Reality Sensing, Mining and Augmentation for Mobile Citizen–Government Dialogue

FP7-288815

D3.2

eGovernance augmented reality application

Dissemination level:	Public (PU)
Contractual date of delivery:	Month 33, 2014-10-31
Actual date of delivery:	2014-10-31
Workpackage:	WP3 - Mobile Augmented Reality
Task:	T3.2 – eGovernance augmentation layer
Type:	Prototype
Approval Status:	Final
Version:	v0.7
Number of pages:	126
Filename:	D3.2-eGovernance augmented reality

Abstract

D3.2 reports on the activities performed during the second phase of the project in the context of WP3. During this period, there has been mainly three different activity tracks constituting WP3 contribution to Live+Gov, namely: a) the enhancement of the platform for mobile augmented reality and the generation of the eGovernance augmented layers, b) the development of a mechanism for personalized content delivery, and c) the review of methods and tools for data aggregation and visualization, as well as the thorough presentation of the analytic tools that have been developed to facilitate the decision makers in making more informed decisions. All three activity tracks are presented in detail placing emphasis on the reasons that have motivated our work and the concrete outcomes of our developments.

The information in this document reflects only the author's views and the European Community is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

This work was supported by the EU 7th Framework Programme under grant number IST-FP7-288815 in project Live+Gov (www.liveandgov.eu)

Copyright

History

Version	Date	Reason	Revised by
0.1	2014-05-27	Draft skeleton - alpha version	S. Nikolopoulos and Y. Kompatsiaris
0.2	2014-09-03	Version for collecting input	S. Nikolopoulos, D. Ververidis, G. Liaros, A. Papazoglou-Chalkias, E. Chatzilari, A. Maronidis, G. Chantas
0.3	2014-09-27	Version incorporating the first round of inputs	L. Niittylä, C. Schaefer
0.4	2014-10-15	Version incorporating the second round of inputs – beta version	L. Niittylä, C. Schaefer, D. Ververidis, G. Liaros, P. Minnigh, M. Terpstra
0.5	2014-10-24	Version incorporating the comments from internal review	S. Nikolopoulos
0.6	2014-10-30	Added a section on the sentiment analysis of social content	S. Nikolopoulos and A. Papazoglou-Chalkias
0.7	2014-10-31	Proof editing and additional refinements – final version	S. Nikolopoulos and Y. Kompatsiaris

Author list

Organization	Name	Contact Information
CERTH	Dimitrios Ververidis	ververid@iti.gr
CERTH	Elisavet Chatzilari	ehatzi@iti.gr
CERTH	Giorgos Liaros	geoliaros@iti.gr
CERTH	Tasos Papazoglou Chalikias	tpapazoglou@iti.gr
CERTH	Spiros Nikolopoulos	nikolopo@iti.gr
CERTH	Yiannis Kompatsiaris	ikom@iti.gr
CERTH	Giannis Chantas	gchantas@iti.gr
CERTH	Anastasios Maronidis	amaronidis@iti.gr
YCT	Pieter Minnigh	p.a.minnigh@yucat.com
YCT	Marjolein Terpstra	m.terpstra@yucat.com
UKOB	Christoph Schaefer	chrisschaefer@uni-koblenz.de
MTS	Laura Niittylä	Laura.Niittyla@mattersoft.fi
BIZ	Maite de Arana	komunikazioa.tekniko1@biscaytik.eu

Executive Summary

D3.2 is organized in four main sections (Section 2-5) that are used to report on the activities that have taken place during the second phase of the project. These sections are complemented by an introductory section that provides an overview of the deliverable and a summary section that concludes with some important insights that we have gained during the Live+Gov experience.

In what refers to the main sections, Section 2 reports on the improvements that we have performed in addressing the project's revised requirements, as well as on the major developments that we have undertaken in order to automatically serve content across platforms and vendors. Section 3 describes the personalization mechanism that has been developed to deliver traffic-related information in a personalized manner. Section 4 reviews some of the most widely established methods and tools for data visualization, while Section 5 provides a thorough analysis of the analytic tools that have been developed to facilitate the extraction of valuable insights for each of the Live+Gov use cases.

Finally, D3.2 incorporates a number of Appendices that provide details about the technical investigations that we have undertaken to select the most appropriate libraries and services for implementing the analytic tools.

Abbreviations and Acronyms

API	Application Programming Interface
AR	Augmented Reality
AREL	Augmented Reality Experience Language
ARML	Augmented Reality Markup Language
CMS	Content Management System
ESA	Explicit Semantic Analysis
HTTP	Hypertext Transfer Protocol
IBS	Image BaSed augmented reality
IST	Information Society Technologies
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
LBS	Location BaSed augmented reality
MVC	Model-View-Controller
NGD	Normalized Google Distance
NRD	Normalized Relevance Distance
REST	Representational State Transfer
SMS	Short Messaging Service
VR	Virtual Reality
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

Table of Contents

List of Figures

List of Tables

1. Introduction

The goal of this document is to report on the activities that have been carried out in the context of WP3 during the second phase of the project. After the completion and evaluation of the first field trials, the goal of WP3 has been twofold. First, to revise, extend and optimize the platform for mobile augmented reality based on the evaluation results, the debugging reports and the revised requirements. Second, to design and implement the necessary tools for data aggregation, interlinking and visualization, and in this way facilitate the decision makers in better understanding the generated data and discovering useful insights.

In this respect, Section 2 departs from the revised requirements and the necessary extensions that has been identified during the first phase evaluation and continues with the description of the technical modifications and optimizations that have been undertaken to address them. Moreover, Section 2 describes also the major developments that have taken place in the augmented reality platform so as to automatically serve AR content across platforms and vendors. These developments were motivated by the need to reach a wider audience and create more impact, which demanded for having the eGovernance aurmentation layers available in the highest possible number of devices. The FastAR module described in Section 2.4 was developed to address this demand. Finally, Section 2 concludes with the description of the eGovernance augmentation layers that were developed to facilitate the execution of Urban Planning and Mobility field trials.

The next section of this deliverable (Section 3) is allocated in describing the algorithms and functionalities that have been developed to address the objective of personalized content delivery. Although our initial intention was to incorporate this mechanism as part of the mobile augmented reality platform, the feedback that we received during the first phase field trials prompt us to repurpose the use of personalization. More specifically, there was a great demand for more personalized services from the testers of the Mobility field trial. They practically mentioned that unless the traffic-related information comes in the appropriate place and time, there is little chance in helping the user to avoid traffic congestions. On the other hand, there was no significant demand for filtering the content presented through AR to avoid the user's overwhelming with data. Motivated by the above, we have decided to repurpose the personalization mechanism so as to learn and utilize the citizen's commuting habits instead of his content preferences. Section 3 describes the algorithms that have been developed to automatically learn the citizen's commuting profile, as well as the functionalities developed based on this profile to deliver traffic-related information in a personalized manner.

The following two Sections (Section 4 and 5) deal with the second goal of this deliverable, which is the design and implementation of analytic tools for helping the decision makers in gaining valuable insights. More specifically, in Section 4 we go through some of the most widely established approaches for data visualization and we review the pros and cons for some of the existing libraries and services implementing these approaches. This review allowed us to adopt the libraries meeting both the fit-for-purpose as well as the technical requirements.

On the other hand, Section 5 places more emphasis on the conceptual motivation for developing these tools, as well as the insights that are expected to arise through their use.

More specifically, after defining a methodological approach consisting of the target user, the intended insight, the internal and external data involved, the processing required and the employed visualization method, we analyze each field trial based on these aspects. Starting from the high-level objective (e.g. improve the transportation network) facilitated by the developed analytics tool, we go on and address each of the aforementioned aspects placing particular emphasis on the intended insights. In this way, we advocate the appropriateness of the developed tools to facilitate the city officials in making more informed decisions.

Finally, Section 6 concludes this deliverable by summarizing the most important experiences that we have gained through Live+Gov, as well as by providing our current understanding for the potential of the employed technologies.

2. Mobile Augmented Reality for eGovernance

2.1 Overview

In this section we report on the AR-related advancements that have taken place since the delivery of the first augmented reality prototype described in deliverable D3.1 [1] of WP3. More specifically, our goal is to layout the revised requirements as derived from the results of the first field trial evaluation and explain how the implemented extensions and improvements succeed in addressing these requirements. While in most of the cases the newly desired functionalities required only small-to-medium effort for their implementation, the requirement for broad reach and higher impact required the implementation of major improvements as detailed in Section 2.4.

The AR solution that has been developed in the context of Live+Gov, although relying on a widely established AR platform, it has been tailored to cover the particular requirements stemming from the domain of eGovernance. The basic requirement for the developed solution has been to present future urban plans in a realistic manner, so as to intrigue citizens to provide their opinion (e.g the preferred location, the architecture type, the eco-friendliness, an estimate for the usage frequency, etc). In satisfying this requirement, AR has been used through two types, namely the location based AR that allows citizens to view the 3D model of the plan in its actual position and the image based AR that allows citizens to scan a promo poster and see the future plan on the poster. The technology developed incorporates the latest versions of commercial AR tools for a professional quality level. However, the need to intrigue citizens has forced us to pay particular attention on the presentation of 3D models, their placement in the scene and, in general, the experience offered to the user. As a consequence, a number of additional functionalities (i.e. dealing with the object's orientation, scale and sensitivity to abrupt location changes) were implemented to cover the identified requirements, which are not part of a standard AR solution. Moreover, the need to broaden our audience and reach a critical mass of citizens has been also the motivation to introduce some new developments outside the usual track of augmented reality solutions. All the above qualifies the Live+Gov solution as the appropriate technology to facilitate augmented reality in the domain of eGovernance.

In the following, we start by summarizing the AR-related revised requirements that are drawn from the related descriptions of deliverable D5.3 [3], as well as the actions that have been undertaken to address them. Then, we describe in detail how the Live+Gov solution for AR has been extended to serve content across platforms and vendors. Finally, we present the eGovernance Augmentation Layers that have been developed to facilitate the execution of the 2nd field trials.

2.2 Revised requirements and extensions compared to the first prototype

During the first phase of the project the AR-related functionalities were tested in the context of the urban planning field trial that was conducted in the region of Gordexola, Spain. The field trial was conducted among a controlled group of testers that were asked to use the application in order to view three future plans and provide their feedback. The scenario involved the use of augmented reality for viewing the 3D models of future plans in their "actual" location, as well as the use of the apps questionnaires to provide their feedback.

Although the evaluation feedback in D5.3 was collected along the typical user evaluation dimensions (i.e. usability, reliability, learnability, utility, memorability, satisfaction and efficiency) in the following we take the developer's perspective and primarily focus on the AR-related requirements. Our intention in Table 2.1 is to provide a rough description of the changes that were deemed necessary for the second field trial. The detailed description of the evaluation results can be found in D5.3 [3].

Table 2.1: Changes that were deemed necessary for the second field trial

Issues in the 1st field trial	
Stability Improve reliability in offered functionalities	The testers experienced many crashes and un-expected behaviors in the promised functionalities. This was due to the loose integration between the AR-library and the base mobile application, as well as the existence of some problematic functional scenarios that were under-estimated.
User Interface Improve Layout and make labels and tags more self-explanatory	The menus and workflows were not sufficiently self-explanatory so as to allow users to easily navigate through the offered information and functionalities. Moreover, there was an additional requirement for bi-language support including both Spanish and the Basque language.
Feedback information Offer meaningful user notifications	The limited number of messages provided by the app was not sufficient for explaining to the user the cause of crashes or un-expected behaviors.
Localization Improve user location-based detection	In a non-trivial number of cases the presentation of 3D models through the augmented reality view failed to offer the intended experience. This was due to the insufficient accuracy on the user's localization that resulted in the misplacement of the objects
Image recognition Improve recognition in image based triggering of AR content	The non-robust recognition of marker images caused the 3D models to frequently disappear, or tremble.
Context awareness	The field trial owners decided that it would be very valuable for them to obtain the context of each submitted vote (i.e. the

Capture and submit additional information about the citizen’s context	location where the vote was casted)
Bandwidth Improve bandwidth usage	Due to the increased size (in MBs) characterizing some of the utilized 3D models, their downloading (especially in bad network conditions) caused the app to either crash or become un-responsive.
Battery consumption	Being a resource-intensive functionality there were many cases where the continuous use of the AR view drained the entire phone’s battery.

All aforementioned requirements were taken as input for engineering the final prototype of the AR-component. Section 2.3 describes in detail the actions that have been implemented to address these requirements.

However, apart from the aforementioned table dealing with the improvement of the user’s experience, the execution of the field trial revealed also the need for higher citizens’ engagement. More specifically, there was a vital requirement to maximize the potential of reaching more citizens and achieve a broader reach. The fact that, during the first trial, the application was available only for Android devices limited the number of users that could participate in the trial. Moreover, the fact that the future plans of Gordexola were only accessible through the custom mobile app developed by Live+Gov raised a considerable barrier for the citizens. This refers to the citizens that, although frequent users of mobile augmented reality (e.g. through the widespread mobile browsers for augmented reality like Junaio, Layar and Wikitude), didn’t have the chance to hear about and download the custom mobile app developed by Live+Gov. As a consequence, the final prototype of the Live+Gov AR component was extended not only to serve content across platform and vendors but also to semi-automate the process of creating AR-views from standard web-sites (see Section 2.4).

2.3 Improving the augmented reality experience

During the second development cycle the integration of the AR library with the base mobile application become much tighter, which resulted in solving a large number of implementation bugs. As a consequence, the AR functionality offered through the mobile app developed for Urban Planning became much more reliable and robust. The robustness of the functionality was further supported by the execution of three testing cycles prior to the beginning of the field trial. The results of this testing procedure as well as the re-actions that were undertaken to tackle the identified problems are described in D4.3 [6]. In the following, we briefly outline some of the modifications that have been undertaken to improve the augmented reality experience.

Improved layout design: The design of the app, and as a consequence of the AR functionality, was significantly improved allowing the user to more easily understand the cause of the presented urban plans and the expected feedback. The button labels were renamed to more accurately communicate the intended message and the interface of both markers and billboards was significantly improved. Moreover, the functional workflows and labels prompting the user for action were made consistent across both Android and iPhone. Finally, with respect to the language, we have decided to determine the menu language based on the language determined by the user in the phone settings. On the other hand, the questionnaires were selected to be bi-lingual so as to comply with the standard practices in the Basque country.

Rich set of messages and notifications: A large number of messages have been included in the app with the aim, on the one hand, to warn the user about the potential non-optimal experience due to weak sensor signals (e.g. low accuracy in positioning, absence of mobile data network or WiFi) and, on the other hand, to instruct him on how to make the situation better (e.g. perform calibration on his compass, turn-on GPS and 3G for better accuracy in positioning).

Improved AR experience of 3D models based on location: One of the most important shortcomings of AR usage during the first trial was the rather poor experience of the user in watching the 3D models of the urban plans jumping from one place to another. This was mainly due to the inability of the phone's locations service (i.e. service that detects the user's position making combined use of GPS, 3G and – if available – WiFi) to specify the user's position with very high accuracy. In our scenario where the user is standing very close to the 3D model of an urban plan, the location service continues to update the user's location even if he is standing still, resulting in a poor user experience. In order to cope with this issue the improved version of the AR library incorporates a mechanism to stop the location updates, after the user's location accuracy stops improving. We decided not to just lock the location on the first update because there might be the case when the GPS is still searching for a good signal, resulting in a very inaccurate location. By stopping the location updates, the 3D models demonstrating the future plans are displayed smoothly eliminating the "jumping" effect, but raises the issue that if the user is moving, the location will not be accurate anymore. To solve this issue the location updates need to be restarted when the user is moving. In order to do this, we have decided to let the user reset the location himself by shaking the device. When a shake gesture is detected the device is unlocked and the position of the 3D models is refreshed.

Improve AR experience of 3D models based on visual recognition: Another shortcoming of the AR functionality that resulted in poor user experience was the sub-optimal performance of the image markers in acting as the visual recognition triggers of the 3D urban plans. The quality of experience was further reduced by the fact that there was no provision in the first prototype in dynamically specifying the scale and orientation of the displayed 3D models. As a consequence, in some cases, the models appeared oversized and heading to the wrong direction. During the second development cycle the AR server[1] was extended to also incorporate the dynamic configuration of the model's size and orientation, allowing the

[1] http://augreal.mklab.iti.gr/

administrator of the AR platform to have full control over the appearance of the 3D model. The orientation around the z-axis and the distance of the object from the image capturing device could be set up from the web portal with a graphic user interface (see Figure 2.1). Furthermore, the Metaio API was configured so as to recognize faster the printed patterns and prohibit (as much as possible) any trembling effects of the 3D models.

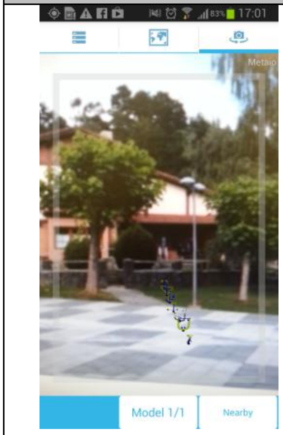
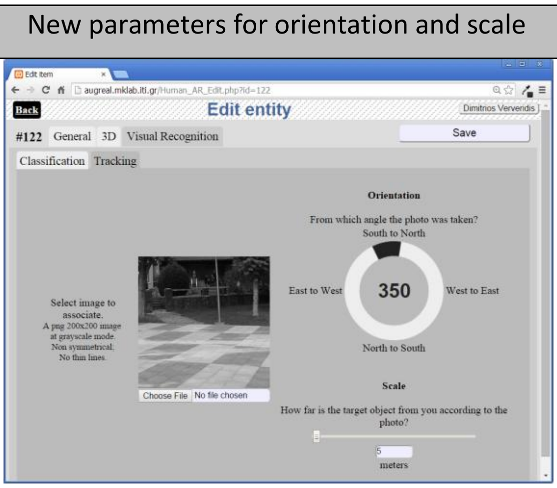
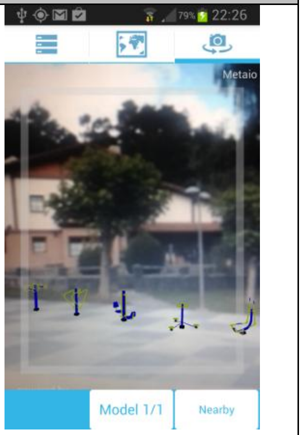
Before	New parameters for orientation and scale	After
	 Edit item augreal.mklab.iti.gr/Human_AR_Edit.php?id=122 Back Edit entity Dimitrios Vervendis #122 General 3D Visual Recognition Save Classification Tracking Select image to associate. A png 200x200 image at grayscale mode. Non symmetrical. No thin lines. Choose File No file chosen Orientation From which angle the photo was taken? South to North East to West 350 West to East North to South Scale How far is the target object from you according to the photo? 5 meters	

Figure 2.1: Enhanced configuration environments for setting the orientation around the z-axis and determining the distance of the object from the image capturing device

In both the location-based and visual-based triggering of the 3D models, we also had to employ a set of refinements dealing with the models' colour and brightness so as to ensure the best user's experience.

Context awareness: The functionality of the app was redesigned so as to collect the user's location information along with his/her answers, when submitting a questionnaire. In this way, each vote is also characterized with a GPS coordinate indicating the location where it was casted.

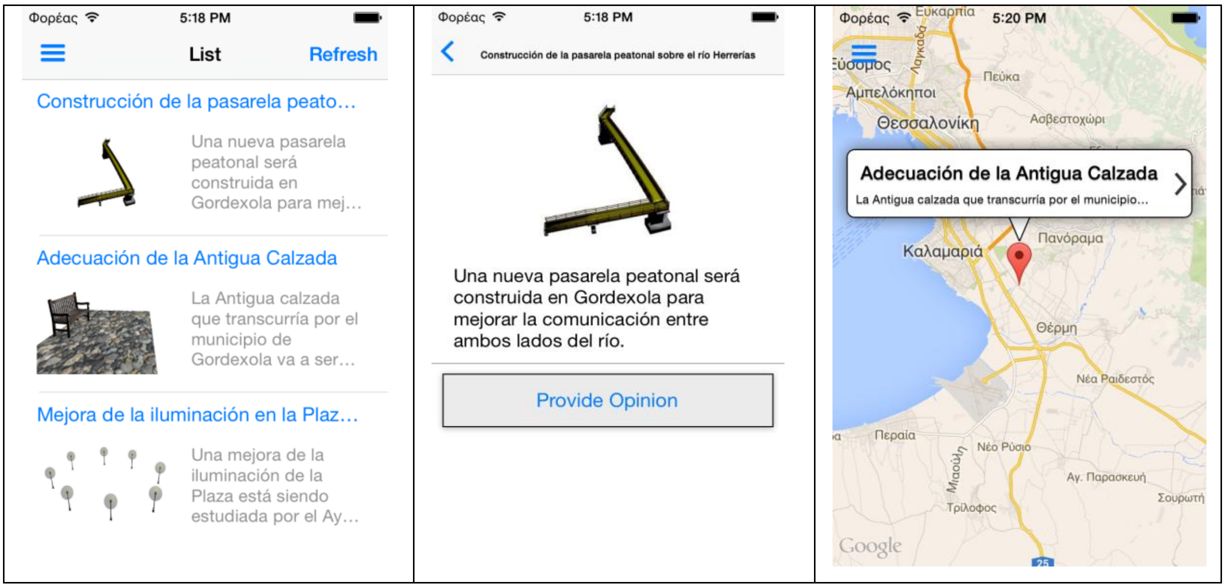
Battery-consumption optimization: The prolonged use of the AR view resulted in heating-up the phone and rapidly consuming its battery, since AR is a resource-intensive functionality. As a consequence, the testers of the first field trial were frequently confronted with the situation of having their battery drained even without making extensive use of the application. This situation was not merely a result of the resource-intensive nature of the AR functionality, but was also caused by the fact that the AR thread remained open in the background even when the user was browsing the map or the list view. Although this strategy achieved very low response times when switching to the AR view, it was far from optimal from the perspective of battery consumption. The strategy adopted for coping with this problem was to pause the AR thread when the AR view was inactive. In this way, we were able to significantly reduce the level of battery consumption when the app was not showing the AR view. Moreover, we have implemented an additional check mechanism that was configured to prompt the user to switch from the AR view, when the temperature of the phone's battery exceeds 25 Celsius degrees.

Bandwidth usage optimization: The increased size of the 3D models demonstrating the urban plans required the app to download a significant amount of content before becoming operational. This size was more or less proportional to the quality of the 3D models. So, there was a trade-off between the quality of experience and the usability of the app. In order to improve the situation we have decided, first, to further compress the quality of the 3D models so as to reduce the size of the downloaded content and, second, to make extensive use of caching and client-server synchronization mechanisms ensuring that almost 90% of the necessary content is downloaded only in the first execution of the app.

2.4 Serving AR content automatically across platforms and vendors

As already mentioned one of the most important requirements that came out of the first field trial was the need for maximizing the potential of reaching more citizens and achieving a broader reach. Motivated by this requirement we have decided to extent the Live+Gov solution for mobile augmented reality along the following axes:

a) Next to the Android version that was used during the first field trial we have decided to make the app available also for iPhone users as a native application (see Figure 2.2).



Construcción de la pasarela peato…

Una nueva pasarela peatonal será construida en Gordexola para mej…

Adecuación de la Antigua Calzada

La Antigua calzada que transcurría por el municipio de Gordexola va a ser…

Mejora de la iluminación en la Plaz…

Una mejora de la iluminación de la Plaza está siendo estudiada por el Ay…

Construcción de la pasarela peatonal sobre el río Herrerías

Una nueva pasarela peatonal será construida en Gordexola para mejorar la comunicación entre ambos lados del río.

Adecuación de la Antigua Calzada

La Antigua calzada que transcurría por el municipio…

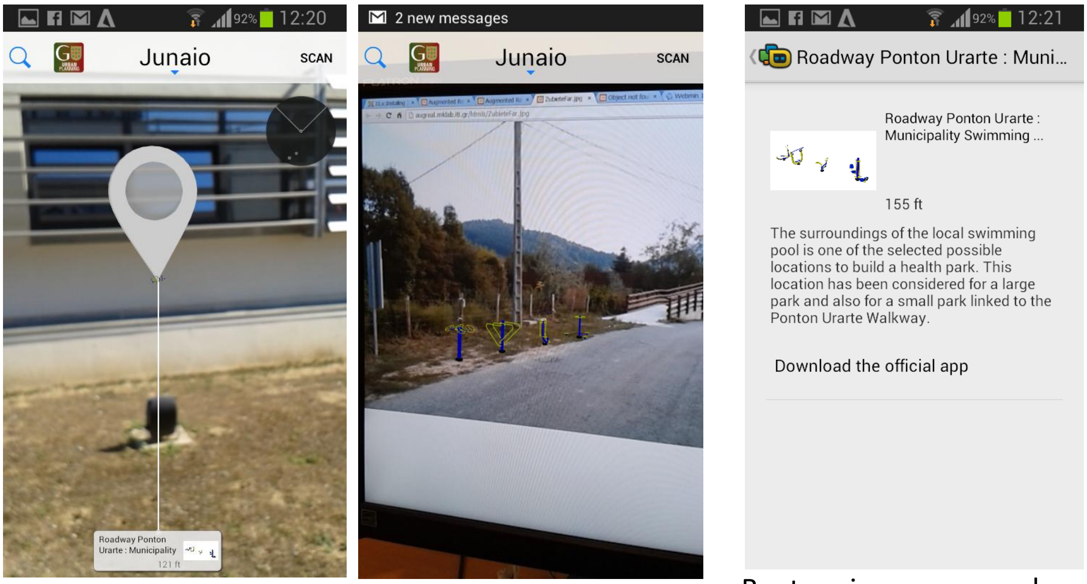
Figure 2.2: Screenshots of the iPhone app that was implemented for the purposes of the Urban Planning field trial

b) Apart from the customized mobile app that was developed within the project for the purposes of the urban planning field trial, we have extended our AR server so as to serve the exact same urban plans though three of the most widely established browsers for mobile augmented reality, namely Junaio, Layar, and Wikitude. All three browsers are available for both Android and iOS systems and are already enjoying significant popularity among mobile users. However, as each of the aforementioned browsers allow only for specific features through its respective API, the functionalities per browser differ. These supported features are outlined in Table 2.2.

Table 2.2: Live+Gov functionalities exported to widely established mobile browsers

	Location based AR	Image based AR
Junaio (http://www.junaio.com/)	Yes	Yes
Layar (https://www.layar.com/)	Yes	No
Wikitude (http://www.wikitude.com/)	Yes	No

Some indicative screenshots from the urban planning channel visualized through the Junaio browser are shown in Figure 2.3.



Roadway Ponton Urarte : Muni...

Roadway Ponton Urarte : Municipality Swimming ...

155 ft

The surroundings of the local swimming pool is one of the selected possible locations to build a health park. This location has been considered for a large park and also for a small park linked to the Ponton Urarte Walkway.

Download the official app

The Gordexola urban planning channel as visualized by the Junaio mobile augmented reality browser.

By tapping on an urban plan the user sees more details and he is prompted to download the official app.

Figure 2.3: The urban plans of Gordexola visualized through the Junaio mobile browser

c) To facilitate the smooth integration of the AR Server's web configuration tool into the already existing portal of a municipality, we have decided to re-package this tool as a Joomla directory that could be very easily installed and become operational even by a non-expert. In other words, if the portal of the municipality is already based on the Joomla CMS system, the administrator of the portal can install the "Urban Planning AR directory" with just a few clicks and obtain the full functionality of the AR server (in terms of generating the urban plans and serving them in AR channels).

d) Finally, next to these project-specific actions that were primarily oriented towards increasing the audience of the urban planning field trial, we have also decided to complement the AR Live+Gov solution with an automatic tool that could help ordinary users to turn their web-site into a fully functional augmented reality channel.

In the remaining of this section we present in detail the implementations that have been undertaken to facilitate the third and fourth bullet.

2.4.1 Motivation

Given that the vast majority of municipalities have already a Content Management System (CMS) that hosts their portal and services, there is a great potential to benefit from the strict structure imposed by those systems and automate the process of generating augmented reality channels/layers. Indeed, one of the most popular CMSs is Joomla[2]. It offers a back-end graphic user interface that allows to easily configure a web-portal and also to install new functionalities with a single click. The basic information unit in Joomla is an Article where someone can write his/her content with an HTML editor. However, the Article is a rather abstract structure that is not sufficient for serving concrete applications. To treat this problem, several Joomla directories have become available for allowing the programmers to make their own database Entry with custom fields. These components can be easily incorporated within a Joomla installation and offer a certain type of functionality. One of the most popular components for Joomla that allows for new customizable fields is Sobipro[3].

Sobipro offers a range of tools and wizards for simplifying the creation of a promotional website but at the same time it imposes a rather strict structure on how the data are organized in the database tables. There are several directories for Sobipro that can be installed for creating the desired database scheme. For instance, Sobipro has the “Business Directory” preinstalled that contains a database scheme for storing companies, as well as the style files for presenting them as a web page. There are several other templates such as the “Restaurant Directory”, the “Real-estate Directory”, etc. Following the same rationale we have decided to create the “Urban Planning AR Directory”. By installing this template, the fields that are necessary for hosting the urban plans are automatically created in the underlying database.

In addition, having ensured a rather strict structure on the underlying content, we were able to proceed with the implementation of FastAR[4], which has been implemented as a Joomla component that is freely available as open source for Joomla 2.5 or later, under the Affero GPL license. In principal, the goal of FastAR has been to exploit the consistency of the underlying data structure in order to automate the process of turning an ordinary website hosted by Joomla-SobiPro, into an augmented reality experience viewed through mobile AR browsers. The logical connection between Joomla, Sobipro, the Urban Planning AR Directory and FastAR is depicted in Figure 2.4.

[2] http://www.joomla.org/

[3] http://extensions.joomla.org/extensions/directory-a-documentation/directory/16649

[4] http://arexporter.mklab.iti.gr

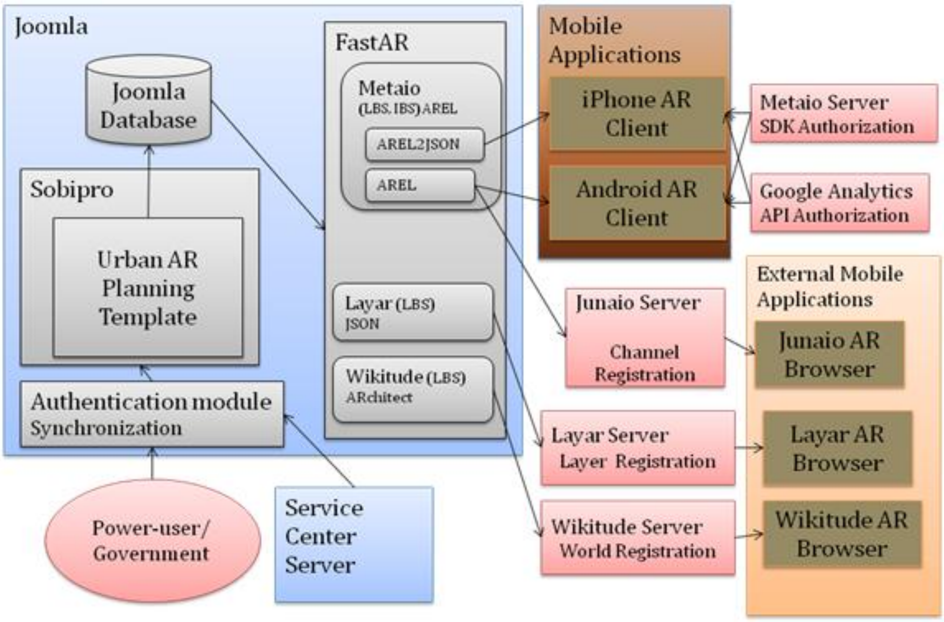


Figure 2.4: Connection between Joomla, SobiPro, “Urban Planning AR directory” and FastAR.

2.4.2 Existing solutions for creating and publishing AR content

In order to render content as part of an augmented reality world there are two principle choices that have to be made, namely how to publish the AR content and how to translate your data in an AR-compatible format. In D3.1 [1] we have reviewed the existing options with respect to their pros and cons. In the following, we briefly summarize the investigated options and motivate the choices adopted for the implementation of FastAR. In detail, there are four options for publishing AR content, namely: a) using a commercial SDK to build a custom AR browser, b) through free AR browsers provided by certain companies, c) using open source toolkits to make a custom AR browser, and finally d) using an HTML5 based AR browser for exploiting the web browser capabilities.

Case (c) was adopted at the early stages of the project but it was abandoned due to the low quality of the result and the excessive amount of required effort. Case (d) was rejected because it leads to a solution with low rendering capabilities for 3D content and it also requires great programming effort. Case (a), on the other hand, resulted in a professional application but its drawback was that it required a significant amount of effort for implementing the mobile application for both Android and iPhone operating systems. Thus, case (b) was adopted in our implementation by making FastAR to comply with the Junaio, Layar and Wikitude APIs. In other words, upon its successful installation FastAR will be able to serve the website content through the AR browsers of all three aforementioned platforms.

The next step relates to the approach used for creating AR-compatible content. In the typical case, one can rely on the tool that is offered by each vendor so as to generate content compatible with its AR browser. Indeed, the majority of the existing platforms provide a desktop or a web based CMS usually named as “Studio” or “Creator” for creating AR-compatible content. Apart from the AR vendors, third party solutions also exist. In Table 2.3, the existing third-party CMSs for AR are outlined. BuildAR is a CMS partnered with Metaio for generating content compatible with the Junaio browser. Visar is a CMS for generating content that is compatible with Layar. Similar is the functionality of Dimple, Hippon, Aurasma, Catchhoom, Zapcode, Poistr and Poiz that are CMSs developed to simplify the process of generating AR content and allow non-expert users to generate their own augmentation views.

Table 2.3: AR Content Management Systems Review

Name	License	Modes	Platform	Content	Comments	Webpage
BuildAR	Commercial	LBS	Junaio	Image,Video,3D	No 3D canvas	buildar.com
Dimple	Commercial	LBS, IBS	Layar	Image, Sound, Video	No 3D support	dimplecms.com
Feedgeorge	Open	LBS, IBS	Layar	Image	No 3D support, Wordpress plugin	wordpress.feedgeorge.com
Hippo	Commercial	LBS	Layar	Image, Sound, Video	No 3D support	onehippo.com
Hoppala	Commercial	LBS	Junaio, Layar, Wikitude	Image, Sound, Video, 3D	No 3D canvas	hoppala-agency.com
Visar	Commercial	LBS, IBS	Layar	Image, Sound, Video, 3D	-	Visar.biz
Poistr	Commercial	LBS	Layar, Junaio	Image	Import from kml, free for less than 30 Pois	poistr.com
Poiz	Commercial	LBS, IBS	Layar	Image	-	poiz.biz
ConnectAR	Commercial	LBS, IBS	Junaio	Image, 3D	Joomla, Wordpress, Typo3 components	connectar.com

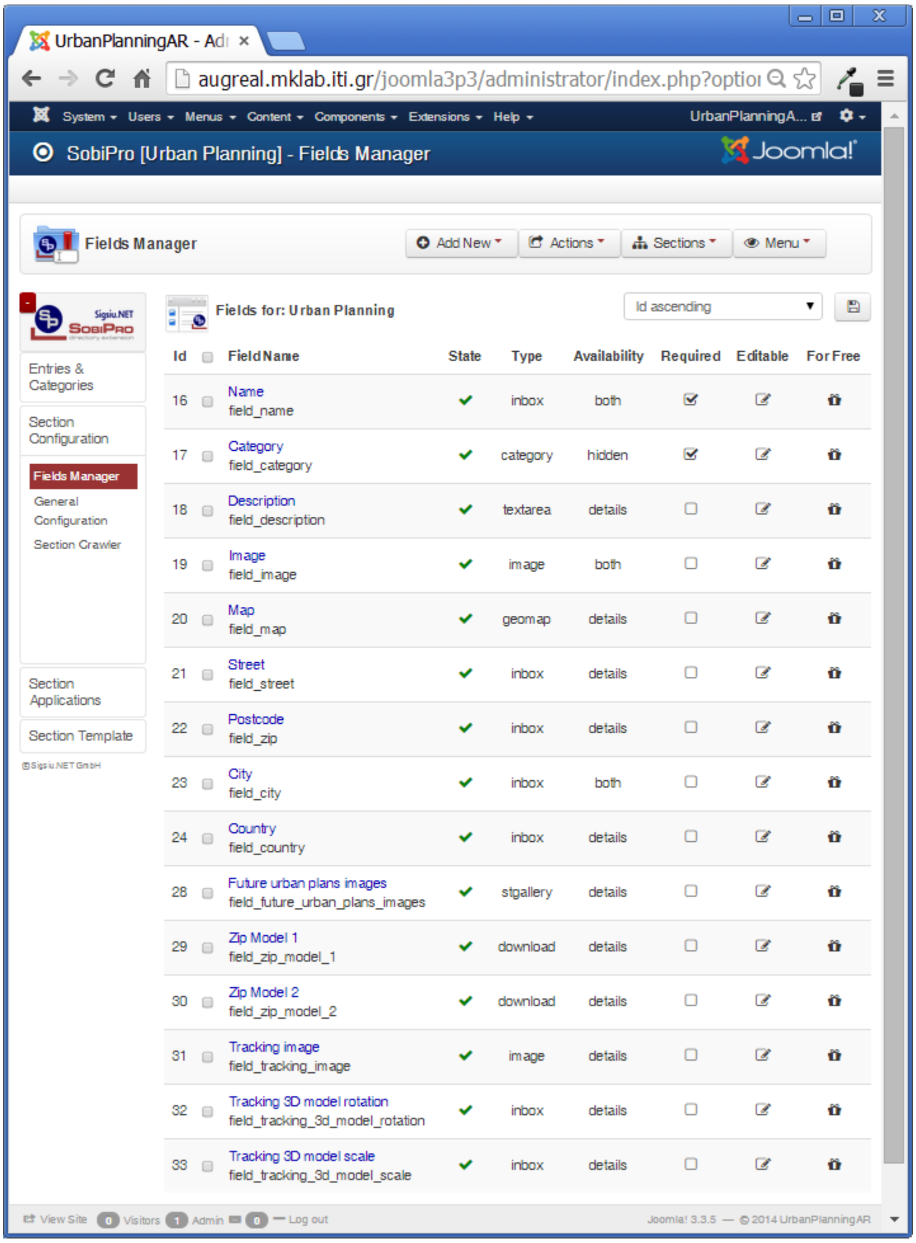
Nevertheless, despite the existence of all these tools, it is only recently where the need to generate AR content in large scale has motivated the development of tools for automating this process. Feedgeorge is an AR plugin for Wordpress that supports location and image based AR through Layar. One can install the Feedgeorge AR plugin in his/her Wordpress CMS and then an AR tab appears for each post in the back-end. In the AR tab one can select the AR resources such as the geographic coordinates of the post or the image for triggering the post through vision recognition. However, the drawback of Feedgeorge is that the data unit information is non-modifiable, i.e. a post witch custom HTML code. On the same principle, ConnectAR is a commercial plugin for several CMSs to export data into Layar and Junaio browsers. It supports Typo3, Joomla, and Wordpress for exporting a page, an article, or a post, accordingly. It additionally supports 3D models with several options for rotation and scaling, as well as previewing in a 3D canvas. ConnectAR suffers also from the problem of non-customizable data units, which is a feature necessary for making new field types.

Motivated by the same trend (i.e. generate AR content in large scale) we decided to proceed with the following developments:

a) Re-package the AR Server infrastructure and web configuration as an "Urban Planning AR directory" that could be easily installed in an already existing installation of Joomla-SobiPro and offer the full AR functionality developed in Live+Gov with just a few clicks
b) Develop FastAR as a complementary tool that can automatically transform already existing web-sites (that have been developed based on Joomla-SobiPro) into AR channels displayed through the mobile browsers of the three major AR platforms (i.e. Metaio, Layar, Wikitude).

2.4.3 Urban Planning AR directory

The Urban Planning AR directory was designed to incorporate the full range of functionalities offered by the Live+Gov AR server (see D3.1 [1]), while taking advantage of the already existing interfaces and functionalities offered by SobiPro. The directory can be exported to a zip file and be installed in other Sobipro based sites. In Figure 2.5, the fields included in the directory for hosting the urban plans are displayed. It is evident that the information incorporated in this directory (i.e. name, description, image, map with GPS coordinates, etc) incorporates all the information necessary for generating AR entities.



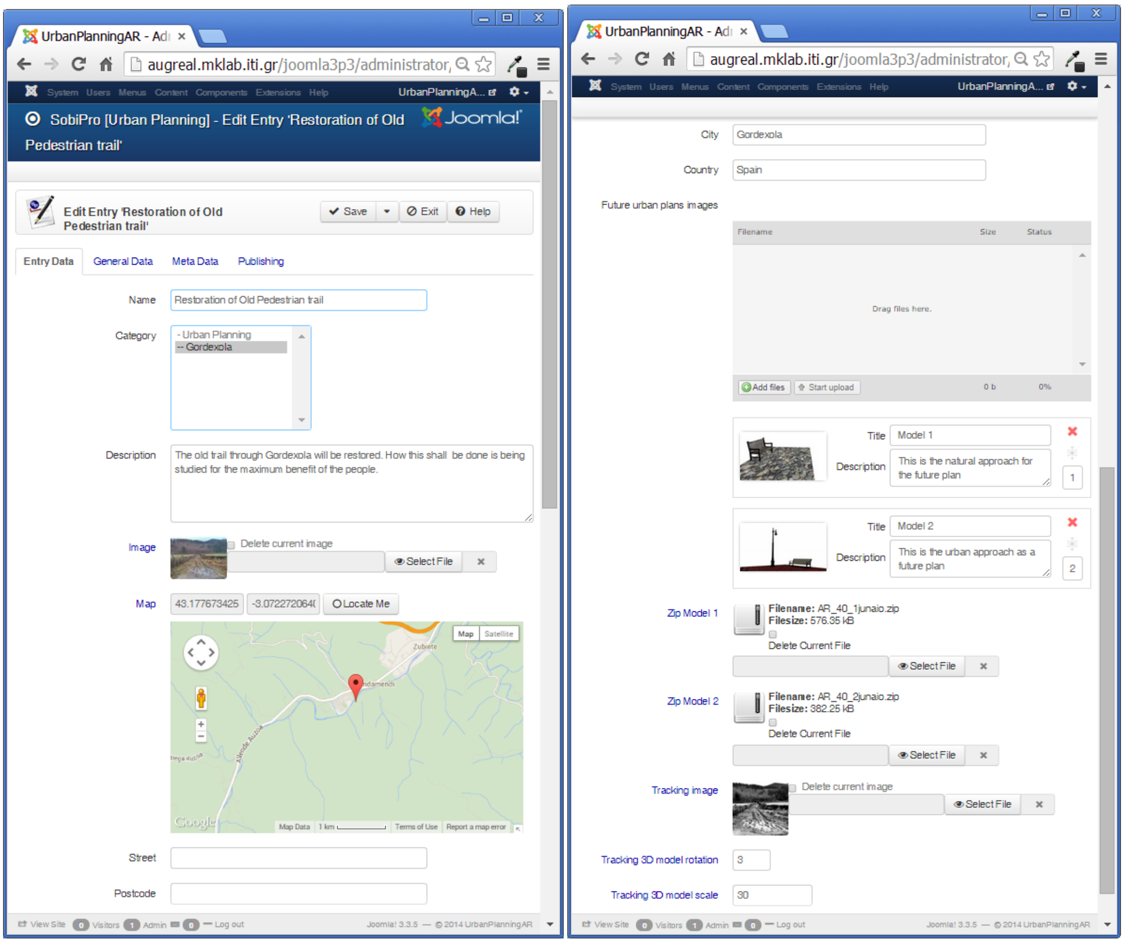
Id	☐	Field Name	State	Type	Availability	Required	Editable	For Free
16	☐	Name field_name	✓	inbox	both	☑	✎	🎁
17	☐	Category field_category	✓	category	hidden	☑	✎	🎁
18	☐	Description field_description	✓	textarea	details	☐	✎	🎁
19	☐	Image field_image	✓	image	both	☐	✎	🎁
20	☐	Map field_map	✓	geomap	details	☐	✎	🎁
21	☐	Street field_street	✓	inbox	details	☐	✎	🎁
22	☐	Postcode field_zip	✓	inbox	details	☐	✎	🎁
23	☐	City field_city	✓	inbox	both	☐	✎	🎁
24	☐	Country field_country	✓	inbox	details	☐	✎	🎁
28	☐	Future urban plans images field_future_urban_plans_images	✓	sigallery	details	☐	✎	🎁
29	☐	Zip Model 1 field_zip_model_1	✓	download	details	☐	✎	🎁
30	☐	Zip Model 2 field_zip_model_2	✓	download	details	☐	✎	🎁
31	☐	Tracking image field_tracking_image	✓	image	details	☐	✎	🎁
32	☐	Tracking 3D model rotation field_tracking_3d_model_rotation	✓	inbox	details	☐	✎	🎁
33	☐	Tracking 3D model scale field_tracking_3d_model_scale	✓	inbox	details	☐	✎	🎁

Figure 2.5: List of field types included in the Urban Planning Template

Furthermore, the administrator can add new fields with the GUI of Sobipro for cases that require additional data. In Figure 2.6, the form to document an urban plan is displayed.

Sobipro provides a standardized GUI based form that is widely known by Joomla users. In this manner, the time required for training at the Urban Planning AR directory is minimized.

Finally, apart from the front-end related functionalities the Urban Plan AR directory incorporates all necessary mechanism for serving the generated content in the appropriate format, so as to be accessible from the major AR platforms (i.e. Metaio, Layar, Wikitude).



SobiPro [Urban Planning] - Edit Entry 'Restoration of Old Pedestrian trail'

Edit Entry 'Restoration of Old Pedestrian trail'

Name: Restoration of Old Pedestrian trail

Category: - Urban Planning; -- Gordexola

Description: The old trail through Gordexola will be restored. How this shall be done is being studied for the maximum benefit of the people.

Map: 43.177673425 -3.072272064(

City: Gordexola

Country: Spain

Future urban plans images

Title: Model 1
Description: This is the natural approach for the future plan

Title: Model 2
Description: This is the urban approach as a future plan

Zip Model 1 — Filename: AR_40_1junaio.zip Filesize: 576.35 kB

Zip Model 2 — Filename: AR_40_2junaio.zip Filesize: 382.25 kB

Tracking 3D model rotation: 3

Tracking 3D model scale: 30

Joomla! 3.3.5 — © 2014 UrbanPlanningAR

Figure 2.6: The form to fill for a new urban plan.

2.4.4 FastAR on Sobipro

As already mentioned, the goal of FastAR is to automate the process of turning an ordinary (and already existing) website hosted by Joomla-SobiPro, into an augmented reality experience viewed through mobile AR browsers. However, developing a software module that could automatically export the content of a website into an AR-compatible format was far from trivial. This is due to the wide variety of programming skills needed for web, mobile and database development, as well as the experience required in using the Application Programming Interfaces (APIs) offered by the existing AR browsers. In developing FastAR we have faced a number of technical challenges. First, we had to deal with the fragmentation of the existing technologies and standards in both creating and publishing AR content. Next, we had to analyze the functional scheme of use for Joomla and SobiPro, so as to derive the core part of the database that remains unaltered independently of the website's nature. Finally, we had to make the necessary adaptations so as for the AR view to be fully integrated and interconnected with the web view. In the following we discuss the solutions that have been adopted to address these challenges.

2.4.4.1 Architecture

The adopted architecture is outlined in Figure 2.7. Joomla CMS serves as a host for Sobipro and FastAR components. This structure is the critical feature exploited by FastAR so as to automate the process of generating AR content. More specifically, FastAR is also implemented as a Joomla component that is capable of undertaking the following two tasks. First, guided by the strict database structure imposed by Sobipro, FastAR is able to read the database entries and generate the AR entities. These AR entities are necessary for synthesizing the AR channel/layer. Second, FastAR interconnects with the servers of the supported AR vendors (i.e. Metaio, Layar and Wikitude) that are responsible for sending the data streams to the corresponding AR mobile browsers. In this way, the database entries that were originally created by the content owner of the website are automatically transformed into AR entities that are now visible over an AR view and through AR browsers that are already installed in millions of mobiles devices.

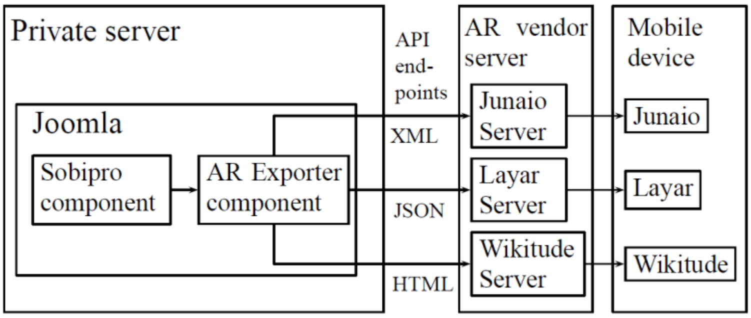


Figure 2.7: The adopted architecture for exporting Sobipro content to AR browsers

In implementing the aforementioned architecture there are two major technical challenges that need to be addressed. The first relates to the generation of fully functional AR entities out of the database entries that have been originally created to support an Internet website. The second has to do with structuring and exporting these AR entities to the appropriate format so as to be compatible with the standards supported by each vendor.

2.4.4.2 Joomla standards for supporting the generation of AR content

Joomla stores html pages in a MySQL database to present them in the web portal, also called as front-end. Joomla is based on the model-view-controller (MVC) scheme. Briefly, the model defines the database schema, the controller defines the actions and view defines the presentation format. However, storing html pages in the database is not an efficient way of saving content when the html pages present the same kind of content, e.g. companies or urban plans. It is typical for content modules like Sobipro to inherit the MCV scheme and make it more specific, so as to simplify the process of generating the necessary html pages. Sobipro stores only the fields that describe the content, e.g. the title, the description, the image but not the whole html page. As a result, it is necessary for a website that has been developed using Sobipro to incorporate the same minimum amount of information for each of unit (e.g. a product, a company, an urban plan, etc). More specifically, the type of information for each unit consists of title and description. The user can set other fields to be mandatory such as telephone, website, image or add new fields such as longitude and latitude.

On the other hand, mobile AR works by overlaying two layers. The bottom layer is the actual world layer which is visible through the device camera. The upper layer consists of the AR entities that are computer graphics such as text, images, 2D drawings, and 3D models in a virtual reality (VR) world. Another essential element of mobile AR is the triggering mechanisms that determine the conditions that must be met in order for an AR entity to appear. The most typical triggering mechanisms are based on location and the proximity between the location of the mobile user and the AR entity (i.e. usually mentioned as location-based channels). Thus, the minimum amount of information that is necessary for building a fully functional AR entity is: title, description, longitude, latitude, and logo image.

It is evident from above that the minimum amount of information per unit, enforced by SobiPro, is sufficient for creating the AR entities that are necessary to support an AR view. As a consequence, FastAR has been designed to automatically retrieve the necessary information from the database of SobiPro and create the AR entities. These AR entities are subsequently organized into the appropriate structures using a set of controllers, as described in the following section.

2.4.4.3 AR standards across vendors

This section provides details about the standards used by each vendor to stream the AR content. It also describes how the proposed design copes with the variety of protocols. AR browsers work by placing a web view on top of a 3D graphics view, and both of them on top of a camera view. In this manner, the interface is loaded in the web view, the 3D graphics view renders the 3D models, and the camera view captures the real world. The necessary resources are provided using html, javascript, and css languages. Text or multimedia data are provided by html; javascript is used to define the human interface actions; and css sets the color style of the web view.

The *Junaio* browser is authored by Metaio Company. The standard used by Metaio is the AR Experience Language (AREL), which is comprised of: a) XML that contains the urls of the resources; b) javascript indicating the scenario of the graphic user interface; and c) css for formatting the style of the interface. Javascript and css are loaded in a web view in a similar way that it is loaded by web browsers, with the difference that the web view has transparent background to allow the camera preview surface to be visible. *Layar* uses javascript object notification (json) format for sending data. The json file contains both the resources (text, location coordinates, image links, etc.) as well as the actions allowed by the user during the immersion (e.g. onClick). The user actions allowed are calls to other applications, namely telephone, sms, e-mail, url, another layer, or share via social networks (through URIs). The user interface is stylized in the channel registration portal. *Wikitude* supports three communication languages: a) the KML (Google Earth) that should be used for a simple location based channel; b) the ARML (OpenARML.org) which is an advanced version of KML that can be used for a location based channel with a custom user interface; and c) ARchitect which is the main language for allowing all features needed for either location or image based channels. ARchitect is formed by HTML, javascript, and CSS in a way similar to the Metaio's AREL language.

In dealing with the variability of the existing AR standards we have decided to adopt an open architecture for our AR Exporter. More specifically, although we have defined a common structure for keeping the data that are extracted from the SobiPro database, the transformation of these data into the appropriate AR language is being performed through vendor-specific controllers. Thus, our current implementation incorporates three controllers that export the content in three different formats, namely XML for Metaio, JSON for Layar and HTML for Wikitude. In a similar fashion, a new controller can be implemented to provide support for an additional vendor. FastAR automatically generates the API Endpoints, i.e. the URLs of the aforementioned controllers so as to be used for registration in the portal (server) of the corresponding AR vendor. Finally, apart from the content, the controllers are also responsible for handling several additional parameters that are received from the mobile AR browser. Such parameters are the longitude-latitude of the device, the radius in meters around the device of the entries to download, the maximum number of entries, and the keywords for searching for an AR entity.

2.4.4.4 Integration of FastAR with the main application

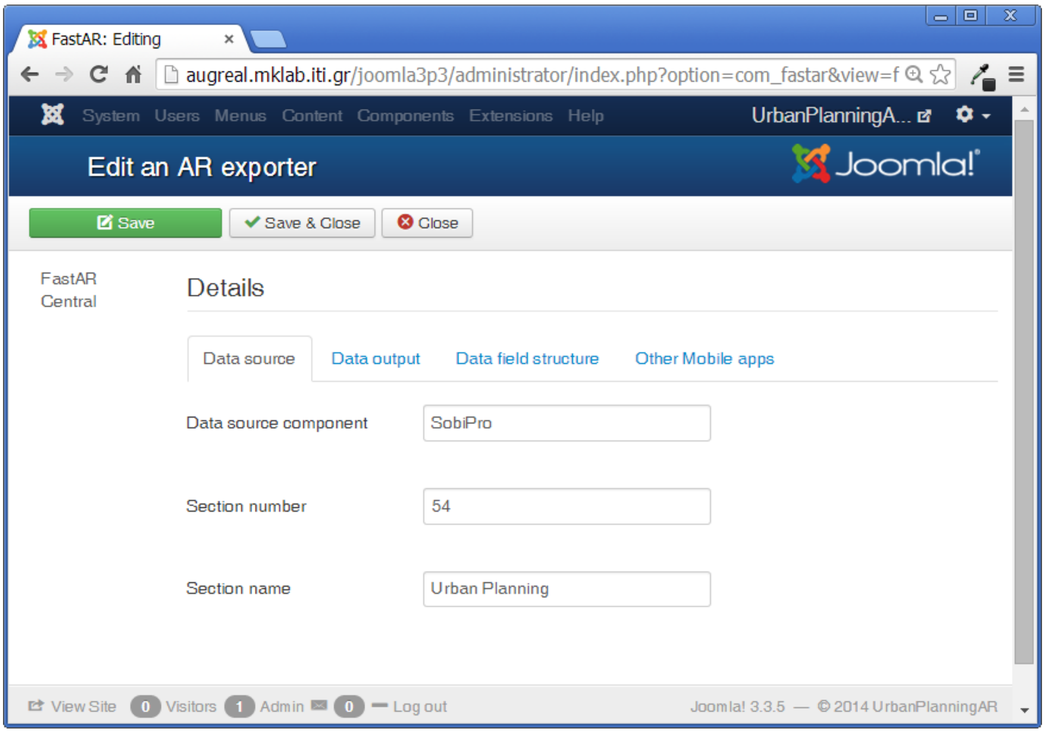
FastAR should be installed into the Joomla framework in order to offer the Augmented Reality functionality. FastAR automatically scans the Joomla framework database in order to find details about the Sobipro sections available. Then, for each Sobipro section an *Augmented Reality Exporter* entry is created in a central panel (see Figure 2.8). In the central panel it is possible to select one AR exporter and edit it. It consists of four tabs. The first tab named as “Data source” contains the details of the Sobipro section form which the data is extracted. The second tab named as “Data output” enlists the AR controllers, i.e. the urls of the API Endpoints for each AR Vendor. These are the urls that should be registered in the registry channel of the AR vendor, so as to become available through the corresponding browsers. The third tab named as “Data field structure” enlists the alias of each Sobipro field that should be exported as AR content. To be more specific, the “Title AR field” is the text that should be used on an AR billboard, the “Icon AR field” is the icon in the billboard and the “Description AR field” is the description to be shown when the user taps on a billboard. Similarly, the “Weblink AR field” is the url that should be accessed when the user presses the button “Continue in web page”. This is also the place where the user gets more details about the AR Entity in a web browser. The longitude and the latitude information are strictly extracted with two options, namely: (a) From plain Sobipro text fields with alias 'field_latitude' and 'field_longitude', or (b) If "geomap" field extension is installed in Sobipro, from a map field with alias 'field_map'.

After entering the necessary information in the FastAR central panel, the next step is to register the API Endpoint URL at each AR Vendor. The registration details depend on each vendor and the user should visit Junaio[5], Layar[6], and Wikitude[7] channel registration sites for more details. Figure 2.9 refers to the Junaio case and demonstrates the place in the corresponding web page where the API endpoint url should be inserted.

[5] http://dev.junaio.com/

[6] https://www.layar.com/my-layers/

[7] http://www.wikitude.com/developer/tools/publish-in-wikitude


FastAR: Editing
augreal.mklab.iti.gr/joomla3p3/administrator/index.php?option=com_fastar&view=f
System Users Menus Content Components Extensions Help
UrbanPlanningA...
Joomla!
Save
Save & Close
Close
View Site
0 Visitors
1 Admin
0
Log out
Joomla! 3.3.5 — © 2014 UrbanPlanningAR

Edit an AR exporter

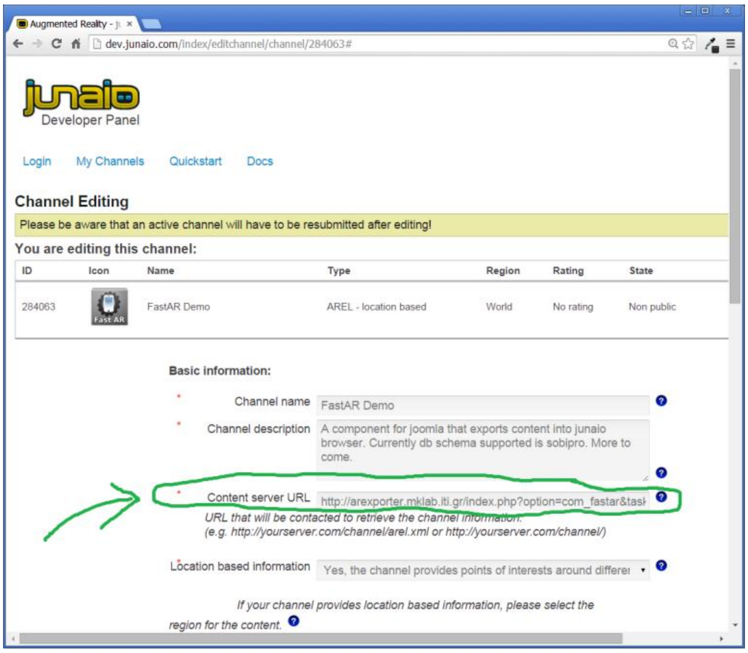
FastAR Central

Details

Data source | Data output | Data field structure | Other Mobile apps

Data source component	SobiPro
Section number	54
Section name	Urban Planning

Figure 2.8: Central panel of FastAR (see http://arexporter.mklab.iti.gr/index.php/installation for more information)



Channel Editing

Please be aware that an active channel will have to be resubmitted after editing!

You are editing this channel:

ID	Icon	Name	Type	Region	Rating	State
284063	Fast AR	FastAR Demo	AREL - location based	World	No rating	Non public

Basic information:

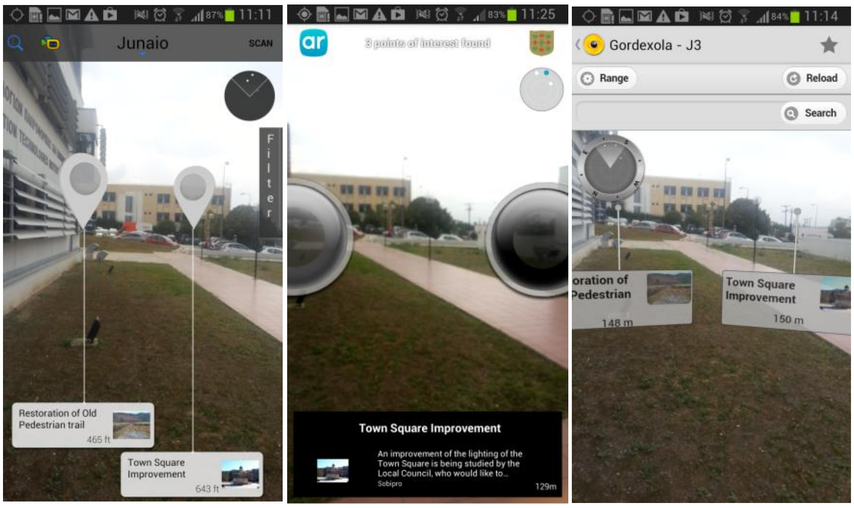
* Channel name	FastAR Demo
* Channel description	A component for joomla that exports content into junaio browser. Currently db schema supported is sobipro. More to come.
* Content server URL	http://arexporter.mklab.iti.gr/index.php?option=com_fastar&task
Location based information	Yes, the channel provides points of interests around differe

URL that will be contacted to retrieve the channel information.
(e.g. http://yourserver.com/channel/arel.xml or http://yourserver.com/channel/)

If your channel provides location based information, please select the region for the content.

Figure 2.9: Registration of the API Endpoint url in the Junaio Developer panel.

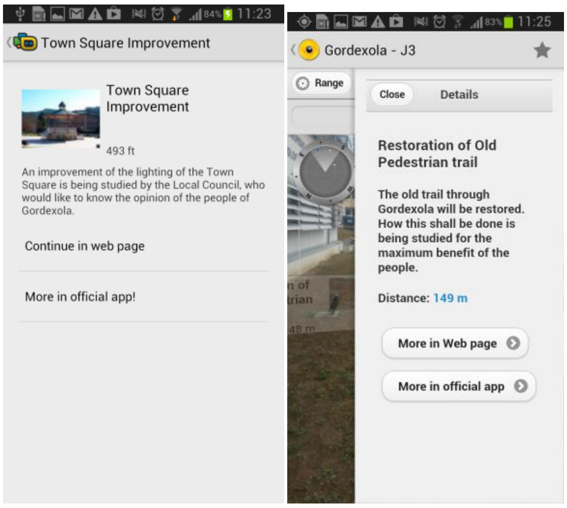
Third, after the successful registration of the API endpoint url to the AR Vendors, the Sobipro data is available as a channel in each AR browser. One should open the browser and select the registered channel by text search or by scanning the QR code given from the AR Vendor. After successfully opening the channel, an AR environment is shown as in Figure 2.10. Each AR Entity is represented by a *billboard* than contains the title, the distance and the image of the AR Entity. By tapping on each billboard, an information page is shown as in Figure 2.11. The user can read the description of the AR Entity as it was defined in the back-end. Then there are two options, namely a) to press "Continue in Web page" where the link defined in the back-end is opened in a new web browser window in order to see furthermore details; or b) to press "More in official app" button where the user can download the custom made application for mobile phones. Additionally to the AR environment the user has the option to view the entities in a map or list view as shown in Figure 2.12.



An improvement of the lighting of the Town Square is being studied by the Local Council, who would like to...

Sobipro

Figure 2.10: AR environment for Junaio, Layar and Wikitude.


Town Square Improvement

Town Square Improvement

493 ft

An improvement of the lighting of the Town Square is being studied by the Local Council, who would like to know the opinion of the people of Gordexola.

Continue in web page

More in official app!

Restoration of Old Pedestrian trail

The old trail through Gordexola will be restored. How this shall be done is being studied for the maximum benefit of the people.

Distance: 149 m

More in Web page

More in official app

Figure 2.11: Information page for Junaio and Wikitude. Layar does not have one.

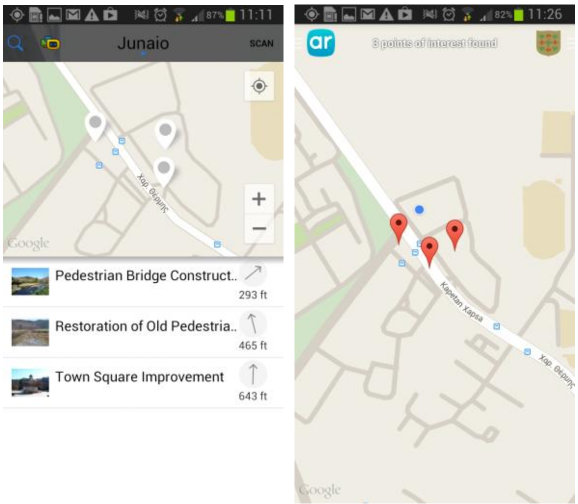


Figure 2.12: Map environment for Junaio and Layar. Wikitude does not have one.

2.4.5 Conclusions

FastAR is a module that allows for generating location based AR channels with as minimum effort as possible. Already existing web portals can exploit this tool and increase their visibility by making their content available through widely-used AR browsers. Although, in its current version, FastAR is limited by the requirement of having the original site hosted in a Joomla-SobiPro installation, we believe that the potential of exploiting the consistency of the underlying data structure in order to automate the process of turning a simple website into an augmented reality experience is a highly promising direction for strengthening the future impact of augmented reality technologies.

2.5 eGovernance Augmentation Layers

The solution for mobile augmented reality that has been developed in the context of WP3 has been employed for the generation of "two" eGovernance augmented layers. More specifically, the developed layers have been used to facilitate the field trials of Urban Planning and Mobility. In the following, we provide technical details about the generation and use of these layers.

2.5.1 Urban planning

The use of augmented reality coupled with the use of 3D objects was envisaged as one of the main enabling technologies in Urban Planning. Indeed, in our effort to use intriguing means to engage citizens in a two-way dialogue with their government, the potential to augment their reality with the 3D representations of future urban plans was considered highly promising. Nevertheless, in ensuring the necessary level of user experience we had to solve a number of technical challenges dealing with the smooth collaboration between the technologies of 3D modelling and augmented reality. In this section, we discuss the approaches that have been adopted.

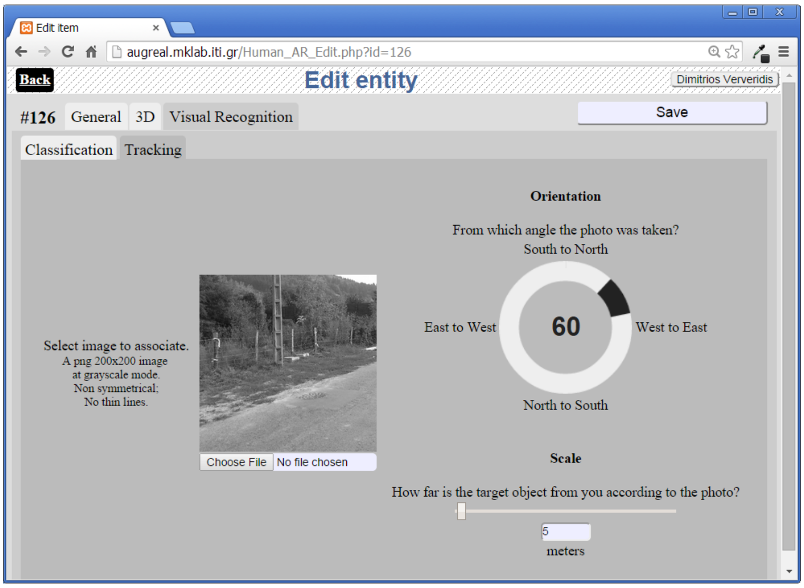
The first technical challenge that we had to face was the construction of a 3D model and a format that should follow specific requirements, in order to be compatible with the specifications of the AR vendors. Aiming for maximum flexibility we have decided to rely on open source tools for designing, modifying and exporting the 3D models. More specifically, the Blender® 3D design tool[8] was used in order to design and modify the 3D models that were used during the field trial. Blender is considered one of the most sophisticated open source tools for 3D design, supporting several features such as importing Autocad/Sketchup models, changing shapes/size/color/material, editing the vertices of the 3D models, combining several 3D models into one, etc. What was particular useful in our case was the ability of this tool to import 3D models from open or commercial sites and to combine them into a single scene. Given that the urban plans that were used during the second field trial (see D5.4 [4]) required the combination of many different pieces of gymnastic equipment, their combination into a single scene was absolutely necessary for achieving a realistic representation of the future plans.

However, apart from combining all different 3D objects into a single scene we had also to deal with the issue of rendering oversized models in augmented reality. Indeed, according to the AR vendors the 3D models should not be too detailed. The first reason is the great bandwidth needed in order to download a 3D model. The second reason is that mobile devices do not have the hardware resources needed to render complicated models. Based on our experience, the 3D model of an urban plan should not exceed the size of 1 MB. Otherwise the mobile device is gradually overheated and frames delays may happen. In order to make complicated 3D models simpler, we have used the *decimation modifier* tool of Blender. This tool can reduce the number of vertices of a 3D model by a given parameter, e.g. 0.5 for 50% reduction. Sometimes, however, decimation is not possible because the models are already parsimonious. In this case manual modifications-deletions are needed.

[8] http://blender.org

The model format that was followed for exporting 3D models was the Wavefront® Object format or OBJ. OBJ is an open source format which is supported from a wide range of 3D design tools and is also compatible with the some of the most important AR Vendors. More specifically, the OBJ format is fully supported by Junaio and MetaioSDK. In Layar, it is supported by using a command based conversion tool from obj to l3d (Layar format) that was included in our setup. Wikitude supports only w3d (wikitude) format, but it does not provide any automatic tool for converting obj to w3d, and therefore in Wikitude only billboards are available without 3D content. The OBJ format consists of a file containing the coordinates of the 3D model, a file for the materials, and of several images that are the textures of the 3D model. The drawback of this format is that it is not amenable to compression resulting in oversized 3D models. In order to overcome this drawback, our AR server was configured to compress the files of the 3D models in a single zip file before transmission, and thus significantly reducing the required bandwidth. Moreover, the mobile app that was developed for the purposes of the Urban Planning field trial employs a caching system that prevents downloading the same file twice, and in this way significantly improving the experience of the end-user.

As already described in D3.1 [1] the AR server features a web portal that allows for uploading, inspecting and configuring the 3D models that are served by the system. However, since the 3D models used during the second field trial needed to offer a much more fine-grained experience, we had to make certain enhancements. More specifically, in image-based recognition apart from the object itself information about its scale and orientation are also crucial. Since the 3D model is overlaid on a photo, the distance from the target and the angle from where the photo is taken are important. In order to cope with these two problems we have added a "scale" and a "rotation" angle parameter in the web interface of the AR server. These parameters allow the operator to modify the model's scale and orientation according to the photo shooting distance and angle. For example, in Figure 2.13 it is seen that the photo was taken with 60 degrees from South to West and with a distance of 5 meters from the origin of the urban plan.



augreal.mklab.iti.gr/Human_AR_Edit.php?id=126

Select image to associate.
A png 200x200 image
at grayscale mode.
Non symmetrical;
No thin lines.

Orientation

From which angle the photo was taken?

South to North

East to West

60

West to East

North to South

Scale

How far is the target object from you according to the photo?

5

meters

Figure 2.13: Passing photo shooting parameters to AR web portal so as to configure image based AR and to present the 3D model correctly.

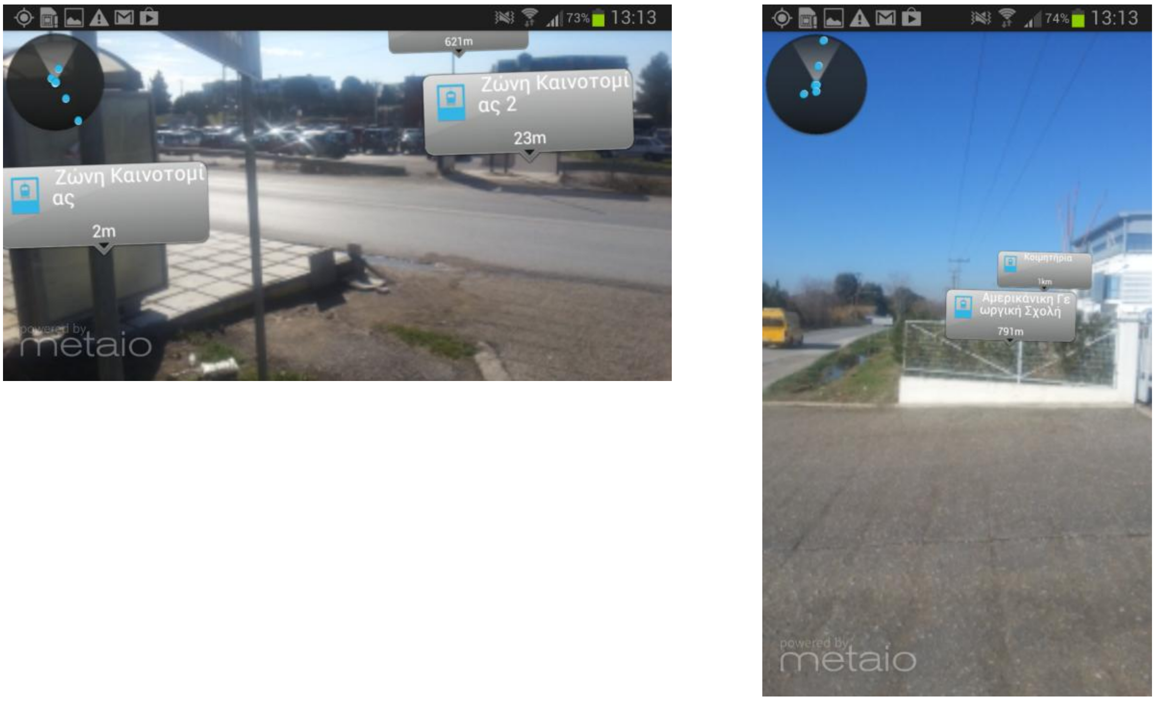
2.5.2 Mobility

Augmented Reality is thought, among other uses, as a way to minimize the effort required by the user to access relevant information. Motivated by this intriguing concept we have decided to use augmented reality as an enabling technology for presenting information related to mobility. More specifically, in the mobile app that was used during the 2nd field trial of the Mobility use case, we have decided to offer an AR-based functionality that would allow the user to quickly access information about his nearest public transit stops, as well as information about the service lines serving these stops. In the following we provide details about the nature of this functionality and motivate its usefulness.

The first part of the AR-based functionality consists in visualizing, in an augmented view, the transit stops located near the user by a given radius (e.g. 1500 meters), as shown in Figure 2.14. In order to understand the usefulness of such functionality, consider the case where the user is searching for the appropriate bus stop to take him to his destination but there is more than one stop around him, or there is no stop within his viewing range. This is typical the case for city visitors (e.g. tourists) that have a rough idea about the city’s public transportation network but are not sure about the exact location of every bus or tram stop. The aforementioned functionality will help a city visitor to quickly orient himself by turning his camera towards a certain direction and view the bus (or tram) stops lying in this direction, along with their distance from his current location (i.e. using an AR billboard). In this way, the user is offered a much more intriguing and convenient way to obtain this type of information compared to the typical map-based interfaces.

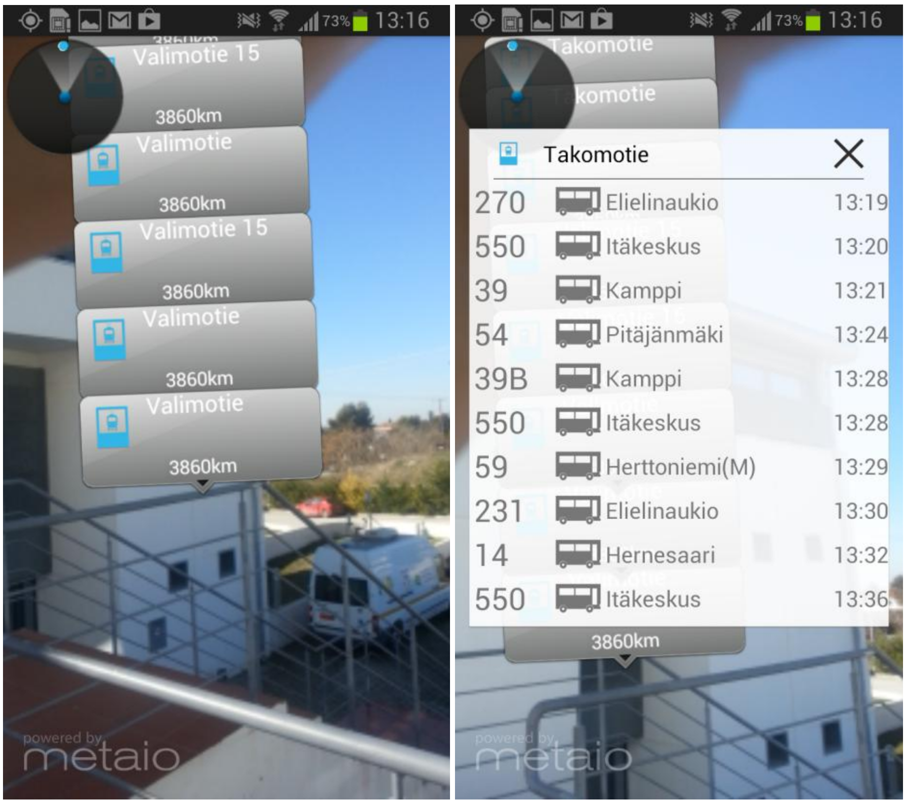
Moreover, given the fact that the stops are presented in the AR view, relevant information can be accessed rather quickly by tapping on the AR billboard. This is actually the enabler for the second part of our AR-based functionality presented in Figure 2.15. More specifically, upon clicking the AR billboard the user is presented with timetable information including the numbers of the service lines crossing this stop, as well as the “time-to-arrival” information of the forthcoming busses (or trams). In this way, the user of the app can make good use of his time that would otherwise spent on the stop waiting for his bus or tram to arrive.

Finally, it is important to mention that one of most critical factors enabling the aforementioned functionality is the availability of open data by the HSL (Helsinki Regional Transport). Information about the location of stops in Helsinki, as well as timetable information about the service lines crossing each stop is accessible through an easy to use API that has been employed in our mobile app. Actually, the use of augmented reality together with HSL data in the context of the Mobility use case has been an excellent example of using external data sources to facilitate novel services.



(a): Two bus stops side by side **(b):** Two distant stops

Figure 2.14: Example screenshots demonstrating the use of AR in the mobility use case



Takomotie

270	Elielinaukio	13:19
550	Itäkeskus	13:20
39	Kamppi	13:21
54	Pitäjänmäki	13:24
39B	Kamppi	13:28
550	Itäkeskus	13:28
59	Herttoniemi(M)	13:29
231	Elielinaukio	13:30
14	Hernesaari	13:32
550	Itäkeskus	13:36

Figure 2.15: Timetable information that appears when tapping on the "Takomotie" billboard

3. Personalized content delivery

3.1 Motivation

One of the most interesting feedback points that we have received during the 1st field trial of the Mobility use case, was that the usefulness of the application would significantly increase if it could become more familiar with the commuting habits of the user and more personalized in the provided suggestions. For instance, it was mentioned that for a commuting suggestion to be valuable it should be communicated to the user before he reaches the point when he is actually stack in a traffic jam. In other words, the information should come to him before leaving his home or his office and not when he is already at the bus stop, which was the case for the prototype used during the 1st field trial. Thus, one of the strategic decisions for enhancing the functionalities of the mobility prototype was to design and implement the necessary algorithms and functionalities for enabling the provision of more personalized information. This was also in alignment with the scientific objective of WP3 aiming to build mechanisms for personalized content delivery.

3.2 Personalized traffic-related information

The first step in implementing a personalized content delivery mechanism is to create user profiles consisting of the criteria that will be used as arguments to discover when and what type of information to provide. In our case, and since we are dealing with the commuting habits of the user, the criteria are the most visited places and the usual departing/arriving times in these places. Our approach has been to automatically extract this information for each user without requesting any explicit input. Subsequently, using the aforementioned info in combination with information about the bus stop locations and timetables, as well as with disruption info about the good condition of the traffic network we have been able to offer personalized suggestions about the optimal commuting options. In the following we describe the approach that we have adopted to implement this functionality.

3.2.1 Extracting the commuting habits

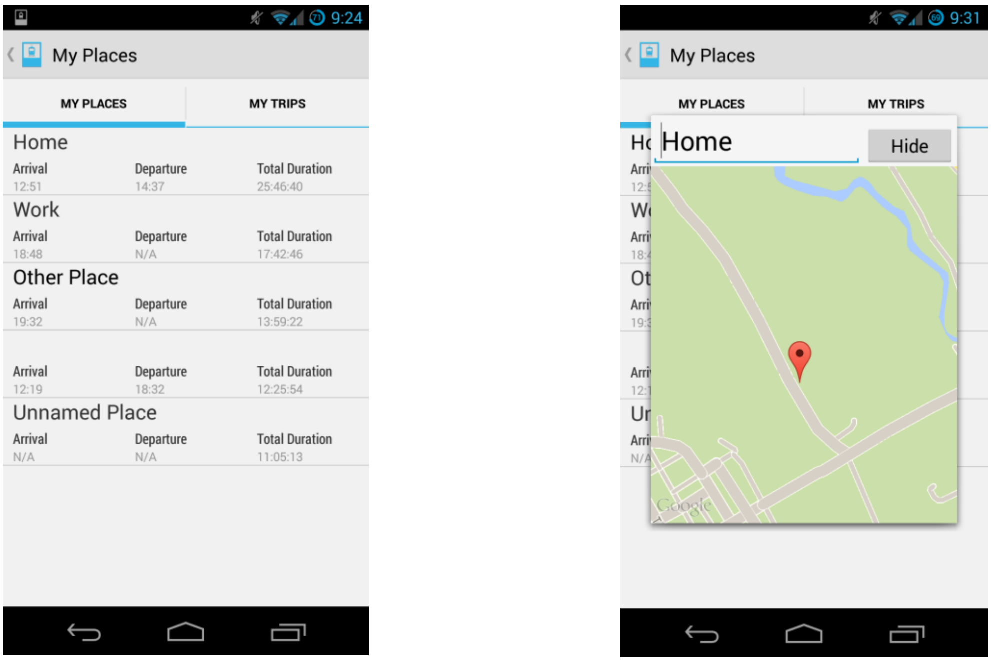
In order to identity the most visited places we have established a data collection service that is scheduled to run in the background and collect location points with a ***15 min*** interval. For each location point the fields *latitude, longitude, accuracy* and *timestamp* are stored in the phone's local database. The service listens to location updates passively so as not to consume any additional power. Subsequently, these points are clustered into visited places by grouping together the points that lie within a ***500m*** radius around the clusters centre.

Then, having identified the visited places we can spot the ones that are most visited. In detail, by using the timestamps associated with location points we can calculate how many times a user has visited a certain place. The timestamps are used to distinguish between legitimate and fake visits. More specifically, a threshold of ***2 hours*** has been selected to perform this distinction. So, if someone has left from a place and returned only after 2 hours we will count 2 visits for this place, otherwise we only count for one. This is to cater for cases where, for example, someone leaves his home or office for some time but he doesn't actually makes any use of the public transportation network. Finally, after counting all legitimate visits we can calculate the approximate time that the user has spent on the place and, based on this information, rank the identified clusters from the most visited place to the less visited one.

The next step is to calculate the arriving and departure time to/from a place. In order to do this we convert all arrival and departure times to “minutes since the start of the day”. Then, in order to find the arrival time, we consider only the timestamp of the earliest visit (out of the group of all visits assigned to a certain place) and we average these timestamps across days. Moreover, in order to cater for the case where the user does not always arrive in a place at a specific time, we calculate the standard deviation of the arrival times. If the standard deviation is found to be over a certain threshold we conclude that the arrival time cannot be estimated accurately for this place, otherwise we consider the averaged timestamp across days as an accurate estimation of the arrival time. Similar is the process for estimating the departure time for each visited place.

3.2.2 Building the user’s profile

Based on the aforementioned process we have managed to automatically extract the following criteria about the user’s profile: a) Coordinates of the places that the user is visiting, b) Time spent on each place, c) Arrival time to each place, and d) departure time from each place. A dedicated view was incorporated into the mobile application in order to visualize the results of this process Figure 3.1(a). Through this view the user has the option to name the detected places or hide them if a place has been falsely detected Figure 3.1 (b). By hiding a place the user will stop receiving notifications related place and is will also disappear from the “My Places” view. However, by pressing “Hide” the place is not totally deleted from the local database, because that would mean that it could appear again in the future. Only a flag is used in the database to indicate that the place should not be taken into account when sending notifications.



Home

Arrival	Departure	Total Duration
12:51	14:37	25:46:40

Work

Arrival	Departure	Total Duration
18:48	N/A	17:42:46

Other Place

Arrival	Departure	Total Duration
19:32	N/A	13:59:22

Arrival	Departure	Total Duration
12:19	18:32	12:25:54

Unnamed Place

Arrival	Departure	Total Duration
N/A	N/A	11:05:13

(a) Automatically detected places

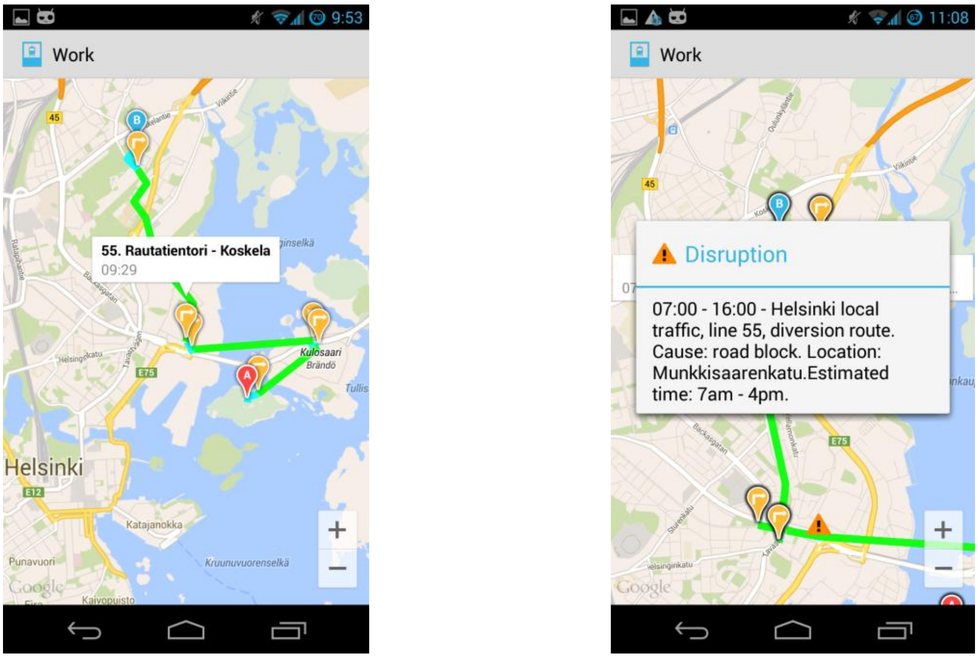
(b) Visualization of a certain place

Figure 3.1: Interface of the mobile app that visualizes information about the citizen’s commuting habits

3.2.3 Generating personalized suggestions

Using these criteria we can offer personalized commuting suggestions by making use of the HSL route planning API[9]. This API receives as parameters two pairs of coordinates (current location and destination) as well as a desired arrival time. The output returned is the suggested route. Given that we have information about the user's upcoming trip (i.e. start point and destination), we are able to call this API with the necessary parameters and obtain information about the public transportation means (i.e. bus or tram lines) that the user is likely to use. Having acquired the bus or tram lines we are also able to call the disruption info API[10] so as to determine if any of the transit lines included in the suggested routes have any disruptions. In this way, we are able to warn the user about potential delays in his upcoming travel.

The suggested route is visualized on the map with a polyline. The "Walking" parts of the route are coloured cyan and the public transit parts are coloured green. A marker is placed at the points where the user changes transportation method. When a user taps on a marker, if the transportation method is walking, then "Walking" is displayed as a title and the arrival time is displayed as a subtitle. If the transportation method is a public transit line then the line number, the departure point and the terminal point is displayed as a title, whereas the departure time from the station according to the timetables is displayed as a subtitle (see Figure 3.2(a)). When applicable, a disruption info marker is also placed in between the stops of the affected line and shows the disruption message if tapped (see Figure 3.2 (b)).



07:00 - 16:00 - Helsinki local traffic, line 55, diversion route. Cause: road block. Location: Munkkisaarenkatu.Estimated time: 7am - 4pm.

(a) Route suggestion

(b) Disruption details

Figure 3.2: Notifications generated by the personalized content delivery mechanism based on the citizen's commuting habits

[9] https://www.hsl.fi/en/information/how-use-public-transport/planning-journey

[10] http://www.reittiopas.fi/en/disruptions.php

These suggestions are presented to the user via the notification system of his mobile phone, without requiring from the user to have the app running. For example, if the app has detected that a user usually goes to work at 11:00 and has also detected the coordinates of his work, it sends him a notification at 9:00 with the message “Trip Update: Departure for Work (11:00)”. If the notification is tapped then the detailed map view will be presented to the user. This will help the user to know: a) what time he should depart or be on the bus stop in order to arrive on time, and b) if there are any disruptions that will affect his journey.

3.3 Privacy

A major concern in implementing this personalized functionality is respecting the user’s privacy. The system is tracking the user’s whereabouts throughout the whole day. Other applications such as Google Now store the user’s location data on the cloud. The data are also linked with the user’s google account, which means that the user can be identified easily. The Mobility app on the other hand does not communicate the data anywhere on the internet. They are exclusively stored and processed in the user’s device. By storing the data on the device the user has the freedom to clear all data through its settings menu, or by uninstalling the app. The data process as well, is performed only when the device is connected to the charger, in order to avoid additional power consumption. The only drawback of this method, as opposed to storing the data on the cloud, is that the data is tied on the device and not to the user, meaning that if a user switch devices from a mobile phone to a tablet he would not have the same experience. We have decided to adopt this approach given that we regard the user’s privacy as more important than tolerating with this minor drawback.

3.4 Related apps and how Live+Gov goes beyond

A number of related apps are already on the market offerning personalized traffic information to their users. In the following, we briefly describe some of the most popular apps belonging to this category and compare them against the mobile app developed for the Mobility field trial.

Waze: Waze is mostly considered as a GPS navigation app, but it has many features that enhance the navigation experience. Using crowdsourcing data from its own users it can provide real time traffic information or other events that affects a user’s mobility, such as accidents, road closures, etc. Each user of the app sends anonymous data like driving speed and position that are used to calculate traffic load for the area. In order to provide reliable information, a significant amount of users are required. To encourage more users contributing to crowdsourced data, waze uses gaming conventions such as earning points and getting rewards depending on how much they drive using the app.

Google Now: Google Now is a personal assistant smartphone application that predicts what you need at a given time and displays information cards such as sports, parking location, traffic, calendar reminders and many others. Although not entirely dedicated to transportation, it has some interesting features that are worth noticing. The application can detect your commuting habbits and timely present information cards containing traffic data, routing information and time required to reach a destination. It also has the ability to automatically detect the transportation means you usually prefer (bike, vehicle or public transit) and adapt these cards accordingly. So, in the case of public transit for transportation, it presents timetables of the nearest stop. By taking advantage of Google's app ecosystem it can also use information retrieved from other apps such as calendar events or recently searched places on Google Maps to enhance the overall experience.

Moovit: Moovit is an application for Android, iOS and Windows Phone that specializes on public transit transportation. Users can access a live map and view nearby stops and stations based on their current GPS location, as well as plan trips across transportation modes based on real-time data. The application differs from traditional public transit information as it is a community-driven application that integrates static transit data from public operators with real-time data collected from users via crowdsourcing. Moovit provides disruption information if supported by the specific city authorities. The users can also store their favourite places in order to access quickly routing information.

INRIX Traffic: INRIX Traffic is an application developed by INRIX, a company based in the US that specializes on road traffic and driver services. INRIX collects trillions of bytes of information about roadway speeds from over 175 million real-time anonymous mobile phones, connected cars, trucks, delivery vans, and other fleet vehicles equipped with GPS locator devices. Their mobile application shows real time or predictive traffic information and can also provide alerts such as traffic congestions, road closures, construction and other events. The application also allows the users to define their home and work locations in order to receive personalised alerts about their daily commute.

It is evident from the aforementioned descriptions that the Mobility app shares some common features with the applications that are already available in the market. For instance, the ability to obtain routing information, as well as to define the favorite places for receiving personalized alerts is supported by all apps. However, it is only the Mobility App and Google Now that are able to detect these places automatically. Similarly, the provision of authority alers seems to be a common feature among the examined apps, but it is only the Mobility app and Waze that are able to receive and distribute user-provided alerts next to the authority alerts. Moreover, we can see that more advanced functionalities like human activity recognition, or the ability to push personalized notifications based on the user's commuting habbits, combined with traffic-related information are only supported by Google Now and the Mobility app. Finally, the integration of augmented reality as the basic mean to communicate information about the nearby busstops together with the timetables of the forthcoming busses, is currently the feature that makes our application unique among the existing competitors. Table 3.1 , provides an overview of our comparison.

Table 3.1: Comperative evaluation of the Mobility app against similar apps.

	Mobility	**Waze**	**Google Now**	**Moovit**	**INRIX Traffic**
User alerts	+	+	-	-	+
Authority Alerts	+	-	+	+	+
Favourite Places	+	+	+	+	+
Automatic detection of favourite places	+	-	+	-	-
Routing Information	+ (HSL)	+	+	+	+

Notifications	+	-	+	-	-
Activity Recognition	+	-	+	-	-
Nearby stops and timetable information through Augmented reality	+	-	-	-	-

4. Visualization tools for data comprehension

4.1 Introduction

Visual Analytics (VA) is a fledgling scientific discipline that combines techniques and methods from the research areas of Human Computer Interaction (HCI), visualization, and automatic analysis techniques, including statistical methods, machine learning, data mining, knowledge discovery, and similar approaches. The need for VA is motivated by the vast amount of today's available digital information, which cannot be processed fully automatically, if the problems are not defined a priori, the scenarios are too complex, or require human knowledge to solve them [20]. The goal of visualization is to aid our understanding of data by leveraging the human visual system's highly-tuned ability to see patterns, spot trends, and identify outliers. Well-designed visual representations can replace cognitive calculations with simple perceptual inferences and improve comprehension, memory, and decision making. The power of visualisation is to help the observer use his perceptual system in order to free up our cognition for higher-level tasks [10] (i.e. assist thinking [14]). This is to make it possible for analysts of data to obtain internal mental models of the information content in datasets; models which subsequently can be used for characterization, prediction, and/or decision making [16]. More generally speaking, the purpose of visualisation is threefold [18]:

1. **Exploration:** the search for hypotheses that consists in: a) Exploring large amounts of data, b) Finding interesting relations, and c) Building hypotheses
2. **Analysis:** the confirmation or rejection of hypotheses that consists in: a) Analysing the identified hypotheses, b) Extracting patterns, and c) Detecting tendencies
3. **Presentation:** the presentation of facts that are fixed a priori, which consists in: a) Presenting the analysis of results to others, and b) Presenting information in an easy-to-understand way.

Moreover, by making data more accessible and appealing, visual representations may also help engage more diverse audiences in exploration and analysis. However, one of the challenges that we need to face is to create effective and engaging visualizations that are appropriate to the data. The data can be of different type [19], such as: i) Bivariate data, ii) Multivariate data, iii) Timedependent data, iv) Networks & Hierarchies, v) Geo-located data, vi) Other types (e.g.: text, audio, video, images). Thus, In order to create a visualization one must determine which questions to ask, identify the appropriate data, and select effective visual encodings to map data values to graphical features such as position, size, shape, and color. The challenge is that for any given data set the number of visual encodings — and thus the space of possible visualization designs — is extremely large.

Apart from selecting the appropriate encoding, there are three additional aspects that must be taken into account, namely: **excellence of the employed graphics**, **analytical capacity** and **anatomy of the design patterns**.

With respect to **graphics**, visualisation of data is not about making the fanciest visualisations. In many cases, infographics degenerate to mere eye-candy visualizations, taking into account all the new possibilities provided by modern computer graphics – colourful and fancy – but, however, pretty meaningless [11]. Edward Tufte refers to this as *chartjunk*. He argues that graphical excellence comes from "clarity, precision and efficiency" and it should focus on graphical decisions, not on fashionable display techniques [16]. This does not mean that visualisations cannot be fancy. A powerful visualisation is right on the thin borderline between functional and graphics excellence [12]. Tufte describes five principles for the excellence of graphics [15]: a) Well-designed presentation of interesting data – a matter of *substance,* of *statics,* and of *design,* b) Complex ideas communicated with clarity, precision, and efficiency, c) Gives to the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space, d) is nearly always a multivariate, and e) Requires telling the truth about the data.

With respect to the **analytical capacity**, Tufte identifies the following six principles for the effective analysis of the data: a) Show **comparisons**, contrasts, differences, b) Show **causality**, mechanism, explanation and systematic structure, c) Show **multivariate data**; that is, show more than one variables simultaneously, d) Completely **integrate** words, numbers, images and diagrams, e) Thoroughly describe the **evidence**, and f) maintain the data **quality**, **relevance** and **integrity**.

Finally the **design patterns** for interactive information visualisation can be contextualised in different ways: design patterns, behaviour patterns and interaction patterns. **Display patterns** are mainly about the visualisation of the data itself. **Behaviour patterns** deal with the dynamic characteristics of interactive infographics. They are about the functionalities the dynamic infographic gives you, like navigating a map, rearranging a graphic, etc. **Interaction patterns** are about the interactive part of infographics: essential interface elements that let the user give input to an information system.

A single design pattern consists of four elements: a) description - states the purpose of the pattern; b) required data - information about the data basis that is needed; c) usage - describes the steps the designer has to perform to apply the pattern, d) rationale - summary of the essential characteristics and argumentation for its usage – why instead of how.

Display patterns are mainly about the visualisation of data itself. Behrens divides them into eight categories [11]:

- Correlations, to identify patterns and other structures in observed phenomena, experimental samples or other statistical data. These are often hard to conclude from a table.
- Continuous quantities, to show some kind of causal relationship between data along a (regular) interval line. Although the data set is the result of data samples taken along the line, they can be treated as a continuous occurrence.
- Discrete quantities, to display sets of discrete data (enclosed and countable items of quantitative data). This is in contrast with the data represented at a continuous interval.
- Proportions, to show the share of certain data to the total quantity.
- Flows, to show the sequential nature of data. Here, it is not about the condition's static totality, but about how this condition has evolved.
- Hierarchies, to show connections in hierarchical relationships.
- Networks, for the display of interconnected entities.
- Spatial configurations, maps and space-related representations.

Behaviour patterns deal with interactive infographics and the way one can navigate through them. There are five basic functionalities:

- Navigation, to identify "where" within the data structure the user currently is.
- Filtering, to select order criteria in the display of the data.
- Arrangement, to examine data from different perspectives.
- Exploration, functionalities to explore the data set.
- Transition, functionalities to actually move the visual representation itself instead of user's eyes.

Interaction patterns, within interactive infographics, are able to give input to the system, for example to select which data is shown and which data is not shown. There are three main categories:

- Boolean selection, where a user selects an item or he does not select an item.
- Linear adjustment of data mapped over a continuous interval, for exploring large interval-based data sets.
- Spatial navigation, to interact with elements in their spatial context.

In the remaining of this section, our goal is to provide an overview some of the most typical visualization methods, placing emphasis on the methods that are considered more relevant for Live+Gov, i.e. map-based, image-based and graph-based visualizations. This is complementary to the study of visualization methods performed in D2.3 [5], since our interest here is not so much on reviewing the available methods for implementing the basic principles of visual representations, but rather to investigate our options from a practical and technical perspective.

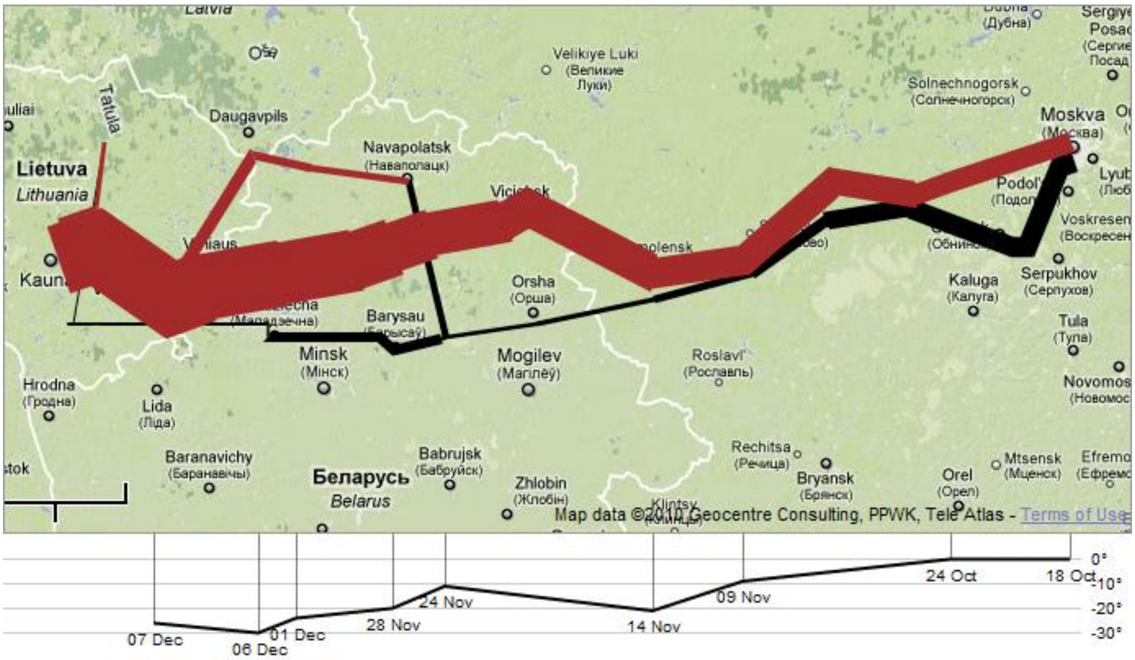
4.2 Map based visualisations

Maps are one of the most appealing approaches to visualize geo-located data, investigate their spatial distribution and detect patterns of trends. Indeed, geo-representations are a very important tool for visualizing data and become even more important in the context of Live+Gov, since the mobile-sensed data are by default geo-enabled. In this sub-section, we briefly discuss the different option for map-based visualization, we mention the available map-based components (both web and mobile) by taking an excerpt from D4.1 [2] and we consider performance indicators.

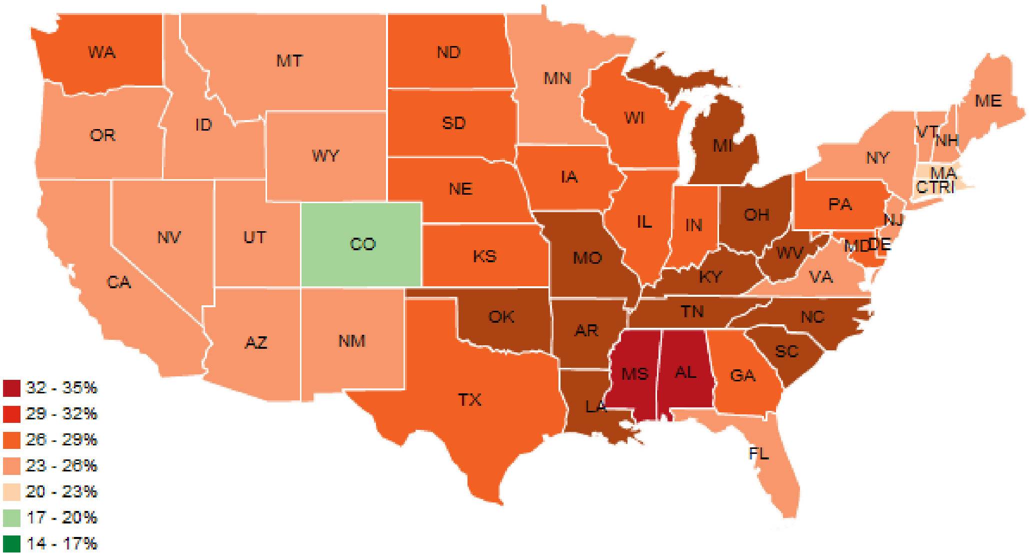
4.2.1 Types of map-based visualizations

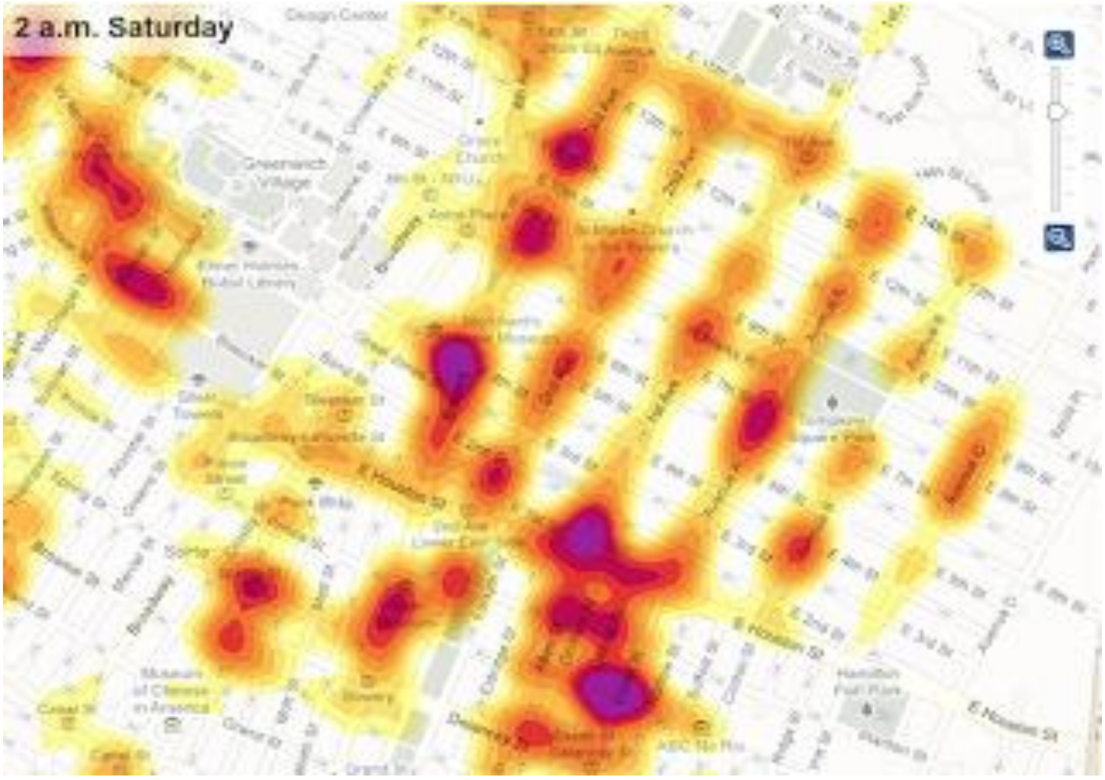
Many maps are based upon a cartographic projection: a mathematical function that maps the 3D geometry of the Earth to a 2D image. Other maps knowingly distort or abstract geographic features to tell a richer story or highlight specific data.

Flow Maps: By placing stroked lines on top of a geographic map, a flow map can depict the movement of a quantity in space and (implicitly) in time. Flow lines typically encode a large amount of multivariate information: path points, direction, line thickness, and color can all be used to present dimensions of information to the viewer. Many of the greatest flow maps also involve subtle uses of distortion, as geography is bended to accommodate or highlight flows.

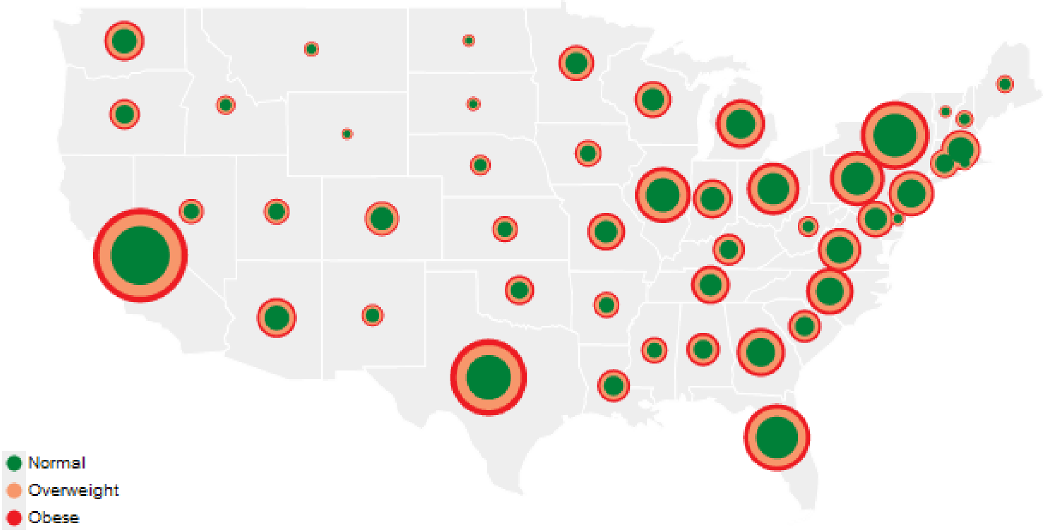


<u>Choropleth Maps:</u> Data is often collected and aggregated by geographical areas such as states. A standard approach to communicating this data is to use a color encoding of the geographic area, resulting in a *choropleth map*. In the figure below, a color encoding is used to communicate the prevalence of obesity in each U.S. state. A specialization of the Choropleth maps can be considered the **heatmaps** where individual numerical values are represented as colors overlaid on a map. Recently, the traffic heatmaps has become a very popular visualization tool for providing an overview of the traffic congestion in a city. However, heatmaps can be used to offer insights on various types of geo-located data such as air or sound pollution measurements, tweets or facebook posts, etc





<u>Graduated Symbol Maps:</u> An alternative to choropleth maps is the graduated symbol map, which instead places symbols over an underlying map. This approach avoids confounding geographic area with data values and allows for more dimensions to be visualized (e.g., symbol size, shape, and color). In addition to simple shapes like circles, graduated symbol maps may use more complicated glyphs such as pie charts. In the figure below, the total circle size represents a state's population, and each ring indicates the proportion of people with a specific Body Mass Index rating.



4.2.2 Map visualizations on the web

As part of D4.1 [2], research has been performed on the available map providers with the aim to assess their pros and cons. More specifically four map components, namely Leaflet, MapBoxJS, OpenLayers and Google Maps API, have been evaluated on the basis of certain criteria including the required mapping functions, data formats, features, look and feel, maps controls, search capabilities and community.

Leaflet (http://leafletjs.com/) is a community-driven project that is open-source and suitable for mobile usage. It is used by websites like Flickr[11], Foursquare[12] and Craigslist[13]. The user experience is very good, and the API for the developer is clean and simple.

Mapbox (https://www.mapbox.com/) is a tile service provider, but also delivers a JavaScript geo-component. It provides well-designed maps, which are also available for other JavaScript components (like leaflet). Advantages of this component are its speed and attractive user-interface.

OpenLayers (http://openlayers.org/) is a mature JavaScript Maps API. It’s large in size and provides many features, but looks a bit out-dated. Although the use of this component is not really recommended by the respective community, we have decided to include it in this comparison due to it long-established use and the fact that it provides the most features.

Google Maps (https://maps.google.com/) API is used by many companies, organisations and people. It is a feature-rich mapping library, which has nice extras, i.e. street-view, satellite view, etc, that are very favourable for research purposes. However, the license does not allow free usage for more than 2500 requests per day.

In search for a suitable geo solution that meets the established criteria (future-oriented, good community, well documented and good look & feel/performance), Leaflet was deemed as the most appropriate choice (see D4.1 [2]) for becoming an integral part of the Live+Gov toolkit. This was further advocated by the fact that since our initial research Leaflet has undergone several improvements, with the most important being the improved support for mobile devices which improved the reliability on different web browsers. On the other hand, the Google Maps API feature-set and community are very rich, supporting a number of innovative services. Thus, it was decided to employ the Google Maps API as part of our research prototypes due to its potential in becoming a facilitator of our research goals.

Finally, it is important to note that in order to ensure the best user experience when using the map-based visualizations; we have thoroughly investigated the use of certain optimizations. In this investigation we have examined the different options to optimize the loading performance, the working performance, as well as the pros and cons for each of the available solutions. A detailed description of our conclusions, as well as the solutions that have been adopted in our implementations are available in Appendix A.

[11] http://www.flickr.com/

[12] https://foursquare.com/

[13] www.craigslist.org

4.2.3 Map visualizations on mobile devices

Given that apart from the web-based visualizations, the field trials envisaged in Live+Gov required the development of mobile apps featured with a map-based module, thorough investigation has been also conducted at the level of map-based components for mobile devices. During this investigation the following aspects were considered: a) Licence: are we allowed to use the library in our application? b) Flexibility/ Usability: How easy is the library to use in an application? Are there any limitations in using the library? Is it possible to make changes in the library (if needed) before we can use it?, and c) online/offline mapdata: what type of mapdata are used and how are they handled in an online and offline mode.

In the context of this investigation we have examined different known map data source for mobile devices, namely Bing maps, Google Maps, Osmdroid and Mapsforge. The results of this investigation are presented in detailed in Appendix B.

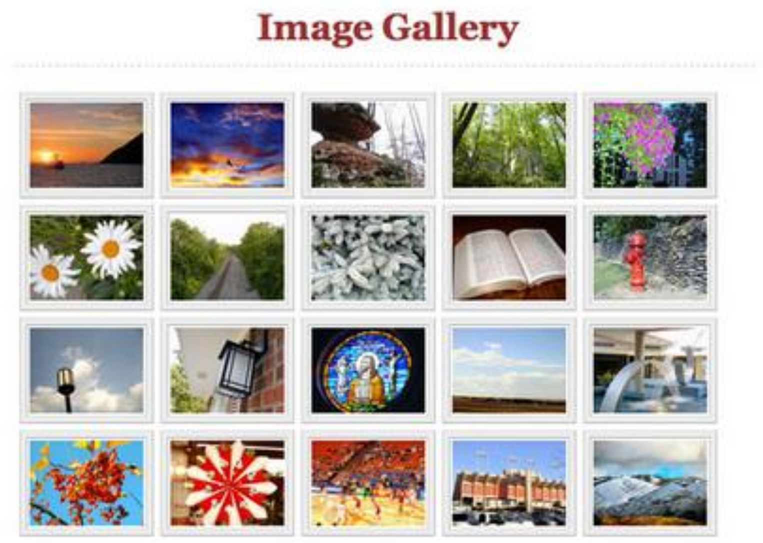
4.3 Image-based visualisations

By image-based visualizations we refer to the visualization means that are used to facilitate the extraction of insights by displaying many images simultaneously. The spatial distribution of the images, their arrangements along a timeline, or even their clustering based on visual charactertistics constitute some examples of image-based visualizations. Of course in order for the image-based visualizations to become effective the underlying dataset should incorporate images associated with auxiliary metadata information, allowing their filtering or their spatio-temporal arrangement. The great benefit of this type of visualizations is their ability to provide a quick and comprehensive glimpse of the available content, the resolution of ambiguites (i.e. textual), as well as the fact that they can be a very powerful tool in discovering the stories that are hidden in the data. In this direction, one of the visualization tools that has been extensively used in Live+Gov is the gallery-view. In the remaining of this section we will discuss the optimizations that have been undertaken to offer a nice user experience to the user of the implemented gallery view.

4.3.1 Types of image-based visualizations

Image-based visualisations are less straightforward than map-based visualisations, given that an image is in itself a visualisation. Therefore, creating a visualisation with multiple images can become quite complex. In this section, we present two types: the well-know and broadly used gallery view, and a pivot-viewer technique, where images are shown in a pivot table.

Gallery view: The gallery view is well-known and broadly used when visualising multiple images. The main functionality is to browse through a set of images, by showing multiple thumbnails on the screen. This makes visual comparison among pictures possible. Combined with filtering and sorting options, this can easily be used to discover patterns and stories within the images. However, a visual judgment by the viewer is always necessary to interpret the images.



<u>PivotViewer:</u> In order to visualize images in large datasets like BuitenBeter, a Picture Pivot Table technique can be used. The Picture Pivot Table has been developed by Microsoft Live Labs. It is called 'PivotViewer' and is based on Microsoft Silverlight. An experiment with this technique has been conducted to estimate the usefulness of this technique. More specifically, two Microsoft Live Labs tools have been used: a) A MS Excel plugin that pre-processes the data to get it in a format (called CXML) that can be used in the PivotViewer, and b) a standalone PivotViewer application (Windows based) that can open a CXML file and show the data in a picture-pivot way[14].

The PivotViewer is based on a technique called DeepZome[15]. This technique is very promising when visualizing large image based datasets. Because of the easy and convenient way of playing around with the data, it encourages users to further interact with the data and in this way discover new relations among them (see Figure 4.1, Figure 4.2 and Figure 4.3).

The main drawback however, is that this tool has been developed in the Microsoft Live Labs but it has never reached maturity (i.e. it has been 'dropped' by the Live labs), meaning no (formal) support and limited availability to resources. This is a major risk when relying on this tool. This problem is already present for the Excel plugin (for preparing the data), which has become obsolete. That was the reason for selecting the gallery-based view to be the sole image-based visualization implemented by the analytic tools of Live+Gov (see Section 5).

[14] This tool can be downloaded at: http://research.microsoft.com/en-us/downloads/dd4a479f-92d6-496f-867d-666c87fbaada/default.aspx. Information about this tool was originally found (but unavailable since unknown): https://getsatisfaction.com/live_labs_pivot/topics/faq_what_to_know_about_pivot

[15] See: http://www.microsoft.com/silverlight/deep-zoom/

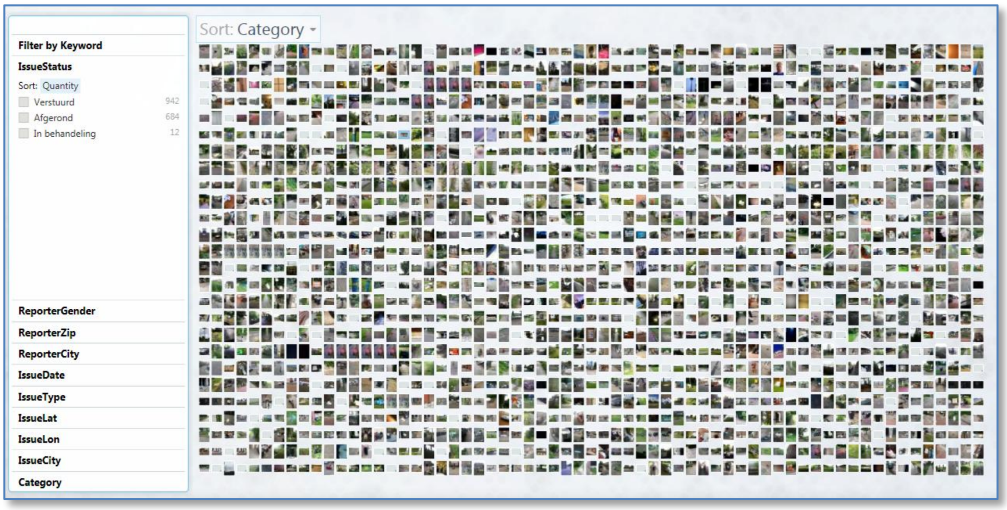


Figure 4.1: All images shown in one overview. Left part: annotations that can be filtered on

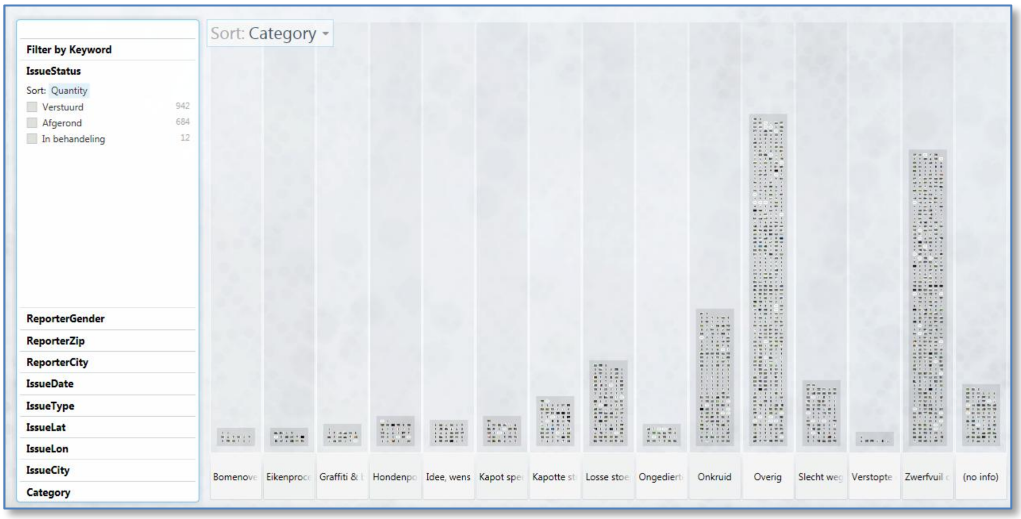


Figure 4.2: All images shown in a pivot-view. One issue-category per column

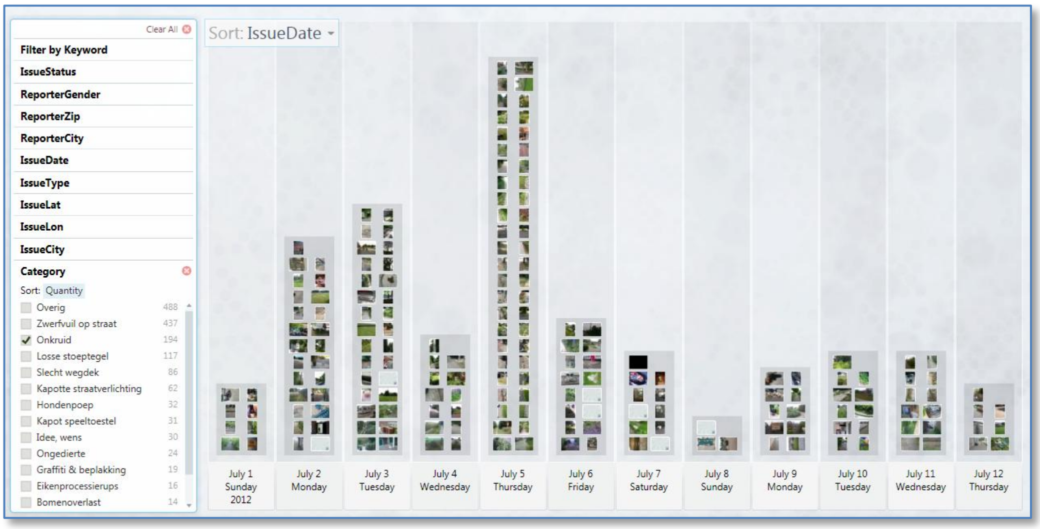


Figure 4.3: Issues filtered on Category ‘Onkruid’ and shown per date (IssueDate)

4.3.2 Performance considerations in gallery-views

Dealing with images is a rather sensitive issue from a technical perspective, especially when the volume of the displayed content becomes large scale. Since the gallery view was one of the main modules composing the Live+Gov solution for visual analytics, we have decided to look into this technical problem in more detail. The website jijmaaktutrecht.nl that was developed for the purposes of the urban maintenance use case, acted as the test basis for this investigation. More specifically, the undertaken tests were mainly focused on displaying how the performance of browsers in displaying a significant amount of photos influences the user experience. In this respect, a number of critical parameters were examined such as the latency in browser response, the memory usage, the utilization of the heap-size, the option of infinite scrolling, etc. In order to obtain a basis for comparison, popular sites like Pinterest, Google+, Instagram and Twitter were examined side-by-side with jijmaaktutrecht.nl. The results of this investigation are presented in detail in Appendix C.

4.4 Graph based visualisations

Another visualisation technique that has been used in Live+Gov to make sense of data is based on graphical representations. Graphs are widely used to make sense out of (statistical) data and complex information. The purpose of graphical representations is to "provide better insight into large sets of data and complicated structures that are otherwise not immediately accessible to a reader or spectator" [11]. The insights that can be determined from graphs may widely vary: from discovering developments in time to distributions, from discovering correlations to showing (relative) quantities, etc. In this section, we initially present some of the most well-known types of graph-based visualisations. Then, we summarize our investigation results on graph-based visualisations. The final section discusses graph visualisations on mobile devices. For both cases the results of our investigation are presented in Appendix D.

4.4.1 Types of graph-based visualizations

Mainly, graphs are a 2D representation where data are opposed on two axes and their intersection is drawn. Four main types of graph-based visualisations are represented in this section.

Scatter plot: In a scatter plot, correlations between two metric variates can be discovered. It visually describes a two-column table with pairs of variables that does not provide much meaningful information in the tabular form, especially when the underlying datasets become large. Each pair of variates is represented by a dot in the two-dimensional Cartesian coordinate system (see Figure 4.4). With a sufficient number of elements, it enables the viewer to identify certain development trends of the data and potentially even point to functional correlations between the observed variables. Also, exceptions from such functional rules become visible, like outliers [11].

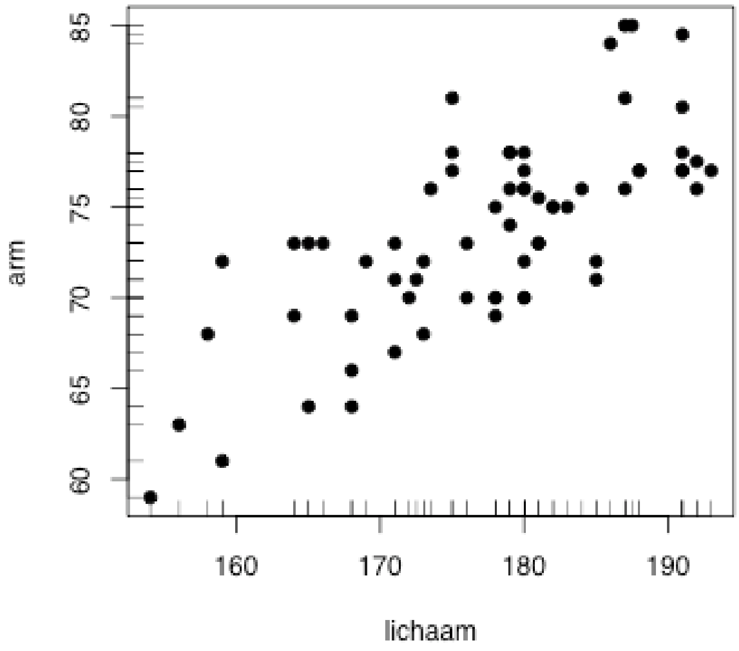


Figure 4.4: Example of a scatter plot: correlation between length of arm and body length.

Line chart: Line charts display the quantitative value of an observed object over a continuous interval (see Figure 4.5). In most cases, this interval is a time span, and the graph describes how the object's variable changes over this time interval. The line chart is widely used, and due to its familiar structure it is easy to grasp. Besides the individual values themselves, the most significant information that can be derived from a line chart is the gradient of the curve, which provides information about the intensity of the attribute's change over time. Also, minimum and maximum values can be easily identified from such a representation [11].

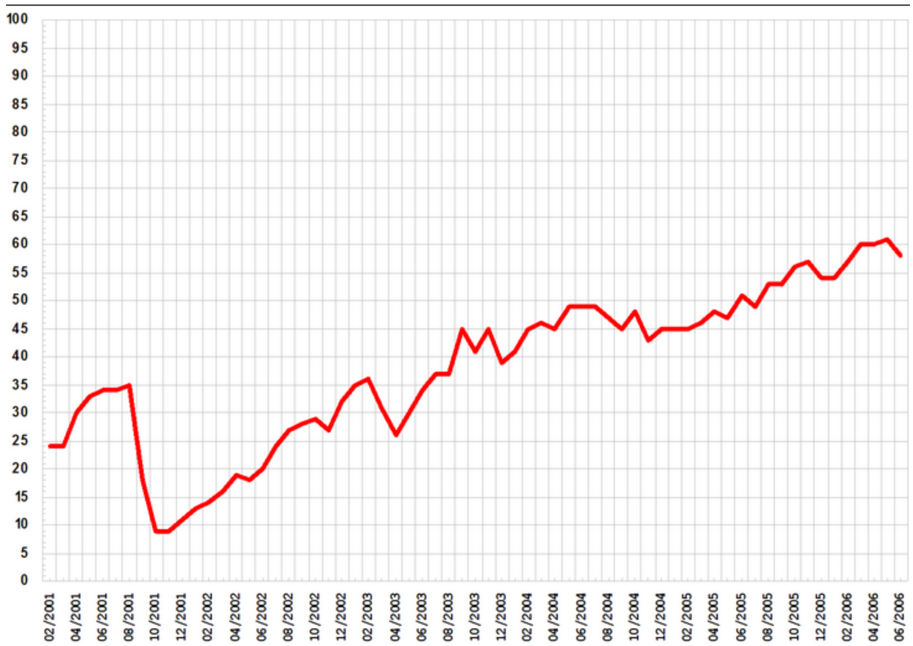


Figure 4.5: Example of a line chart.

Several variations on the simple line chart can be made, for example multiset line charts (displaying multiple lines in one chart area; see Figure 4.6) and stacked area charts (where the area between the y-axis and the line is filled; see Figure 4.7).

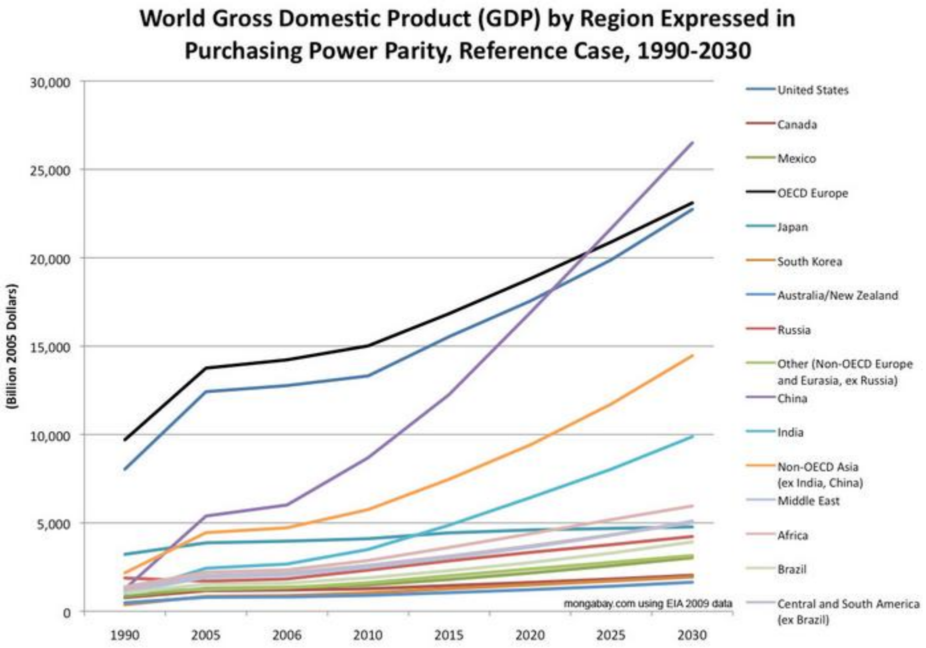


Figure 4.6: Example of a multiset line chart

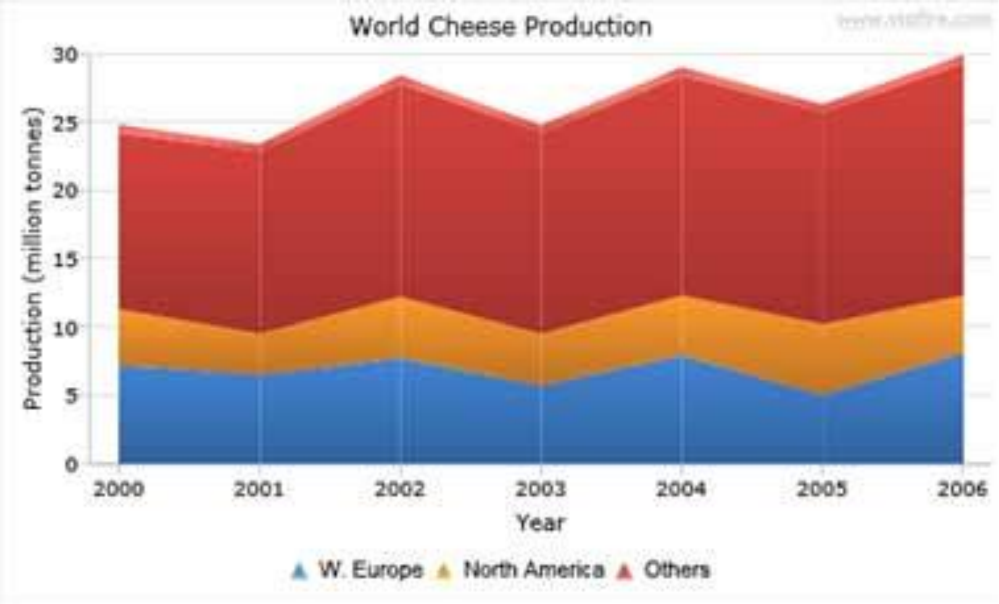


Figure 4.7: Example of a stacked area chart

Bar chart: Bar charts are used to visualise absolute magnitudes of nominal data items, i.e. discrete quantities (see Figure 4.8). They can theoretically consist of only a single data item, but in most cases are used to additionally compare the quantitative value of several entities with each other. Bar charts are distinguished from line charts and pie charts as they do not display continuous developments over an interval, but measure the values of discrete data items. Also, they display absolute numerical values rather than proportions [11].

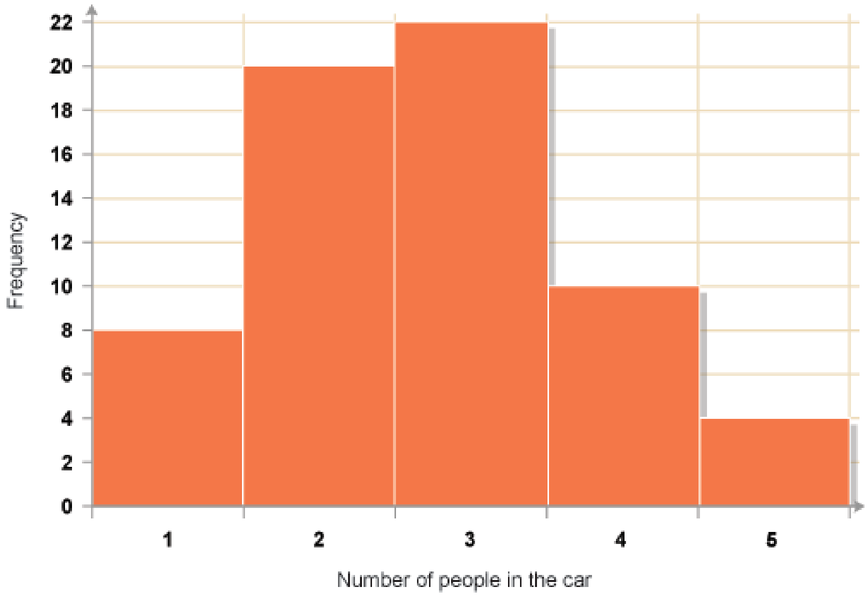


Figure 4.8: Example of a simple bar chart

Variations on the simple bar chart are: multiset bar chart (see Figure 4.9), stacked bar (see Figure 4.10) chart, isometric bar charts and span charts.

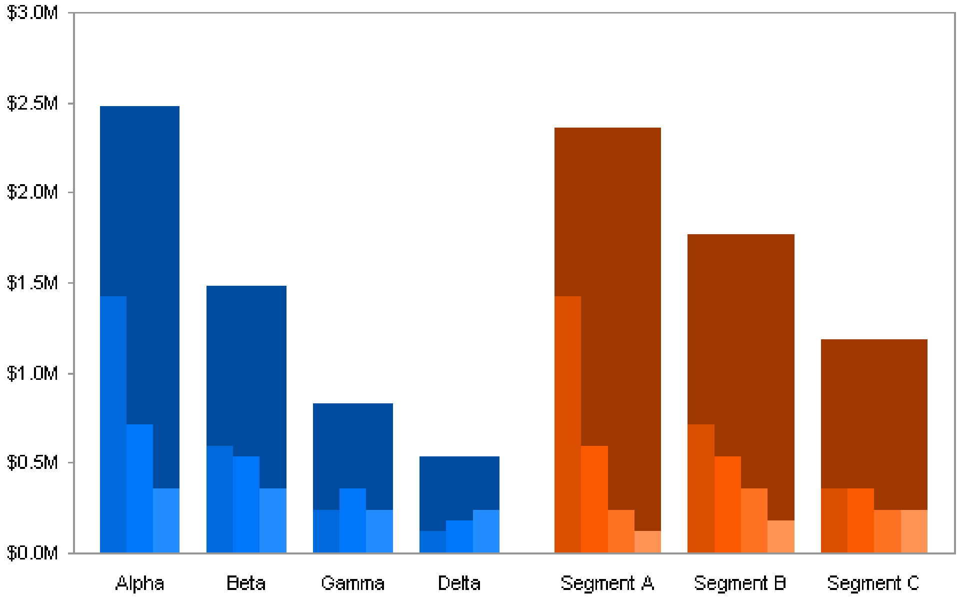


Figure 4.9: Example of a multiset bar chart

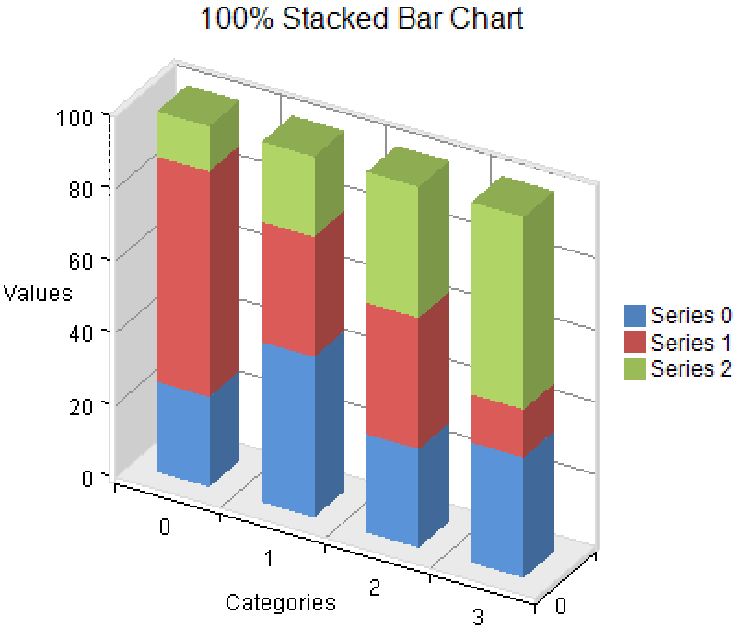


Figure 4.10: Example of a stacked bar chart

Pie chart: A pie chart is a circular object divided into multiple polar segments (see Figure 4.11). It displays the relative magnitude of several quantitative values compared to each other, or, in other words, the distribution of several values that belong to the same dataset. The full circle represents the total magnitude of this dataset, equal to 100 percent, while each segment stands for the magnitude of one particular variable. Segment area, arc length and arc angle of each segment are proportional to the value the segment represents. The segments of a pie chart are usually labeled with percentage numbers rather than total values (although they can feature both for the sake of understanding) [11].

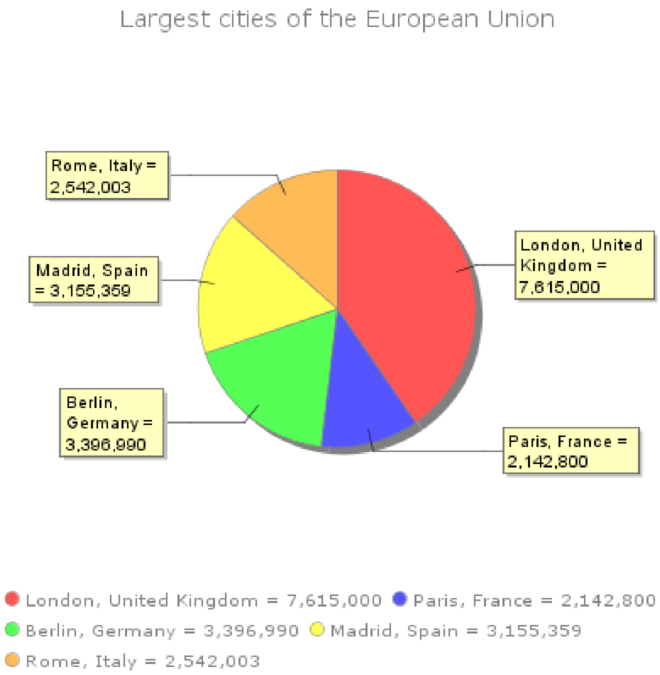


Figure 4.11: Example of a pie chart

A variation on the pie chart is the ring chart. In a ring chart, proportions are shown in multiple rings. In this way, a third dimension can be added to the data (see Figure 4.12).

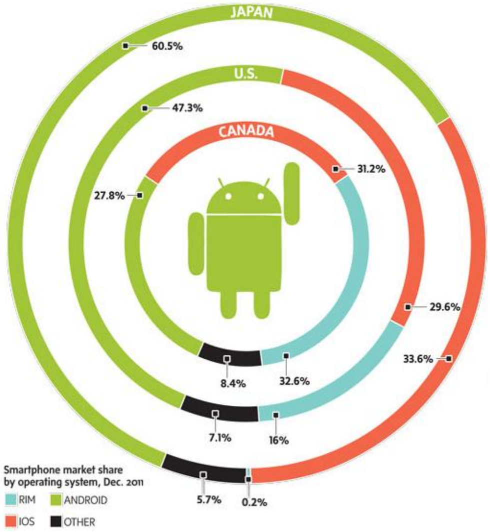


Figure 4.12: Example of a ring chart

4.4.2 Graph-based visualizations on the web

For graph-based visualisations on the web, a comparative study among different frameworks and libraries has been performed. In more detail, two main libraries have been investigated: D3.js[16] and Flotcharts[17]. The criteria that were set for the libraries are twofold: a) it should

[16] http://d3js.org/

[17] http://www.flotcharts.org/

offer possibilities to create different charts (of which at least a stacked bar chart), and b) it should not have a complicated license. The comparative study is included in Appendix D.

4.4.3 Graph-based visualizations on mobile devices

Given the mobile context of the Live+Gov project, explorative investigation about displaying charts on mobile devices has been also conducted. The investigation was specifically tunneled towards examining the libraries that are available in the market. In this investigation, the following aspects have been considered: a) adoption of the library: is it still used/developed and how old is the current version? and, b) license: are we allowed to use the library? The results on this explorative research are described in detail in Appendix D.

5. Data aggregation and visual analytics

5.1 Motivation and approach

The activities described in this section are motivated by the cardinal objective of transforming the information sensed by mobile citizens into valuable insights about the city functioning. In achieving this objective the foreseen activities were primarily oriented towards the summarization of citizen-sensed information into aggregated views and comprehensive summaries. These views and summaries would subsequently allow the decision makers to discover consistent patterns and knowledge structures inside the data. However, as already mentioned in Section 4, in order to come up with a meaningful visualization you must first determine which question to ask, what data are appropriate to answer this question and which visual encoding is the most suitable for the intended purpose.

In applying this rationale in the context of the Live+Gov we have decided to describe the visual analytics tools that have been developed for each use case, along the following dimensions: a) **Target user:** Receiver of the insight, b) **Intended insight:** What are we looking for, c) **Citizen data involved:** The type of data that are appropriate for extracting this insight, d) **External data involved (if applicable):** For cases where the citizen-sensed data are combined with external data to offer a more rich landscape for extracting decisions, e) **Processing required:** Explaining the kind of processing that the data should undergo in order to yield the intended insight when coupled with the appropriate visualization encoding, f) **Visualization encoding:** Describing the visualization tools that are used to display the processed data, as well as the potential options for interacting with the data (facets).

5.2 Visual analytics in Live+Gov

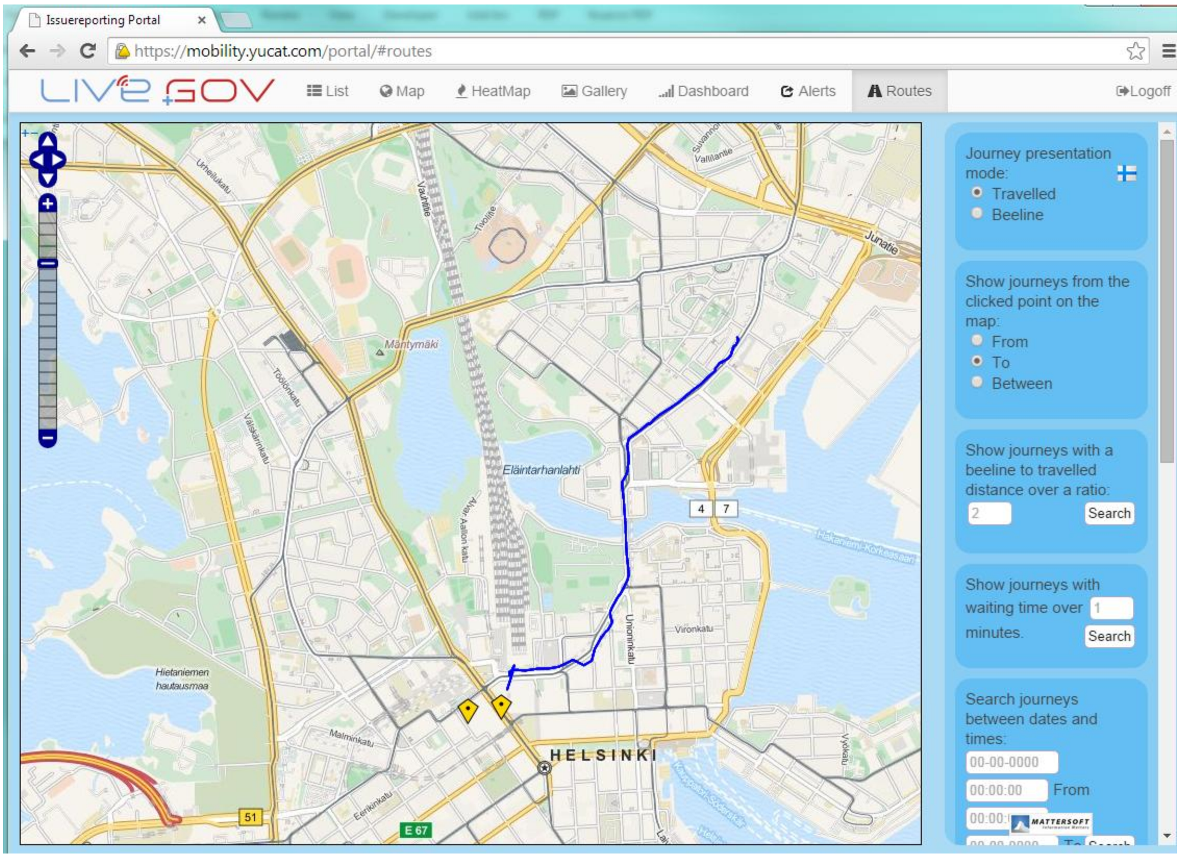
By visual analytics we refer to all processes, methods and tools that have been developed to facilitate the decision makers in gaining valuable insights. Working along the aforementioned dimensions and using a set of screenshots as triggers, our aim in this section is to clearly demonstrate the advancements that have been achieved by Live+Gov in each use case.

5.2.1 Mobility

The goal of the Mobility use case has been to establish a two-way information stream about the mobility status of the city between citizens and authorities. Using the mobile app that has been developed for this trial the citizens are able to constantly sense the mobility status as they move within the city by uploading their every-day routes, as well as by detecting and reporting traffic-related issues. Subsequently, the decision makers (in this case the Helsinki Regional Transport) are given the opportunity to access the aggregated and anonymized users' data. Thus, the insights that we are looking for in this field trial are expected to come from the recordings that have been uploaded and the traffic-related issues that have been reported by the citizens. These insights can either help the decision makers to improve the city's transportation network, as well as obtain a good overview of how traffic-related problems evolve in time and space. Information on how to access the analytics tool that has been developed for the purposes of the Mobility field trial can be found in D5.4 [4].

5.2.1.1 Improving the transportation network

In order to help public authorities in improving the city's transportation network we have developed a visual analytics tool that is capable of visualizing information about the routes recorded during the field trial. Thus, the **target user** of this tool is the authorities (in our case the HSL public transport planners), that are seeking to extract detailed insights about the type of journeys citizens make. Moreover, it is evident that the **citizen-data involved** are the travel data that have been submitted by the citizens through the app. The data are further **processed** either by the human-activity recognition module that is able to classify the different segments of a journey into certain activities (i.e. walking, running, tram, bus, subway, ferry), or by a statistical module that is able to filter the displayed journeys based on a pre-defined set of criteria. As a result, by interacting with this tool the authorities can gain valuable **insights** emerging from the citizen's movements, such as detect the most popular routes, identify the time(s) in the day when there is high demand for public transport services, or even discover if there are areas which are difficult to reach smoothly. As shown in Figure 5.1:, the **visualization method** that has been employed to facilitate the extraction of the aforementioned insights is based on a map that can be operated through a set of filters, imposing certain criteria on the displayed content.



Journey presentation mode:
● Travelled
○ Beeline

Show journeys from the clicked point on the map:
○ From
● To
○ Between

Show journeys with a beeline to travelled distance over a ratio: 2 Search

Show journeys with waiting time over 1 minutes. Search

Search journeys between dates and times:
00-00-0000
00:00:00 From
00:00:

Figure 5.1: Map-based visualization of the recorded citizens' routes based on the applied filters

By modifying these filters the HSL authorities can discover repeating patterns in the passenger journeys. In this way, they can visualize the passenger route chains from one location to another or from one time instance to another, embedding spatial context and time-awareness on their understanding on the use of the transportation network. As a consequence, it rather easy to infer if there are needs for new routes or for shortening the passengers waiting time on the stops, which is not only valuable, but well needed information that has cannot be available for them effortlessly. In the HSL case this type of information is currently gathered manually with different types of surveys, where the survey method has mainly been through phone interviews on a 2-4 year intervals. With this visual analytics tool, the information that is gathered automatically via the mobile app can be easily translated into the necessary insights without the need for a separate survey. Thus, the resources required are significantly less and the decision makers can further benefit from the advantage of having new information received constantly.

The analytics tool relies on a central database where the mobile app logs the user routes and route details. For each route recorded, there is GPS-information, timestamp of the recording, mean of transport and information about the utilized service line. Based on this information, a set of predefined filters can be applied in order to visualize only routes that match the criteria specified by the decision maker. The definition of these filters was done in close co-operation with HSL in the beginning of the project, so as for the analytics tool to offer the filtering functionalities that were considered useful for the public transport planners. In D5.4 [4] (Sect. 3.2.1) there is a detailed description for each of the implemented filters, the corresponding criteria, as well as information about what types of routes and details are returned with the result.

5.2.1.2 Landscape of traffic-related issues

The landscape of traffic related issues offers the authorities a valuable insight on the issues that are regularly addressed in public transportation services and facilities. In this case, the goal of our visualization is to provide the authorities with information on the issues citizens report while using the transportation network. For the reports, a set of categories have been pre-defined allowing the citizens to choose the best category characterizing their issue. Thus, the **target user** in this case is the city authorities that are interested in gaining valuable information on the different types of problems and obstacles passengers face in public transport. The **citizen-data involved** are the reports that have been made by the citizens that may belong to each of the pre-defined categories. The **indented insights** differ greatly based on the category of the reported issues.

Issues related to staff usually deal with bus/tram driver behaviour, such as driving style or communication with passengers. Both types of issues strongly affect the customer experience and the citizens' view on these fields can be used to improve customer satisfaction by creating or improving the instructions and guidelines provided to the staff.

Stop issues include details regarding stop conditions (e.g. broken shelters, litter, etc) or the information that is displayed on the stops (broken displays, outdated information). Issue reports that point out the problems that passengers experience while waiting for the vehicle to arrive, strongly relate to the travel experience and comfort. With the details provided by this type of reports, existing problems are quickly brought to the notice of the responsible people allowing the problems to be fixed faster than ordinary. Also, the information enclosed in the reports (description, location and image) allows the authorities to better understand which tasks need to be performed on site and what will be necessary for fixing the identified issues. In this way, the maintenance persons can be more prepared and better equipped when going on site, making the overall process more efficient.

Service-related reports provide information about the service level in different areas, problems, improvement needed and general opinions that can be used to enhance the passenger satisfaction in the given area. These reports thus, provide important information to the public transport planners about the opinions of the citizens that frequently travel to/from the given area.

Vehicle-related reports are meant for providing operators' information about vehicle conditions. Many of the technical issues can easily be noticed and reported by the driver on duty, but there are several other types of issues, such as broken seats or malfunctioning stop-buttons, that might not be noticed in the superficial daily inspections. These reports could help the operators to more timely notice and react to the issues, thus improving the passenger satisfaction.

Reports related to timetables enclose information about the problems in scheduled arrival times, such as constantly delayed vehicles, chaining of vehicles in certain areas, or unreliable driving times. This information is valuable for the public transport planners who can use it to better design the timetables in problematic areas.

There is also a category "Other" for the type on issues that do not belong to any of the pre-defined categories. In these reports, general feedback on the public transport is usually given, including citizen opinions on the services, general suggestions, etc. All issues are formed based on the details users provide with their reports, such as location, category, description and images. This fact allows the employment of various **visualization methods** that are able to filter the displayed issues based on spatio-temporal criteria and facilitate the identification of patterns leading to the aforementioned insights. These methods are similar to the ones that have been used in the urban maintenance use case, such as map-based view, gallery view, graph-based aggregation, etc. More details about these methods are provided in Section 5.2.2.

5.2.2 Urban Maintenance

The aim of this field trial has been to offer a public place where both citizens and officials can communicate, share and re-act upon essential information about their city. Ranging from issue reports and citizens initiatives, all the way to co-maintenance spots and government-initiated participation projects, the information collected and hosted in the developed infrastructure has the potential to accurately capture and vividly describe "what is happening in the city". Thus, in this case the insights are expected to derive from the data provided by both citizens and officials, while the insights are once again intended not only for the officials but also for the citizens themselves. In the following, we describe how the visual analytics tool developed for urban maintenance succeeds in facilitating the extraction of the intended insights. Information on how to access the analytics tool that has been developed for the purposes of the Urban Maintenance field trial can be found in D5.4 [4].

5.2.2.1 Discover spatio-temporal patterns in the collected data

One of the most effective ways to understand "what is happening in the city" is to follow the spatio-temporal patterns that are formulated by the contributed data. By visualizing these patterns you can easily detect phenomena like the existence of outliers, or segments in time or space with high activity. Given that such phenomena deviate from the ordinary, they usually foster interesting insights. Thus, the **target user** in this case is both the citizens and city officials that are interested in understanding which is the "ordinary" pattern of the city functioning and whether there are any "anomalies" that may be of interest to them. The **data involved** in this case consists of all four types (i.e. citizens' initiatives, co-maintenance spots, reported issues and government-initiated participation projects), while the **processing** required for revealing the aforementioned phenomena pertains to statistical calculations like histogram-based aggregation, averaging, grouping, etc. Finally, the **visualization method** that is most suitable for presenting the results of this processing is based on maps that are further enhanced with the ability to apply spatial, temporal and textual filters. The map-based visualizations that have been developed for the visual analytics tool of the urban maintenance use case are cluster-enabled maps (that are further enriched with category-based color coding) and heat-map based maps (see Figure 5.2, and Figure 5.3). These map-based tools have been implemented so as to incorporate facet browsing techniques. More specifically, multiple filters can be applied on the displayed data allowing users to set their own criteria. The filtering capabilities consists, among others, in setting the date range, the address, the name of the municipality, the category characterizing the report, and even performing free text search in the description associated with each report. Deliverable D5.4 [4], provides a detailed description about the filtering capabilities of the developed tool.

Based on the above, the **intended insights** that could be facilitated by the developed visual analytics tool may range from general observations like the distribution of reports in certain areas (revealing which neighbourhoods are more active, as well as what is happening in a neighbourhood, which citizen or government initiative is happening, where influence can be added), all the way to topic specific observations like the spatio-temporal distribution characterizing the issues related to broken street lamps (i.e. whether there are certain neighbourhoods that seem to be more heavily affected by this issue, or whether there are certain times in the year where this issue becomes more frequent).

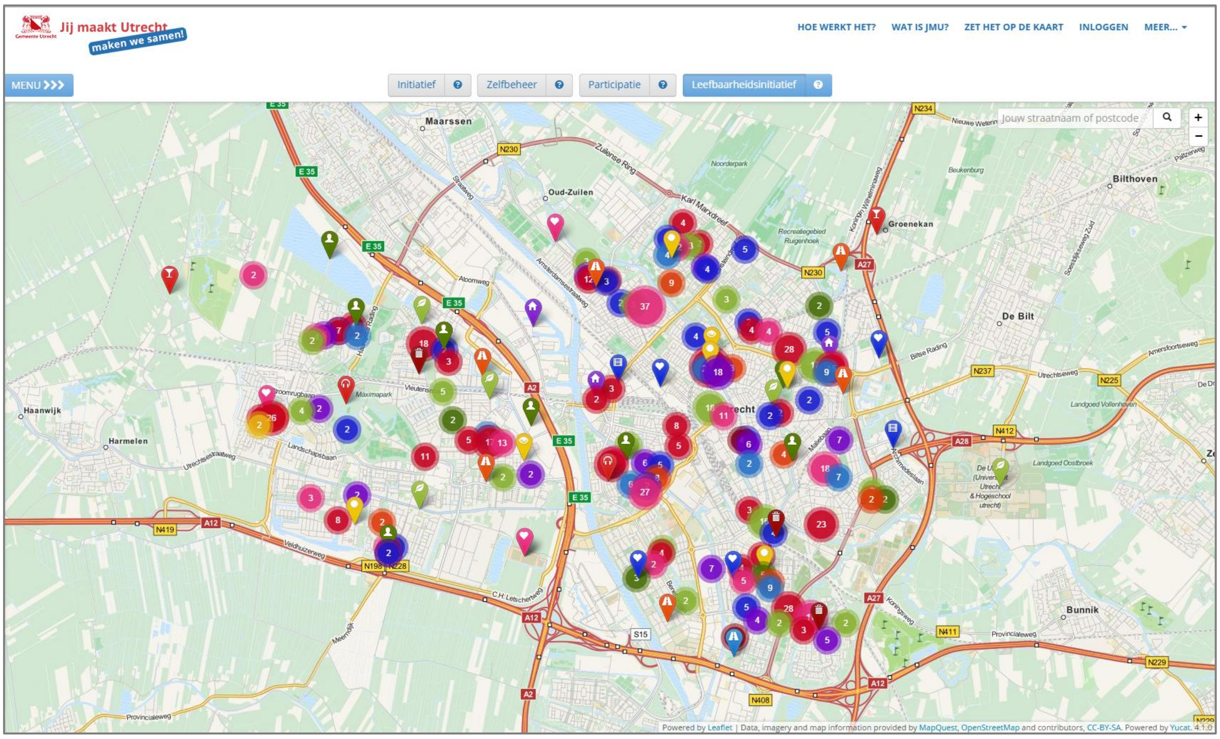


Figure 5.2: Cluster-enabled maps with category-based color coding

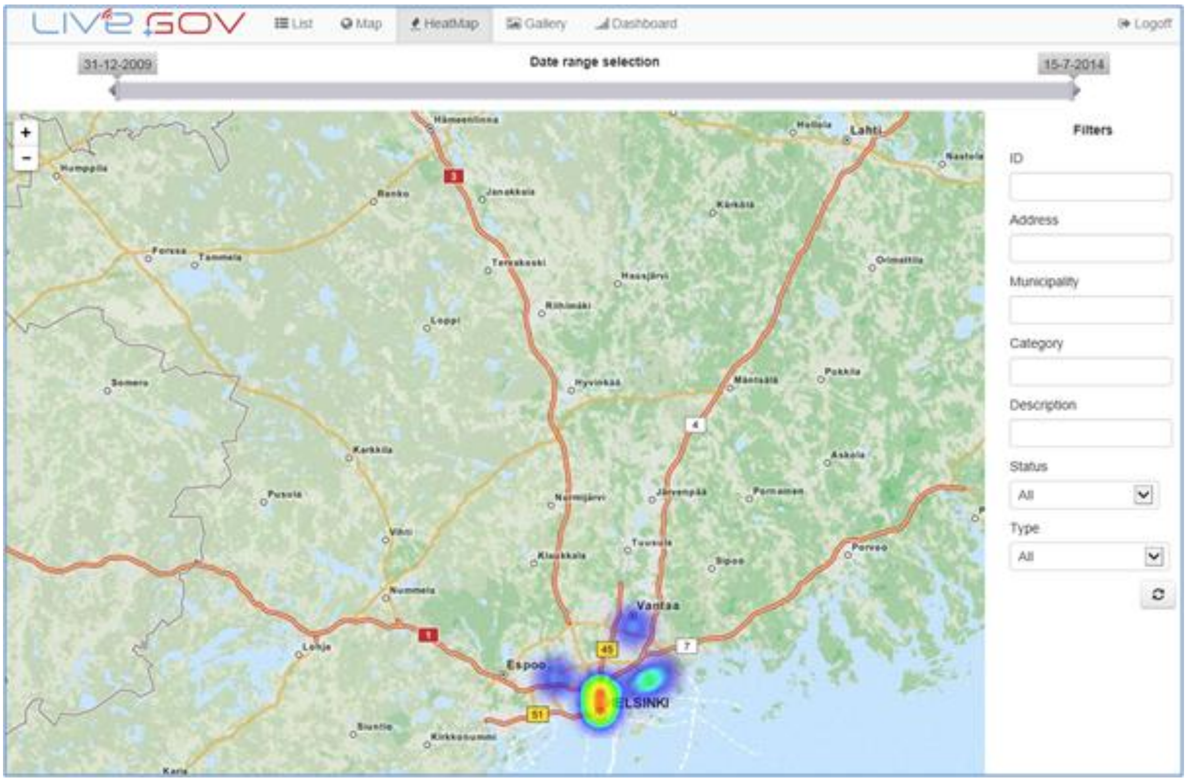
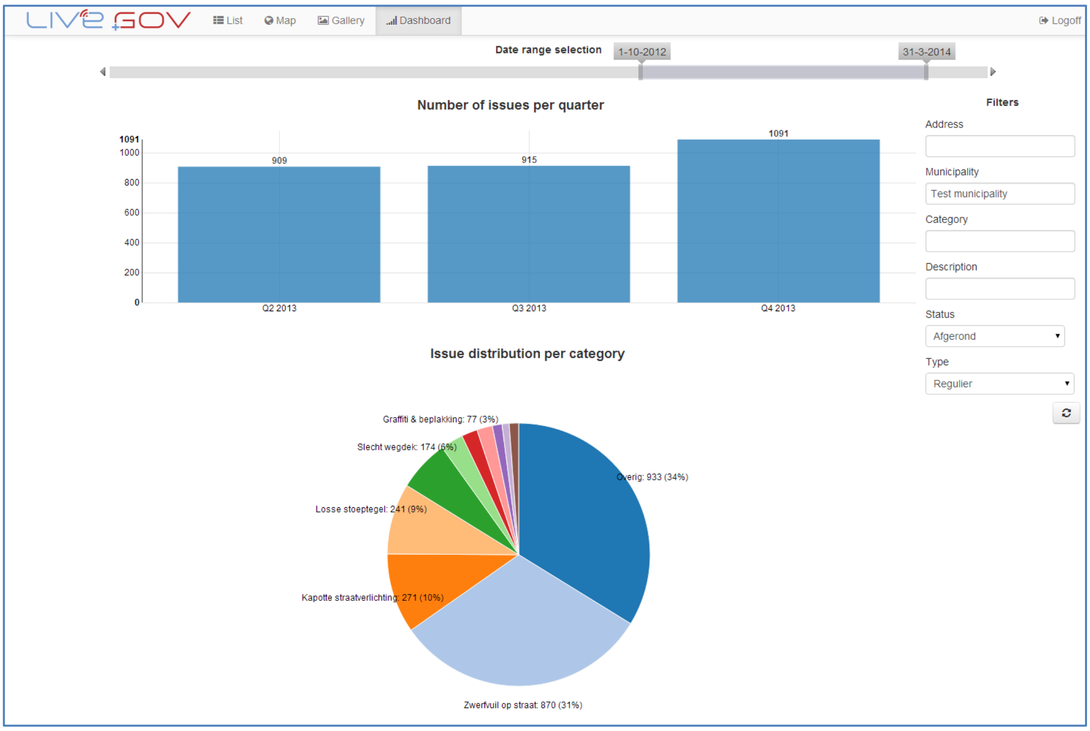


Figure 5.3: Heatmap-based maps allowing to discover the neighbourhoods with high activity

5.2.2.2 Understanding the statistical tendencies of the data

Live+Gov has developed a graph visualisation dashboard view as can be seen in Figure 5.4. Graph based visualisations provide statistical information that is easy to digest. In issue reporting, the **target group** is the public officials of the municipality. Charts are available on the number of issues per quarter, so the date of issue reports is used. Next to that, categories are shown in a pie chart. The **data involved** is the data generated in the Urban Maintenance field trial through mobile input from Issue Reporting. The offered bar and pie-charts are implemented with the use of a Data-Driven Document (D3) JavaScript library. The webapplication makes use of specific queries that are optimised for streamlined data requests from the server. This way, the data transmissions between client and server are stripped from surplus information. With this **processing**, the Issue Reporting web module can offer fluent graph-presentations that are rich and fast in response, which is crucial for a good user experience and effective management report tooling. Users have different options to **interact** with the data visualisation. Most important are the filter options provided on each of the following data or on a combination of the following data: i) Issue Report ID (melding id), ii) Address of the reported issue, ii) Municipality, Category of the reported issue, iv) Description of the reported issue, and v) Status. Furthermore, the user can select the data based on time-range with the time-slider. In this view, the user is able to get aggregated presentations of the reported issues, e.g. distribution over time or topic. With the included options to filter, the graphs are a great tool for management reports and trend analysis. With the combined use of different filters and aggregation options, **insights** as trend-lines over time, season-specific issue reporting topics, distribution of issue types are easily discovered and can be of valuable input for optimising work distribution and development of new or adjusted policies.


List
Map
Gallery
Dashboard
Logoff
Number of issues per quarter
1091
1000
800
600
400
200
0
909
915
1091
Q2 2013
Q3 2013
Q4 2013
Issue distribution per category
Graffiti & beplakking: 77 (3%)
Slecht wegdek: 174 (6%)
Overig: 933 (34%)
Losse stoeptegel: 241 (9%)
Kapotte straatverlichting: 271 (10%)
Zwerfvuil op straat: 870 (31%)
Filters

Date range selection 1-10-2012 31-3-2014

Address

Municipality: Test municipality

Category

Description

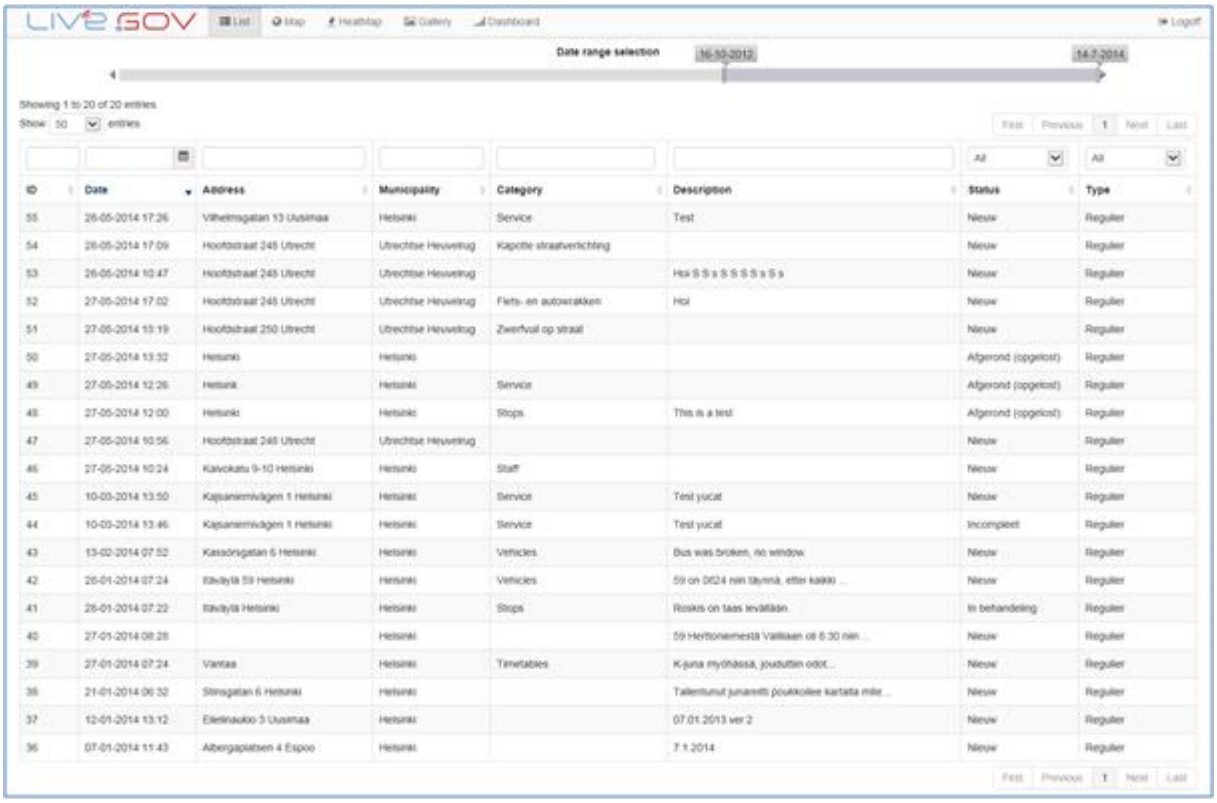
Status: Afgerond

Type: Regulier

Figure 5.4: Dashboard: Graphs of Issue Reporting web module

5.2.2.3 Monitoring and administering the flow of data

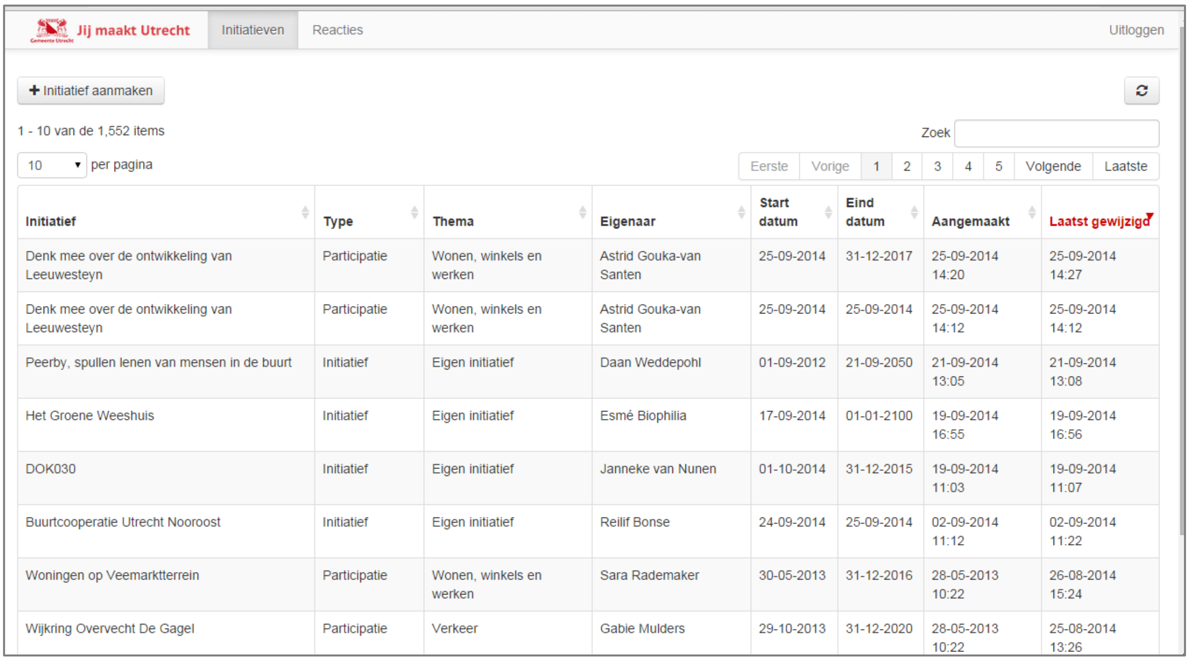
One rather basic visualisation of data is a list-view, a representation of data in a data-grid. This basic representation is important for the authorities involved with the data. The view is not particularly well-suited for citizens. The **target users** for the list-view are public officials of the municipality, and more specifically those responsible for monitoring and moderation of Issues or citizen initiatives. Examples of the list view as used in Issue Reporting and *JijMaaktUtrecht* can be seen in Figure 5.5 and Figure 5.6.



Showing 1 to 20 of 20 entries
Show 50 entries

ID	Date	Address	Municipality	Category	Description	Status	Type
						All	All
55	26-05-2014 17:26	Vilhelmsgatan 13 Uusimaa	Helsinki	Service	Test	Nieuw	Regulier
54	26-05-2014 17:09	Hoofdstraat 245 Utrecht	Utrechtse Heuvelrug	Kapotte straatverlichting		Nieuw	Regulier
53	26-05-2014 10:47	Hoofdstraat 245 Utrecht	Utrechtse Heuvelrug		Hoi $ $ s $ $ $ $ s $ s	Nieuw	Regulier
52	27-05-2014 17:02	Hoofdstraat 245 Utrecht	Utrechtse Heuvelrug	Fiets- en autowrakken	Hoi	Nieuw	Regulier
51	27-05-2014 15:19	Hoofdstraat 250 Utrecht	Utrechtse Heuvelrug	Zwerfvuil op straat		Nieuw	Regulier
50	27-05-2014 13:32	Helsinki	Helsinki			Afgerond (opgelost)	Regulier
49	27-05-2014 12:26	Helsink	Helsinki	Service		Afgerond (opgelost)	Regulier
48	27-05-2014 12:00	Helsinki	Helsinki	Stops	This is a test	Afgerond (opgelost)	Regulier
47	27-05-2014 10:56	Hoofdstraat 245 Utrecht	Utrechtse Heuvelrug			Nieuw	Regulier
46	27-05-2014 10:24	Kaivokatu 9-10 Helsinki	Helsinki	Staff		Nieuw	Regulier
45	10-03-2014 13:50	Kajsaniemivägen 1 Helsinki	Helsinki	Service	Test yucat	Nieuw	Regulier
44	10-03-2014 13:46	Kajsaniemivägen 1 Helsinki	Helsinki	Service	Test yucat	Incompleet	Regulier
43	13-02-2014 07:52	Kasadorsgatan 6 Helsinki	Helsinki	Vehicles	Bus was broken, no window	Nieuw	Regulier
42	28-01-2014 07:24	Itäväylä 59 Helsinki	Helsinki	Vehicles	59 on 0624 niin täynnä, ettei kaikki ...	Nieuw	Regulier
41	28-01-2014 07:22	Itäväylä Helsinki	Helsinki	Stops	Roskis on taas levällään.	In behandeling	Regulier
40	27-01-2014 08:28		Helsinki		59 Herttoniemestä Vallilaan oli 6:30 niin ...	Nieuw	Regulier
39	27-01-2014 07:24	Vantaa	Helsinki	Timetables	K-juna myöhässä, jouduttiin odot...	Nieuw	Regulier
38	21-01-2014 06:32	Stinsgatan 6 Helsinki	Helsinki		Tallentunut junareitti poukkoilee kartalla mile...	Nieuw	Regulier
37	12-01-2014 13:12	Elielinaukio 3 Uusimaa	Helsinki		07.01.2013 ver 2	Nieuw	Regulier
36	07-01-2014 11:43	Albergaplatsen 4 Espoo	Helsinki		7.1.2014	Nieuw	Regulier

Figure 5.5: List-view of Issue Reporting web module

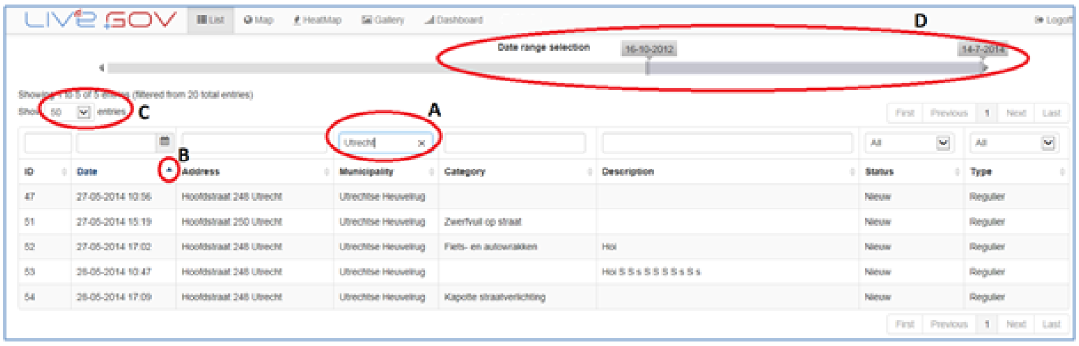


1 - 10 van de 1,552 items
10 per pagina

Initiatief	Type	Thema	Eigenaar	Start datum	Eind datum	Aangemaakt	Laatst gewijzigd
Denk mee over de ontwikkeling van Leeuwesteyn	Participatie	Wonen, winkels en werken	Astrid Gouka-van Santen	25-09-2014	31-12-2017	25-09-2014 14:20	25-09-2014 14:27
Denk mee over de ontwikkeling van Leeuwesteyn	Participatie	Wonen, winkels en werken	Astrid Gouka-van Santen	25-09-2014	25-09-2014	25-09-2014 14:12	25-09-2014 14:12
Peerby, spullen lenen van mensen in de buurt	Initiatief	Eigen initiatief	Daan Weddepohl	01-09-2012	21-09-2050	21-09-2014 13:05	21-09-2014 13:08
Het Groene Weeshuis	Initiatief	Eigen initiatief	Esmé Biophilia	17-09-2014	01-01-2100	19-09-2014 16:55	19-09-2014 16:56
DOK030	Initiatief	Eigen initiatief	Janneke van Nunen	01-10-2014	31-12-2015	19-09-2014 11:03	19-09-2014 11:07
Buurtcooperatie Utrecht Nooroost	Initiatief	Eigen initiatief	Reilif Bonse	24-09-2014	25-09-2014	02-09-2014 11:12	02-09-2014 11:22
Woningen op Veemarktterrein	Participatie	Wonen, winkels en werken	Sara Rademaker	30-05-2013	31-12-2016	28-05-2013 10:22	26-08-2014 15:24
Wijkring Overvecht De Gagel	Participatie	Verkeer	Gabie Mulders	29-10-2013	31-12-2020	28-05-2013 10:22	25-08-2014 13:26

Figure 5.6: Interface for moderating by municipality officials of citizen initiatives

The **data involved** depends on the specific use case scenario. For initiatives in JMU, this includes Title, Type, Theme, Owner, Start date, End date, Date and time initiative added to JMU, Date and time initiative was last modified. Because of the **intended aim** of the visualisation to be able to monitor and moderate reactions, Date and time when the reaction was placed, Name of the person who placed the reaction, the Message itself and the Initiative the reaction was added to are displayed as well. For Issue Reporting the **data** involved is Issue Report ID, Date of the report, Address of the reported issue, Municipality, Category of the reported issue, Description of the reported issue, Status, and Type of the report. In general, for the list-view not much **processing** is required as it is database data that is presented in a data-grid. In order to be able to make use of a list-view, **interaction possibilities** required. In the list, a filter option is added, based on a standard search query. In the upper right corner of the screen, a moderator can search on free text, e.g. for a specific initiative, type, theme, or owner. In the list, filter options are added to each column, based on a standard search query (A in Figure 5.7). In the fields above each column, a user can search on free text. Arrangement of the data is possible in two ways. The first arrangement possibility is offered by sortable columns – a user can sort the data by any column in ascending or descending order (B in Figure 5.7). The second arrangement possibility is offered by boundary filters – on the left above the list, the number of rows shown on screen can be set (C in Figure 5.7). In this way, a moderator can scroll through the data by pages or by scrolling down the screen. Overall, the user can select the data based on time-range with the time-slider (D in Figure 5.7); a feature that is active for all the different views of the Issue Reporting web module. Exploration of the data in more detail is possible by clicking on a specific row. The details of that issue are opened.



Showing 1 to 5 of 5 entries (filtered from 20 total entries)

Show 50 entries

Utrecht

ID	Date	Address	Municipality	Category	Description	Status	Type
47	27-05-2014 10:56	Hoofdstraat 248 Utrecht	Utrechtse Heuvelrug			Nieuw	Regulier
51	27-05-2014 15:19	Hoofdstraat 250 Utrecht	Utrechtse Heuvelrug	Zwerfvuil op straat		Nieuw	Regulier
52	27-05-2014 17:02	Hoofdstraat 248 Utrecht	Utrechtse Heuvelrug	Fiets- en autowrakken	Hoi	Nieuw	Regulier
53	28-05-2014 10:47	Hoofdstraat 248 Utrecht	Utrechtse Heuvelrug		Hoi S S s S S S S s S s	Nieuw	Regulier
54	28-05-2014 17:09	Hoofdstraat 248 Utrecht	Utrechtse Heuvelrug	Kapotte straatverlichting		Nieuw	Regulier

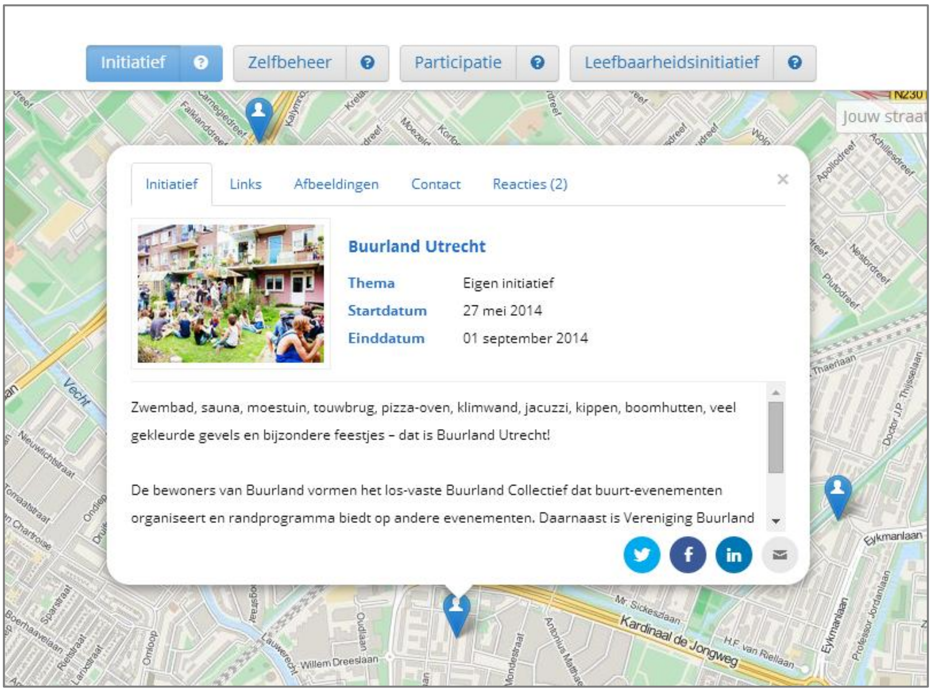
Figure 5.7: Sort, search, and filter options in the issue reporting list-view

The information in the list-view is presented in a straightforward layout, typically meant for quick overviews on the data, e.g. for statuses overview and monitoring of issue reports. The **insights** are based on offering a quick way to monitor and moderate initiatives and issue reports. The presentation offers an easy way to check which initiatives or issue reports are new since the last visit and thus need judgement on their appropriateness. These easily obtained insights make monitoring and moderation effective. In the list view from Reactions on initiatives, a moderator can see the latest reactions added to initiatives on *JijMaaktUtrecht*. This offers insights on whether or not moderation is necessary on reactions. Overall, the presentation of data makes quick workflow processing within administrations possible.

5.2.2.4 Understanding the semantic evolution of a report

An issue report or an initiative contains information that is specific for the particular entity. This can be compared with a specific advertisement on eBay, or properties on a real estate agency website. Where in most visualisations the goal is to aggregate information, with the detail-view the purpose is to offer all the specific information of the entity itself. Here, the **targeted users** (for initiatives these are citizens and public officials, for issue reports the target group is public officials) are provided with a single location which encompasses all the relevant information of one entity. The **involved data** in the detail view contains the following information per issue report: Issue Report ID (melding id), Date of the issue report, Address of the reported issue, Category of the reported issue, Description of the reported issue, Status, User details, Feedback (read and add), Photo, Location of the issue report on the map. For J*ijMaaktUtrecht*, the detail information of an initiative includes: Location (required), An initiative is part of a Theme, a Theme is part of certain Type (required), Title (required), Description (required), Start- and end date (required), Links (optional), Images (if more than one; optional), Contact information of the 'owner' (owner is required, showing contact information is optional), Reactions (optional). Because multiple user groups (citizens as well as authorities) make use of the detail-view of initiatives, precaution is taken whether or not specific privacy sensitive user information is displayed or not. Each initiative has, in principle, an owner. This can be a municipality official in the case of participation projects, as well as a citizen or an employee of any other organisation. The owner manages the 'content' of his/her initiative and is the contact person of that particular initiative. An owner can decide for himself if contact information is showed in the tab 'Contact' with his initiative. The **intended insight** for the initiative detail-view is to explore an initiative in more depth. On the map and in the gallery, the title and short description are shown. However, given that there is more information about an initiative, the detailed view offers access to this information. For an impression of the detail view of an initiative, see Figure 5.8. From the viewpoint of the initiative 'owner' (the person that published the initiative on JMU), intended insights are relating to a semantic and contextual evaluation of initiatives or issue reports. Examples of the offered functionality, information, and context embedding are: communication about the initiative to visitors of JMU, relations of the initiative with others via social media or e-mail, lead visitors to the web page of the initiative (via 'Links'), provoke reactions about the initiative (via 'Reactions'), find interested persons/parties (via 'Reactions' or 'Contact'). With the detail-view, for citizens it is possible to gather insights on what an initiative is about, see when the initiative is active (via 'Start and end date'), see where more information about the initiative is available (via 'Links'), share the initiative via social media or e-mail, e.g. to engage others. In short, the goal of this detail-view is to see more information about the initiative, find out where more (digital) information is shown and share the initiative or give a reaction via JMU directly.

Connections to external data sources are made when sharing an initiative via social media or e-mail. This can be done by clicking on the corresponding symbol in the lower right corner of the detail-view (see Figure 5.9; shown is only the information on one of the different tabs). Clicking on these opens a popup for Twitter, Facebook, LinkedIn or takes the visitor to his e-mail client.



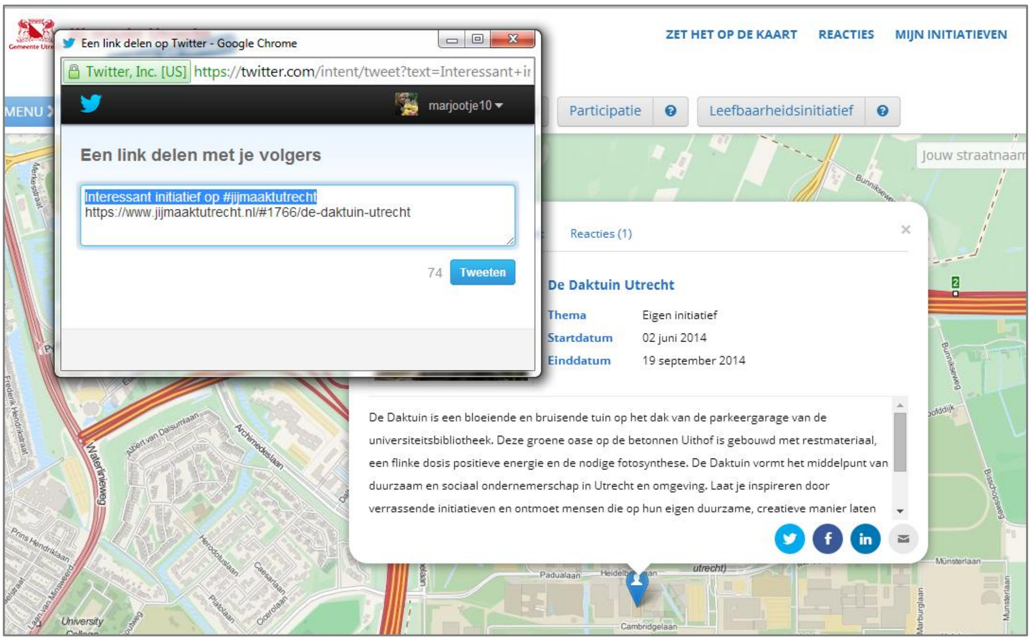
Buurland Utrecht

Thema	Eigen initiatief
Startdatum	27 mei 2014
Einddatum	01 september 2014

Zwembad, sauna, moestuin, touwbrug, pizza-oven, klimwand, jacuzzi, kippen, boomhutten, veel gekleurde gevels en bijzondere feestjes – dat is Buurland Utrecht!

De bewoners van Buurland vormen het los-vaste Buurland Collectief dat buurt-evenementen organiseert en randprogramma biedt op andere evenementen. Daarnaast is Vereniging Buurland

Figure 5.8: Detail view of an initiative: short summary of an initiative in tab 'Initiatief'



Een link delen met je volgers

Interessant initiatief op #jijmaaktutrecht
https://www.jijmaaktutrecht.nl/#1766/de-daktuin-utrecht

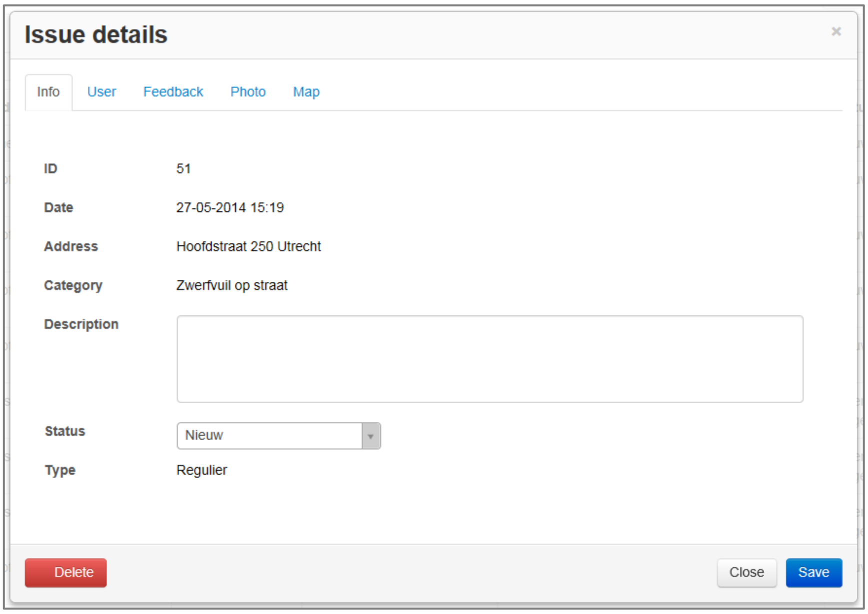
De Daktuin Utrecht

Thema	Eigen initiatief
Startdatum	02 juni 2014
Einddatum	19 september 2014

De Daktuin is een bloeiende en bruisende tuin op het dak van de parkeergarage van de universiteitsbibliotheek. Deze groene oase op de betonnen Uithof is gebouwd met restmateriaal, een flinke dosis positieve energie en de nodige fotosynthese. De Daktuin vormt het middelpunt van duurzaam en sociaal ondernemerschap in Utrecht en omgeving. Laat je inspireren door verrassende initiatieven en ontmoet mensen die op hun eigen duurzame, creatieve manier laten

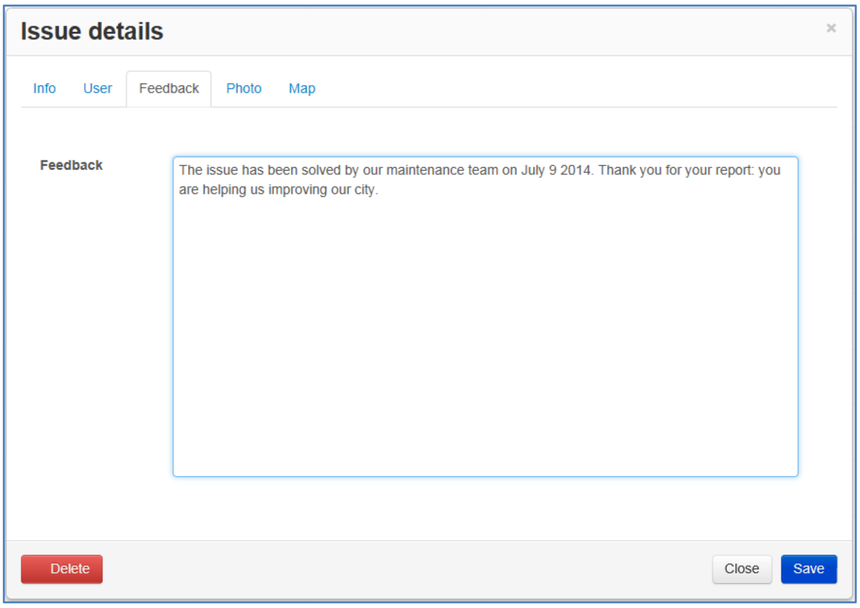
Figure 5.9: Share an initiative via Twitter

For issue reports, the targeted users are within the municipality administration, in a professional environment. Here the report details give specific insights and options to do professional processing of a filed issue. In Figure 5.10 the details are shown, including the different tabs at the top. Specifically the status and feedback options (see Figure 5.11) are of importance in the use case, as this functionality provides a two-way communication channel between citizen and municipality. The other information in the detail view makes proper processing and next workflow actions possible.



ID	51
Date	27-05-2014 15:19
Address	Hoofdstraat 250 Utrecht
Category	Zwerfvuil op straat
Description	
Status	Nieuw
Type	Regulier

Figure 5.10: Detail-view of an issue report



Feedback: The issue has been solved by our maintenance team on July 9 2014. Thank you for your report: you are helping us improving our city.

Figure 5.11: Extended feedback mechanism integrated in the Issue Reporting web module

As described above, the detail-view offers contextual information and offers functionalities as well. In the detail-view, **interaction** is supported by detail on demand: via the specific tabs, a user can see the details and interact on them, e.g. by providing the issue report with feedback. This feedback is automatically updated in the mobile client of the citizen. The detail-view of an initiative can be accessed by clicking on a marker in the map view or by clicking on an image in the gallery view. In the detail-view itself, not all information is showed at once in order not to overflow the user with information. The information is grouped in different tabs Initiative, Links, Images, Contact, Reactions.

5.2.2.5 Obtain a coherent view of the topic through visual manifestations

One of the most powerful ways to offer a real-life impression on what is happening in the city is an image-based presentation. The great benefit of this type of visualization is the ability to provide a quick and comprehensive glimpse of the available content, the resolution of ambiguities (i.e. textual), as well as the fact that they can be a very powerful tool in discovering the stories that are hidden in the data. In this direction, one of the visualization tools that has been extensively used in Live+Gov is the gallery-view as can be seen in Figure 5.12 and Figure 5.13.

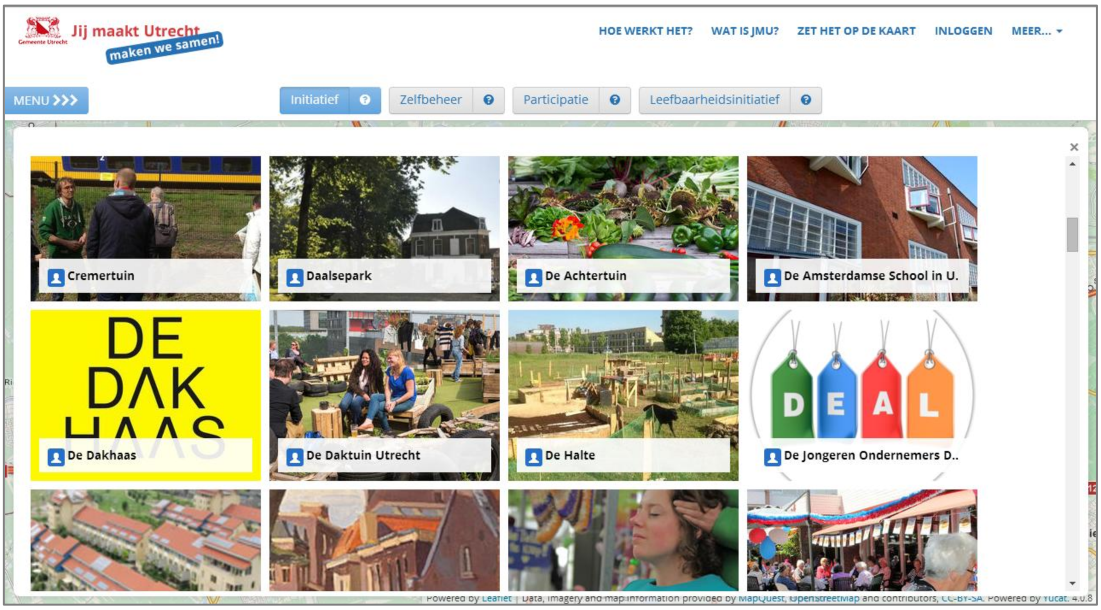


Figure 5.12: Gallery view in *Jij Maakt Utrecht*

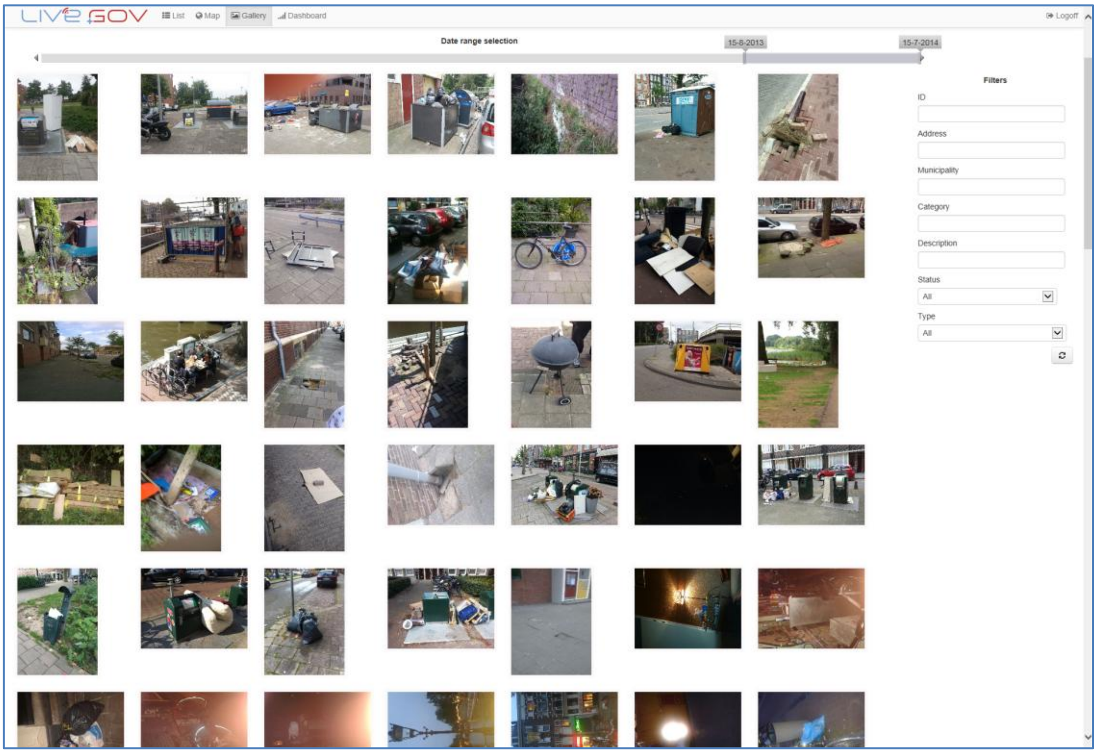


Figure 5.13: Gallery-view of Issue Reporting web module

This visualisation is **targeted** to public officials of the municipality in the case of issue reporting. For JijMaaktUtrecht the target audience are both public officials and citizens. The **presented data** are the images from issue reports and the images of initiatives. These images are accompanied with other data of the respective entity. This makes **interaction options** possible, for example search and filter functionality. With these interaction possibilities, the image gallery provides users with easy to use methods to generate a visual impression on categories and/or time based selections. Filtering involves standard search query on free text – by typing any word, e.g. ‘green’, all initiatives within the current Type that use this word (in their title, description or Theme) will be filtered out in the gallery; Boundary filter for actuality – below the textual search, an actuality filter can be set by clicking on one or more out of the three options “Geweest”, “Actueel” or “Binnenkort”. This filter is based on the start and end dates at an initiative. Selecting all options will show all initiatives (of no other filters are set) in the gallery; Filtering on Themes – below the actuality filter in the menu, themes can be turned on or off by clicking on the icons. For issue reporting, the filter on category is particularly powerful, but also filters on status and date will help to create the desired sub-sets of images.

The visual presentation offers possibilities to generate **insights** on the variety of topics within a category. Furthermore, the visuals directly make clear what the initiative is about and well-chosen images are able to bring enthusiasm and can motivate to participate. Within administrations, having access to an image based presentation, the municipality is provided with a great tool for storytelling, which is difficult to achieve with pure textual analytics. With images it is very easy to give people less involved in the matter, a quick and adequate impression of real-life scenarios. A potential downside of images is the performance of a gallery representation. In order to be a powerful tool, it is required that browsing through images is fluent and offered in user-friendly manner. One important measure that has been taken in Live+Gov is the **processing** of images. In Appendix C we report on the optimizations that have been employed to maintain a high level of user experience.

5.2.2.6 Assess and prioritize issues based on external context

The rationale behind the urban maintenance use case is to administer and analyze the data contributed by citizens. As already became clear from the previous sub-sections, the analysis of collected content is performed through a web-based analytics tool that employs a number of visualization (i.e. heatmaps, gallery view) and spatio-temporal filters to facilitate both citizens and officials in gaining valuable insights about the city functioning. However, even though the aforementioned visualizations are able to present the collected data in comprehensive forms, they are provided in a rather poor context (i.e. bare map presenting information only about the application-specific data) which is not always adequate to facilitate the process of turning simple observations into well-advocated decisions. Driven by this fact we have investigated the option of enhancing our analytics tools by allowing to easily plug-in and out geo-enabled information layers of governmental data.

In this respect, the **target users** of this enhancement are, once again, both the citizens and city officials that are interested in detecting patterns in the functioning of their city, but they are further interested in visualizing additional information that may help them interpret these patterns. The **data involved** are, as in the previous case, all four application-specific data types, which are further complemented by a number of **external datasets**. More specifically, our ambition was to benefit from on-going initiatives in making non-personal government data, freely available to everyone to use and republish as they wish. These data are typically generated by governmental departments (either regional of federal) and may relate to: a) infrastructures (e.g. position of schools, hospitals, city parks, public facilities, etc), or b) statistics deriving from the departments of economy (e.g. average income per area), health (e.g. satisfaction index on health care services), environment (e.g. population density, or electricity consumption), transport (e.g. road traffic), police (e.g. criminality), education (e.g. success rates for higher education per area) and many more.

This type of information is of great value when correlated with the user submitted issues revealing, otherwise difficult to discover, **insights**. For instance, by overlaying geo-enabled information related to the geographical position of schools, hospitals and critical infrastructures, or by overlaying traffic related information along with a density-based heatmap of the submitted issues, critical decisions like: “which issues to resolve first?”, “how to plan my resources for next year?”, or “where should I invest more at the level of infrastructures?” could be made on a much more informed basis. The **visualization method** to present the external data are plain maps (in either topographical or heatmap view) and the **processing** required entails in making the data compatible with the utilized map visualization module (e.g. following the KML standard). The only additional requirements that should be satisfied to offer a meaningful visualization experience is for the enhanced tool to support keywords based searching with the long list of available datasets, as well as the ability to overlay more than one data layers simultaneously.

The following screenshot demonstrates the enhancement that has been implemented to visualize external datasets on top of the citizen-contributed content. On the left-side of the panel the user can search for the appropriate dataset by providing keywords. Then, by clicking on the corresponding dataset the geo-located information is displayed on the map. For instance, Figure 5.14 displays the “City Council Districts” allowing for the user to understand which of the council members is more heavily loaded with requests during the specific time-period.

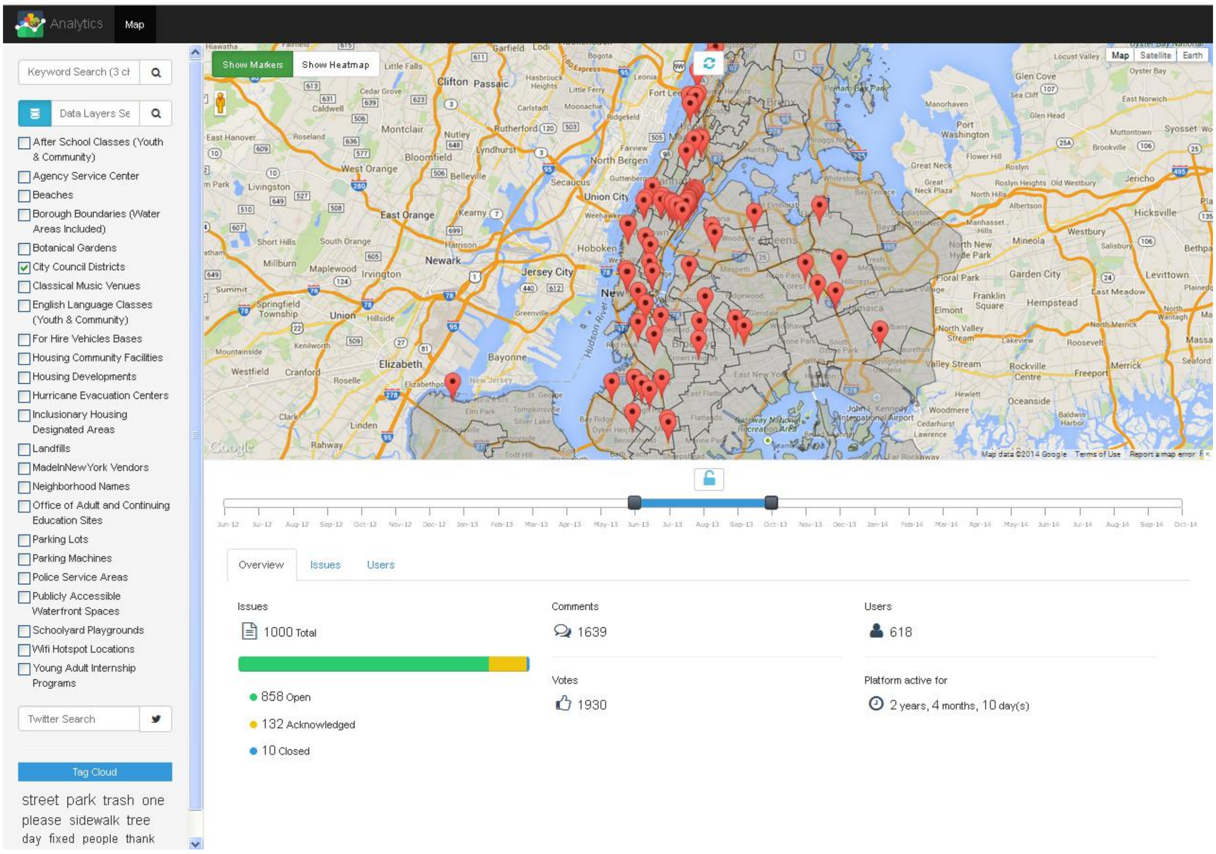


Figure 5.14: The City Council Districts are displayed in combination with the user submitted issues (http://lganalytics.mklab.iti.gr)

Similarly, in Figure 5.15 the “Schoolyard Playgrounds” are depicted allowing the city administrator to prioritize the issue that has been reported right in front of the playground and maybe jeopardize the safety of small kids.

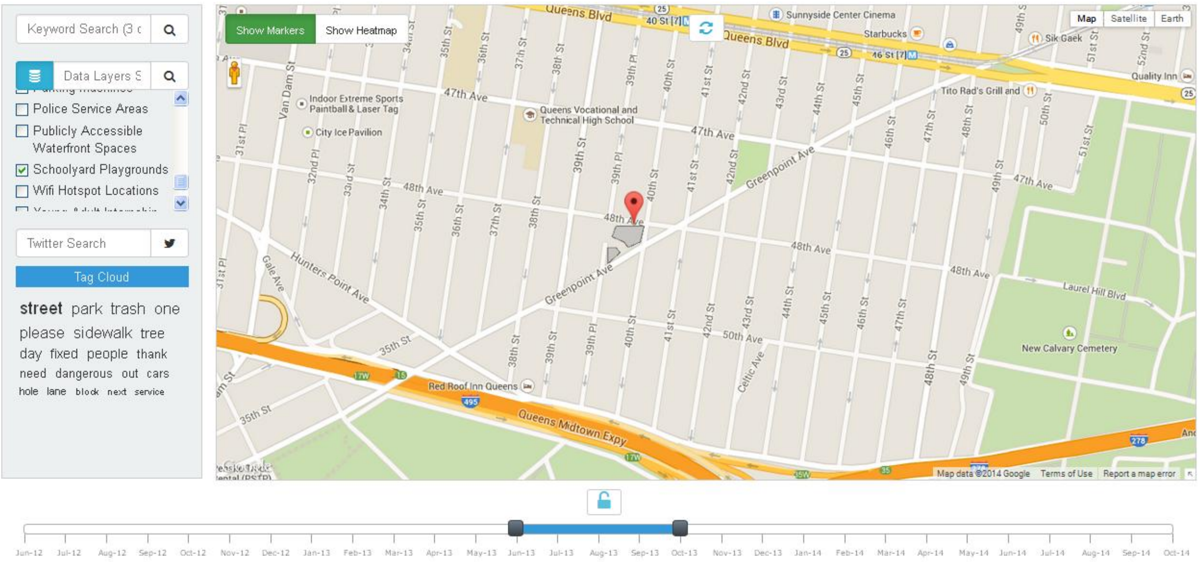


Figure 5.15: Schoolyard playgrounds are depicted next to the reported issues allowing the administrators to better prioritize their resolution (http://lganalytics.mklab.iti.gr)

5.2.2.7 Enriching the issues' context using social media

Apart from the data layers that have been made public from governmental agencies, another very rich source of contextual information is the social media stream that is generated by user-contributed content. Twitter messages, facebook posts and foursquare check-ins constitute some of the most promising sources of content in acquiring the desired context. However, in order for this user-generated content to become valuable in our case, it should feature geo-located information allowing its positioning on the city map. The direct consequence of this restriction was that the number of geo-located facebook posts and twitter messages that were found to relate with the reported issues was too small to facilitate the extraction of a meaningful contextual layer. On the contrary, the information obtained from foursquare was not only geo-located but also characterized with one of the categories included in the long list of foursquare category tree[18]. This fact made foursquare the perfect candidate for generating contextual layers and overlaying them on top of the user reported issues.

As in the previous case, the **target users** are both the citizens and city officials that are looking to further assess the impact of the reported issues and determine their importance/priority based on the information submitted in foursquare. The **data involved** are the citizen-reported issues or initiatives, while the **external datasets** that are used in this case is the check-in information contributed by the users of foursquare. As already mentioned, foursquare offers a large volume of check points characterized with one of the existing categories. These categories include among others schools, theatres, parks, airports, restaurants, and in general highly visited places. Thus, it is evident that knowing the most highly visited places you can automatically infer the issues with the highest impact. These are the **insights** that we are aiming to facilitate with the foursquare layers functionality depicted in Figure 5.16. Through the "Foursquare Layers" functionality the user can search through the extended tree of categories supported by foursquare and plug-in or -out one or more of the foursquare categories. Moreover, additional options like displaying the most highly visited places, may offer even stronger indicators for assessing the impact of every reported issue, i.e. a broken lamp on a street that hosts some of the most visited places in town maybe prioritized compared to a broken lamp in a less popular street. Finally, the **visualization method** that is used to facilitate the decision maker in performing the right kind of impact assessment is based on maps and the **processing** required entails in injecting the foursquare data to the utilized map visualization module. For instance, Figure 5.16 demonstrates the functionality of searching through the foursquare categories for "playgrounds" and overlaying the retrieved venues together with the citizen reported issues. Given that the band condition of playgrounds may be dangerous for small kids, the issues reported in the area may receive the highest priority.

[18] https://developer.foursquare.com/categorytree

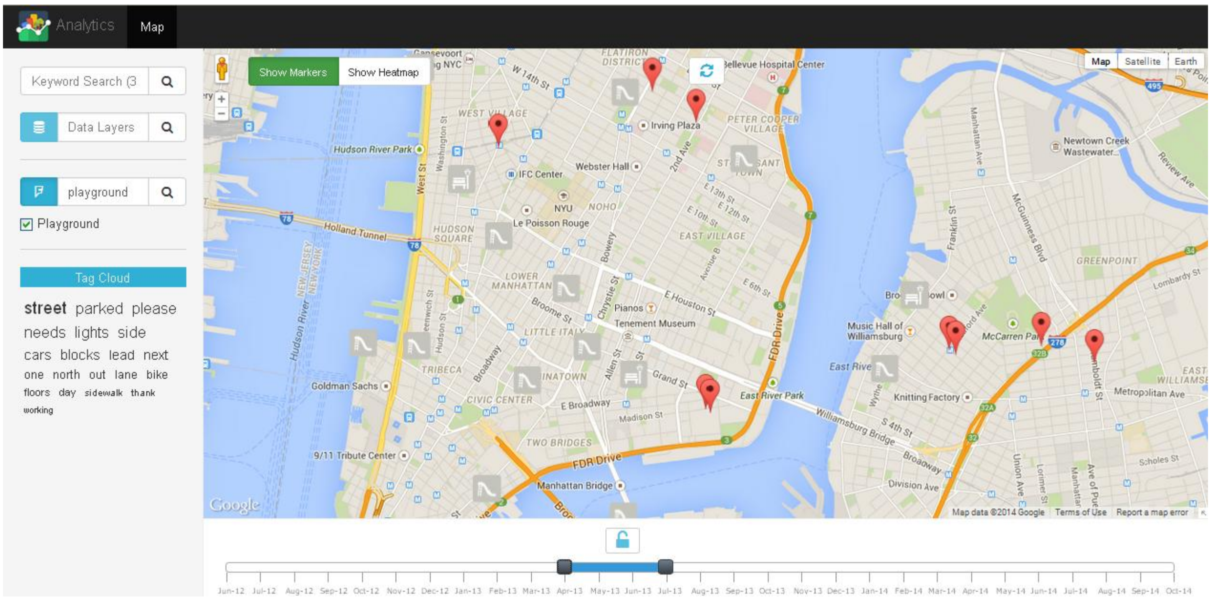

Analytics
Map
Keyword Search (3
Data Layers
playground
☑ Playground
Tag Cloud
street parked please
needs lights side
cars blocks lead next
one north out lane bike
floors day sidewalk thank
working
Show Markers
Show Heatmap
Map
Satellite
Earth
Jun-12 Jul-12 Aug-12 Sep-12 Oct-12 Nov-12 Dec-12 Jan-13 Feb-13 Mar-13 Apr-13 May-13 Jun-13 Jul-13 Aug-13 Sep-13 Oct-13 Nov-13 Dec-13 Jan-14 Feb-14 Mar-14 Apr-14 May-14 Jun-14 Jul-14 Aug-14 Sep-14 Oct-14

Figure 5.16: Foursquare layers displayed simultaneously with user reported issues (http://lganalytics.mklab.iti.gr).

5.2.2.8 Harvest the opinion of social users

Another very intriguing potential for acquiring more context is to harvest the opinion of social users. Given that a non-trivial number of the submitted issues/suggestions/initiatives maybe of general interest, it may be likely that a relevant discussion is taking place among the users of social networks. Thus, it could be an interesting additional source of information if we could spot the relevant discussions and analyze them with respect to their sentiment or prevailing topics. However, the task of spotting a set of discussions that are related to a specific issue/suggestion/initiative has proved particularly challenging mainly due to the following reasons: a) unless the crawling mechanism can be narrowed down to a specific set of user accounts that are likely to share information of potential interest, the information collected through the general purpose APIs is rarely relevant with the intended topic of interest; b) the information context that is included in the description of an issue/suggestion/initiative is usually insufficient to enable the deployment of an efficient text-based analysis algorithm that would succeed in detecting meaningful matches between the description of the issue/suggestion/initiative and the social media discussions. As a direct consequence of the aforementioned reasons, our efforts to rely on the general twitter and facebook APIs for detecting relevant discussions didn't result in usable content.

In overcoming this obstacle and driven also by the contextually-rich nature of foursquare data, we have decided to establish the connection between a certain issue/suggestion/initiative and a set of use contributed discussions in an implicit manner. The foursquare application, apart from checking-in and categorizing a venue, offers also the option of providing "tips" (i.e. opinions) about a certain venue. Thus, if the issue/suggestion/initiative that has been reported by the citizen can be tightly linked to an existing foursquare venue (e.g. geographical proximity, or explicit reference of the venue name), then the "tips" that have been contributed by the foursquare users to characterize this venue can be consider to form a set of discussions related to the initial issue/suggestion/initiative. Based on this implicit interlinking we have been able to harvest the opinion of social users by performing sentiment analysis on the venue “tips”.

Sentiment analysis refers to the process of automatically assessing the positive or negative attitude of a sentence based solely on natural language processing techniques. It is typically used to estimate whether people are talking positively or negatively about a certain topic. In our case, we employed a sentiment analysis algorithm to classify the “tips” contributed for a venue in five different classes, namely *very negative*, *negative*, *neutral*, *positive* and *very positive*. Subsequently, by aggregating the analysed “tips” based on their sentiment in a histogram-like view, we are able to easily obtain an understanding of whether this venue (and thus the issue/suggestion/initiative that has been linked with this venue) is seen positively or negatively by social users. In addition, by allowing the user to go though the entire list of the analyzed “tips” along with their assigned sentiment tags, we further support the decision maker in understanding the cause of this positive or negative attitude.

For instance, by browsing through the submitted issues the decision maker can easily spot a case similar to the one depicted in Figure 5.17, where a number of citizens have reported problems about a park. Indeed, by digging into the reported issues we come across issues of the type “Bags of garbage,mattresses,loose trash”, “Large parties, loud music, and bbq smell”, or “people feeding waterfowl”. However, if the decision maker would like to verify whether this negative feeling about the park is permanent or incidental, he could choose to overlay the foursquare layer related to Playgrounds. Then, by clicking on the corresponding venue icon he can obtain a rough overview about the people’s sentiment with respect to this park. The histogram-like view presented in Figure 5.18 (a) reveals that the opinion of social users is primarily neutral, but leaning more towards negative rather than positive. In addition, the decision maker can further dig into the text of the sentiment-tagged “tips” to discover if there is a common reason that generates this negative attitude towards this park (see Figure 5.18 (b)).

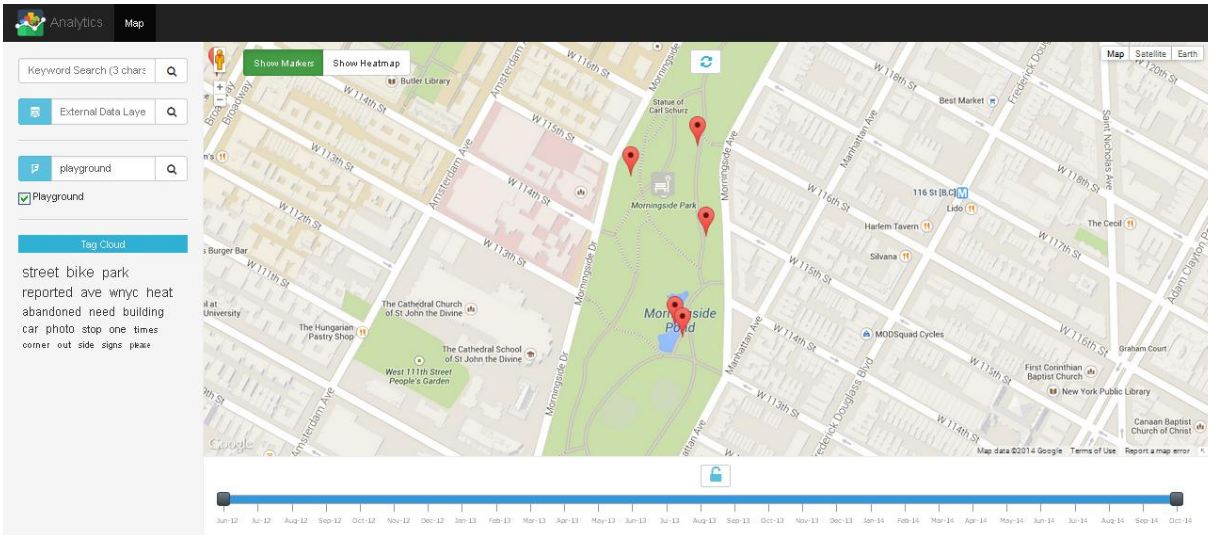
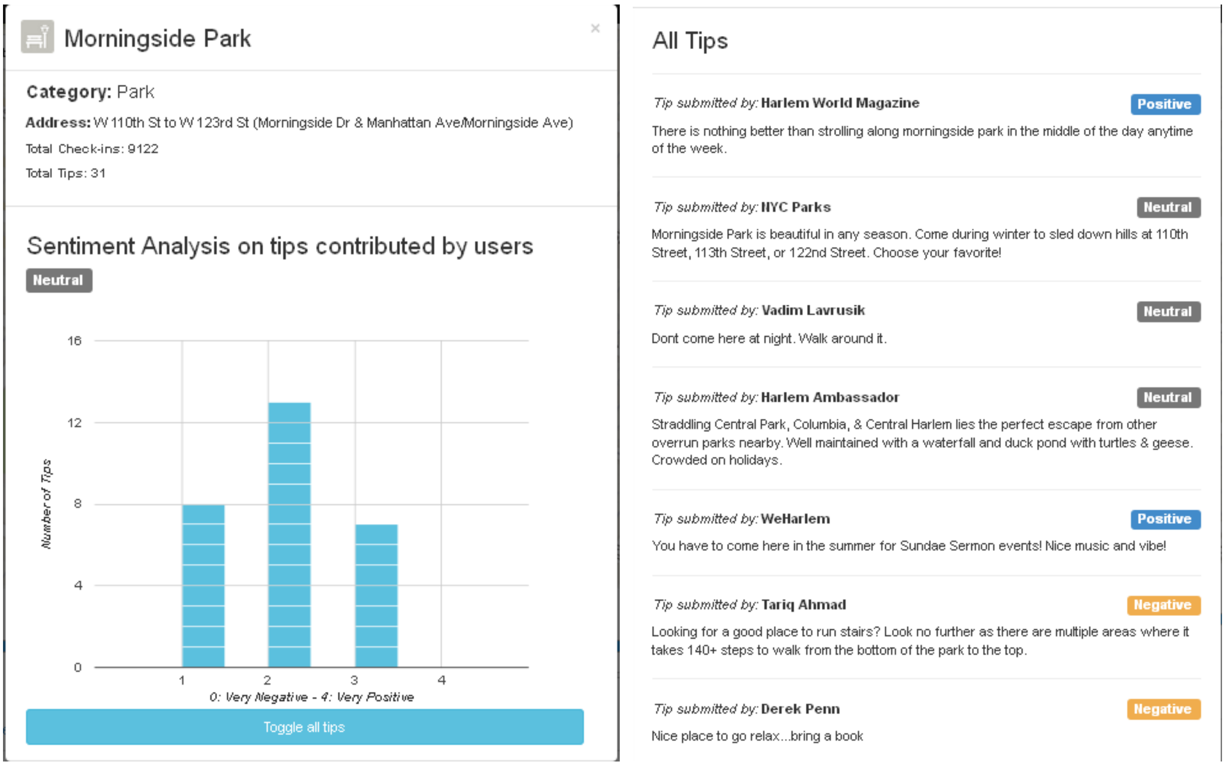


Figure 5.17: Joint visualization of citizen-reported issues and the foursquare layer related to Playgrounds that reveals the connection between the reported issues and the foursquare venue (http://lganalytics.mklab.iti.gr)



Morningside Park

Category: Park

Address: W 110th St to W 123rd St (Morningside Dr & Manhattan Ave/Morningside Ave)

Total Check-ins: 9122

Total Tips: 31

Sentiment Analysis on tips contributed by users

Neutral

All Tips

Tip submitted by: **Harlem World Magazine** Positive

There is nothing better than strolling along morningside park in the middle of the day anytime of the week.

Tip submitted by: **NYC Parks** Neutral

Morningside Park is beautiful in any season. Come during winter to sled down hills at 110th Street, 113th Street, or 122nd Street. Choose your favorite!

Tip submitted by: **Vadim Lavrusik** Neutral

Dont come here at night. Walk around it.

Tip submitted by: **Harlem Ambassador** Neutral

Straddling Central Park, Columbia, & Central Harlem lies the perfect escape from other overrun parks nearby. Well maintained with a waterfall and duck pond with turtles & geese. Crowded on holidays.

Tip submitted by: **WeHarlem** Positive

You have to come here in the summer for Sundae Sermon events! Nice music and vibe!

Tip submitted by: **Tariq Ahmad** Negative

Looking for a good place to run stairs? Look no further as there are multiple areas where it takes 140+ steps to walk from the bottom of the park to the top.

Tip submitted by: **Derek Penn** Negative

Nice place to go relax...bring a book

(a)

Histogram-like view of the sentiment analysis results applied on the "tips"submitted for a venue

(b)

Text of sentiment-tagged "tips" allowing the decision maker to discover whether there is a common reason behind negative or positive fealings

Figure 5.18: The results of sentiment analysis applied on the "tips" submitted for a certain venue (http://lganalytics.mklab.iti.gr)

The **target users** in this case are primarily the decision makers that are interested in harvesting the opinion of social users so as to verify or enhance their understanding about a certain topic. The **data involved** are the citizen reported issues/suggestions/initiatives, while the **external data** in this case is the "tips" provided by users of foursquare. The **intended insights** is the positive or negative feeling of social users with respect to a certain venue (and as a consequence the issues/suggestions/initiatives that are connected with this venue), as well as the discovery of the underlying reasons that maybe generating this positive or negative attitude. Finally, the **visualization method** that has been used consists in a histogram-like aggregation of the "tips" based on their sentiment, while the **processing required** relates to the textual analysis of the contributed "tips" so as to classify them based on their sentiment. The sentiment analysis algorithm presented by Socher in [21] and implemented by the Stanford natural language processing group as part of the Suite of CoreNLP tools [22] has been used in our implementation.

5.2.2.9 Automatic categorization of un-classified issues

A major problem with the data collected through the urban maintenance use case is that a significant number of reported issues have not been assigned to any of the pre-defined categories. They were simply assigned to the default category “others”. In the past, uncategorized issues led to additional workload in the subsequent steps by the city council. The issues had to be forwarded manually to those departments which are responsible for the respective issue category. Therefore, given the existence of a large volume of un-categorized issues, the **insight** that emerged as a research question was the potential of automatically categorizing these issues to one of the “known” categories. The **target user** for this insight is clearly the city officials that would like to remove the additional burden of having to manually process the un-classified issues. The **citizen-data involved** are the reported issues that have not been manually classified to any of the pre-defined categories. However, in this case, the employed approach was not driven by a visualization method facilitating the observation of meaningful patterns. Instead, the use of a machine learning method was employed to mine the category information out of the issue textual data.

In the following, we provide the experiments that have been undertaken to assess the effectiveness of the employed approach. The data set which forms the basis for the following evaluation contains 13,811 reported issues collected by BuitenBeter. Each issue contains a GPS coordinate (inside the border of The Netherlands), a photo, a timestamp, a description text entered by the user, a category like “litter on the street”, and anonymized user data. Figure 5.19 depicts one of these issues where the correct category would be “litter on the street” but the user specified the category “others” instead.

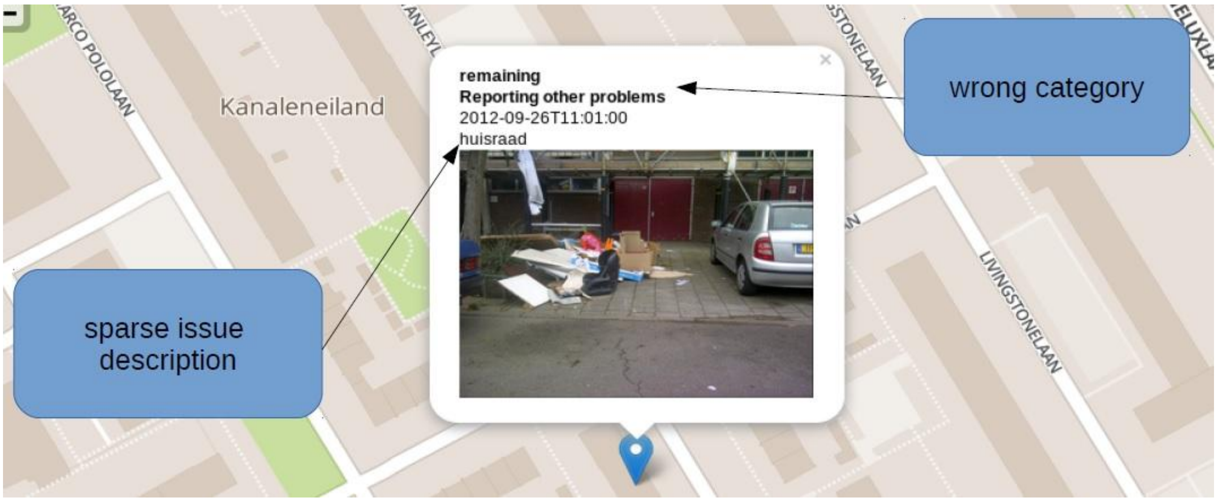


Figure 5.19: Reported issue where the description is very short and the user did not assign the correct category “litter on the street”

Other problems we encountered during the data analysis phase were reports of users which only tested the app (e.g. "proef melding", "test"), reports without description, and reports where the description didn't contain information about the issue (e.g locations "Carolusweg 192" or "zie foto / see picture"). Another problem encountered within the data set is the existence of highly related categories like “bad roads” and “loose paving stone”, “weed” and “Nuisance trees”, or “Litter on the street” and “full container”.

Classification Approach: For an automatic categorization process, a classifier is required which is able to assign issues to one of the categories. Therefore a simple text classifier was trained on the description texts of the reported issues. A variety of approaches comes into consideration when it comes to text classification. Among those approaches are e.g. Decision Trees, Naive Bayes or Maximum Entropy. Another way to deal with text classification is to use the so called Vector Space Model [8]. In this model, natural language, like the issue text, is transferred into a space of vectors by counting word frequencies. This numeric representation enables mathematical calculations like computing the distance between two texts. With the help of the vector space model, some more advanced classifiers like the Rocchio classifier, K-Nearest-Neighbours or Support Vector Machines can be used. These classifiers were used as a baseline for comparison with our own, more problem-specific solution.

Normalized Relevance Distance: Our approach serves to overcome the problem that the issue descriptions, given by the users, are often quite sparse. Thus, training the classifier only on the aggregated problem descriptions would probably not lead to the desired success. Therefore a distance metric called Normalized Relevance Distance (NRD) [9] was used. This distance metric builds upon a method named Explicit Semantic Analysis [7]. ESA is a well-known method to compute the semantic relatedness between two texts where Wikipedia is used as a corpus for indexing. In this approach the vector space, into which the issue description texts are transformed, consists of all articles from Wikipedia. Since the issue descriptions are written in Dutch language, only the Dutch part of Wikipedia is used as the indexing corpus. The weights of a vector representing an individual word is then calculated by counting the occurrences of this word in the respective Wikipedia articles and a whole issue description text again is computed by calculating the centroid of all word vectors of the issue text.

Our experiments have shown that the accuracy of the classifier depends on the right choice of the text corpus on which the background knowledge vector space is build. According to the original ESA approach we started with the Dutch version of Wikipedia. However, it has been found that the issue descriptions itself lead to a higher accuracy in the classification. Thus, the resulting NRD KNearest-Neighbours classifier for classifying an issue in the category "others" is as follows:

```
// Build the background vector space on the issue descriptions.
for each word in the issue data set:
        for each issue in the issue data set:
                compute the tf-idf value.
        store all tf-idf for the word as "word vector"
        in the vector space.

// Build the category vectors.
for each category:
        compute category centroid
        (weighted centroid of N randomly selected word vectors
        occurring in this category)

// Classify an issue.
for each category:
        compute distance (NRD) to the category centroid

assign category with highest similarity (NRD)
```

Some of the k nearest and farthest issues for each category centroid are presented in Table 5.1. The issues which are farthest away from a certain category are most likely to be spam issues, or issues from another category. However, this does not work when a great number of similar spam descriptions, like the term “test”, are contained in one category. These would then falsely be treated as characteristic issue descriptions for this category. In order to prevent this unwanted behaviour, we filter out issues from the category centroid, which have a high similarity to all category centroids. This filters out descriptions like "test" or "proef" provided that these are distributed evenly over all categories.

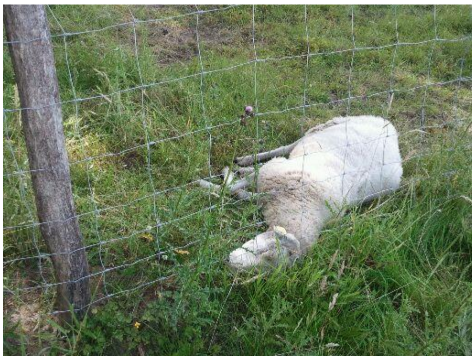
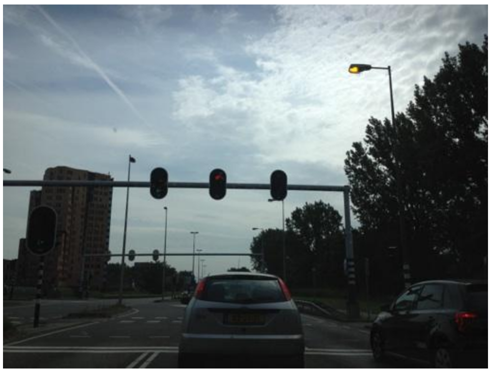
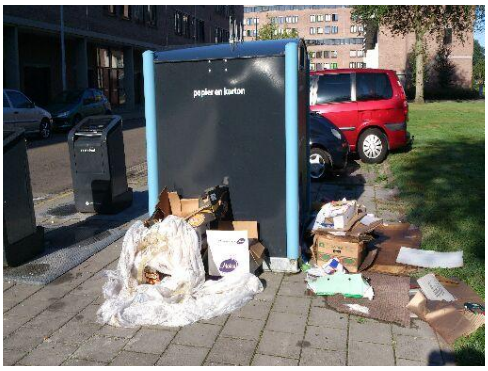
Table 5.1: Examples for issues near and far from the category centroid

Category	**near to centroid**	**far from centroid**
Liter on the street	“Veeeel afval buiten de container? Kan het opgehaald worden?” “Plastic afval is niet opgehaald.” “Dumpen van afval”	“olweversgaarde to 168” “Bla bla” “Het os ear”
Dogshit	“Hondenpoep op het speelveldje” “Hondeigenaars laten hun honden op straat en groen poepen” “Poep op de stoep”	“Dat werkt bij de gemeenthe“ „Bij de school“ “Prullen bak weer in elkaar getrapt Hj janssen”
Loose paving stone	“Losse tegel en gat eronder“ “Er liggen na werkzaamheden nog allemaal losse tegels en een hoop zand.“ „Zou hier maar gekeken kunnen worden diverse tegels op fietspad zitten los“	„fghyyh” “Van no: 101 tot no: 143” “Bla”
Pests	“Dode duif” “Ratten in de tuin” „Wespen nest“	“Theekopje op tafel” “Dode boom vanuit tuin al eerder bericht gestuurd mahlerlaan 9” "Vuilniszakken gedeponeerd waar ongedierte op afkomt“

In Table 5.2 we show some example issues taken from the category “others” and classified with the NRD KNearest-Neighbours classifier.

Table 5.2: Example issue taken from the category “others’ along with their classification scores

Example issue	**NRD to category centroids (smaller distance means more similar)**

 Description: "dode geit in de wei. geit heeft geel oormerk nr 0047"	pests **0,595** broken streetlights 0,601 broken playset 0,603 dogshit 0,617 weed 0,621 Idea wish 0,633 bad roads 0,640 loose paving stone 0,653 Litter on the street 0,663 Graffiti & coverings shall 0,710
 Description: "Rechterverkeerslicht doet het niet"	broken streetlights **0,278** weed 0,366 Litter on the street 0,383 loose paving stone 0,384 pests 0,395 Idea wish 0,401 dogshit 0,422 bad roads 0,422 broken playset 0,473 Graffiti & coverings shall 0,513
 Description: "Vuilnis"	Litter on the street **0,392** Graffiti & coverings shall 0,543 dogshit 0,550 weed 0,558 broken streetlights 0,562 pests 0,571 Idea wish 0,572 bad roads 0,580 loose paving stone 0,589 broken playset 0,596

 Description: "Iemand heeft hier op 10m afstand van speeltuin professioneel draad met SCHEERMESJES aangebracht! Dit is met zoveel kinderen in de buurt onacceptabel en misdadig buiten alle proporties. Graag SPOED laten weghalen en reactie"	broken playset	**0,508**
	Litter on the street	0,522
	loose paving stone	0,529
	Graffiti & coverings shall	0,530
	weed	0,533
	broken streetlights	0,533
	Idea wish	0,534
	dogshit	0,534
	bad roads	0,537
	pests	0,549

We evaluated the categorization performance by computing the recall for each category. The results are shown in Figure 5.20. On the x-axis we took an increasing number of correct user-classified issues to build the representative category centroids. All representative issues were chosen randomly from the data set and not reused in the recall computation. The recall on the y-axis shows the average recall over all categories. Depending of the amount of issues and uniqueness of the used terms in the individual categories, this value varied between 0.98 and 0.69.

The best performance with a recall value of 0.98 is archived in the biggest category “litter on street” which contains 2555 issues. The second best performance with a recall value of 0.92 is archived in the second biggest category “broken streetlights” with 1140 issues in our test set. The worst category “bad roads” with a recall value of 0.69 is not clearly distinct from the category “loos paving stone”. The assertion of missing distinction is also supported by a very small distance between category centroid of both categories. The second worst category “dogshit” with a recall value of 0.72 contains only 184 issues. In this case, a larger data set would be interesting to reduce the weight of outliers.

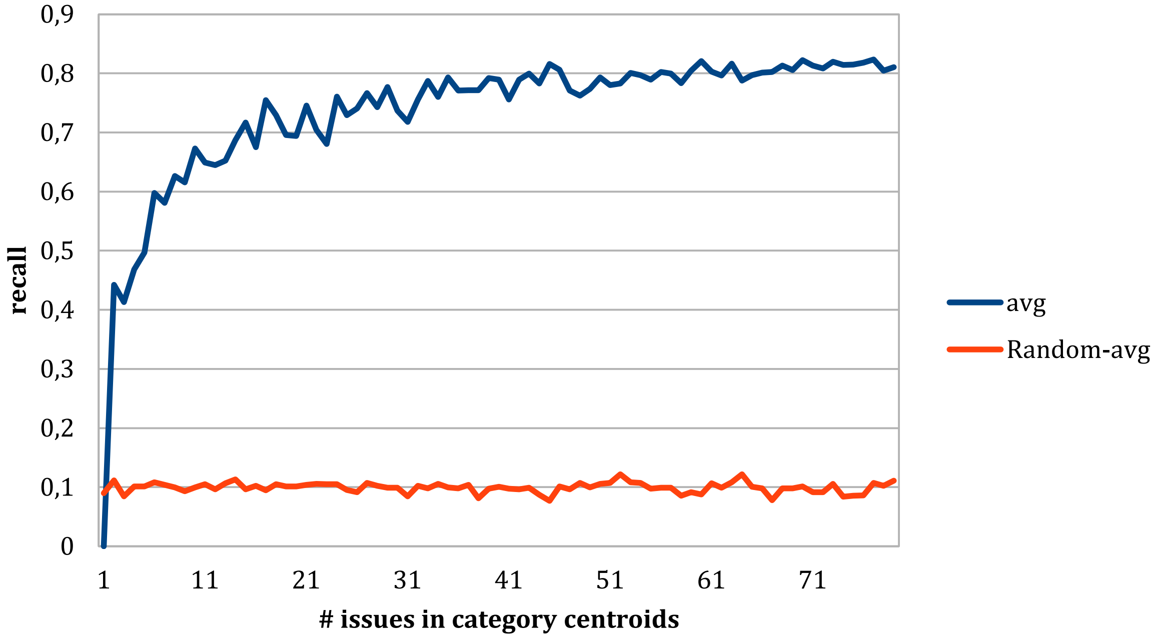


Figure 5.20 Categorization performance of NRD KNearest-Neighbours classifier

5.2.3 Urban Planning

The main aim of the Urban Planning field trial has been primarily the user's participation and therefore the visualization is slightly different than the above mentioned use cases. In this case it is mainly handled in order to provide transparency and to allow citizens to receive the results in real time on their mobile device, while offering a global impression of the results to administrators who see aggregated views on a computer. This way they get an understandable summary of the results of participation and they are also enabled to filter the results and extract further knowledge from the collected information.

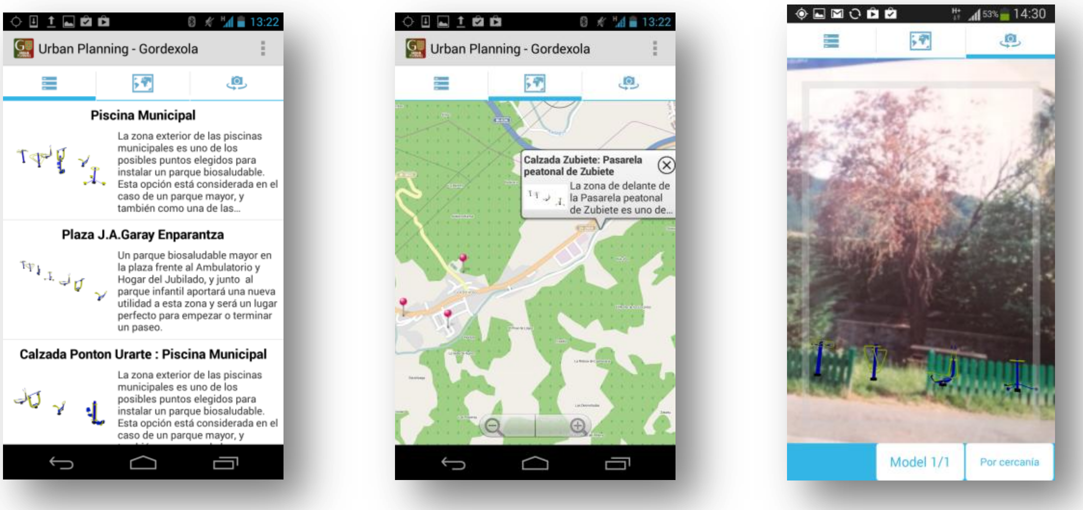
Therefore this establishes a new communication channel for administrators and citizens who can share information related to plans with an important dependency on the visualization of the plans. The mobile app developed for this use case gives information provided by administrators to citizens with the intention to harvest their opinion or preferences about the presented plans. Moreover, the citizens are given the opportunity to view the general results of the opinion that is gathered from the public at any time, thanks to the visualization of the aggregated results on the smart phone. On the other hand, the administrators can also gain access to this information that can be further filtered based on the statistical characteristics of the participants. This is achieved via the web application for visual analytics that has been developed to facilitate the faceted presentation of the results in a desktop computer.

The insights that are planned to be extracted within this field trial are also along the aforementioned tracks. Firstly, the visualization of the current status of the on-going opinion poll will allow citizens to obtain a clear view of the opinion of their neighbours about the presented issues. Secondly, the administrators can get more insights about the collected opinions by grouping the users in various facets. This can allow decision makers to get a better idea of citizens' opinion, and of certain segments of the population, that can help them make more informed decisions.

The Urban Planning use case is composed by a mobile application and a web application, as described in D5.4 [4], and visualization has a key role in both communicating the necessary information, as well as showing the results. In the following we discuss these two cases separately.

5.2.3.1 Communicating plans and feedback to mobile citizens

In the case of the mobile application, the presentation of the plans on the app itself can be considered the first visualization for this use case. The three views presented to the citizens (**target users**), the list view, the map view and the AR view (see Figure 5.21), allow them to easily obtain interesting information about the future plans, such as a description, the location on a map, or the enhanced view using the camera (**data involved**). The **intended insights** here would be a comprehensive understanding of the future projects or plans that is necessary for every citizen to provide his opinion.



13:22

Urban Planning - Gordexola

Piscina Municipal

La zona exterior de las piscinas municipales es uno de los posibles puntos elegidos para instalar un parque biosaludable. Esta opción está considerada en el caso de un parque mayor, y también como una de las...

Plaza J.A.Garay Enparantza

Un parque biosaludable mayor en la plaza frente al Ambulatorio y Hogar del Jubilado, y junto al parque infantil aportará una nueva utilidad a esta zona y será un lugar perfecto para empezar o terminar un paseo.

Calzada Ponton Urarte : Piscina Municipal

La zona exterior de las piscinas municipales es uno de los posibles puntos elegidos para instalar un parque biosaludable. Esta opción está considerada en el caso de un parque mayor, y

13:22

Urban Planning - Gordexola

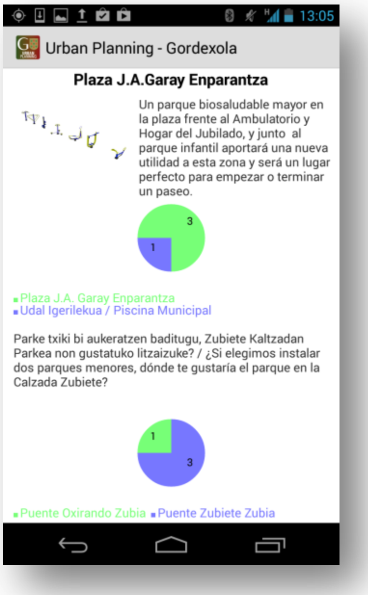
Calzada Zubiete: Pasarela peatonal de Zubiete

La zona de delante de la Pasarela peatonal de Zubiete es uno de...

14:30

Figure 5.21: Visualization of plans in the Urban Planning Gordexola mobile app. List view, map view and AR view.

Then in a second step, after the filled in questionnaire has been submitted by the user, he is given the opportunity to get an overview of the current status of the on-going opinion harvesting campaign. Thus, the **target user** is once again the mobile citizen, while the **intended insight** in this case is the opinion of the group of users who have showed interest in a certain plan and have already provided their opinion. Given that the information is rather general, the **visualization method** that is employed to provide the necessary feedback is pie charts, which display the percentages of opinion for each of the response options, as well as the total numbers for each option (see Figure 5.22). In this case, the **data involved** is the answers to the questionnaire themselves.



13:05

Urban Planning - Gordexola

Plaza J.A.Garay Enparantza

Un parque biosaludable mayor en la plaza frente al Ambulatorio y Hogar del Jubilado, y junto al parque infantil aportará una nueva utilidad a esta zona y será un lugar perfecto para empezar o terminar un paseo.

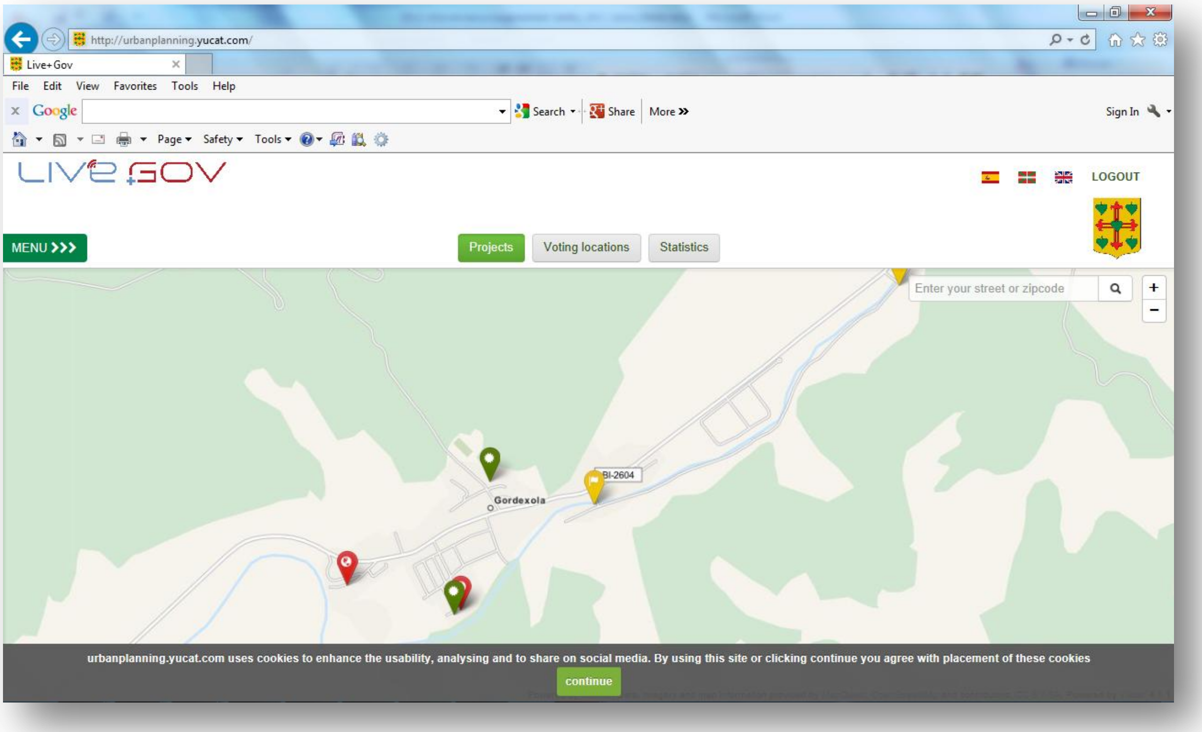
3
1
▪Plaza J.A. Garay Enparantza
▪Udal Igerilekua / Piscina Municipal

Parke txiki bi aukeratzen baditugu, Zubiete Kaltzadan Parkea non gustatuko litzaizuke? / ¿Si elegimos instalar dos parques menores, dónde te gustaría el parque en la Calzada Zubiete?

Figure 5.22: Visualization of the results of participation in the Urban Planning Gordexola mobile app.

5.2.3.2 View citizens' opinion from different perspectives

The web application called Urban Planning Gordexola Reporter was created for the visualization of the results by the decision makers. The goal of this Reporter has been to visualize the same kind of information but in this time filter the displayed data based on the profile information obtained from users. Thus, the **data used** in this case are not only the citizens' opinion but also the information provided to describe their profile, such as their age, gender, area of residence and submission location of the questionnaires. Based on these data, the **visualization methods** that are employed by the Reporter consists in a map-based panel (see Figure 5.23) that offers information about the plans, their location, as well as the spatial distribution of the places from which participants have sent their questionnaires. The intended insight in this case are related to the action of the participation itself, aiming to analyze the patterns of participation.



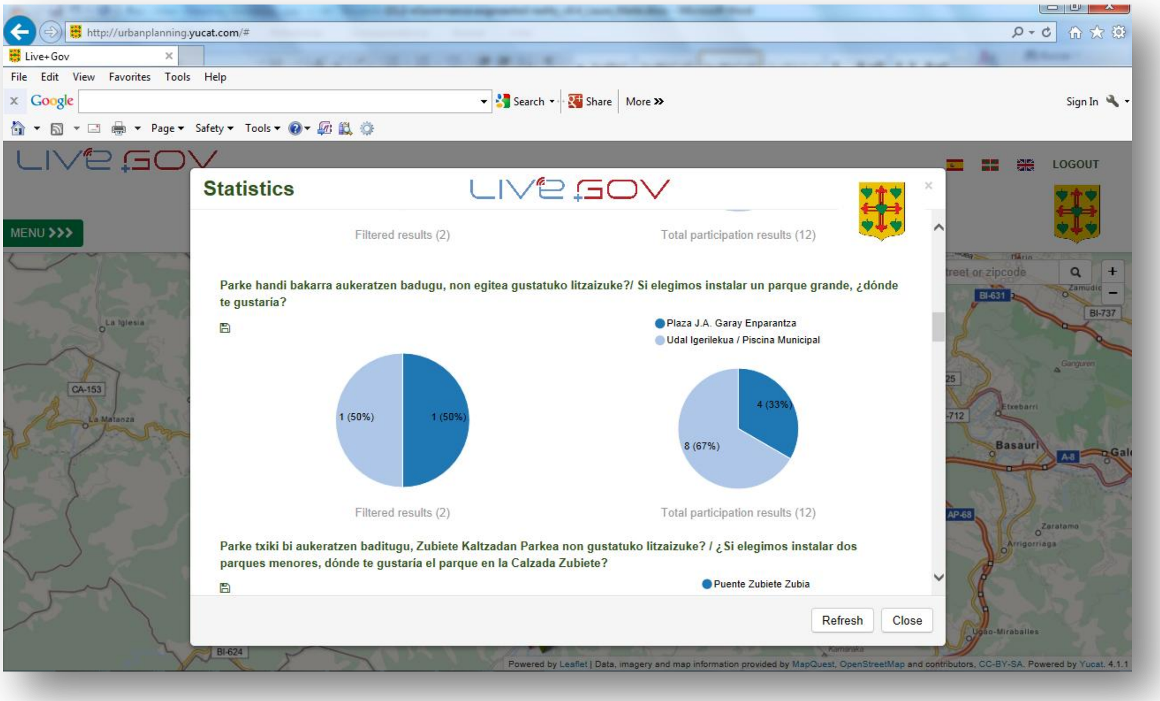
urbanplanning.yucat.com uses cookies to enhance the usability, analysing and to share on social media. By using this site or clicking continue you agree with placement of these cookies

continue

Figure 5.23: Visualization of the location of plans on the Urban Planning Gordexola Reporter web application.

However, the most important **visualization method** of the Reporter is the graph-based statistically-aggregated presentation of the participation results. In this case, the general results are filtered based on the citizens' profile information, allowing the administrators to obtain information about the interest of different groups of citizens. For instance, in the case of "Park", which is the plan presented in the second trial, they are able to study the results collected by age range. For example, this filtering option will allow them to discover whether the construction of a park is particularly important for a certain age group, so as to decide the installation of the park near the Senior Citizen facilities. If the filtered results show that it is considered as favourable or not favourable by this group, the decision makers could take it into account as they would be affected by the plan in a way.

The **citizen data involved** in this case consists of the answers to the questionnaire, which are related to the plans, and the user profile information, which is related to the user and contains demographical information. By modifying the filters and facilitate by a wide variety of faceted visualization (see Figure 5.24 and Figure 5.25), the decision makers can obtain the **insights** related, for instance, to the level of interest or participation by age group, or by gender, or by a certain age group that are residents in a certain area. The ability to produce these visualizations can offer interesting insights when it comes to making a decision. Therefore, this tool allows the administrative users to perform an in depth analysis of the collected opinions based on the different characteristics of the participants. This is accomplished by selecting the filtering options that can best help them to look for the **insights** that may be hidden in the collected information, but cannot be straightforwardly derived from the totals.



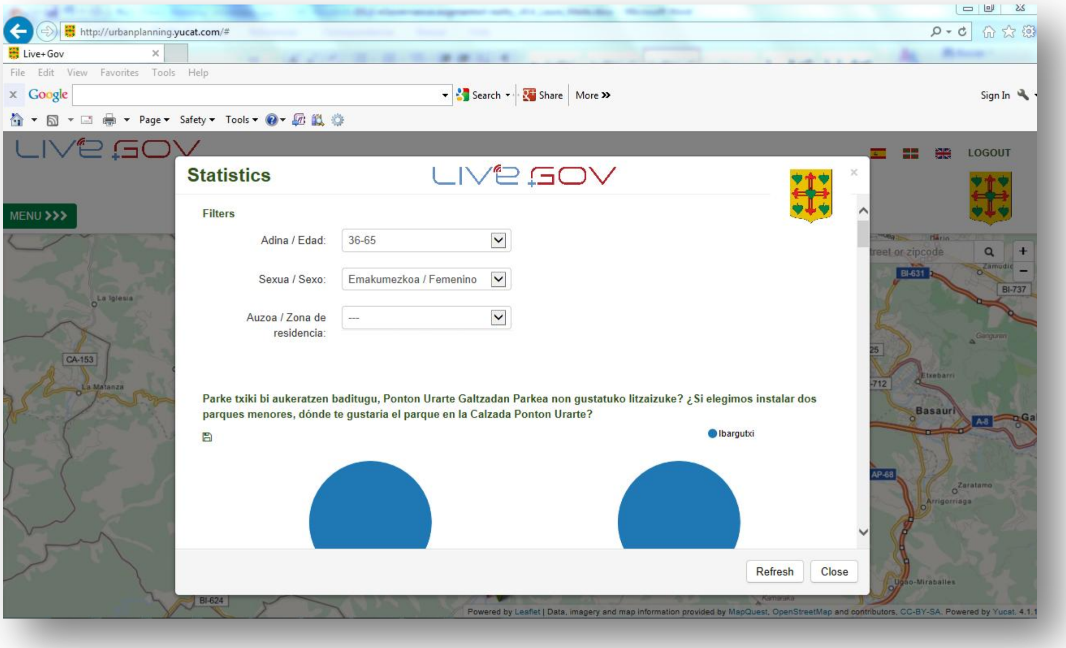
Statistics

Parke handi bakarra aukeratzen badugu, non egitea gustatuko litzaizuke?/ Si elegimos instalar un parque grande, ¿dónde te gustaria?

Parke txiki bi aukeratzen baditugu, Zubiete Kaltzadan Parkea non gustatuko litzaizuke? / ¿Si elegimos instalar dos parques menores, dónde te gustaría el parque en la Calzada Zubiete?

http://urbanplanning.yucat.com/#
Live+Gov
File
Edit
View
Favorites
Tools
Help
Search
Share
More »
Sign In
Page
Safety
Tools
LOGOUT
MENU >>>
Filtered results (2)
Total participation results (12)
● Plaza J.A. Garay Enparantza
● Udal Igerilekua / Piscina Municipal
1 (50%)
1 (50%)
4 (33%)
8 (67%)
Filtered results (2)
Total participation results (12)
● Puente Zubiete Zubia
Refresh
Close
Basauri
Powered by Leaflet | Data, imagery and map information provided by MapQuest, OpenStreetMap and contributors, CC-BY-SA. Powered by Yucat. 4.1.1

Figure 5.24: Visualization of the filtered results of participation by gender and age range to the question about installing a health park in front of the senior citizen facilities.



Statistics

Filters

Adina / Edad: 36-65

Sexua / Sexo: Emakumezkoa / Femenino

Auzoa / Zona de residencia: ---

Parke txiki bi aukeratzen baditugu, Ponton Urarte Galtzadan Parkea non gustatuko litzaizuke? ¿Si elegimos instalar dos parques menores, dónde te gustaria el parque en la Calzada Ponton Urarte?

● Ibargutxi

Figure 5.25: Filtering options for the statistical presentation of results in the Urban Planning Gordexola Reporter web application

6. Summary

In concluding this deliverable we would like to pin-point some of the most important insights that we have gained through the Live+Gov experience.

The first relates to the value of augmented reality as an intriguing approach to engage users. What we have realized from our intensive collaboration with the use case owners is that it is difficult to come across a concrete scenario where the use of augmented reality has a significant advantage over a map-based visualization from the perspective of informativeness. For instance, many were the cases where the potential scenarios for augmented reality were abandoned under the argument of having little to offer compared to a map-based visualization. However, in the cases where we did manage to come across such a scenario (i.e. the 3D visualization of future urban plans, or the augmentation of the citizen's view with the nearby bus stops) the engagement of the users was impressive.

The second important insight is also related to augmented reality but has a technical aspect. The impact of the GPS accuracy in offering a smooth experience for the augmented reality users was largely underestimated. More specifically, in the urban planning scenario we have envisaged the presentation of future plans as 3D objects, embedded in their real environment and viewed from a rather short distance. However, with the accuracy of the GPS system ranging from a few meters to a few dozen of meters it has been particularly difficult to ensure a smooth experience across devices, operating systems and bandwidth conditions. As a result of inaccurate and constantly changing GPS position, the users experienced a jumping behaviour of the presented plans, which acted against their engagement. It was only after making the assumptions mentioned in Section 2.3 that we were able to offer a smooth experience.

However, apart from underestimating the impact of some features over the augmented reality experience, we have also underestimated the potential of the currently existing technologies in making your AR content accessible to a significant number of devices. In this respect, we were positively surprised by the publishing capabilities offered by the major AR vendors in terms of: a) registering your own-served AR channel to their servers, and b) exposing the content of this channel through their widely established mobile browsers for augmented reality. Identifying the great potential of these capabilities we were motivated to contribute in advancing the technological boundaries of current state of the art and develop a solution for automatically transforming existing web sites into AR channels.

Finally, the last point has to do with the repurposing of some tasks from what was originally foreseen in the description of work. This was the case with the personalized content delivery mechanism. Despite the fact that this mechanism was originally intended to facilitate the filtering of content presented through augmented reality, it was eventually employed to learn profiles about the commuting habits of each citizen. Similar was the case with the visualization of the aggregated views and comprehensive summaries where despite the fact that our initial intention was to use these visualizations for addressing the citizens, during the requirement analysis of the use cases it became evident that the majority of these tools would make more sense for the decision makers.

7. References

[1] Nikolopoulos, S., et al., (2013) “Platform and prototype application for augmented reality”, Live+Gov Deliverable D3.1, July 2013

[2] Thiele, F., et al., (2013) “Report on Live+Gov toolkit requirements and architecture”, Live+Gov Deliverable D4.1, July, 2013

[3] Minnigh, P.A., et al., (2014) “Results of first trial and revised requirements”, Live+Gov deliverable D5.3, February 2014.

[4] Minnigh, P.A., et al., (2014) “Prototype / demonstrator for second trials”, Live+Gov Deliverable D5.4, July 2014

[5] Kovats, L., et al., (2014) “Visualization on data injection from mobile sensing”, Live+Gov Deliverable D2.3, January 2014

[6] Thiele, F., et al., (2014) “Report on the Phase 2 Integrated toolset and AIV Test Report”, Live+Gov Deliverable D4.3, October 2014

[7] Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In IJCAI, volume 7, pages 1606–1611.

[8] Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. Communications of the ACM, 18(11):613–620.

[9] Schaefer, C., Hienert, D., and Gottron, T. (2014). Normalized relevance distance – a stable metric for computing semantic relatedness over reference corpora. In ECAI’14: Proceedings of the 21st European Conference on Artificial Intelligence

[10] Meyer, M. (2007), Information Visualisation for scientific discovery. Waterloo: TEDx Talk. Available at http://www.youtube.com/watch?v=Sua0xDCf8MA

[11] Behrens, C. (2008), The form of facts and figures: Design patterns for interactive information visualization. Master’s Thesis, Potsdam University of Applied Sciences.

[12] Kuiper, G.J. (2010), De database als vertelvorm, maar hoe vertel je het verhaal? Blog: http://www.denieuwereporter.nl/2010/02/de-database-als-vertelvorm-maar-hoe-vertel-je-het-verhaal/

[13] McCandless, D., Information is beautiful. Available at http://www.informationisbeautiful.net. Accessed July 24th, 2013.

[14] Davies, J., Map projections. Available at https://www.jasondavies.com/maps/. Accessed July 25th, 2013.

[15] Tufte, E. (2001), *The visual display of quantitative information*. Graphics Press, Cheshire, CT. Second edition.

[16] Tufte, E. (1990), *Envisioning Information*. Graphics Press, Cheshire, CT.

[17] Nagel, H.R. (2006), Scientific Visualization versus information visualization, workshop on state-of-the-art in scientific and parallel computing Umeå, Sweden. PDF from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.164.319&rep=rep1&type=pdf

[18] Ruppert, T., J. Bernard, J. Kohlhammer (2013), Bridging Knowledge gaps in policy analysis with information visualization, Presentation at eGov conference in Koblenz, September 19, 2013.

[19] Ruppert, T. (2013), Bridging Knowledge Gaps in Policy Analysis with Information Visualization, Presentation at the EGOV conference, 19 September 2013, Koblenz

[20] Heer, J., Bostock, M., and Ogievetsky, V., (2010). A tour through the visualization zoo. Commun. ACM 53, 6 (June 2010), 59-67

[21] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng A., and Potts, C., (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Conference on Empirical Methods in Natural Language Processing (EMNLP 2013, Oral).

[22] Manning, Christopher D., Surdeanu, Mihai, Bauer, John, Finkel, Jenny, Bethard, Steven J., and McClosky, David. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55-60.

A Appendix

A.1. Performance of map-based visualisations

The good performance of a geo component is very important for offering a smooth user experience. In particular, when a lot of markers are displayed on the map, the performance of the geo component will suffer. In Live+Gov, we have performed an extensive investigation on the performance of the visualization modules. In this Appendix we report on our findings regarding this investigation.

One of our questions has been: how many markers can be displayed on a map with acceptable performance? Furthermore, several possible performance optimisations were analysed. A distinction between the loading of the application and the performance of the markers when moving around and zooming on the map was made.

A.1.1 Loading performance

At the first time one opens a map-based component, all available data is downloaded and loaded into the front-end. The technique of a single-page application requires this initial load. This has a large advantage: when the user clicks to activate a layer, the application will almost immediately show the markers on the map. However, a drawback is that depending on the amount of data the initial load can take relatively much time. In this section, possible optimisations for (initial) loading performance are described.

Possible Optimisations for loading performance:

- **Activate GZIP on the server:** GZIP will compress the output and makes it much smaller. For example a theme that is around 847kb uncompressed (~1200 features) will be 106kb when gzipped. This will dramatically increase download speed.
- **Activate client-side caching in headers:** This will improve the performance when data is reloaded, or when a visitor visits for the second time (see Table 7.1). When there is cached version available on the client side, the browser will not do an http request.

Table 7.1: Client side caching optimisation for leading performance

Client-side caching	Initial load in seconds	Total features	Features per second	Simultaneous fetching
Disabled	21.754 seconds	4992	240	yes
	32.355 seconds	4992	161	no
Enabled	4.556 seconds	4992	1095	no

- **Implement server-side caching:** Implement server-side caching for data that has not changed. This will dramatically increase average loading time because no sql-queries have to be fired against the database.
- **Display the layers when data is being fetched:** At this time a spinner will show up when data is being fetched from the server. It will go away when all data is fetched. This can be a long delay. The solution could be that users are immediately being presented with the layers, while the application fetches the data in the background. When a user clicks a

theme that is not yet fetched it will set this theme to higher priority, so it will be fetched as soon as possible.

A.1.2 Working performance

The working performance is heavily influenced by the amount of objects that the program needs to display on the map. More DOM-objects means slower performance of the JavaScript application. So the amount of objects should be as low as possible. The interaction with the map begins to feel a bit sluggish at around 400-500 objects. More than 1000 objects makes the application feel slow.

Working performance benchmarks

The working performance is heavily influenced by the amount of objects that the program needs to display on the map (see Table 7.2).

Amounts of objects on map	Activate on map (in milliseconds)
19	9ms
35	16ms
120	43ms
205	70ms
279	110ms
1292	730ms
3970	4670ms

Table 7.2: Working performance benchmarks: amount of objects on map

Possible Optimisations for working performance

- **Cluster the markers:** This way fewer objects are added to the DOM and it will respond quickly. The clustering will increase performance depending on the current zoom level. Clustering a theme that has ~1200 markers, will show 17 objects at the highest zoom-level.
- **Remove objects that are not displayed from the DOM:** Some markers are outside of the viewport. Some improvement in performance will occur by not showing these. The MarkerCluster plugin has implemented this performance improvement. When completely zoomed-in upon a theme, there are only 40-50 objects on the map. In a mid-size zoom (the map shows more & smaller clusters than a few zoom-levels higher) there are around 200-300 objects.
- **Only allow 1 or 2 themes to be activated on the map at the same time:** Because every theme is clustered separately, the performance gain by clustering will be reduced. With one theme activated the maximum average of objects will be 200-300. With two or three themes this will be: 400-500 / 800-900. Three themes activated at the same time has the risk of feeling slow (but still is completely dependent on the amount of markers in there).
- **Cluster all the themes together:** When the clustering is combined, the maximum amount of objects displayed will still be around 200-300, so it will perform very well while having all the themes activated. The disadvantage of this solution is that it could

be less clear what themes are activated because they are behind the same clustering marker (the round marker with a number in it). Of course this could be tackled by using a more informative marker (a little pie chart for example).

Loading and working performance are not the only problems why the user experience decreases when showing a large amount of data on the map. In Figure 7.1 the other problem is shown: there are so many markers on such a small space that you can't see all the markers. Markers overlap and they disperse behind other markers, giving an overall poor user-experience for which the user can quickly lose the overview.

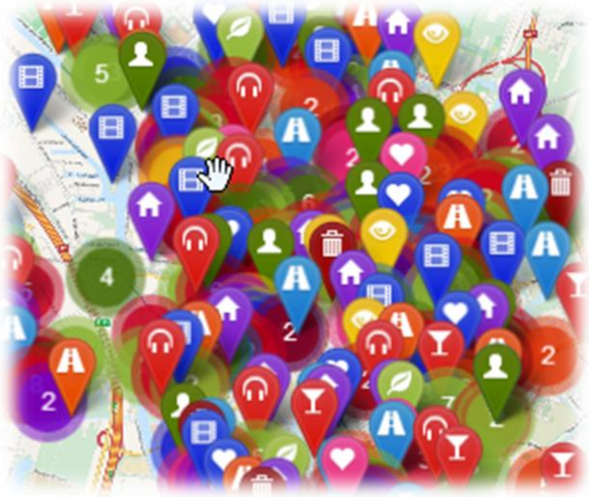


Figure 7.1: A lot of data on small space

A.1.3 Different solutions on the problem of visualising a large amount of data on a map

Several different solutions have been investigated in order to tackle the aforementioned issues regarding visualising larger amounts of data on a map. In this section, we provide an overview of the different solutions.

Do nothing: One option is to do nothing and just show all the available data. This is probably the easiest approach to show data on a map. However the performance of application is closely related to the amount of available data. When there is a large amount of data the application's performance will decrease heavily. Next to that, markers overlap and disperse behind other markers. The user can quickly lose overview when multiple markers are shown on a small space.

Table 7.3: Pros and Cons for adopting the Do nothing approach

Do nothing	
Pro	**Con**

• Easy to implement	• Performance issues when having a large dataset (>500). • Markers can overlaps and dispersing when close to each other. • Have to download all the data, even if we only interested in a small amount of data.

Client-side clustering: The clustering method is used to combine markers that are close to each other and display them as one “marker” as can be seen in Figure 7.2. Users can easily see how many markers there are in a specific area. This makes it for the user easier to see all the data, without markers overlaps and dispersing behind others.

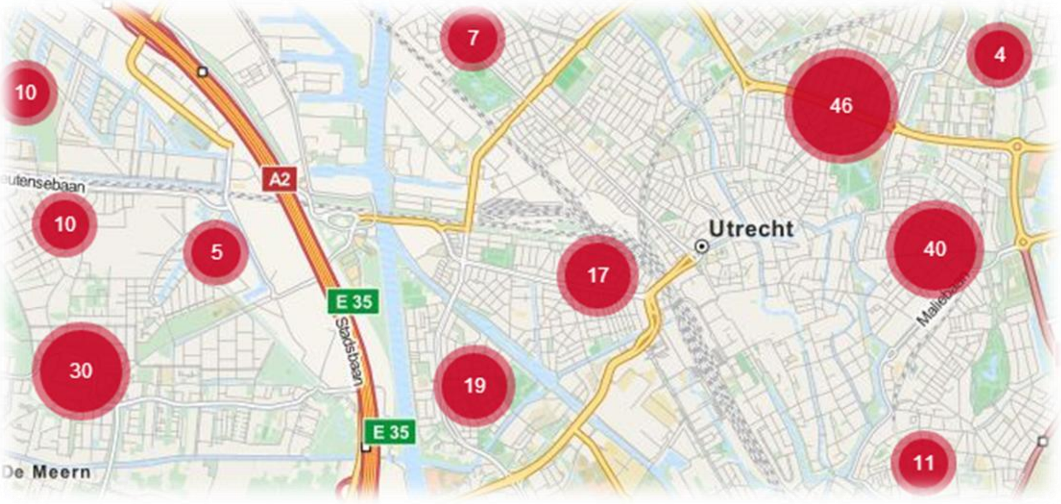


Figure 7.2: Client-side clustering

Second, it can handle more data than the “do nothing”-approach because it does not show every marker. Moreover, since the application does not need to paint every marker, it can save a lot of time when painting all the markers/clusters. Computing those clusters cost time, but this task only has to perform when the data and/or zoom level changes and not, in the case of painting, when the map moves. In most cases computing a cluster is also less compute-power intensive than painting markers on a map.

Although this can reduce the performance problems quite drastically, there are still problems that clustering does not solve. The application still downloads all available data, even if the user only needs a small part of it. Furthermore, all the data is stored in memory. This can be problematic when application does not have a lot of memory available (for example on a mobile phone) or when having a dataset that is larger than the available memory.

Table 7.4: Pros and Cons for Client-side clustering

Client-side clustering	
Pro	**Con**

• Markers do not overlap • Less performance issues, because not all many markers need to be painted individually	• Have to download all the data, even if one is only interested in a small amount of data. • It can use a lot of memory. • Need to calculate the clusters.

Client-side clustering has been tested using the data from the Urban Maintenance Use Case. The difference between clustering and non-clustering is quite large. When clustering is disabled, it took the application 2.12 seconds (see Figure 7.3) to show all the markers on the map. With clustering enabled, it took the application 532 milliseconds to show all clusters (see Figure 7.4). In this case, clustering improves showing all markers on the map by almost four times.

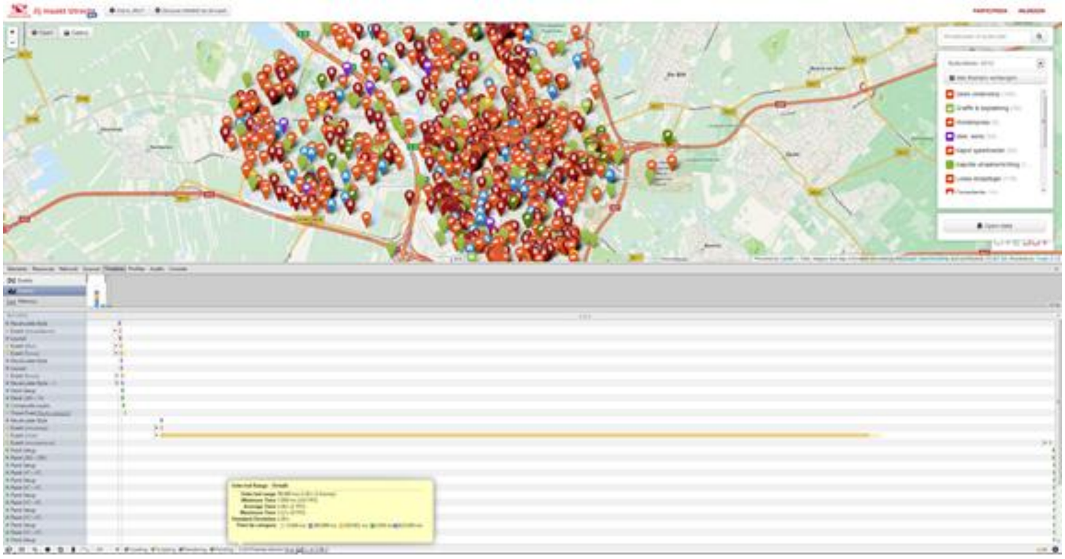


Figure 7.3: No optimisation. Execution time: 2.12 seconds

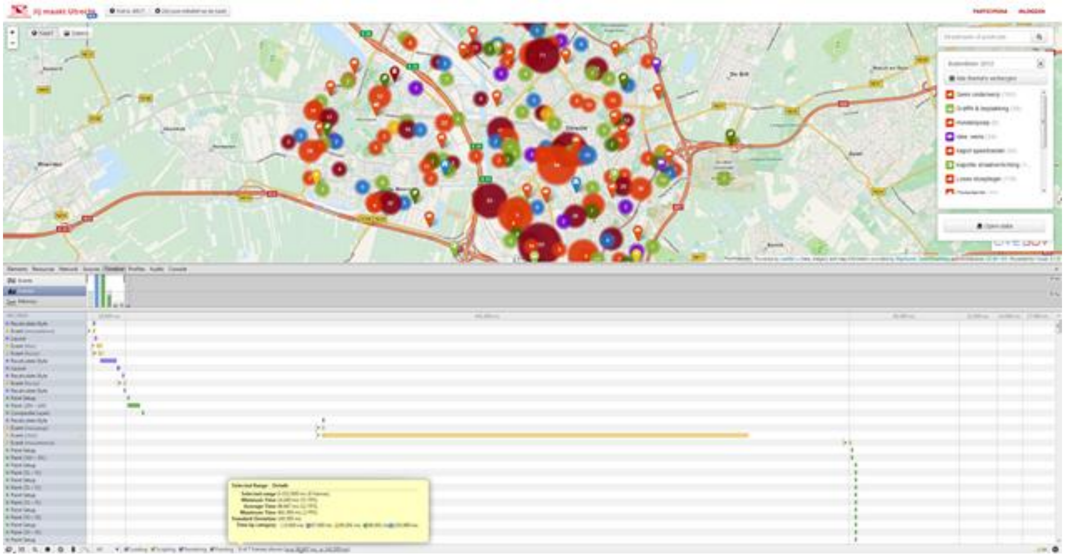


Figure 7.4: Client-side clustering. Execution time: 0.532 seconds

Server-side clustering: The idea of server-side clustering is almost identical to client-side clustering, combining markers that are close to each other. This will save time because the application does not have to paint all objects. The main difference between server-side and client-side clustering is that server-side clustering will be performed on the server while client-side clustering is calculated on the client. When clustering on the server, the server only needs to sends back all the necessary information. This means for example that server only needs to send “3 clusters on a specific location x,y,z with 5000, 1200, 10000 markers” instead of “16200 markers on location x,y,z”. This will have a positive effect on all three cons on the client-side clustering. First, the client does not have to download a lot of data that probably never will be used. Second, the client does not have to calculate the clusters because the server already did this. And finally the client does not use a lot memory because it received less objects from the server.

When using server-side clustering, a connection to the server is always needed. If the application zooms in or out or the application needs new data, it needs to send a request to the server. Because the server only sends information to the client on the current view/zoom level, it has to wait for a response from the server. The performance while zooming is therefore dependent on the performance of the server and the current internet connection. Because the clustering is performed by the server, the server needs to do more work in order to serve the client the requested data.

Server-side clustering	
Pro	**Con**
• Can handle a lot of data (> 1.000.000) • Markers do not overlap • Less performance issues, because not all many markers need to be painted individually • The client does not need to calculate the clusters • Use less memory of the client, compared to client-side clustering. • Client only downloads the necessary data.	• More burden on the server, because clusters are computed there. • The client needs a (stable) internet connection. • Performance is based on the server and internet connection.

Table 7.5: Pros and Cons for Server-side clustering

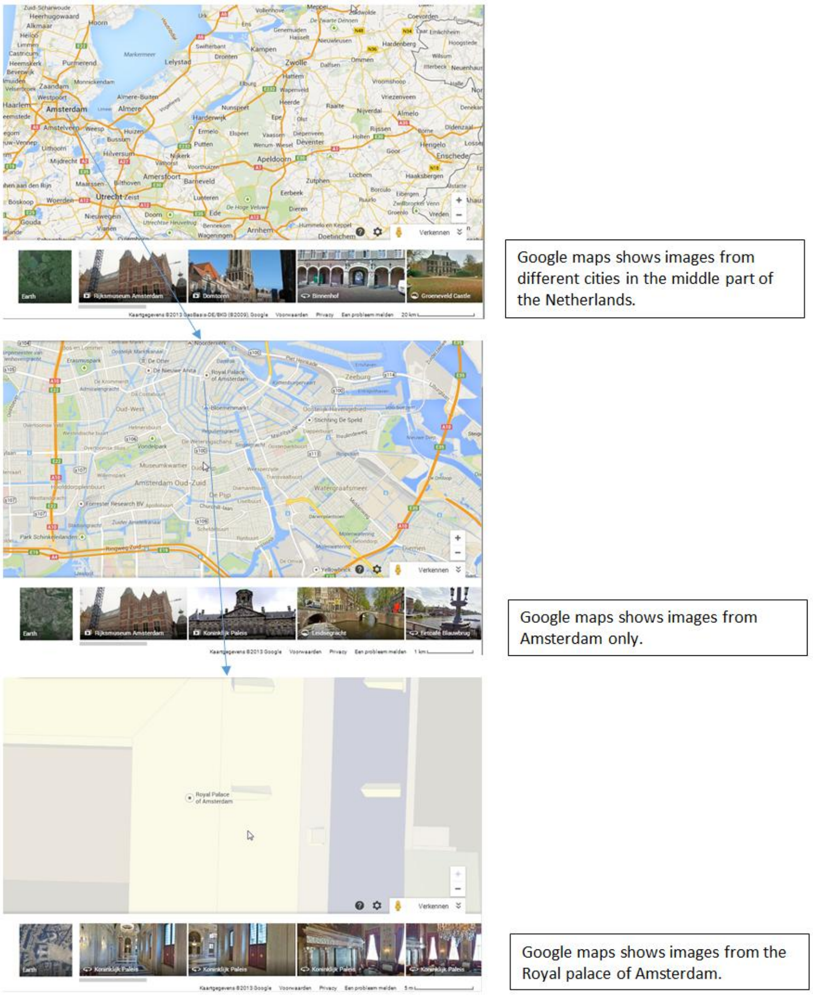
Returning a Subset of data: Returning a subset of the data can dramatically decrease the size of the data needed for download. This subset method can be based on different criteria. Probably the most common criterion is returning a subset based on the view of the user. For example, if the user is zoomed in to the Netherlands, he does not need the data that is located in Belgium, because this data is not used in the current view. This means however that if the user pans to Belgium, the application needs to separately download the subset of Belgium. This idea can also be applied on other levels, like city or even street level. Moreover, this can also be applied to categories or other criteria.

This method can be easily combined with other methods, for example clustering. It can be used to only return the markers/clusters of the current area.

Table 7.6: Pros and Cons for returning a Subset of data

Returning a Subset of data	
Pro	**Con**
• Use less memory, because it uses a subset of all data. • Only downloads the needed data.	• The client needs to request new data more often, e.g. by every panning-action.

Returning Necessary data only: Like the method of returning a subset of data, this method also returns a selective amount of the data. A good example of where this method is used, is Google Maps. When zoomed out, Google only shows a couple of the probably millions of photos taken in the area. Most of the photos will never been shown. When zooming in the principle stays the same, the application only shows a selection of the photos in that area. However, the area getting smaller and smaller and you get more and more photos of the area you want to see (Figure 7.3).



Google maps shows images from different cities in the middle part of the Netherlands.

Google maps shows images from Amsterdam only.

Google maps shows images from the Royal palace of Amsterdam.

Figure 7.5: Example of returning Necessary data only: Google Maps

Returning Necessary data only	
Pro	**Con**
• Use less memory, because it shows not all data • Only downloads small amount of data	• Does not show all the data. • Cannot be used in combination with client-side clustering, because the total number of items is unknown.

Table 7.7: Pros and Cons for returning Necessary data only

Image based layers: Another option is to render an image of the markers/clusters and then send the image to the client. The client only needs to place an overlay over the current map to show the markers/clusters. In most map frameworks this is probably the easiest technique to show clusters/markers.

Table 7.8: Pros and Cons for Image based layers

Image based layers	
Pro	**Con**
• Use less memory, because the client does not need to store all the points in memory • Only downloads necessary images. • Easy to implement in most map frameworks.	• The client does not have any information about markers/clusters. • Need an active (stable) internet connection to download new images

Threshold on the accepted amount of results: In this method the server does not return any results when the amount of result is higher than a predefined threshold. By limiting the amount of results send back, memory problems at the client can be prevented.

Table 7.9: Pros and Cons for Threshold on the accepted amount of results

Threshold on the accepted amount of results	
Pro	**Con**
• Control the amount of data send to the client.	• There is a possibility that the client will not receive any information from the server.

A.1.4 How much data can each method handle?

The exact amount of data each of the methods described above can handle is, hard to pinpoint exactly. There are many different factors that can influence the maximum amount of data. Next to that, user experience cannot be easily grasped into objective measures like maximum amount of data. Some users don't mind waiting longer when loading the page, while others want a faster loading and don't mind if zooming-functionality takes a bit longer. Other factors like offline/online databases, internet connecting and type of device can have large influence on how much data a method can handle.

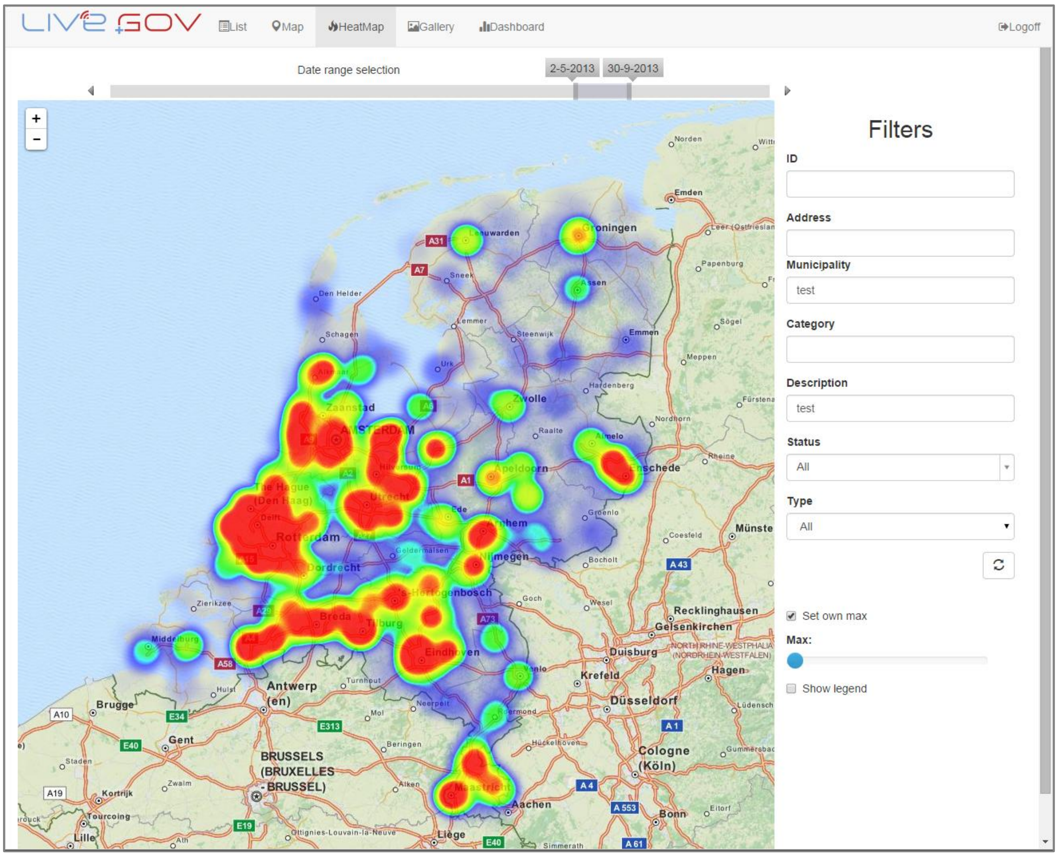
In order to give an idea how much data each method can handle, we categorised the methods in three different groups, based on the user experience on a computer with a good internet connection (see Table 7.10).

Table 7.10: Categorisation on the amount of objects that can be handled for the optimisation methods

Handling of **0 - 2.000 objects**	**Handling of** **2.000 – 50.000 objects**	**Handling of** **50.000+ objects**
Do nothing	Client-side clustering	Server-side clustering
		Image based layers
		Returning Necessary data only

A.1.5 Heat map

Another type of a map-based visualisation are (geographical) heat maps. Heat maps are used in Live+Gov to easily see the density of issue reports on a map in colours (see Figure 7.6). Experiments on heat maps are implemented in both the Urban Maintenance and the Mobility use cases.



Date range selection 2-5-2013 30-9-2013

Filters

ID

Address

Municipality: test

Category

Description: test

Status: All

Type: All

☑ Set own max

Max:

□ Show legend

Figure 7.6: Heat map on test data of issue reports

Because the goal of a heat map is quite specific, namely see the density of issue reports and not all individual issue reports, aggregation of the information for the visualisation itself has other aspects than described above for topographic maps. Insights on these aspects are reported in this section.

Relative character of heat maps: Heat maps in general have a relative character. The heat concentrations are relative to the area shown on the screen. This means that the legend can change as the zoom level on the map changes or a user pans to a different area on the map. Red areas on the map are always relative to the other areas. For the Netherlands as a whole, a red area can mean >10.000 issues, whereas in a smaller city a red area can represent >5 issues (see Figure 7.7).

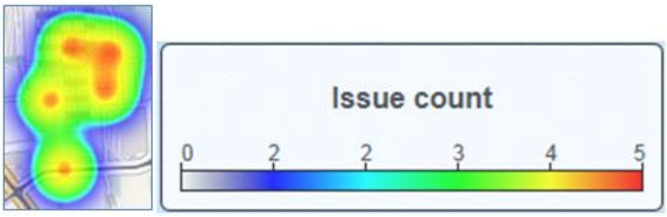


Figure 7.7: Heat map legend on small area

A related issue to this is the influence of certain places on the rest of the map. A heat map shows relative density. If a lot of issues are made in one place, this will make this place red, whereas the rest of the map is empty.

Performance issues: Both on server side and client side, performance issues arise with large amounts of data (>5,000 issues). The JSON parser on the server could handle up to 5,000 issues. With a larger amount of issues the application crashed with a thrown exception. Next to that, the computer itself had a hard time to process all issues in a heat map.

To solve performance issues, the possibilities to use clustering techniques in creating a heat map have been researched. However, this would quickly lead to the situation that there was not enough data to create a nice heat map. With some adaptations to the cluster thresholds, this could be solved.

Render issues: Making a heat map on top of a geographic map requires that the concentrations are smoothly shown to a user. However, with the chosen heat map plugin, there is a render bug in Leaflet. The issue with this is that rendering sometimes is cut off at the edges of a tile on the map. As a heatmap presentation is always relative on an area for multiple dots, instead of a single dot on one exact location, tiling techniques have to be used for heatmap aggregations.

B Appendix

B.1. Map visualisations on mobile devices

B.1.1 Comparative study

For the Urban Planning Use Case and Urban Maintenance Use Case it might be required to have a ‘Geographic information system’-module (GIS-module) available. Research has been done on possibilities for GIS modules for mobile devices. In this investigation, the following aspects have been analysed for the different options, namely Bing maps, Google maps, Osmdroid and Mapsforge:

- Licence: Are we allowed to use the library in our application?
- Flexibility/ Usability: a) How easy is the library to use in an application?, b) Are there any limitations in using the library? and c) Is it possible to make changes in the library (if needed) before we can use it?
- Online/offline mapdata: Type of mapdata

For this research, we have investigated different known map data sources which can be used on a mobile platform. In Table 7.10 we summarize our findings:

Feature	Bing maps	Google maps	Osmdroid	Mapsforge
Licence	http://bingmapsandroidsdk.codeplex.com/license	No real license, however there is a Terms of Service which includes a licence paragraph: https://developers.google.com/maps/terms	GNU Lesser GPL http://www.gnu.org/licenses/lgpl.html	GNU Lesser GPL http://www.gnu.org/licenses/lgpl.html
Are we allowed to user the library in our application?	Yes, this SDK is developed under the Microsoft Public License (Ms-PL). However we are required to have a BingMaps key.	Yes, however a Google Maps Android API key (v2) is required. https://developers.google.com/maps/documentation/android/start#the_google_maps_api_key	Yes	Yes
Flexibility/Usability				
How easy is the library to use in an application	The BingMapView is created as a component in a way that it can be added anywhere in an application.	The MapView is dependent on a MapFragment (or SupportMapFragment). The use of fragments complicates the implementation a little; however it should still be doable.	The MapView is created as a component in a way that it can be added anywhere in an application.	The implementation is similar to the Google Maps API v1 sdk. The use of a MapActivity is required, which limits the implementation.

Are there any limitations in using the library?	No.	Other than the usage of fragments, none.	No	The use of an Activity limits the use to a fullscreen application or a tabbed view.
Is it possible to make changes in the library (if needed) before we can use it?	Yes, the source is available.	No, the SDK is closed source	Yes, the source is available	Yes, the source is available.
Online & offline map data				
Online/offline mapdata	Online only	Online only	Both	Technically both
Online	Yes	Yes	Required tiles are downloaded over the air if a connection is available.	It is possible to download the .map files over the air. These files can however be rather large (Germany is 800MB!!).
Offline	No	No	The downloaded tiles are stored on the device.	After the .map files are downloaded the application map data can be used offline.
Type of map data				
Type of map data	Apparently a combination of raster tiles and vector graphic. Could not find a solid reference for confirmation	Vector graphics	Tiles only	Vector graphics

Satellite images	Yes	Yes	If a source hosting the images is available it can be added.	This is possible if a .map file containing satellite data is available.
Bonus				
Shape support (e.g. point/polygon/multipoint)	Yes, this library has native support for adding features.	Yes, this library has native support for adding features.	No native shape support	An overlay API is available
Geocoding	Yes, this library uses the Bing geocoding API.	Yes, this library uses the Google geocoding API.	No native geocoding support. There is a 'bonus pack' available for this library which used a Nominatim based service. According to the creator of the bonus pack Nominatim is suboptimal.	There is no native geocoding support.

Table 7.11: Feature summary of different map data providers for Android

In Table 7.12 we perform a pro/con analysis of the different map libraries that we have investigated:

Table 7.12: Pro-Con analysis for different map libraries

Client/library	PRO	CON
Bing Maps	• Vector graphics for smooth transition • Satellite images • Support for shapes • Geocoding	• Online only • No caching • Required API key • SDK has some problems on Android version 3.x and 4.x
Google Maps	• Vector graphics for smooth transition • Satellite images • Support for shapes • Geocoding	• Online only • No caching • Requires API key • No source is available
Osmdroid	• Only download tiles for area which are visited using the map viewer • Offline caching • Support for multiple sources of mapdata	• Requires internet connection for initial download of map tiles • No support for shapes • No native geocoding api
MapsForge	• Vector graphics • Full offline viewer • Has an overlay API for adding shapes	• Required .map files containing a lot of map data which the user might not even need. • No geocoding api • Performance is lacking when the map is zoomed out a lot

B.1.2 Making a decision

Each of the aforementioned options have their own *pro and cons*. The pivot point is the requirement of offline caching.

Offline: When offline caching of map data is required the choice is between Osmdroid, Mapsforge and the GIS-module developed at Yucat. Between these three implementations the Osmdroid is the preferedone. This decision is based on the following feature points; implementation of offline storage and performance.

Offline storage: Mapsforge requires the user to download big mapfiles containing e.g. a whole country or the whole of Europe. Considering the fact that the mapfiles per country can range from 1MB up to 800MB, and automatic switching between country files is not supported, this will cause an inconvenience for the enduser. Osmdroid and the Yucat GIS-module only download the area which the user visits using the mapviewer. The Yucat GIS-module has functionality for batchdownloading map tiles.

Performance: Performance between the Mapsforge and Osmdroid implementation is at an equal level when Osmdroid has the required tiles downloaded. Between the both of them

Osmdroid has a more stable performance on all zoomlevels, Mapsforge responses a bit slower at lower zoomlevels (more zoomed out). The Yucat GIS-module clearly loses out on performance; it also has some issues with drawing the map.

Online: If offline caching is not required the choice would be between BingMaps, Google Maps and Osmdroid. BingMaps would not be a recommended choice because of the problem it has on Android 3.x and 4.x implementations. In the choice between Google Maps and Osmdroid the winner would be Google Maps, as the offline caching of Osmdroid would be just a bonus. Google Maps also has native support for displaying shapes and the use of geocoding.

In a nutshell, for offline requirements Osmdroid has our preference. If offline usage is not required, Google Maps comes on top.

C Appendix

C.1. Performance of gallery views

C.1.1 Initial research on gallery performance

As for gallery views, performance is a crucial topic. Images in the gallery view need to be loaded and this puts a heavier load on the internet connections than loading only the markers on a map. Extensive investigation on gallery performances has been conducted. In this appendix we report on this investigation, where the Urban Maintenance use case Webapplication of jijmaaktutrecht.nl has been used for testing.

Jij Maakt Utrecht has an image gallery where the user can view images of selected layers. The gallery shows images of the initiatives and the title of the initiative. When the user clicks on an image, the website opens a popup with more information about the initiative.The performance test is mainly focused on how the performance influences the user experience. The main question for this investigation is: At how many initiatives (with images) is the user experience in the gallery-view affected by performance issues? A second question is: What is the limit of initiatives before the user experience starts to degrade considerably?

Research setup

For this test we used two different *Jij Maakt Utrecht* servers. The first server is the production server that can be found on www.jijmaaktutrecht.nl (as part of the Live+Gov Urban Maintenance Use Case). Because it is a live server, it is not possible to change the data on the server for these tests. In order to make this possible, a copy of the complete production server is made. This server runs on a local development server. Because the development server is running locally it does not connect over the internet. This makes the local server probably a lot faster with downloading the images. For this reason we are using a complete copy of the production server, so we can do the same tests and compare those tests.

To make the comparison more real from the user point of view, three different browsers are used for testing, namely Chrome, Firefox and Internet Explorer 11 on a Windows 7 computer. The computer is iMac from Mid-2011 and has an I7 cpu with 8 GB of RAM. In Chrome and Internet Explorer 11 we used the development-tools that are already available in the browser. In Firefox we used the third-party plugin called Firebug. The internet connection to the production server has a download speed of around the 6 Mbps.

Initial results of the quick scan on performance of gallery views

The following tests were performed on the actual production server of *Jij Maakt Utrecht*. First, a test with a high ping and slow download speed. Second, in order get more real-life results, a test with the internet ping and download speed at normal levels is performed.

Preconditions for the tests:

- Website is loaded completely. So it will only load in the images for this test.
- All features are already loaded.
- Client-side caching is turned off unless mentioned otherwise.

An overview of the results is presented in Table 7.13. If the user loads a layer in "Jijmaaktutrecht.nl" with a lot of initiatives, like the 1059 initiatives of the "Leefbaarheidsinitiatief 2012"-layer, browsers can behave sluggish. Especially when the browser is still loading the page and the user scrolls down to the end of the gallery. All of the tested browsers have some trouble and become sluggish. Once Chrome and Firefox have loaded all the images, they behave normal again. However, IE 11 still behaves slow and sluggish. IE 11 will take about a second to react on the user input to scroll the page. As expected, performance decreases even further when the amount of initiatives increases. With 2000 initiatives in a layer, IE 11 it becomes almost impossible to work with it. It takes a couple a seconds before it even reacts on a scroll input. When IE 11 finally reacts on the user input, it just jumps to the position the user scrolled too: IE doesn't show an "animation" when it scrolls. Chrome and Firefox only had trouble with loading the initiatives. Once loaded, the scrolling was smooth again. The dev-tool of Chrome however had some trouble and freezes a couple of times, but this did not affect the webpage. Firebug in Firefox was not able to show all the information because it has a limit on how much debug-information it can store and show.

Initial findings on the performance of gallery views

On the *Jij Maakt Utrecht* production server it takes around 32 seconds to download 452 photos. The download time increases linearly. This means that it probably takes around one minute to download 1000 photos and 2 minutes to download 2000 photos. However, 32 seconds is already on the limit of acceptable for a user. More than one minute waiting time is assumed to be unacceptable for the user.

Not only is the download time an issue. The browser responsiveness is also a problem with a lot of images in the gallery. Firefox and Chrome behave sluggish when loading for the first time. However, scrolling is smooth when the images are fully loaded. Even when having 2000 images in the gallery, scrolling is smooth. IE 11 however is a complete other story and behaves sluggish, even when images are fully loaded. Scrolling the gallery with 1000 images makes the gallery slow and react for about a second later. When having a gallery with 2000 images, it is almost impossible to work with it. It takes a couple a seconds before it even reacts on a scroll input.

Test	Test 1 (high ping and slow download speed)	Test 2 (regular ping and download speed)
Ping to Google.nl	- Between 300 and 500 milliseconds	- Between 23 and 26 milliseconds.
Total loading time of all thumbnails (48 thumbnails)	- Firefox: 4,4 seconds - Longest blocking time: 3,99 seconds	- Firefox: 1,5 seconds - Longest blocking time: 1,2 seconds
Loading of “Leefbaarheidsinitiatief 2012” (total of 1059 initiatives) *Note: None of these initiatives have images. He will load a “no image”-image for all the initiatives.*	- Firefox: 324 milliseconds - Longest blocking time: none	- Firefox: 35 milliseconds - Longest blocking time: none
Loading of “beheer door bewoners” (total of 464 initiatives)	- Firefox: 45,1 seconds ○ Longest blocking time: 35,7 seconds ○ Amount of MBs to load: 20,8 MB - Chrome: ~41 seconds ○ Amount of MBs to load: 20,9 MB - IE 11: ~45 seconds ○ Amount of MBs to load: 20,9 MB (21.910.513 bytes)	- Firefox: 32,4 seconds ○ Longest blocking time: 27,2 seconds ○ Amount of MBs to load: 20,8 MB - Chrome: ~32 seconds ○ Amount of MBs to load: 20,9 MB - IE 11: ~32 seconds ○ Amount of MBs to load: 20,9 MB (21.910.513 bytes
Total loading time of all thumbnails out of the cache (html-code: 304) (451 thumbnails)	- Firefox: 12,8 seconds ○ Longest blocking time: 10,2 seconds ○ Amount of MBs to load: 0 MB - Chrome: ~20 seconds ○ Amount of MBs to load: 0 MB	- Firefox: 3,1 seconds ○ Longest blocking time: 1,8 seconds ○ Amount of MBs to load: 0 MB - Chrome: ~4 seconds ○ Amount of MBs to load: 0 MB

Table 7.13: Test results of gallery performance (high ping and slow download speed VS regular ping and download speed)

C.1.2 Follow up research on performance of gallery views

Following our initial investigation, we have expanded our previous research of the *Jij Maakt Utrecht* photo gallery. In the gallery the user can view images of selected layers. The gallery shows all initiatives with images of the initiatives, when available. When the user clicks on images, the website opens a popup with more information about the initiative.

There are two problems that would make the *Jij Maakt Utrecht* implementation unusable when increasing the amount images in the gallery. The first problem is the download time of the images. It probably takes around one minute on a 'normal' internet connection to download 1000 images. When loading them all at the same time, the browser starts to become sluggish. When the users scrolls all the way down, the user have to wait till all the images are downloaded in order to see the last images.

The second problem is that browsers do not like scrolling when having a large DOM with a lot of images. The browsers become sluggish and react slowly on user scroll inputs. Some browsers like IE 11 having this problem faster than others like Chrome, but eventually when having a large enough DOM, all browsers have this problem.

Infinite scrolling

Infinite scroll has been called autopagerize, unpaginate, endless pages. But essentially it is pre-fetching content from a subsequent page and adding it directly to the user's current page. Infinite scrolling is a technique that's relative new and is used to replace "page navigation". Websites like Twitter, Pinterest and Facebook are currently using infinite scrolling.

Memory problems: As reported in the initial research, browsers can behave sluggish with a lot of initiatives. With 2000 initiatives, IE 11 even becomes almost impossible to work with. This problem will not be solved by using infinite scrolling. Like the current implementation, infinite scrolling just adds the pre-fetched images to the bottom of the page. This will increase the amount of images on current page when there is a new pre-fetched page added. Every time new images are added, the DOM size increases and the browsers become more and more sluggish.

We have seen this memory problem also on some popular websites like Pinterest. When the user scrolls down and activate the infinite scrolling for a couple of times, the DOM size and the amount of images increases rapidly.

Pinterest: The Pinterest website makes use of the infinite scrolling technique. When the user is scrolling down to the bottom of the page, the website will load some new content and add it to the bottom. Figure 7.8 shows the memory uses and the amount of Nodes in the DOM when scrolling through the Pinterest website. The light-blue filled area 2D chart shows the memory uses over time of the Pinterest website. The green line shows the amount of nodes in the DOM. When recording the information of Figure 7.8 the website scrolled a couple of time down to load in new content using infinite scrolling.

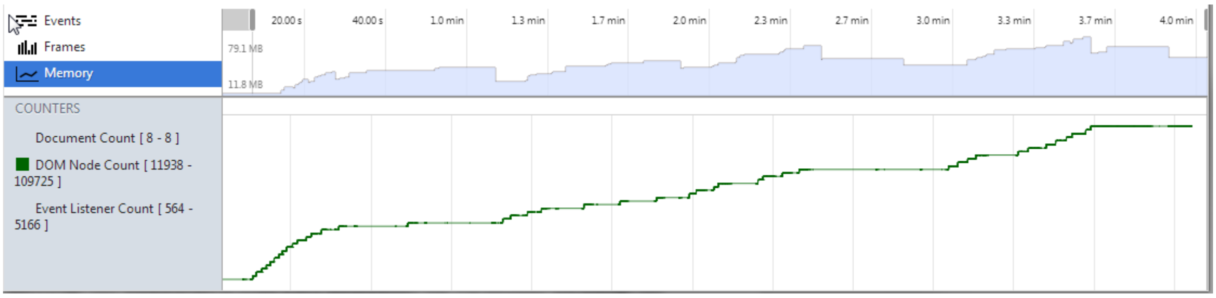


Figure 7.8: Pinterest: Memory uses and DOM node Count

When looking at Figure 7.8 it is clear that if you keep scrolling, the amount Nodes only increases. This means that the data currently outside the view is not removed and still existed in the DOM. Also the memory uses increases slowly over time. When scrolling down a couple of times, the Pinterest website showed comparable behaviour as seen before at *Jij Maakt Utrecht*. The scrolling became sluggish at around 1.000 images. On IE 11 it took around a second to react on the user input to scroll the page. Chrome and Firefox where not as that bad as IE 11, but it was clearly more sluggish then as the website just loaded in.

Heap: When comparing the heap snapshots of the start and end of this measure session it gives the same results as Figure 7.8. The heap size increases from 11.4 MB to 52.5 MB. The amount of IMG-elements in the DOM also increases from 127 to 1203 IMG-elements. For example, the DIV-elements increased from 567 to 5287.

Other websites: Not only Pinterest has the problem of the increased amount of Nodes in the DOM. The DOM-size of other websites like Twitter, Google+ and Instagram also increases when scrolling. Figure 7.9, Figure 7.10 and Figure 7.11 are the results when scrolling on Google+, Instagram and Twitter.

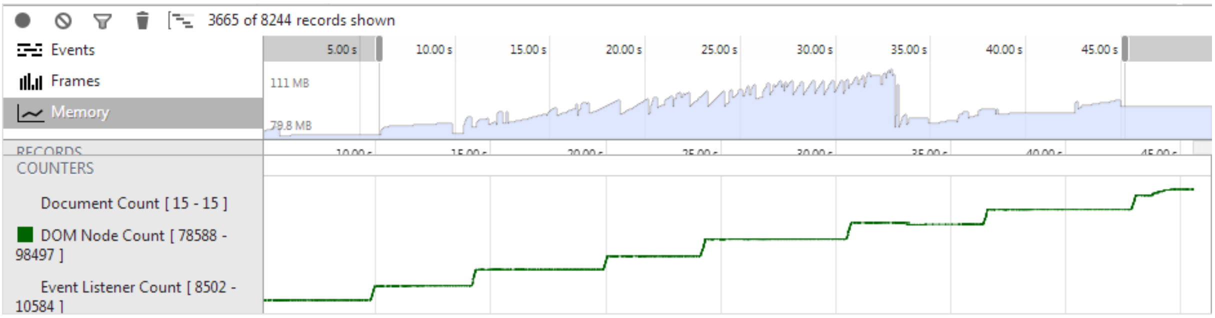

3665 of 8244 records shown
Events
Frames
Memory
5.00 s
10.00 s
15.00 s
20.00 s
25.00 s
30.00 s
35.00 s
40.00 s
45.00 s
111 MB
79.8 MB
RECORDS
COUNTERS
Document Count [15 - 15]
🟩 DOM Node Count [78588 - 98497]
Event Listener Count [8502 - 10584]

Figure 7.9: Google+: Memory uses and DOM node Count

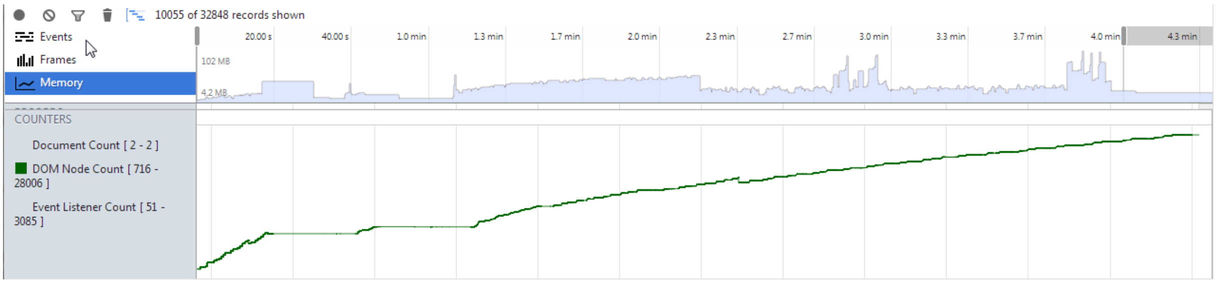


Figure 7.10: Instagram: Memory uses and DOM node Count

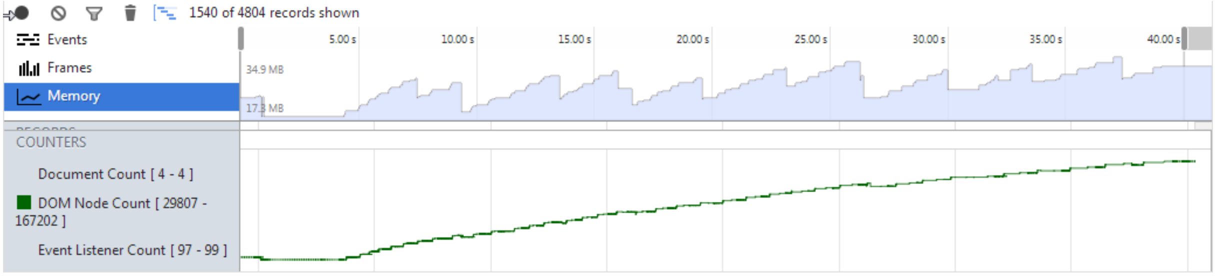


Figure 7.11: Twitter: Memory uses and DOM node Count

However, there are some different between those websites. Instagram behaves comparable to Pinterest, the amount of IMG-elements increases as more content is loaded. However, Twitter and Google+ show different behaviour. Although the DOM-size increases, the IMG-elements do not increase. This means that somehow the images are unloaded when they are outside of the view. The unloading of the images is visible in the used amount of memory graph. The memory goes up and drops again after a while. The DOM-size still increases over time because of the increase of other elements like DIVs. Not only did the DOM-size increased, also the heap-size increased fast for those websites. For example Google+ started at a heap-size of 19.7 MB and increased to 62.8 MB when scrolling down a couple of times.

Although increasing the DOM and heap-size, scrolling felt a lot smoother than for example Pinterest. When scrolled down a couple of times the browser stayed smooth and did not become sluggish. Even IE 11 didn't become unusable and stayed relative smooth. Although it is clearly smoother than for example a Pinterest or *Jij Maakt Utrecht*, it still becoming less smooth when new content is loaded in.

Removing content: In order to make an image gallery than can show a large amount of images, it is necessary that it stays responsive with a large dataset for an acceptable user experience. When looking at websites like Google+ and Twitter it is possible to show a large amount of data without decreasing the performance. This is achieved by only removing the images outside of the current view. This means that the DOM-size still increases because of the other elements like DIV-elements. This makes the scrolling smoother, but still fills up the DOM quickly.

Another approach is to not only remove the images, but also removing the other elements outside of the view. This not only reduces the used memory, but also reduces the DOM-size. In order to test this, we used some kind of lazy-loading script that only loading the current view. Elements that are outside the current view were removed.

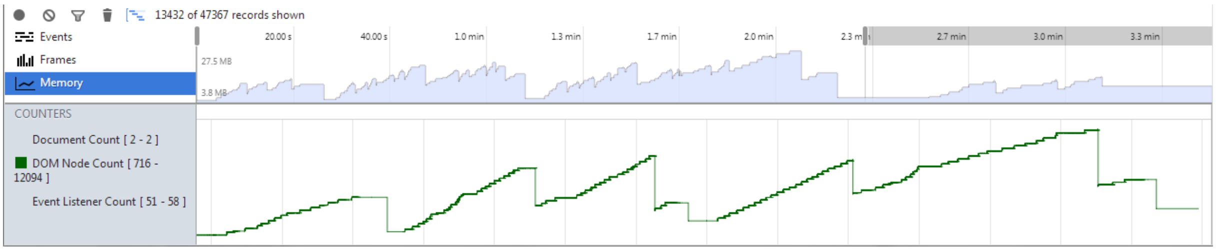


Figure 7.12: lazy-loading script: Memory uses and DOM node Count

In Figure 7.12 it looks the light-blue memory graph a bit like the Twitter and Google+ graph. The amount of memory increases when scrolling and drops again after a while. The difference with the lazy-loading script and Google+ and Twitter is that the amount of nodes in DOM drops in the lazy-loading script.

Scrolling: When looking at the scrolling performance of this script, there is much difference between the other mentioned websites. Because the DOM-size does not increase when the user has scrolled a couple of times, the responsiveness of the website was constant to when it was first loaded in. Even above 2500+ images there were no performances issues at all, even when using IE 11.

Measurements: When looking at the measurements from Chrome, it possible to confirm this difference in smoothness (see Table 7.15).

Website	Amount of images	Average time to render a frame
Lazy-loading Script	40 images	40 milliseconds
	4000 images	44 milliseconds
Lazy-loading Script, without removing images outside the view	40 images	40 milliseconds
	4000 images	3 seconds
Pinterest	20 images	30 milliseconds
	1200 images	500 milliseconds

Table 7.14: Frame render-times

Table 7.14 contains the average render times of a frame while scrolling. By comparing the lazy-loading script and Pinterest it is clear that Pinterest becomes slower over time, while the lazy-loading script does not slow down that much. When turning off the images removing in the lazy-loading script we see that the render time increases fast and became completely unusable.

Website	CPU Idle time
Lazy-loading Script	90%
Lazy-loading Script, without removing images outside the view	9%
Pinterest	32%

Table 7.15: CPU idle time while scrolling

Another tool tells the same story as the render time. In Table 7.15 we can see the idle time of the CPU while scrolling the page. The lazy-loading script handle's it quite easily with a CPU idle time of 90%. The Pinterest website takes more CPU to render and has an idle time of 32%. The lazy-loading has only got a CPU idle time of 9%.

Pros and cons of infinite scrolling

Infinite scrolling is not the optimal solution for pagination, nor is it worse than pagination. It is just another way to present data to the user. In Table 7.16 the pros and cons of infinite scrolling are described[19].

[19] http://designshack.net/articles/navigation/to-infinite-scroll-or-not-to-infinite-scroll-where-weve-come-so-far/

Other optimisations for image based views

One additional technique that can be used to optimise the performance of the gallery views is pagination, which essentially limits the amount of returned images.

Pagination is probably one of the most used techniques on the internet and is used to divide results into multiple pages. Pagination is available in for example the search results of Google and in most forums.

Pagination makes it easy to limit the amount of images that needs to be downloaded. It only needs to download the images on the current page. Not only does pagination reduce the amount of images that need to be downloaded at once, it also reduces the DOM size. Each time the user goes to a new page, the entire page is reloaded. This means that the DOM size does not increase when opening a new page. Because the DOM size does not increase when switching pages, this technique can be used with an unlimited amount of images. The only limitation is the amount of images on a single page.

The second method is to **limit the amount of images** send to the client. With this method it is not possible to show a large amount of images. It is important to make it possible to filter the images; otherwise the user only sees a selected amount of images. These filters can be based on for example location or the image title. A good example can be found in Google Maps. On Google Maps the user sees a couple of images based on currently visible map. For example when the user is zoomed out to country level, the user sees random images of that country. When the user zooms in to a city, it only sees a selection of images of that specific city.

Infinite scrolling	
Pro	**Con**
• The uninterrupted attention a user maintains when more content is provided automatically is at the core of the attractiveness of the infinite scrolling. The reader will not have to stop to think where to find the next button or which number of pages should come, following an improved attention. • In psychological terms, infinite scrolling seems to trigger automatic responses based on curiosity and the alleviation of the expectation produced while waiting for new information, which causes a kind of excitement and willingness to continue scrolling to see what comes up. This psychological mechanism deserves sufficient comprehension as the initial advantage can turn into a problem. For example, the findings of McKinley suggest that endlessly scrolling the search results page somehow shocked or confused the users, overloading them with more and more information to the point that using the search was avoided. • Template designs can benefit with more cleanness, more room for content and less distracting elements like the list of numbers for paging.	• Both seeing the footer disappear and being unable to reach can be a "traumatizing" experience for the user. Also, all the content of the footer, and therefore, the function of the footer itself will vanish. • Lack of orientation and spatial reference: In a paginated scheme users can set a simple visual reference to orient themselves through the content of the page and mark the places where something of interest is found, so it is possible to quickly return there later. There could be users who feel lost or confused not knowing where they really are or missing what they were looking for. • Loss of the user's last position in the stream of data, when the refresh or the back button is pushed. Since the infinite scrolling aims to show large amounts of entries, it should be implemented also a way to retrieve the actual position in the list, to avoid frustrating situations. Most of complaints against endless pages refer to these two last points as the users were losing control over the page they visit. • Bookmarks tend to be useless since a point of interest will not be marked on a discrete page but floating somewhere in the flow of entries. • While it is not necessarily a bad thing, rankings could vary greatly on search pages since those results confined after page 2 will then appear on page 1 just by scrolling down long enough.

Table 7.16: Pros and Cons for Infinite scrolling

D Appendix

D.1. Graph visualisations on web

The research presented in this section is a comparison of different libraries/frameworks to visualise graphs on the web[20]. Requirements where the library or framework should comply to are the following: a) The possibility to show data in different ways. A stacked bar chart is an absolute must, other diagrams are not determined beforehand; b) The library/framework should preferably be free, be not too expensive and/or have a complicated license. Two main interesting libraries and frameworks are researched: D3.js and Flotcharts. They are presented in the following two sections.

D.1.1 D3.js

D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG and CSS. D3's emphasis on web standards and gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.[21]

D3.js is a very flexible framework. With this framework, many types of charts can be created. However, the flexibility also has a large drawback: it is not really straightforward to make a visualisation. The developer is responsible for the design of the chart and the processing of the data. D3 offers some great tooling that can speed up the development. One of the functionalities that could be of great use, like "create a bar chart with this data", should however be developed. In the next section, examples of D3 libraries are discussed. More information about D3 can be found in several resources.[22]

7.1.1.1 D3 libraries

D3 is a framework, not a library to visualise data. Because of this it is not possible to call a certain function that will create, for example, an on-screen bar chart. To overcome this problem, third parties have developed libraries that simplify the creation of a specific chart. In most cases, a chart can be created by using a few lines of code on screen.

DC: dc.js is a JavaScript charting library with native cross-filter support and allowing highly efficient exploration on large multi-dimensional dataset (inspired by crossfilter's demo). It

[20] This research is based upon information presented at different websites that compare graphs and chart libraries:

- http://socialcompare.com/en/comparison/javascript-graphs-and-charts-libraries
- http://techslides.com/50-javascript-charting-and-graphics-libraries/
- http://www.fusioncharts.com/javascript-charting-comparison/
- http://maskniga.blogspot.nl/2012/06/comparison-of-javascript-data.html
- http://en.wikipedia.org/wiki/Comparison_of_JavaScript_charting_frameworks

[21] http://d3js.org/

[22] See for example these ebooks:

- https://leanpub.com/D3-Tips-and-Tricks/read#leanpub-auto-introduction-to-dcjs
- https://speakerdeck.com/binx/d3-dot-js-plus-mobile
- https://www.dashingd3js.com/binding-data-to-dom-elements

leverages d3 engine to render charts in css friendly svg format. Charts rendered using dc.js are naturally data driven and reactive therefore providing instant feedback on user's interaction. The main objective of this project is to provide an easy yet powerful javascript library which can be utilized to perform data visualization and analysis in browser, as well as on mobile device.[23]

DC is a library that combines D3 and Crossfilter libraries. The Crossfilter library is developed to filter data at the client. This combination (DC), allows the user to filter data via different types of charts, by using D3 and Crossfilter. A screenshot is shown in Figure 7.13.

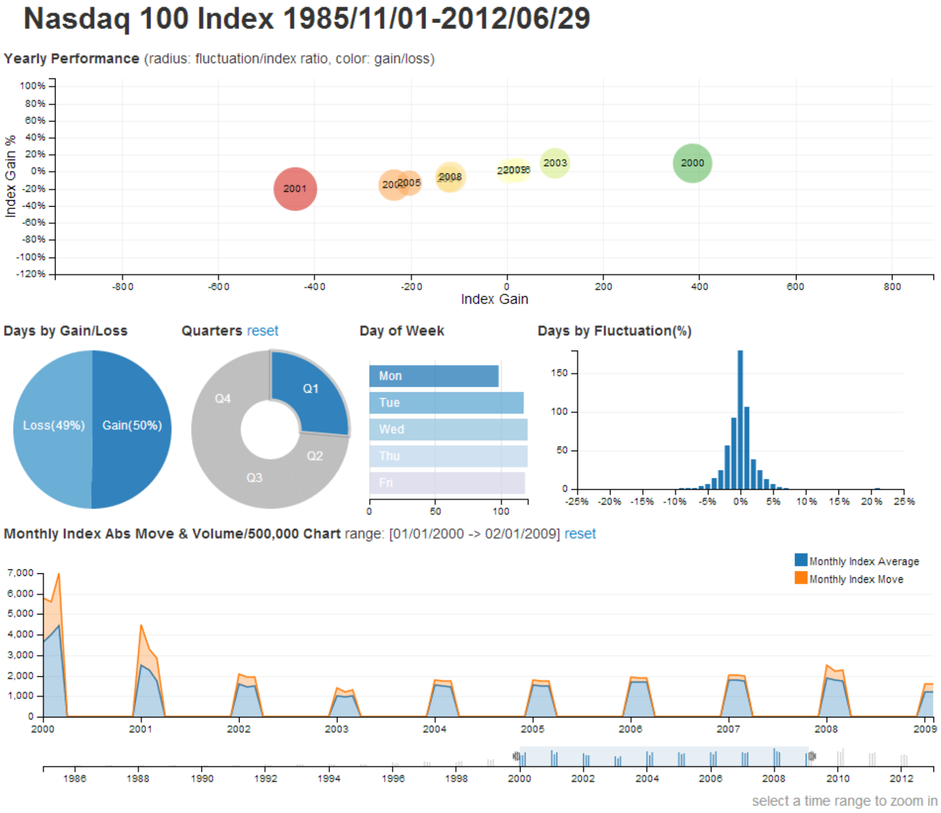


Figure 7.13: Screenshot of DC.js library

<u>Dimplejs:</u> The aim of dimple is to open up the power and flexibility of d3 to analysts. It aims to give a gentle learning curve and minimal code to achieve something productive. It also exposes the d3 objects so you can pick them up and run to create some really cool stuff. The intention is to make the basics easier, while still exposing the d3 components so that you can go off-piste when you get a bit more familiar. The target audience for this is advanced analysts who don't necessarily consider themselves highly proficient in JavaScript but want to build some axis based visualisations.[24]

With Dimple, one can easily create really nice charts. This library supports the use of line charts, bubble charts, scatter plots, bar charts and area charts. Figure 7.14 show examples for some of these charts in Dimple. However, the library lacks the possibility to create one

[23] https://github.com/dc-js/dc.js

[24] http://dimplejs.org/

widely used chart, namely pie charts. Furthermore, the library adds functionalities like tooltips and when you hover on a certain point in the chart, a line is drawn towards the x- and/or y-axes. Dimple is a recently developed D3 library and is updated on a regular basis.

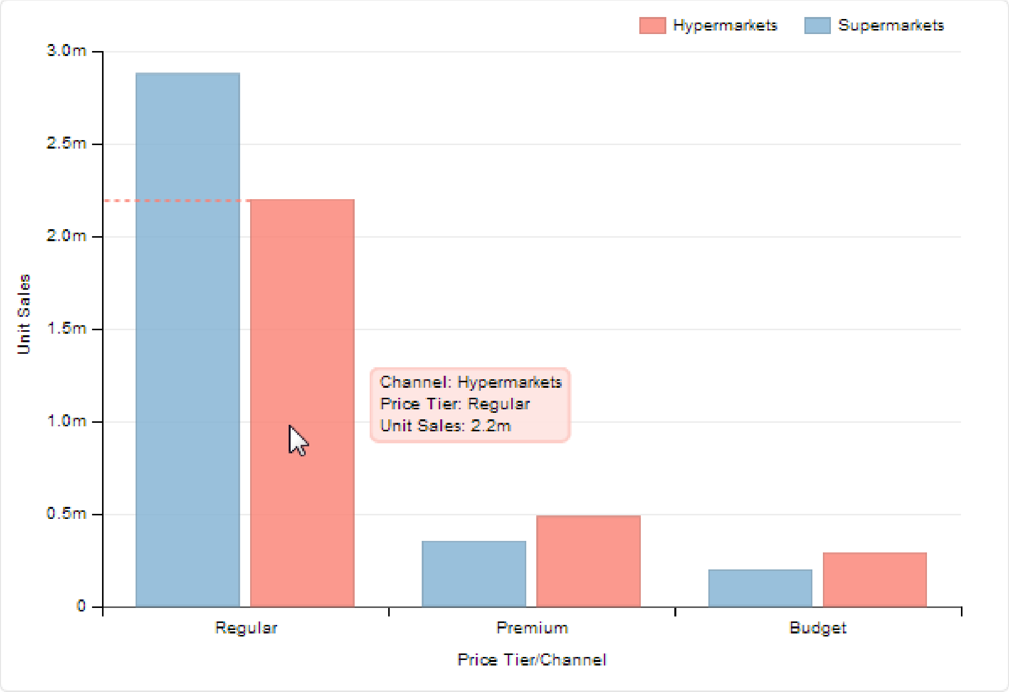


Figure 7.14: Screenshot of a simple bar chart in Dimple

NVD3: This project is an attempt to build re-usable charts and chart components for d3.js without taking away the power that d3.js gives you. This is a very young collection of components, with the goal of keeping these components very customisable, staying away from your standard cookie cutter solutions. [25]

NVD3 is a very extensive chart library for D3. With just a few lines of code, NVD3 offers the opportunity to build an entire chart that in addition is dynamic. It automatically takes the size of your screen into account and has the possibility to filter the data. This library supports well-known charts such as line charts, pie charts and bar charts (see. Moreover, it supports functionalities like labels and tooltips. Figure 7.15 and Figure 7.16 shows some example charts that can be generated with this library. The library is in place for some time and has proved itself to be a useful library.

[25] http://nvd3.org/index.html

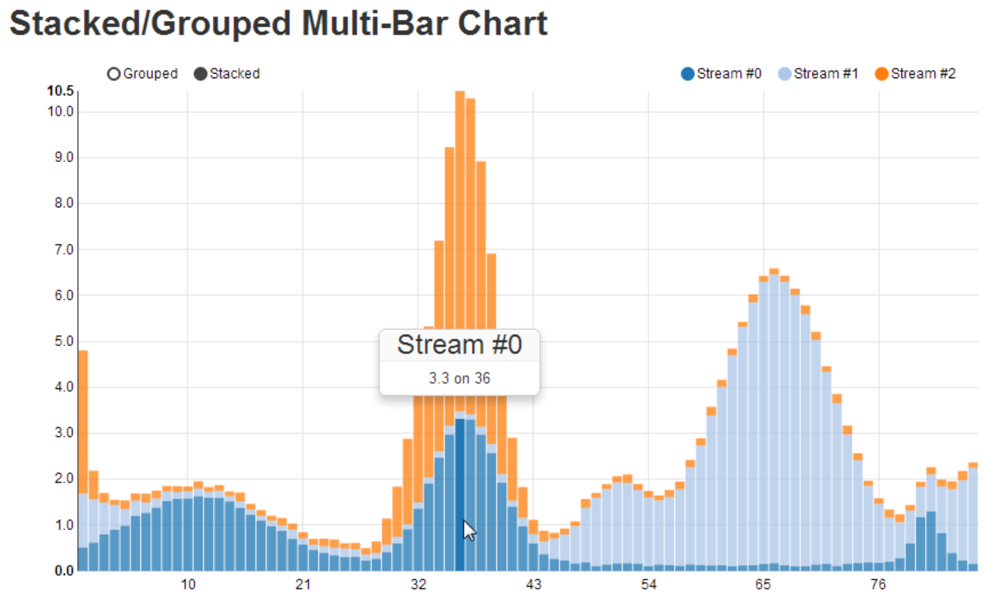


Figure 7.15: Screenshot of a stacked multiple bar chart in NVD3

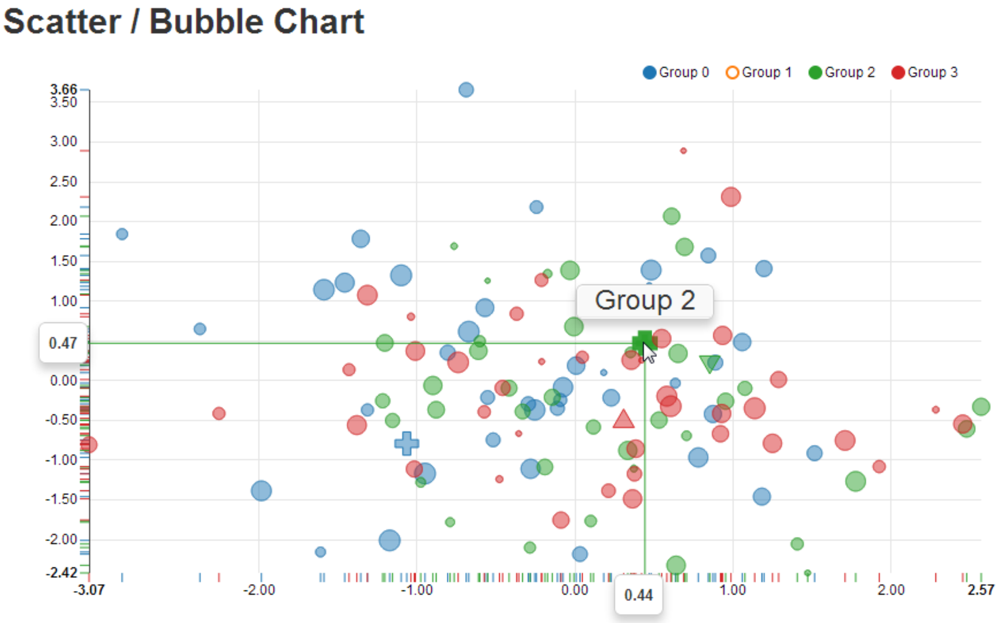


Figure 7.16: Screenshot of scatter plot/bubble chart in NVD3

D.1.2 Flotchart

Flot is a pure JavaScript plotting library for jQuery, with a focus on simple usage, attractive looks and interactive features[26]. In contrast to D3, Flotchart is a library that is specifically focused on generating charts and making this as simple as possible. Flot supports many different types of charts: well-known and broadly used ones like line charts, bar charts and

[26] http://www.flotcharts.org

pie charts (see Figure 7.17 and Figure 7.18). Furthermore, it also supports less known charts like spider charts and span charts. Moreover, plugins can be downloaded that further extend this supply.

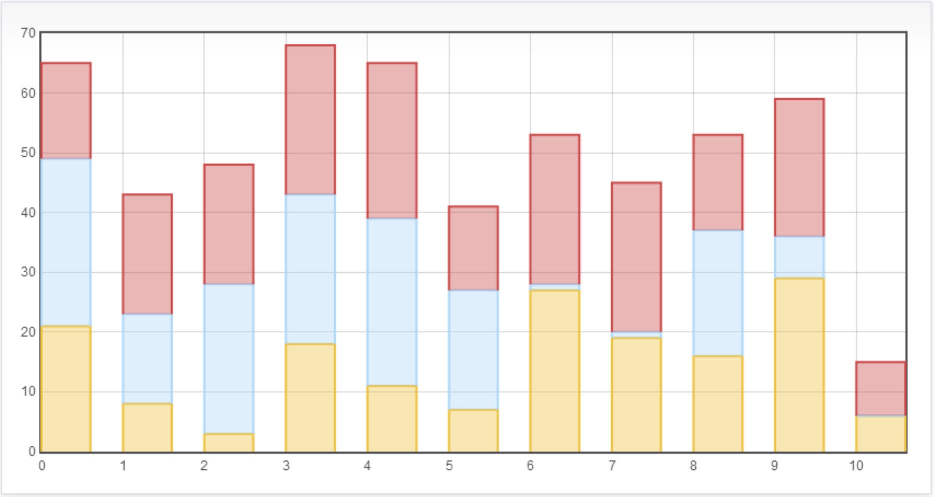


Figure 7.17: Screenshot of a stacked bar chart in Flot

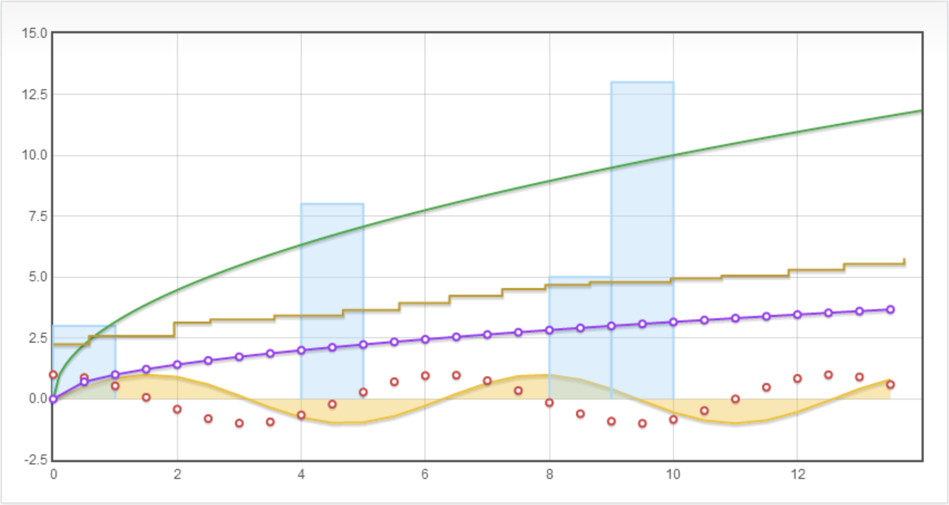


Figure 7.18: Screenshot of different line charts in Flot

D.1.3 Conclusion

This investigation has explored different frameworks and libraries to create charts for web sites. D3 and Flotchart are both decent solutions to generate several charts. Both offer opportunities to create nice and dynamic charts and support different types of charts. The primary goals between D3 and Flotchart differ. Where D3 is more targeted to visualise data in general, Flotchart is mainly targeted to generate charts. As a consequence, D3 has a steeper learning curve and is much more flexible than Flotchart. On the other hand, Flotchart is easier to use at first. To overcome the steeper learning curve in D3, a third-party library can be used. When comparing different third-party D3 libraries, NVD3 seemed the most appropriate since it is relatively simple, quite extensive and has proved its usefulness in the community. Thus, the NVD3 library was adopted for the purposes of Live+Gov.

D.2. Graph visualisations on mobile devices

The investigation presented in this section is aimed to find out which mobile libraries are available in the market for showing charts. We have investigated the following libraries.

D.2.1 Native libraries

Core Plot: Core Plot is a plotting framework for OS X and iOS. It provides 2D visualization of data, and is tightly integrated with Apple technologies like Core Animation, Core Data, and Cocoa Bindings. Some examples of Core Plot graphs can be seen in Figure 7.19.

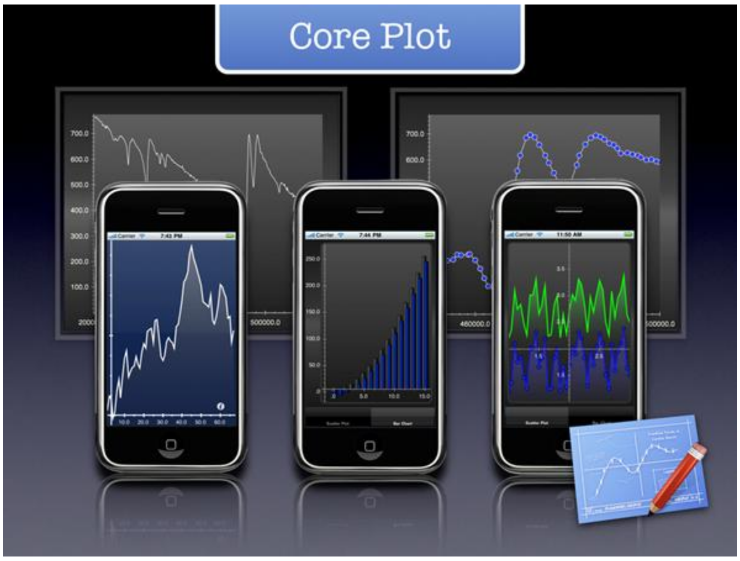


Figure 7.19: Core Plot graph examples for mobile charts

Shinobicontrols: Shinobicontrols is another very interesting library that can be purchased for a standards fee. It is available for both iPhone and Android. Some example plots are depictred in Figure 7.20.

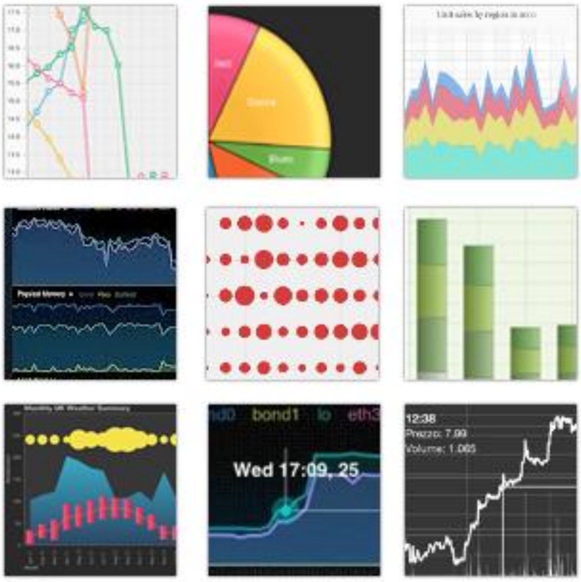
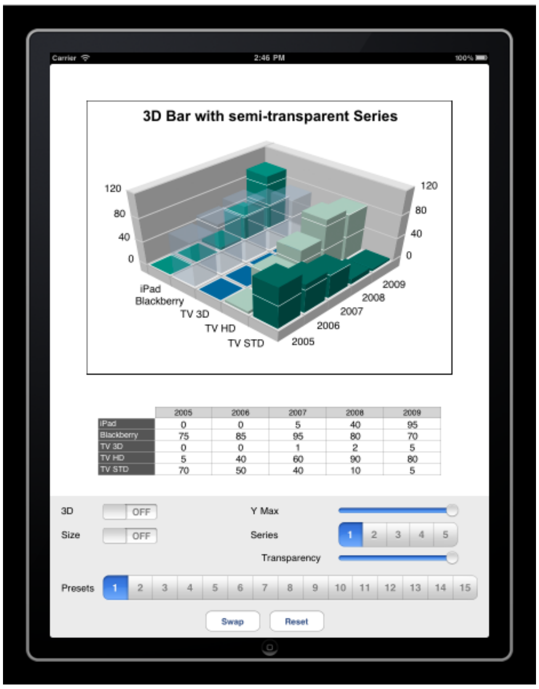


Figure 7.20: Example plots that can be implemented with Shinobicontrols

THREE D GRAPHICS: Is a library that is offered in 3 different versions (intro, standard and enterprise) and it offers the potential to have: unlimited number of developers; unlimited number of applications; unlimited number of users; unlimited distribution. Figure 7.21 displays some of the visualizations that can be implemented with THREE D GRAPHICS.



	2005	2006	2007	2008	2009
iPad	0	0	5	40	95
Blackberry	75	85	95	80	70
TV 3D	0	0	1	2	5
TV HD	5	40	60	90	80
TV STD	70	50	40	10	5

3D OFF
Size OFF
Y Max
Series 1 2 3 4 5
Transparency
Presets 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Figure 7.21: Example plots that can be implemented with THREE D GRAPHICS

INFRAGISTICS (NUCLiOS): It is a useful library that has an active support community and contains also a grid and a rich text label component. Some of the special graphs that can be supported by this library are depicted in Figure 7.22.

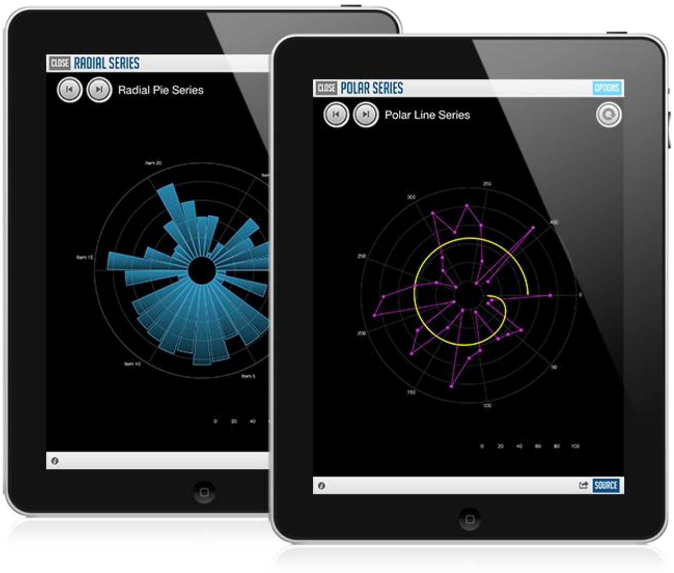


Figure 7.22: Graphs supported by INFRAGISTICS

Hongcheng: Is an open source library for both iOS and Android that seems to be no longer subject of active development. The type of pie charts that can be generated with this library are depicted in Figure 7.23.

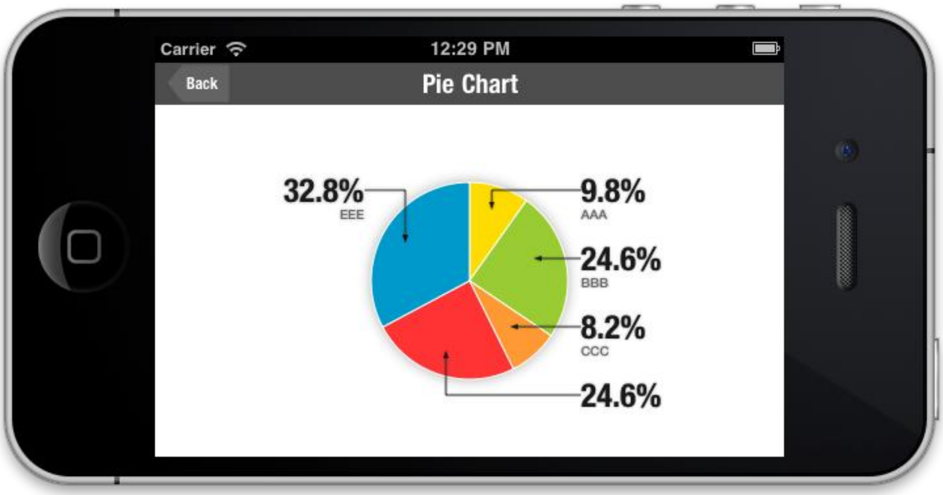


Figure 7.23: Example of a Pie chart generated with the Hongcheng library

D.2.2 HTML / JavaScript

Another approach to draw charts is by the use of HTML and JavaScript. An online or offline solution is possible. The UIWebView component can be used to show the HTML code. Examples of JavaScript libraries for drawing charts are:

- http://www.fusioncharts.com
- http://www.jqplot.com
- http://www.highcharts.com
- https://developers.google.com/chart/

A small demo application was developed for evaluating the HTML and JavaScript approach. For this demo application the highcharts library was used and was linked as a resource to mobile base application, so the application can draw the charts offline.

D.2.3 Summary

This study looked at mobile libraries available in the market which can be used to display charts. The following chart libraries were examined:

- Core Plot (open source)
- ShinobiCharts (starting at $395)
- THREE D GRAPHICS (starting at $995)
- KeepEdge (starting at $999)
- INFRAGISTICS NUCLiOS (starting at $295)
- iOS Plot (open source)

Based on the outcome of our evaluation the Core Plot and INFRAGISTICS NUCLiOS library seemed to be the most interesting libraries. The option of using HTML instead of a native library was also considered as a good choice. There are several JavaScript libraries that are available in the market for drawing charts, like: i) fushioncharts, ii) jqplot, iii) highcharts, and iv) google charts. For the purposes of Live+Gov we have decided to rely on Core Plot.