be represented using W3C standards (such as RDF and SPARQL 108) to identify the entities and model data.

★★★★★ Data should be linked to other dataset published on the Web, to define a context and provide high serendipity. For example, if the data to be published describes the level of pollution in the city of Madrid, it is good practice to link our data with the element represented by Madrid in open databases such as DBPedia (http://dbpedia.org/resource/Madrid) so that our data does not remain isolated.

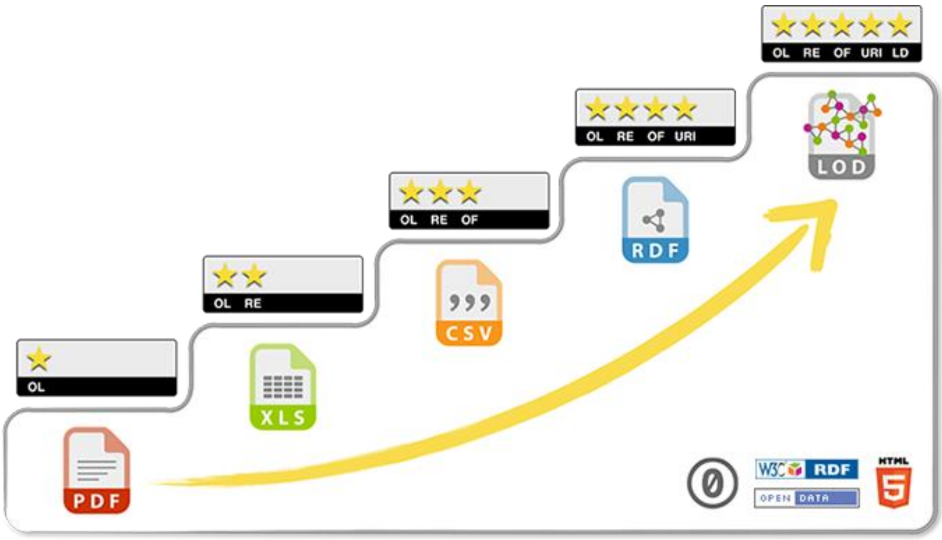


Figure 55: 5-stars linked open data.

107 http://linkeddata.org/

108 SPARQL Query Language, https://www.w3.org/TR/sparql11-query/

There is already a large amount of structured data accessible on the Web through Web 2.0 APIs such as eBay, Amazon, Yahoo, and Google Base APIs. Compared to these APIs, Linked Open Data has the advantage of providing a single data model and a small set of standardized access mechanisms rather than relying on different interfaces and data formats. One such data access mechanism is based on content negotiation. Content negotiation is a feature of the HTTP protocol which allows a client to request a resource in a specific format. An HTTP client can send a specific header with each a request to a URL resource to request a particular data format either as RDF (e.g. in the Turtle serialisation) or in a different format, such as a human readable HTML page.

The server parses the header and selects an appropriate representation. If, for example, the HTTPS client is a web browser, the header would probably request an HTML representation of the entity identified by the URL. The server will generate an HTML document that describes the entity so that it can be consumed by a traditional web browser. However, if the HTTP client is a semantic application which prefers an RDF representation, this application will specify the desired RDF serialization in the header of the request. The server could then return an RDF resource in the specified serialisation. It should be noted that content negotiation is not guaranteed to succeed. Common reasons of failure are a server not supporting content negotiation, or simply not having a representation of the resource in the required format.

As an example to describe the content negotiation mechanism, we can consider the case of two applications accessing a resource: a semantic web application and a traditional browser. We can imagine these resources making a GET request to a Linked Open Data resource: for example the entity Spain in the DBPedia identified by the URL Http://dbpedia.org/resource/Spain.

In the case of a traditional web browser, the GET request will indicate in the header that the expected form is an HTML document, so the browser can display it in a way that can be consumed by a human user:

```
curl -i -H "Accept: text/html"  -X GET http://dbpedia.org/resource/Spain
```

The server could respond with a 303 http code informing the client that the requested resource can be found under another URL, which is indicated in the response returned by the server (highlighted in yellow):

```
HTTP/1.1 303 See Other
Date: Tue, 08 Feb 2016 12:45:04 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 0
Connection: keep-alive
Server: Virtuoso/07.20.3217 (Linux) i686-generic-linux-glibc212-64  VDB
Location: http://dbpedia.org/page/Spain
Expires: Tue, 07 Mar 2017 12:45:04 GMT
Cache-Control: max-age=604800
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Allow-Headers:           DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Accept-Encoding
```

The client understands the HTTP 303 code and consequently repeats the same GET request, but at the URL specified in the "Location" parameter, so it automatically executes the request:

```
curl -i -H "Accept: text/html"  -X GET http://dbpedia.org/page/Spain
```

With this request, the server returns an HTTP code 200 together with the HTML document expected by the browser:

```
HTTP/1.1 200 OK
Date: Tue, 08 Feb 2016 12:50:19 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 1727888
Connection: keep-alive
Vary: Accept-Encoding
Server: Virtuoso/07.20.3217 (Linux) i686-generic-linux-glibc212-64  VDB
Expires: Tue, 07 Mar 2017 12:50:19 GMT
Cache-Control: max-age=604800
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Allow-Headers: DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Accept-Encoding
Accept-Ranges: bytes
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:dbpprop="http://dbpedia.org/property/"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   version="XHTML+RDFa 1.0"
   xml:lang="en"
><!-- header -->
<head profile="http://www.w3.org/1999/xhtml/vocab">
   <meta charset="utf-8" />
   <meta http-equiv="X-UA-Compatible" content="IE=edge" />
   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
   <title>About: Spain</title>
   …
</head>
<body about="http://dbpedia.org/resource/Spain">
….
</body>
</html>
```

In the case of a semantic application, the HTTP client makes a GET request to the same resource, specifying the type of serialization, for example JSON-LD:

```
curl -i -H "Accept: application/ld+json"  -X GET http://dbpedia.org/resource/Spain
```

The server responds with a 303 code that tells the client that the resource that was requested is located under another URL, which is indicated in the response that the server returned:

```
HTTP/1.1 303 See Other
Date: Tue, 08 Feb 2016 12:56:09 GMT
Content-Type: application/ld+json; qs=0.5
Content-Length: 0
Connection: keep-alive
Server: Virtuoso/07.20.3217 (Linux) i686-generic-linux-glibc212-64  VDB
TCN: choice
Vary: negotiate,accept
Alternates: {"/data/Spain.atom" 0.500000 {type application/atom+xml}}, {"/data/Spain.jrdf" 0.600000 {type application/rdf+json}}, {"/data/Spain.jsod" 0.500000 {type application/odata+json}}, {"/data/Spain.json" 0.600000 {type application/json}}, {"/data/Spain.jsonld" 0.500000 {type application/ld+json}}, {"/data/Spain.n3" 0.800000 {type text/n3}}, {"/data/Spain.nt" 0.800000 {type text/rdf+n3}}, {"/data/Spain.ttl" 0.700000 {type text/turtle}}, {"/data/Spain.xml" 0.950000 {type application/rdf+xml}}
Link: <http://creativecommons.org/licenses/by-sa/3.0/>;rel="license",<http://mementoarchive.lanl.gov/dbpedia/timegate/http://dbpedia.org/resource/Spain>; rel="timegate"
Location: http://dbpedia.org/data/Spain.jsonld
Expires: Tue, 07 Mar 2017 12:56:09 GMT
Cache-Control: max-age=604800
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Allow-Headers: DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Accept-Encoding
```

The client understands the HTTP 303 code and consequently repeats the same GET request, but at the URL specified in the "Location" parameter, so it automatically executes the request:

```
curl -i -H "Accept: application/ld+json"  -X GET http://dbpedia.org/data/Spain.jsonld
```

With this request, the server returns an HTTP code 200 together with the RDF graph of the serialized resource of Spain in the form expected by the client semantic application:


```
HTTP/1.1 200 OK
Date: Tue, 08 Feb 2016 12:57:31 GMT
Content-Type: application/ld+json
Content-Length: 53137
Connection: keep-alive
Server: Virtuoso/07.20.3217 (Linux) i686-generic-linux-glibc212-64  VDB
Expires: Tue, 07 Mar 2017 12:57:31 GMT
Link: <http://creativecommons.org/licenses/by-sa/3.0/>;rel="license",<http://dbpedia.org/data/Spain.xml>; rel="alternate"; type="application/rdf+xml"; title="Structured Descriptor Document (RDF/XML format)", <http://dbpedia.org/data/Spain.n3>; rel="alternate"; type="text/n3"; title="Structured Descriptor Document (N3/Turtle format)", <http://dbpedia.org/data/Spain.json>; rel="alternate"; type="application/json"; title="Structured Descriptor Document (RDF/JSON format)", <http://dbpedia.org/data/Spain.atom>; rel="alternate"; type="application/atom+xml"; title="OData (Atom+Feed format)", <http://dbpedia.org/data/Spain.jsod>; rel="alternate"; type="application/odata+json"; title="OData (JSON format)", <http://dbpedia.org/page/Spain>; rel="alternate"; type="text/html"; title="XHTML+RDFa", <http://dbpedia.org/resource/Spain>; rel="http://xmlns.com/foaf/0.1/primaryTopic", <http://dbpedia.org/resource/Spain>; rev="describedby", <http://mementoarchive.lanl.gov/dbpedia/timegate/http://dbpedia.org/data/Spain.jsonld>; rel="timegate"
X-SPARQL-default-graph: http://dbpedia.org
Cache-Control: max-age=604800
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Allow-Headers: DNT,X-CustomHeader,Keep-Alive,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Accept-Encoding
Accept-Ranges: bytes
{ "@graph": [
   { "@id": "http://dbpedia.org/resource/Spain",
     "@type": [ "http://dbpedia.org/class/yago/YagoGeoEntity",
       "http://dbpedia.org/class/yago/YagoPermanentlyLocatedEntity",
       "http://dbpedia.org/class/yago/WikicatAfricanCountries",
       "http://dbpedia.org/ontology/Place",
…
       "http://umbel.org/umbel/rc/PopulatedPlace" ],
     "http://www.w3.org/2000/01/rdf-schema#label": [ { "@value" : "\u30B9\u30DA\u30A4\u30F3" , "@language" : "ja" },
       { "@value" : "Espanha" , "@language" : "pt" },
       { "@value" : "Spanien" , "@language" : "de" },
       { "@value" : "Espagne" , "@language" : "fr" },
       { "@value" : "\u0418\u0441\u043F\u0430\u043D\u0438\u044F" , "@language" : "ru" },
       { "@value" : "Spagna" , "@language" : "it" },
       { "@value" : "Espa\u00F1a" , "@language" : "es" },
       { "@value" : "Hiszpania" , "@language" : "pl" },
       { "@value" : "\u897F\u73ED\u7259" , "@language" : "zh" },
       { "@value" : "\u0625\u0633\u0628\u0627\u0646\u064A\u0627" , "@language" : "ar" },
       { "@value" : "Spain" , "@language" : "en" },
       { "@value" : "Spanje" , "@language" : "nl" } ],
…
```


Figure 56 presents the communication flow in the case of a semantic application that makes a GET request to a LOD resource, to obtain the RDF graph serialized as RDF/XML.

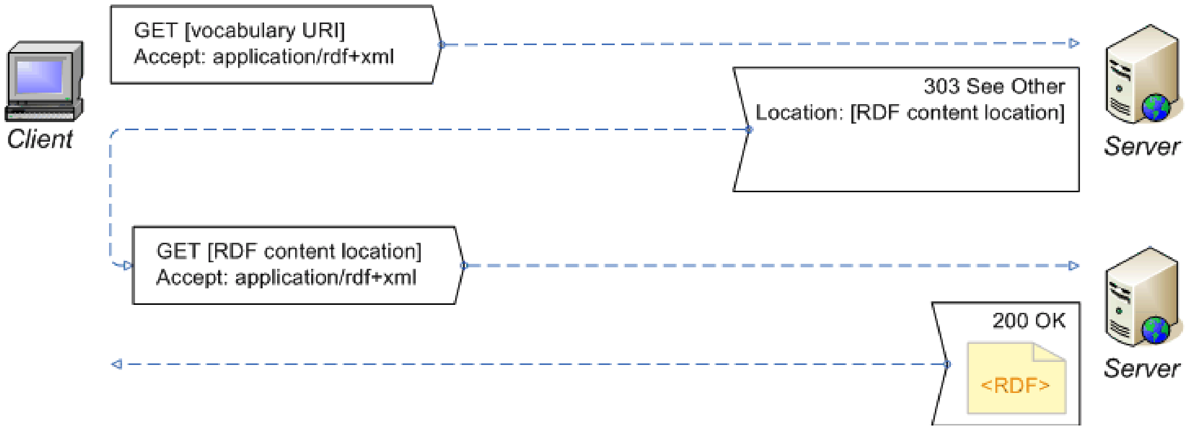


Figure 56: Content negotiation in case of a semantic web application client.

7.2. Ontologies for Smart cities

In the area of smart cities there are several domains of interest that have taken advantage from semantic web data representation. In these areas we can identify sensor networks of the city and building representations as relevant domains for the CitiSim project.

In this section therefore we will identify a set of suitable semantic models that can be used as starting point for the definition of the model used in CitiSim. These models can be defined as an extension of the models presented in this section or as an instantiation of elements in these models.

7.2.1. Semantic Sensor Network Ontologies

The CitiSim project relies heavily on Internet of Things (IoT) and sensors in order to provide contextual information about the situation in the Smart City: as an example to identify a dangerous situation, to detect a fire, to detect positions of the persons that are involved in an evacuation plan and so on. Therefore, it is necessary to model semantically sensor information to provide well defined semantics related to sensor observations. A number of different IoT middleware have been developed to provide a unified view of heterogeneous sensors and devices. (109) An early attempt at standardizing the implementation of multiple sensor networks comes from the Global Sensor Networks (GSN) framework. (110) GSN models heterogeneous sensors as *virtual sensors*. Virtual sensors provide a layer of abstraction over different sensor implementations and convert their outputs into a single format that can be used by the GSN system. Another attempt to provide a uniform view heterogeneous sensors at the level of data access comes from the Device Profile for Web Services (DPWS).(111) Using DPWS, each sensor can be modelled as a set of services that can be called to obtain sensor data. DPWS has the advantage of building upon standards such as SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language). While providing a uniform format, DPWS and GSN do not provide any semantic information on the

[109] Díaz, Manuel, Cristian Martín, and Bartolomé Rubio. "State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing." Journal of Network and Computer Applications 67 (2016): 99-117.
[110] Aberer, Karl, Manfred Hauswirth, and Ali Salehi. The Global Sensor Networks middleware for efficient and flexible deployment and interconnection of sensor networks. No. LSIR-REPORT-2006-006. 2006.
[111] Sleman, Ayman, and Reinhard Moeller. "Integration of wireless sensor network services into other home and industrial networks; using device profile for web services (DPWS)." Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on. IEEE, 2008.

sensors. In order to address this limitation, the XGSN (112) extension was developed to incorporate Semantic Sensor Network Ontology (113) (SSNO) descriptions into the GSN framework.

The Semantic Sensor Network Ontology (114) (SSNO) is an ontology defined by the W3C Semantic Sensor Networks Incubator Group (SSN-XG) to describe sensors and observations that can be used as baseline for the definition of the sensor network and observations semantic model. This ontology describes sensors in terms of capabilities, measurement processes, observations and deployments. The ontology can describe sensors, the accuracy and capabilities of such sensors, observations and methods used for sensing. Also concepts for operating and survival ranges are included, as these are often part of a given specification for a sensor, along with its performance within those ranges. Finally, a structure for field deployments is included to describe deployment lifetime and sensing purpose of the deployed macro instrument.

The SSNO is defined based on an Ontology Design Pattern (ODP) (115) describing the relationships between sensors, stimulus, and observations, this design pattern is called Stimulus–Sensor–Observation (SSO) pattern (116). According to this design pattern, the ontology can be seen from four main perspectives:

- A sensor perspective, with a focus on what senses, how it senses, and what is sensed;
- An observation perspective, with a focus on observation data and related metadata;
- A system perspective, with a focus on systems of sensors and deployments; and,
- A feature and property perspective, focusing on what senses a particular property or what observations have been made about a property.

---

[112] albimonte, Jean-Paul, et al. "XGSN: An Open-source Semantic Sensing Middleware for the Web of Things." *TC/SSN@ ISWC*. 2014.

[113] M. Michou, A. Bikakis, T. Patkos, G. Antoniou and D. Plexousakis, "A Semantics-Based User Model for the Support of Personalized, Context-Aware Navigational Services," 2008 First International Workshop on Ontologies in Interactive Systems, Liverpool, 2008, pp. 41-50.

[114] M. Michou, A. Bikakis, T. Patkos, G. Antoniou and D. Plexousakis, "A Semantics-Based User Model for the Support of Personalized, Context-Aware Navigational Services," 2008 First International Workshop on Ontologies in Interactive Systems, Liverpool, 2008, pp. 41-50.

[115] A. Gangemi *Ontology design patterns for semantic web content* ,in: Y. Gil, E. Motta, R. Benjamins, M. Musen (Eds.), 4th International Semantic Web Conference, Lecture Notes in Computer Science, ISWC 2005, vol. 3729, , Springer (2005), pp. 262–276

[116] Krzysztof Janowicz and Michael Compton. 2010. *The stimulus-sensor-observation ontology design pattern and its integration into the semantic sensor network ontology*. In Proceedings of the 3rd International Conference on Semantic Sensor Networks - Volume 668 (SSN'10), Kerry Taylor, Arun Ayyagari, and David De Roure (Eds.), Vol. 668. CEUR-WS.org, Aachen, Germany, Germany, 64-78.

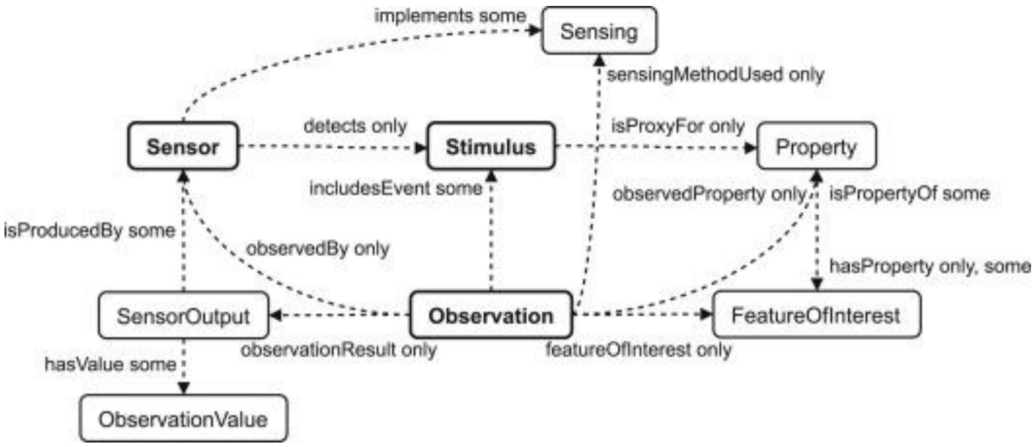


Figure 57: Stimulus-Sensor-Observation design pattern

SSNO takes a liberally inclusive view of what a sensor is: anything that observes; and allows such sensors to be described at any level of detail, for example, allowing sensors to be seen simply as objects that play a role of sensing, as well as allowing sensors to be described in terms of their components and method of operation. Humans and also simulations can be modelled as sensors.

The list of concepts contained in the SSNO, together with the semantic definition as provided by the W3C Working Group, is described in the following table. For saving space, we use the namespace prefix notation in the concepts definition by replacing the URI http://purl.oclc.org/NET/ssnx/ssn# with SSNO. The ontology consists of 41 concepts and 39 object properties, SSNO includes elements from the DOLCE+DnS Ultralite (DUL) (117) ontology, by taking into consideration 11 concepts and 14 DUL object properties, we will use DUL prefix (used to replace the full URI http://www.loa-cnr.it/ontologies/DUL.owl#) to identify such elements.

NAMESPACE	CONCEPT	Semantics
DUL	DesignedArtifact	A PhysicalArtifact that is also described by a Design. This excludes simple recycling or refunctionalization of natural objects. Most common sense 'artifacts' can be included in this class: cars, lamps, houses, chips, etc.
DUL	Event	Any physical, social, or mental process, event, or state. More theoretically, events can be classified in different ways, possibly based on 'aspect' (e.g. stative, continuous, accomplishement, achievement, etc.), on 'agentivity' (e.g. intentional, natural, etc.), or on 'typical participants' (e.g. human, physical, abstract, food, etc.).
DUL	InformationObject	A piece of information, such as a musical composition, a text, a word, a picture, independently from how it is concretely realized.

[117] http://ontologydesignpatterns.org/wiki/Ontology:DOLCE%2BDnS_Ultralite

NAMESPACE	CONCEPT	Semantics
DUL	Method	A method is a Description that defines or uses concepts in order to guide carrying out actions aimed at a solution with respect to a problem. It is different from a Plan, because plans could be carried out in order to follow a method, but a method can be followed by executing alternative plans.
DUL	Object	Any physical, social, or mental object, or a substance. Following DOLCE Full, objects are always participating in some event (at least their own life), and are spatially located.
DUL	PhysicalObject	Any Object that has a proper space region. The prototypical physical object has also an associated mass, but the nature of its mass can greatly vary based on the epistemological status of the object (scientifically measured, subjectively possible, imaginary).
DUL	Process	This is a placeholder for events that are considered in their evolution, or anyway not strictly dependent on agents, tasks, and plans. See Event class for some thoughts on classifying events. See also 'Transition'.
DUL	Quality	Any aspect of an Entity (but not a part of it), which cannot exist without that Entity. For example, the way the surface of a specific PhysicalObject looks like, or the specific light of a place at a certain time, are examples of Quality, while the encoding of a Quality into e.g. a PhysicalAttribute should be modeled as a Region. From the design viewpoint, the Quality-Region distinction is useful only when individual aspects of an Entity are considered in a domain of discourse. For example, in an automotive context, it would be irrelevant to consider the aspects of car windows for a specific car, unless the factory wants to check a specific window against design parameters (anomaly detection).
DUL	Region	Any region in a dimensional space (a dimensional space is a maximal Region), which can be used as a value for a quality of an Entity. For example, TimeInterval, SpaceRegion, PhysicalAttribute, Amount, SocialAttribute are all subclasses of Region. Regions are not data values in the ordinary knowledge representation sense.
DUL	Situation	A view, consistent with ('satisfying') a Description, on a set of entities. It can also be seen as a 'relational context' created by an observer on the basis of a 'frame' (i.e. a Description). For example, a PlanExecution is a context including some actions executed by agents according to certain parameters and expected tasks to be achieved from a Plan; a

NAMESPACE	CONCEPT	Semantics
		DiagnosedSituation is a context of observed entities that is interpreted on the basis of a Diagnosis, etc.
SSNO	FeatureOfInterest	A feature is an abstraction of real world phenomena (thing, person, event, etc.).
SSNO	Observation	An Observation is a Situation in which a Sensing method has been used to estimate or calculate a value of a Property of a FeatureOfInterest. Links to Sensing and Sensor describe what made the Observation and how; links to Property and Feature detail what was sensed; the result is the output of a Sensor; other metadata details times etc.
SSNO	Property	An observable Quality of an Event or Object. That is, not a quality of an abstract entity as is also allowed by DUL's Quality, but rather an aspect of an entity that is intrinsic to and cannot exist without the entity and is observable by a sensor.
SSNO	Sensing	Sensing is a process that results in the estimation, or calculation, of the value of a phenomenon.
SSNO	Sensor	A sensor can do (implements) sensing: that is, a sensor is any entity that can follow a sensing method and thus observe some Property of a FeatureOfInterest. Sensors may be physical devices, computational methods and a laboratory setup with a person following a method, or any other thing that can follow a Sensing Method to observe a Property.
SSNO	SensorInput	An Event in the real world that 'triggers' the sensor. The properties associated to the stimulus may be different to eventual observed property. It is the event, not the object that triggers the sensor.
SSNO	SensorOutput	A sensor outputs a piece of information (an observed value), the value itself being represented by an ObservationValue.
SSNO	Stimulus	An Event in the real world that 'triggers' the sensor. The properties associated to the stimulus may be different to eventual observed property. It is the event, not the object that triggers the sensor.
SSNO	SensingDevice	A sensing device is a device that implements sensing.
SSNO	SensorDataSheet	A data sheet records properties of a sensor. A data sheet might describe for example the accuracy in various conditions,

NAMESPACE	CONCEPT	Semantics
		the power use, the types of connectors that the sensor has, etc. Generally a sensor's properties are recorded directly (with hasMeasurementCapability, for example), but the data sheet can be used for example to record the manufacturers specifications verses observed capabilities, or if more is known than the manufacturer specifies, etc. The data sheet is an information object about the sensor's properties, rather than a direct link to the actual properties themselves.
SSNO	Accuracy	The closeness of agreement between the value of an observation and the true value of the observed quality.
SSNO	DetectionLimit	An observed value for which the probability of falsely claiming the absence of a component in a material is X², given a probability X± of falsely claiming its presence.
SSNO	Drift	A, continuous or incremental, change in the reported values of observations over time for an unchanging quality.
SSNO	Frequency	The smallest possible time between one observation and the next.
SSNO	Latency	The time between a request for an observation and the sensor providing a result.
SSNO	MeasurementCapability	Collects together measurement properties (accuracy, range, precision, etc.) and the environmental conditions in which those properties hold, representing a specification of a sensor's capability in those conditions. The conditions specified here are those that affect the measurement properties, while those in OperatingRange represent the sensor's standard operating conditions, including conditions that don't affect the observations.
SSNO	MeasurementProperty	An identifiable and observable characteristic of a sensor's observations or ability to make observations.
SSNO	MeasurementRange	The set of values that the sensor can return as the result of an observation under the defined conditions with the defined measurement properties. (If no conditions are specified or the conditions do not specify a range for the observed qualities, the measurement range is to be taken as the condition for the observed qualities.)

NAMESPACE	CONCEPT	Semantics
SSNO	Precision	The closeness of agreement between replicate observations on an unchanged or similar quality value: i.e., a measure of a sensor's ability to consistently reproduce an observation.
SSNO	Resolution	The smallest difference in the value of a quality being observed that would result in perceptibly different values of observation results.
SSNO	ResponseTime	The time between a (step) change in the value of an observed quality and a sensor (possibly with specified error) 'settling' on an observed value.
SSNO	Selectivity	Selectivity is a property of a sensor whereby it provides observed values for one or more qualities such that the values of each quality are independent of other qualities in the phenomenon, body, or substance being investigated.
SSNO	Sensitivity	Sensitivity is the quotient of the change in a result of sensor and the corresponding change in a value of a quality being observed.
SSNO	System	System is a unit of abstraction for pieces of infrastructure (and we largely care that they are) for sensing. A system has components, its subsystems, which are other systems.
SSNO	Input	Any information that is provided to a process for its use.
SSNO	Output	Any information that is reported from a process.
SSNO	Process	A process has an output and possibly inputs and, for a composite process, describes the temporal and dataflow dependencies and relationships amongst its parts.
SSNO	ObservationValue	The value of the result of an Observation. An Observation has a result, which is the output of some sensor, the result is an information object that encodes some value for a Feature.
SSNO	Condition	Used to specify ranges for qualities that act as conditions on a system/sensor's operation. For example, wind speed of 10-60m/s is expressed as a condition linking a quality, wind speed, a unit of measurement, metres per second, and a set of values, 10-60, and may be used as the condition on a MeasurementProperty, for example, to state that a sensor has a particular accuracy in that condition.

NAMESPACE	CONCEPT	Semantics
SSNO	Device	A device is a physical piece of technology - a system in a box. Devices may of course be built of smaller devices and software components (i.e. systems have components).

Table 14: Concepts and DUL object properties

The relations between concepts defined in the SSNO are reported in the following table together with the semantic definition as provided by the W3C Working Group:

NAME SPACE	RELATION	DOMAIN CONCEPT	RANGE CONCEPT	Semantics
DUL	describes	DUL:Description	DUL:Entity	The relation between a Description and an Entity: a Description gives a unity to a Collection of parts (the components), or constituents, by assigning a Role to each of them in the context of a whole Object (the system). A same Entity can be given different descriptions, for example, an old cradle can be given a unifying Description based on the original aesthetic design, the functionality it was built for, or a new aesthetic functionality in which it can be used as a flower pot.
DUL	hasLocation	DUL:Entity	DUL:Entity	A generic, relative spatial location, holding between any entities. E.g. 'the cat is on the mat', 'Omar is in Samarcanda' and ‘the wound is close to the femoral artery'. For 'absolute' locations, see SpaceRegion
DUL	hasPart	DUL:Entity	DUL:Entity	A schematic relation between any entities, e.g. 'the human body has a brain as part', '20th century contains year 1923', 'World War II includes the Pearl Harbour event'. Subproperties and restrictions can be used to specialize hasPart for objects, events, etc.
DUL	hasParticipant	DUL:Event	DUL:Object	A relation between an object and a process, e.g. 'John took part in the discussion', 'a large mass of snow fell during the avalanche', or 'a cook,

NAME SPACE	RELATION	DOMAIN CONCEPT	RANGE CONCEPT	Semantics
				some sugar, flour, etc. are all present in the cooking of a cake'.
DUL	hasQuality	DUL:Entity	DUL:Quality	A relation between entities and qualities, e.g. 'Dmitri's skin is yellowish'
DUL	hasRegion	DUL:Entity	DUL:Region	A relation between entities and regions, e.g. 'the number of wheels of that truck is 12', 'the time of the experiment is August 9th, 2004', 'the whale has been localized at 34 degrees E, 20 degrees S'.
DUL	includesEve nt	DUL:Situation	DUL:Event	A relation between situations and events, e.g. 'this morning I've prepared my coffee and had my fingers burnt' (i.e.: the preparation of my coffee this morning included a burning of my fingers).
DUL	includesObj ect	DUL:Situation	DUL:Object	A relation between situations and objects, e.g. 'this morning I've prepared my coffee and had my fingers burnt' (i.e.: the preparation of my coffee this morning included me).
DUL	isDescribed By	DUL:Entity	DUL:Descrip tion	The relation between an Entity and a Description: a Description gives a unity to a Collection of parts (the components), or constituents, by assigning a Role to each of them in the context of a whole Object (the system). A same Entity can be given different descriptions, for example, an old cradle can be given a unifying Description based on the original aesthetic design, the functionality it was built for, or a new aesthetic functionality in which it can be used as a flower pot.
DUL	isLocationO f	DUL:Entity	DUL:Entity	A generic, relative localization, holding between any entities. E.g. 'Rome is the seat of the Pope', 'the liver is the location of the tumor'. For 'absolute' locations, see SpaceRegion

NAME SPACE	RELATION	DOMAIN CONCEPT	RANGE CONCEPT	Semantics
DUL	isObjectIncludedIn	DUL:Object	DUL:Situation	Object included into another object
DUL	isParticipantIn	DUL:Object	DUL:Event	A relation between an object and a process, e.g. 'John took part in the discussion', 'a large mass of snow fell during the avalanche', or 'a cook, some sugar, flour, etc. are all present in the cooking of a cake'.
DUL	isQualityOf	DUL:Quality	DUL:Entity	A relation between entities and qualities, e.g. 'Dmitri's skin is yellowish'.
DUL	isRegionFor	DUL:Region	DUL:Entity	A relation between entities and regions, e.g. 'the color of my car is red'.
DUL	isSettingFor	DUL:Entity	DUL:Situation	A relation between situations and entities, e.g. 'this morning I've prepared my coffee with a new fantastic Arabica', i.e.: the preparation of my coffee this morning is the setting for (an amount of) a new fantastic Arabica.
DUL	satisfies	DUL:Situation	DUL:Description	A relation between a Situation and a Description, e.g. the execution of a Plan satisfies that plan.
SSNO	detects	SSNO:Sensor	SSNO:Stimulus	A relation from a sensor to the Stimulus that the sensor can detect. The Stimulus itself will be serving as a proxy for (see isProxyOf) some observable property.
SSNO	featureOfInterest	SSNO:Observation	DUL:Entity	A relation between an observation and the entity whose quality was observed. For example, in an observation of the weight of a person, the feature of interest is the person and the quality is weight.
SSNO	forProperty	SSNO:Sensor	SSNO:Property	A relation between some aspect of a sensing entity and a property. For

NAME SPACE	RELATION	DOMAIN CONCEPT	RANGE CONCEPT	Semantics
				example, from a sensor to the properties it can observe, or from a deployment to the properties it was installed to observe. Also from a measurement capability to the property the capability is described for. (Used in conjunction with ofFeature).
SSNO	hasProperty	SSNO:FeatureOfInterest	SSNO:Property	A relation between a FeatureOfInterest and a Property of that feature.
SSNO	implementedBy	SSNO:Process	DUL:Entity	A relation between the description of an algorithm, procedure or method and an entity that implements that method in some executable way. For example, between a scientific measuring method and a sensor the senses via that method.
SSNO	implements	DUL:Entity	SSNO:Process	A relation between an entity that implements a method in some executable way and the description of an algorithm, procedure or method. For example, between a Sensor and the scientific measuring method that the Sensor uses to observe a Property.
SSNO	isPropertyOf	SSNO:Property	SSNO:FeatureOfInterest	Relation between a FeatureOfInterest and a Property (a Quality observable by a sensor) of that feature.
SSNO	isProxyFor	SSNO:Stimulus	SSNO:Property	A relation from a Stimulus to the Property that the Stimulus is serving as a proxy for. For example, the expansion of the quicksilver is a stimulus that serves as a proxy for temperature, or an increase or decrease in the spinning of cups on a wind sensor is serving as a proxy for wind speed.
SSNO	observationResult	SSNO:Observation	SSNO:ObservationValue	Relation linking an Observation (i.e., a description of the context, the Situation, in which the observation was made) and a Result, which

NAME SPACE	RELATION	DOMAIN CONCEPT	RANGE CONCEPT	Semantics
				contains a value representing the value associated with the observed Property.
SSNO	observedBy	DUL:Situation	DUL:Object	Relation between a situation and an observing object
SSNO	observedProperty	SSNO:Observation	SSNO:Property	Relation linking an Observation to the Property that was observed. The observedProperty should be a Property (hasProperty) of the FeatureOfInterest (linked by featureOfInterest) of this observation.
SSNO	ofFeature	DUL:Entity	SSNO:FeatureOfInterest	A relation between some aspect of a sensing entity and a feature. For example, from a sensor to the features it can observe properties of, or from a deployment to the features it was installed to observe. Also from a measurement capability to the feature the capability is described for. (Used in conjunction with forProperty).
SSNO	sensingMethodUsed	SSNO:Observation	SSNO:Process	A (measurement) procedure is a detailed description of a measurement according to one or more measurement principles and to a given measurement method, based on a measurement model and including any calculation to obtain a measurement result [VIM 2.6]
SSNO	observes	SSNO:Sensor	SSNO:Property	Relation between a Sensor and a Property that the sensor can observe. Note that, given the DUL modelling of Qualities, a sensor defined with 'observes only Windspeed' technically links the sensor to particular instances of Windspeed, not to the concept itself - OWL can't express concept-concept relations, only individual-individual. The property composition ensures that if an observation is made of a particular quality then one can infer that the sensor observes that quality.

NAME SPACE	RELATION	DOMAIN CONCEPT	RANGE CONCEPT	Semantics
SSNO	hasMeasurementCapability	SSNO:Sensor	SSNO:MeasurementCapability	Relation from a Sensor to a MeasurementCapability describing the measurement properties of the sensor.
SSNO	hasMeasurementProperty	SSNO:MeasurementCapability	SSNO:MeasurementProperty	Relation from a MeasurementCapability to a MeasurementProperty. For example, to an accuracy (see notes at MeasurementCapability).
SSNO	madeObservation	SSNO:Sensor	SSNO:Observation	Relation between a Sensor and Observations it has made.
SSNO	observationResultTime	SSNO:Observation	DUL:Region	The result time is the time when the procedure associated with the observation act was applied.
SSNO	observationSamplingTime	SSNO:Observation	DUL:Region	The phenomenon time shall describe the time that the result applies to the property of the feature-of-interest. This is often the time of interaction by a sampling procedure or observation procedure with a real-world feature.
SSNO	qualityOfObservation	SSNO:Observation	DUL:Quality	Relation linking an Observation to the adjudged quality of the result. This is of course complimentary to the MeasurementCapability information recorded for the Sensor that made the Observation.
SSNO	hasSubSystem	SSNO:System	SSNO:System	hasPart relation between a system and its parts.
SSNO	hasInput	SSNO:Process	DUL:Entity	Relation between a Process and its input Entity
SSNO	hasOutput	SSNO:Process	DUL:Entity	Relation between a Process and the output generated
SSNO	isProducedBy	DUL:Entity	DUL:Entity	Relation between a producer and a produced entity: for example, between a sensor and the produced output.

NAME SPACE	RELATION	DOMAIN CONCEPT	RANGE CONCEPT	Semantics
SSNO	hasValue	SSNO:Observation	SSNO:ObservationValue	Relation between an observation and its value
SSNO	endTime	SSNO:Observation	DUL:Region	Relation between an observation and its starting time
SSNO	startTime	SSNO:Observation	DUL:Region	Relation between an observation and its ending time
SSNO	inCondition	SSNO:MeasurementCapability	SSNO:Condition	Describes the prevailing environmental conditions for MeasurementCapabilites, OperatingConditions and SurvivalRanges. Used for example to say that a sensor has a particular accuracy in particular conditions. (see also MeasurementCapability)

Table 15: Relations between concepts defined in the SSNO

CitiSim project can take advantage of the SSNO by instantiating sensors and observations elements for each measurement executed by the sensor network. This ontology is well suited for different types of sensors, so that it is possible, when a new sensor appears to just add the new instances and its observations, without the need for reconfigure or reinitialize the platform.

It is also worth to mention that the Spatial Data on the Web Working Group is working on a revision of the SSNO (118) that will be taken into consideration at the moment of implementing the platform: depending on the level of maturity of the work from the working group the project consortium will chose to use this new revised version or to stick with the original SSNO. The working group, has also extended the Stimulus-Sensors-Observation design pattern, called Sensor, Observation, Sample, and Actuator (SOSA) ontology as the basic design pattern for the new SSNO.

Drawing on considerable implementation and application experience with SSNO, the revised version of the SSNO plans to address several shortcomings of the initial SSNO revision:

- Addressing changes in scope and audience:
    - The initial SSN revision was developed with ontology engineers as targeted audience. Due to the widespread adaptation of Semantic Sensor Networks, the strong focus on lightweight vocabularies by the Linked Data community, and vocabularies such as Schema.org, the new version of the SSNO ontology has been revised in this sense (lightweight approach).
    - Additional classed such as Actuator, Actuation, Sampling, Sampler, and Sample have been added to SOSA and SSN in order to address sampling of observations and IoT sensors actuators.
- Addressing shortcomings of the initial SSNO revision:

118 https://www.w3.org/TR/vocab-ssn/

- The new SSNO reduced the need for the old Device, Platform, and Systems classes.
- The old SSNO was perceived as too heavyweight (on its axiomatization) and too dependent on OWL reasoning by some users. In this sense the axioms and expressiveness of the underlying description logics has been reduced.
- To improve alignment with user expectations, as well as to follow a consistent modeling strategy for observations, sampling, and actuation, the Observation class defined in SOSA and the new SSNO are now conceptualized as activities (not Situations anymore).

▪ Addressing technical developments:
  - The new SSNO and SOSA makes use of the domainIncludes and rangeIncludes annotation properties defined in Schema.org. These had not been available before.
  - Given the increased interest in using Semantic Web technologies directly on the level of individual sensors, actuators, or platforms, many of the complex language elements, axioms and logic constraints introduced by the initial revision of the SSNO have been removed.

SSNO has been applied successfully in several domains: in the framework of the SECURE (119) project the ontology has been used to model semantically the sensors network of a robot targeted to be used in rescue operations as a way to provide semantic abstraction layers on top of the plethora of sensors installed on the robot. Another European project making use of SSNO is Spitfire (120), an FP7 research project where the SSNO was used to support users with no expertise in sensor data in making sense of sensors observations, by making use of Linked Data and semantic annotations. Moreover, in the Envidat (121) project, SSNO was used to model the data gathered by environmental monitoring stations.

The SSNO is arguably the most comprehensive and widespread ontological standard to date that can model sensors in the IoT. However, other concrete ontologies have been developed, such as the SENSEI and the SensorML ontologies.

SENSEI was an integrated project of the EU's Seventh Framework Programme in Information and Communication Technology designed to create an open business-driven architecture to integrate heterogeneous sensor and actuators networks through common interfaces (122). The semantic layer of the SENSEI project is based on the Observation and Measurement (O&M) ontology, which is available in the OWL format (123).

The Sensor Model Language (SensorML) is an approved Open Geospatial Consortium standard to represent sensors, actuators and processes that involve these components, such as post-processing observation values (124). The SensorML model is designed to generate XML schema descriptions. Ontological versions of SensorML in OWL have been created. One such ontology has

[119] Pratikkumar Desai, Cory Henson, Pramod Anatharam, and Amit Sheth. 2011. Demonstration: *SECURE -- semantics empowered resCUe environment*. In Proceedings of the 4th International Conference on Semantic Sensor Networks - Volume 839 (SSN'11), Kerry Taylor, Arun Ayyagari, and David De Roure (Eds.), Vol. 839. CEUR-WS.org, Aachen, Germany, Germany, 115-118.
[120] Myriam Leggieri, Alexandre Passant, and Manfred Hauswirth, inContext-Sensing: LOD augmented sensor data available at:
[121] http://www.envidat.ch/
[122] Presser, Mirko, et al. "The SENSEI project: Integrating the physical world with the digital world of the network of the future." IEEE Communications Magazine 47.4 (2009): 1-4.
[123] http://purl.oclc.org/net/unis/ontology/sensordata.owl
[124] http://www.opengeospatial.org/standards/sensorml

been created by Le-Phouc et al. (125) but the ontology file was not made available. Another ontological version was created by the Marine Metadata Interoperability project (126) and was made available online (127).

The Human-Aware Sensor Network Ontology (HASNetO) (128) is a recently developed ontology for scientific measurements and empirical measurements and it is available online (129). The HASNetO ontology is not specifically designed for the representation of sensors, but it can be used to represent data collected manually, or data collected by sensor networks operated by humans.

7.2.2. City Ontologies

Modeling cities with Semantic Information is a technical challenge which has already been tackled in several occasions.

LinkedGeoData (130) is probably the most famous example of Linked Dataset providing information about urban areas. LinkedGeoData extends the OpenStreetMap spatial data collection to create a large spatial knowledge base. It consists of more than 3 billion nodes and 300 million ways and the resulting RDF data comprises approximately 20 billion triples. The data is available according to the Linked Data principles and interlinked with DBpedia (131) and Geo Names (132).

LinkedGeoData dataset is periodically updated so that the information that is contained in the available datasets needs to be carefully analysed before relying on it. This point is indeed an important shortcoming about Linked Open datasets. The CitiSim platform needs to rely to up-to-date information or real-time data flows, therefore these models need to be filtered out with uncertain information and extended with reliable data.

CityGML ontology (133) is an OWL version of the CityGML (134) standard, which has been created by generating classes, properties and axioms from the CityGML XML Schemas. This work is related to modelling cities with different level of detail, however a usable version of the ontology is not available anymore, making this model of little use in the framework of the CitiSim project.

A W3C working group published a set of best practices (135) regarding the publication of spatial data in the web with a section focused on publishing spatial Linked Data.

In the framework of the City4Age (136) H2020 European project, the university of Deusto developed an ontology schema that can be used to modelling urban elements and their relations (137): the

---

[125] Le-Phuoc, Danh, and Manfred Hauswirth. "Linked open data in sensor data mashups." Proceedings of the 2nd International Conference on Semantic Sensor Networks-Volume 522. CEUR-WS. org, 2009.
[126] Fredericks, Janet, et al. "Integrating Quality Assurance and Quality Control into Open GeoSpatial Consortion Sensor Web Enablement." Proceedings of OceanObs 9 (2009).
[127] http://www.sensorml.com/ontologies.html
[128] Pinheiro, Paulo, Deborah L. McGuinness, and Henrique Santos. "Human-aware sensor network ontology: semantic support for empirical data collection." *arXiv preprint arXiv:1704.01806* (2017).
[129] http://hadatac.org/ont/hasneto/
[130] http://linkedgeodata.org/About
[131] http://dbpedia.org/
[132] http://www.geonames.org/
[133] http://smartcity.linkeddata.es/ontologies/cui.unige.chcitygml2.0.html
[134] https://www.citygml.org/
[135] https://www.w3.org/TR/2017/NOTE-sdw-bp-20170330/
[136] http://www.city4ageproject.eu/
[137] https://morelab.deusto.es/ontologies/city4age

ontology does not provide instances, which need to be created for each urban environment but it is a well-defined semantic model that can be instantiated and extended for the case of CitiSim.

Concluding there are not datasets about geo spatial information regarding cities and building that can be used off-the-shelf. A suitable solution is to use already available datasets such as LinkedGeoData and extend them with more detailed information that is relevant with the project domain additionally in the case of building, that need more level of detail the use of semantic schemas such as the ontology developed for the City4Age project can be adopted for the CitiSim project.

The Human-Aware Data Collection ontology for Smart Cities (HASNetO-SC) is an extension of the HASNetO ontology for smart cities (138). This extension focuses on four main aspects: people, mobility, environment and living. Although potentially relevant to the CitiSim project, the full HASNetO-SC ontology does not seem to be available online.

The KM4City ontology (139) developed by Bellini et al., instead, aims to describe smart cities by focussing on 7 main topics, or *macroclasses*. These topics are (1) administration, (2) street guide, (3) point of interest, (4) local public transport, (5) sensors, (6) temporal and (7) metadata. The KM4City ontology development was guided by the goal of integrating a number of public datasets available in the Tuscany region in Italy and related to the topic of smart cities. The KM4City ontology could be considered for reuse by the CitiSim project because it provides a quite comprehensive set of classes and relations in the smart city domain, and because it is readily available online as an OWL ontology (140).

A number of ontologies relevant to the description of urban environments come from the area of Ambient Intelligence. A review of such ontologies has been made by Stavropoulos et al. in a paper (141) where they introduce the BOnSAI ontology (142). BOnSAI is an OWL ontology to describe smart buildings, and it includes descriptions for common appliances, such as computers and air conditioners, about locations, such as buildings, rooms and floors, and about parameters, such as light, humidity and temperature. The BOnSAI ontology could be relevant to the CitiSim project, although it appears to be more limited in scope than other ontologies such as KM4City.

An example of a real-world application of a city ontology is the Neighbourhoods of Winnipeg (NOW) ontology (143). The NOW ontology powers a website used by the city of Winnipeg to provide information about the city. This ontology contains a large number of classes, as it can be observed by the available graphical visualisations of the ontology. However, structured data about this ontology does not appear to be available. Another example of a smart city ontology is the SCRIBE ontology (144) developed by IBM. Like NOW, ontology files for SCRIBE do not appear to be openly available. Another attempt to model the domain of smart cities comes from the Smart City Ontology (SCO)145. The development of SCO was guided by previous existing smart city ontologies, such as NOW and SCRIBE. The SCO ontology is available online146 and it contains a set of classes and

[138] Santos, Henrique, et al. "Contextual data collection for smart cities." *arXiv preprint arXiv:1704.01802* (2017).
[139] Bellini, Pierfrancesco, et al. "Km4City ontology building vs data harvesting and cleaning for smart-city services." *Journal of Visual Languages & Computing* 25.6 (2014): 827-839.
[140] https://github.com/disit/servicemap
[141] Stavropoulos, Thanos G., et al. "Bonsai: a smart building ontology for ambient intelligence." *Proceedings of the 2nd international conference on web intelligence, mining and semantics*. ACM, 2012.
[142] http://lpis.csd.auth.gr/ontologies/ontolist.html#bonsai
[143] http://now.winnipeg.ca/about/now-ontology
[144] Uceda-Sosa, Rosario, Biplav Srivastava, and Robert J. Schloss. "Building a highly consumable semantic model for smarter cities." *Proceedings of the AI for an Intelligent Planet*. ACM, 2011.
[145] Komninos, Nicos, et al. "Smart City Ontologies: Improving the effectiveness of smart city applications." *Journal of Smart Cities* 1.1 (2016).
[146] https://www.dropbox.com/s/q7tz39jjeeibhzl/2015-SMART%20CITY%20ONTOLOGY-V01.owl?dl=0

relations that could be relevant to the CitiSim project, such as a classification of types of buildings, urban landmarks, typical urban organisations and sensors.

8. SoTA conclusions

8.1. Sensors, IoT and Big Data

The time of Big Data, a large amount of data which is available, making the data to be stored by product of fast growing online source. Machine-to-machine interference, like metering, calls detail recording, environmental sensing, develop their own sea of data. Big Data takes into consideration the requirement of data quality and governance for implementing analytics into operational systems. Talking about big data security is one of the main concern usually most of the data is important and confidential. Scaling is a process where the machine speeds are static because of the power constraints. Big data helps improving better security, privacy, and regulatory compliance. Internet of things (IoT) is an idea that allows availability in the surrounding of many things or objects via wireless or wired connections. The main purpose of IoT is to permit things to be connected anytime, anyplace, with the help of network or pathway and service. IoT plays an important role in developing the smartness of cities by introducing applications which help in monitoring, for example for the parking areas in the certain cities, monitoring buildings, and in some areas sound monitoring. Reliable transmission mainly concerns for the mechanisms control frame, informs peer frames. It automatically repeats request, where the sender waits for the acknowledgment before advancing. Preserve order where it delivers data in the same range to receiver's network layer that sender's network layer is intended. The sensors are made up of sensing board, imote2 by product processing board, battery board, and an external antenna to transmit the data. Sensors convert the physical parameter into signals which are measured electrically.

8.2. Data visualization frameworks

At the beginning of this work It has been explained that the frameworks that accomplish the criteria would have deeper description in addition they will be compared to each other so, as result of this work it has been made a framework comparison matrix series regarding some concrete key issues that are important to evaluate. It has been analyzed the four open source and free software, that are Birt, Jaspersoft, Pentaho and ELK. These are the topics covered by these matrices and their results:

8.2.1. Framework Basic Information

	BIRT	Jaspersoft	Pentaho	ELK
Open Source Website	eclipse.org/birt	community.jaspersoft.com	reporting.pentaho.com	elastic.co/community
Commercial Web Site	developer.actuate.com	jaspersoft.com/reporting	pentaho.com	elastic.co
License	Eclipse Public License (EPL)	JasperReports Lib LGPLV3	Pentaho Reporting LGPL V2.1	Apache License, version 2 ("ALv2")
Report Designer	BIRT Report Designer 4.4.1	JasperSoft Studio 6.0.1	Pentaho Report Designer 5.2.0-GA	Kibana 5.1
Designer Platforms	Windows, Linux, Mac OS X	Windows, Linux, Mac OS X	Windows, Linux, Mac OS X	Windows, Linux, Mac OS X
Eclipse Plug-in Available	YES	YES	NO	NO
NetBeans Plug-in Available	NO	NO	NO	NO
Standalone Java Client Available	YES	YES	YES	NO
Design Paradigm	web page design, frames, tables and lists	banded reports pixel positioning	banded reports pixel positioning	web page design, frames, tables and lists
Report Compilation	Not required	Required	Not required	Not required

	BIRT	Jaspersoft	Pentaho	ELK
Report Format	XML (.RPTDESIGN)	Report design files (.JRXML) compile to Java Byte Code (.JASPER) Deploy/Run .JASPER files	XML Report file (.PRPT) is a ZIP of Design and other resources	JSON or JavaScript script

Table 16: Framework Basic Information

8.2.2. Report Designer Components

	BIRT	Jaspersoft	Pentaho	ELK
Common Report Designer Components	Y	Y	Y	Y
Geometric Shapes	Y	Y	Y	Y
Barcodes	Y	Y	Y	Y
Callouts/notes in design mode	Y	Y	N	Y
Table of Contents as standard component	Y	Y	Y	Y
Sub-reports	Y	Y	Y	N
Sid-by-side report components	Y	Y	Y	Y
Tables	Y	Y	N	Y
Cross-tabs	Y	Y	N	Y
Horizontal Panning	Y	N	N	Y
Newspaper/multi-column layout	N	Y	N	Y
Hyperlinks within a report	Y	Y	Y	Y
Actionable charts	Y	Y	Y	Y
Cascading Style Sheets (CSS controlled format)	Y	Y	Y	Y
Conditional Formatting	Y	Y	Y	Y

Table 17: Report Designer Components

8.2.3. Data Sources

	BIRT	Jaspersoft	Pentaho	ELK
Multiple data sources and queries per report	Y	via sub-reports/charts	via sub-reports/charts	yes, but no with the wizard
Support for joining multiple data sources in the Designer	Y	N	Y	No due noSql database
Report can further manipulate query data re-sort filter group	Y	Y	Y	Y
Non JDBC Data Sources				
Cassandrea	Y	Y	Y	Y
CSV	Y	Y	Y	Y
Custom Data Adapter	Y	Y	Y	Beats
EJB	Y	Y	Y	N
Excel	Y	Y	Y	Y
Haddop Hive	Y	Y	Y	Y
Hbase	Y	Y	Y	Y
Hibernate	Y	Y	N	N
JNDI	Y	Y	Y	N
JSON	Y	Y	Y	Y
Mondrian	Y	Y	Y	N
Mongo	Y	Y	Y	Y

	BIRT	Jaspersoft	Pentaho	ELK
POJO	Y	Y	Y	N
Script	Y	Y	Y	Y
Web Services	Y	Y	Y	Y
XML	Y	Y	Y	N
XML/ A Server	Y	Y	Y	N
JDBC Drivers	Y	Y	Y	N
Query Designer	Y	Y	Y	Y
Graphical Query Designer	Y	Y	Y	Y
Scripting	JavaScript Java Event Handlers	JavaScript Groovy Java	JavaScript -Bean Script Framework (BSF) -Bean-Script Host (BSH) -Single Value Query -Metadata data-source scripting extension	Javascript
RabbitMQ	N	N	N	Y
Logs	N	N	N	Y

Table 18: Data Sources

8.2.4. Output Formats

	BIRT	Jaspersoft	Pentaho	ELK
Paginated HTML	Y	Y	Y	Y
Unpaginated HTML	Y	Y	Y	Y
XHTML	Y	Y	Y	Y
PDF	Y	Y	Y	Y
Excel (XLS 7 XLSX)	Y	Y	Y	Y
XML	Y	Y	Y	Y
Plain Text	Y	Y	Y	Y
Rich Text (RTF)	Y	Y	Y	Y
Powerpoint (PPT)	Y	Y	N	N
CSV	Y	Y	Y	Y
Postscript	Y	Y	N	Y
OpenOffice report types (doc + sheet)	Y	Y	N	Y
Flash (SWF)	N	Y	N	N
Custom Formats	Y	Y	Y	Y

Table 19: Output Formats

8.2.5. Charts

	BIRT	Jaspersoft	Pentaho	ELK
Chart wizard	Y	Y	N	Y
Chart Interactivity	mouse-over -tool tips -drill-through -hyperlinks -hide/show series -etc.	hyperlinks	mouse-over -tool tips -drill-through -hyperlinks -hide/show series -etc.	mouse-over -tool tips -drill-through -hyperlinks? -hide/show
Chart themes	Y	Y	N	Y
Precise control over format of all control elements	Y	Y	N	Y
Common Chart Types	Y	Y	Y	Y
Study charts	Y	N	N	Y
Ring chart	Y	N	Y	Y
Tube chart	Y	N	N	N
Cone chart	Y	N	N	N
Pyramid	Y	N	N	N
Time Series	Y	Y	Y	Y
Meter/ Gauge	Y	Y	Y	Y
Waterfall	N	N	Y	N
Step Area	N	N	Y	Y

	BIRT	Jaspersoft	Pentaho	ELK
Step	N	N	Y	Y
Difference	Y	N	Y	Y
Radar / Spider	Y	Y	Y	N
Thermometer	N	Y	Y	N
Candlestic/ Stock Chart (High/Low)	Y	Y	N	N
Bar Sparkline	Y	Y	N	Y
Line Sparkline	Y	N	Y	Y
Pie Sparkline	Y	Y	Y	Y
Maps	N	Y	N	Y
SVG Charts	Y	Y	N	Y

Table 20: Charts

8.2.6. Report Parameterization

	BIRT	Jaspersof t	Pentaho	ELK
Static Parameters select parameter values from a hard-coded list of values	Y	Y	Y	N
Dynamic Parameters users select parameters from a list of values that came from a database	Y	Y	Y	Y
Cascading parameters	Y	Y	Y	Y
Calendar date-picker for parameters of type date.	Y	Y	Y	N
Can specify default values	Y	Y	Y	Yes, but is a plugin or must be developed
Drop-down list boxes	Y	Y	Y	"
Radio buttons	Y	Y	Y	"
Check boxes	Y	Y	Y	"
Combo Boxes	Y	Y	Y	"

Table 21: Report Parameterization

8.2.7. Aggregations - Summary Data

Birt	JasperSoft	Pentaho	ELK
Average	Average	Average	Average
Count	Count	Count	Count by Page
Distinct Count	Distinct Count	Count by Page	Extended Stats Ag.
First	Sum	Group Count	Geo Bounds Agg.
Is-Bottom-N	First	Sum	Geo Centroid Agg.
Is-Botton-N-Percent	Minimum	Minimum	Minimum
Is-Top-N	Maximum	Maximum	Maximum
Is-Top-N-Percent	Standard Deviation	Sum Quotient	Percentile
Last	Variance	Sum Quotient Percent	Percent-Rank
Max	System	Calculation	Scripted Metric Agg.
Median	User defined Functions	Count for Page	Stats Agg.
Min		Sum for Page	Sum
Mode		Sum (Running)	Top hits
Moving Ave		Count (Running)	Value Count
Percentile		Group Count (Running)	Bucket Agg.
Percent-Rank		Count Distinct (Running)	
Percent-Sum		Average (Running)	

Birt	JasperSoft	Pentaho	ELK
Quartile		Minimum (Running)	
Rank		Maximum (Running)	
Running Count		Percent of Total (Running)	
Running Sum		User defined Functions	
Standard Deviation			
Sum			
Variance			
Weighted Average			
User defined Functions			

Table 22: Aggregations - Summary Data

8.2.7.1. BIRT

BIRT's continues to provide the strongest report development tool. BIRT's greatest strengths are in its ease of use and the completeness of features. If you are looking for a tool that allows report developers to create reports using a thick client application which will be deployed into an existing Java application framework, it is hard to beat BIRT. BIRT provides the easiest way to create reports that are focused on delivery over the web.

BIRT has two main weaknesses. First, BIRT is primarily focused on reporting instead of analytics, if you are looking to work with OLAP data, BIRT will not be appropriate. Second, BIRT lacks an open source server component and therefore if you are looking for a complete web-based BI solution BIRT is not the best choice.

8.2.7.2. Jaspersoft Studio

Jaspersoft Studio provides an outstanding and widely used report development tool that can easily be deployed either through the Jaspersoft Server community edition or through the JasperReports Library to an existing application. A particular strength of Jaspersoft is the way it works with data passed to the report as plain old java objects (POJOs). Jaspersoft is also the best product if your primary focus is to deliver printed reports.

Jaspersoft's chart engine is significantly weaker than BIRT although we prefer Jaspersoft's implementation of JFreeCharts over Pentaho's. Jaspersoft's Table of Contents for large report navigation is also more difficult to use than either BIRT or Pentaho. Finally, we found Jaspersoft's SQL editor to be the least developer friendly.

8.2.7.3. Pentaho Report Designer

Pentaho Report Designer has particular strengths around its outstanding reporting wizard and the ease of use to create simple to moderate reports. Its ability to connect to and manipulate virtually any data through its Pentaho Data Integration (Kettle) data source is a distinguishing feature.

Pentaho's charts implementation and crosstab component (experimental) need to improve to reach parity with the other products. If you are looking for one tool that is reporting-focused, we would choose either BIRT or Jasper over Pentaho.

8.2.7.4. ELK

Advantages

It has a large number of built-in graphics types. The control over them was initially limited, in the version 4 the ability to save items was added, and use "drag and drop" elements to edit dashboards. It is customizable from the web interface, which is the one used for the dashboard itself, so the dashboard creation tool is the dashboard itself.

The speed and power of Elasticsearch.

Dashboards are updated in real time as the data changes.

Very good documentation of all components as well as huge community and support offered by its maintenance team and users.

Drawbacks

In the last years, the relational databases have been dominating the way the data is modelled and stored, so the SQL language to retrieve and query data became a standard for these. Due to the naturality of a NoSql and no relational database, the different query syntax might be an issue to be learned usually from zero, also it is found that is more complicated to construct advanced query than SQL. But this is not a drawback itself it is just a matter of time that it turns over the usage of this kind of search engines.

8.3. GIS Technologies and data sources

A geographical information system (GIS) is a machine-based tool for mapping and help to make analyses for the things that exist and events that happens on the earth. The main purpose of GIS technology is that it articulates the database operations like query and statistical analysis determination and geographic analysis which are supported by the maps. The Spatial data discusses the absolute and correlating location of geographic features. There is a broad range of private and open source technologies regarding the implementation of Spatial Data Infrastructures (SDI) with the help of Geoportals which are basically the web mapping applications. CAD is a good example, it is derived to make designs and plans for the architecture of buildings and infrastructures. These systems need rules to identify how the components can be assembled and restrict its limitations capability. Carto DB helps designing and publishing maps. Besides this it is used in developing applications and helps to make leaflet and Google maps. OrbisGIS is a cross-platform open-source GIS made by the research and for the purpose of research. OrbisGIS introduces new methods and techniques to model, represent, process and share spatial data, which helps to monitor the geographical territories and creates their own evolution. Another Program GeoDa which is carried by the new users into spatial data analysis helps in statistics. Perceptive users have more knowledge about the geographic generalized data which is depending on the scale and resolution on which the data were originally produced. National Spatial Data infrastructure is able to realize if the data for a specific location is available. The geomatics solutions are web-based which are ruling over the usual desktop atmosphere.

Regarding Data Sources, the conclusions are that OpenStreetMap is the best source of open data about free geographic data for the world. It includes multitude of points of interest, buildings and also natural features and land use information.

8.4. 3D City Modelling and simulation

3D city models offer an excellent dataset for automatized and reliable heating demand diagnostics. If detailed thermal building information is available and accurate, the error for simulating the total district heating demand may lie under 10% like in the case of the district model Karlsruhe-Rintheim. If the building data availability and quality are low, the use of local building typology libraries relative to building/refurbishment years allow a reasonable district heating demand error of around 20% to be maintained, but can lead to high uncertainties for single buildings. Accurate data collection that is applicable at the city scale without being overly time-consuming, represents the major challenge for the generalization of 3D city model use. For this purpose, crowd sourcing could be an interesting path. Private owners could provide essential building data like construction standards, window areas and types, refurbishment measures undertaken and type of heating systems on a web server. The task consists of setting incentives to provide this data, as well as to control the data quality. Another option consists of geo-localised infrared thermography's, which allow for the automatic extraction of window areas, as well as refurbishment status and thermal properties using image segmentation algorithms. This is one of the goals of the new project SimStadt (SimStadt, 2013), which aims at providing cities and energy suppliers a useable simulation environment in the next three years. In addition to heating demand diagnostics, which serves as a calibration phase, 3D city models offer opportunities to simulate energy scenarios, supporting city planners and municipal managers in the development of long-term urban energy strategies. 3D city models could also directly address the building owners or tenants and allow them to calculate their energy savings potential and the investment costs of required refurbishment. Whatever the application, 3D city models have the potential to facilitate and support a holistic city energy strategy and thereby, become a keystone of the energy transition147.

8.5. Visualization 3D in Browsers

When choosing a definitive framework to visualize 3D objects on the web, it will be necessary to set some bases, which depending on these would be demarcated to a greater or lesser extent by the use of one of these frameworks and formats exposed. In these bases, we would define the minimum and maximum scales to visualize our work, that is, whether a situation is only local or relative to the environment, or a more global or regional situation is required, including territory, communications, reliefs, etc. Then the objects would be defined and detailed levels to be displayed. Another important variable would be the format of the data or services that are arranged to visualize the data. Then, as usually happens in most cases, depending on these variables would be what would be used to decide the best framework to use in a project. For example, in the case of wanting to represent elements on a global or far-reaching scale.

Cesium would be a good choice as it offers high resolution aerial images, coverage and offers terrain reliefs, being a good choice for a geographic context of our 3D objects, in addition it has an extensive support of all formats and archives of most widely used geospatial information. Cesium is a 3D visualization engine focused on geospatial visualization. This engine is capable of handling geographic and flat coordinate systems, different types of maps and projections, and addresses the problems that arise when using very large world coordinate systems. Taking advantage of these capabilities would be very beneficial in time, resources and costs. It is also capable of making general 3D graphics and these capabilities seem to be increasing. For jobs requiring 3D graphics

[147] https://en.wikipedia.org/wiki/3D_city_models.

and a certain amount of geospatial visualization it is recommended to examine whether Cesium is able to meet all requirements. It is the most used and most advanced terrestrial globe library to construct 3D scenes on it. It is also the one that more 3d formats and geospatial services supports, besides being the promoter of the new standards on which they are working and developing at the present time.

The main disadvantage of Cesium is that because it uses the globe to draw on it the scenarios, it is a heavy library compared to other frameworks such as OSMBuildings, which on the contrary is very light.

On the other hand, if what concerns us is a relative environment, in which the external situation is not so important, then libraries like three.js, or Babylon can be a good alternative, even being able to be better, since they have perhaps more tools for creating complex and more developed 3D scenarios.

The following table shows the support of certain characteristics and the state of development of the most important balloon representation projects:

8.5.1. Format and platform support review

	Cesium	Glob3	OpenWeb Globe	ReadyMap	OSMBuildings	VizyCities
Android	No	Yes	No	No	No	No
iOS	No	Yes	No	No	No	No
Web (WebGL)	Yes	Yes	Yes	Yes	Yes	Yes
KML	KML to CZML	KML to OGR DataModel	No	No	No	No
SHP	SHP to CZML	SHP to OGR DataModel	No	No	No	No
Development	+++ advanced	In progress	stopped (more than 1 year)	stopped (> 2 year)	Just started	Just started

Table 23: Format and platform support review

At the local context library level this table shows support for different formats.

8.5.2. Format support review

	ThreeJS	Copperlicht	SceneJS	BabylonJS
.3ds	No	Yes	Yes	No
.lwo	No	Yes	No	No
.md2	No	Yes	Yes	No
.obj	Yes	Yes	Yes	Yes
Development	Active	Active	Stopped from 4 months	Active

Table 24: Format support review

8.5.3. Augmented Reality

The Augmented Reality frameworks are improving quickly for devices with RGB camera, offering recognition by markers and new functionalities, such as localization and simultaneous mapping in real time (currently known as SLAM). This allows simple devices such as smartphones to have a set of tools for this type of technology although, step by step, new devices and prototypes such as Microsoft HoloLens and Google Tango are being released and entering the market successfully thanks to their own capabilities of spatial mapping provided by new integrated depth cameras which allow these new devices to offer a better experience, compared to other devices without depth sensor, and fit the real needs of the public both present and future.

8.6. Ontologies for Smart Cities

CitiSim project can take advantage of the SSNO by instantiating sensors and observations elements for each measurement executed by the sensor network. This ontology is well suited for different types of sensors, so that it is possible, when a new sensor appears to just add the new instances and its observations, without the need for reconfigure or reinitialize the platform.

Concluding there are not datasets about geo spatial information regarding cities and building that can be used off-the-shelf. A suitable solution is to use already available datasets such as LinkedGeoData and extend them with more detailed information that is relevant with the project domain additionally in the case of building, that need more level of detail the use of semantic schemas such as the ontology developed for the City4Age project can be adopted for the CitiSim project.