Renewable Energy Production Distribution Map of Catalan Homes

by

Pau Perez Fabregat

Submitted to the Department of Service and Information System Engineering
in partial fulfillment of the requirements for the degree of

Informatics Engineering

at the

UNIVERSITAT POLITÈCNICA DE CATALUNYA

February 2014

Carles Farré Tost
Associate Professor
Thesis Supervisor

Renewable Energy Production Distribution Map of Catalan Homes

by

Pau Perez Fabregat

Submitted to the Department of Service and Information System Engineering
on Jan 10, 2014, in partial fulfillment of the
requirements for the degree of
Informatics Engineering

Abstract

In this thesis...

Thesis Supervisor: Carles Farré Tost
Title: Associate Professor

March 11, 2014 **DRAFT**

Acknowledgments

This is the acknowledgements section. You should replace this with your own acknowledgements.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

The Center for Ecological Research and Forestry Applications (CREAF) is a public research institution that was created in 1987. The members of the Governing Council of CREAF are the Generalitat of Catalonia, the Autonomous University of Barcelona (UAB), University of Barcelona (UB), the Institute for Research and Technology (IRTA), the Institute of Catalan Studies (IEC) and the Spanish National Research Council (CSIC).

Its objective is to generate knowledge and create new methodological tools in the field of terrestrial ecology in order to improve environmental planning and management in rural and urban areas with special emphasis on forest ecology. This is achieved, among other means, through the development of methodological and conceptual tools designed to facilitate decision-making and improve environmental management.

Since its creation, CREAF has made very important contributions to the field of terrestrial ecology and towards a sustainable management of the environment. This has been achieved through research, development, training and technology transfer. Some of its most relevant contributions include the design and implementation of the Ecological and Forest Inventory of Catalonia (EFIC), innovative at the international level due to the incorporation of new ecological parameters, the production of the Land Cover Map of Catalonia (MCSC), a high-resolution digital map for environmental assessment and territorial planning and management and the development of the MiraMon©Geographic Information System, widely adopted in Catalan administration and currently used in over thirty countries around the world.

1.1 Motivations

The motivation for this master thesis stems from the idea that the wide development of web technologies, DevOps and infrastructure services make possible the implementation of powerful systems with significant cuts on budget and maintenance costs. Moreover, these technologies often entail considerable improvements in maintainability, performance and reduced complexity.

I am firmly convinced that these technologies and tools can improve research in centres such as CREAF, providing them with better and affordable infrastructures reducing the timespan of common processes. Furthermore, they provide new means for the dissemination of the resulting data.

Nonetheless, I think that computer engineering should be conceived as a tool to push forward the development of other sciences. As recently graduated engineers we can contribute back to society with our knowledge attempting to solve problems that benefit us all. Hence, I wish to develop a project in which the outcome could improve the work of some public research centre thus, becoming a useful tool.

Given my discovered interest in topics like distributed computing, sensor networks and resilience systems arouse in my recent stay in University of Antwerp. I am eager in expanding my knowledge further and dig deeper into these fields in a real world use case.

Within the context of volunteered geographic information (VGI) and renewable energies, CREAF wants to solve the problem of knowing the distribution of renewable energy produced in Catalan homes. Nowadays, its performance, time evolution and distribution is unknown thus, complicating the decision-making regarding renewable energy sources in Catalonia.

On the other hand, CREAF wishes to expand its methodological tools by adopting sensor web. The reduced cost of hardware devices like Raspberry Pi and Arduino and their general-purpose features make them an affordable and versatile solution as sensor devices. Their considerable computational power also facilitates the development of services clients in widely adopted languages. For these reasons, CREAF plans to deploy its own sensor devices in wild nature in the near future.

Although CREAF already took some design decisions, the architecture of the system and the software the devices would be shipped with were still to be determined.

Given the mutual interest in the project outlined by CREAF we set out to design and implement a prototype as a first working solution.

1.2 Project Goals

The Renewable Energy Production Distribution Map of Catalan Homes (REDCH) is aimed at developing a system that offers features to registered users, those who freely share their data, and some other features publicly available. It will visualize the energy production of the clients system and its contribution to the whole Catalan renewable energy production in a real time map, while offering a private analytics dashboard to registered users where they can figure out the actual performance of their system.

Given the extent of the desired product with this thesis we are going to develop a proof of concept. A distributed computing system that will provide essential features, being a simple but functional prototype of the final product. Once built, the system results and metrics will be evaluated and its architecture may eventually become the standard infrastructure basis for future CREAF projects that demand sensor data. As a consequence, all features for registered users are out of the scope.

To sum up, these two general objectives are translated into the following specific goals:

- Provide a command line interface that allows the simulation of the sensors functionality
- Develop a functional system that stores and processes the sensor observations
- Implement a simple public web application to display the observations in a real time map

1.3 Methodolgy

1.3.1 Iterative Development

Although advocating for agile software methodologies the concept-proof nature of the thesis, which is developed by just one person make them an unsuitable choice. We vote for a custom adaptation of iterative development instead.

In conjunction with incremental development Iterative development is a way of breaking down the development of a software system in smaller chunks and repeated cycles. In each cycle, known as iteration, the slice of functionality is designed, developed, tested, deployed and evaluated. This allows software developers to apply the knowledge acquired in previous iterations, so the first implementation whose goal is to build a bare minimal functional system is iteratively enhanced so as to meet the requirements.

Nevertheless, iterative and incremental development are the basis for Agile development. Therefore, by adhering to them we attempt to avoid the agile practices and constraints that may be pointless in this case. Doing so, we aim to progressively enhance the codebase in subsequent iterations, gain insight into the architecture and improve any weak points we may identify until eventually meeting the requirements. Additionally, early results can be achieved and evaluated by CREAF resulting in a smoother collaboration.

1.3.2 Test-Driven Development

The chosen methodology also includes Test-Driven Development, which is a developer practice that involves writing tests before writing the code being tested. The initially failing test defines the behaviour of the code to be written, then the developer writes the minimum amount of code required to pass the test. Once it passes, it is time to refactor it to remove any duplication. This cycle must be repeated as many times as it required to further extend the responsibilities of the code.

Besides validating the correctness of the code, by driving the design of the program in small chunks through test cases the developer is mainly concerned with the interface of the program rather than its actual implementation.

What we aim for while using TDD in this project is to obtain a more modularized, maintainable, and extensible code. The development of the software in small units leads to smaller, more focused and loosely coupled classes and cleaner interfaces. The main benefit we may get by this means, however, is a greater level of confidence in the code caused by the fact that all code written is covered by at least one test.

Additionally, in this early stage of the project to have a test-covered code is basic practice for the successful evolution of the project. So it can be ensured that it keeps its intended behaviour and any defects are caught early in the development process, what has a considerably minor impact on costs than in later stages.

Chapter 2

Technology research

This chapter aims to synthesize the research carried on the main knowledge areas involved in the development of this project. It analyses technologies, paradigms and different insights the design of the project is based upon. This includes the main areas of concern: Public Interface and Interoperability, Database, Real time, Web Technologies, Event logging.

2.1 Public Interface and Interoperability

2.1.1 Client devices

IoT Raspberry Pi, Arduino

2.2 Database

Ever growing data set, very small data units and mainly write ops.

2.2.1 Relational Databases and NoSQL

NoSQL MongoDB most mature pros: - scalabiltiy (auto sharding capabilities) - performance - administration ease - several new options, not very production ready, stability

cons: - migration from relational design - non-ACID - eventual consistency (although, non achievable in a distributed system)

2.3 Real Time

for frontend redirect to web technologies

2.3.1 Asynchronous Messaging Queue

RabbitMQ ZeroMQ

impact on client. Async app server. Reactor pattern.

2.3.2 Tuple Space

redis

2.4 Web Technologies

Since the Sir Tim Berners-Lee's first draft of the World Wide Web back in 1989 and his first proposal for the HyperText Markup Language (HTML) [?] [1] in 1991 the WWW has experienced a tremendous evolution. Since then, HTML has gone through many revisions. The World Wide Web Consortium (W3C) published many iterations of the standard until the specification HTML 4.01 in 1999. It was not until 2004, when the Web Hypertext Application Technology Working Group (WHATWG) was founded by individuals of Apple, the Mozilla Foundation, and Opera Software, that HTML was proposed to be extended so as to allow the creation of web applications. WHATWG published the First Public Working Draft of the HTML5 specification in 2008. Although parts of HTML5 have been implemented in browsers it hadn't been until 2012 that W3C designated HTML5 as a Candidate Recommendation and it is planned to be released as a stable Recommendation by the end of 2014.

Ajax

HTML5 - what is it - who - future - APIs: Canvas, SVG, ... particularly the ones involved in Redch

Browser vendors - dramatical improvement of JS engines - battle as new explorer engine - prefixed CSS properties and non-standarized prefixed APIs

- Firefox OS, Node.js (JS in the backend), Win8 JS apps, - rich applications GDocs, Chrome book - Towards application platform Canvas, SVG

JS Libraries - to solve cross-browser issues/compatibility, old browsers. jQuery - D3, widely used. Standard for data visualization. As being the 5th most starred repository in github, more than rails or angular.js, shows. - Chart.js

JS apps -¿ Redch web application problem of JS apps -¿ old browsers. Autoupdate - Backbone, underscore -¿ Angular.js, Ember - full ecosystem of MVC components, templates systems, etc.

2.4.1 Server Sent Events

specification over standard HTTP. RFC

2.4.2 Web Sockets

buzzword, trend, full duplex Too often misunderstood, lower level protocol than HTTP, whole new protocol

2.5 Event Logging

Need to get insight of the system functioning. Misbehaviour. Analysis tool.

Chapter 3

Analysis

3.1 Requirements

They have been obtained throughout some meetings with CREAF researcher in charge of the project.

Real use-case data (number of owners) has been obtained to "scope" the system qualities properly.

3.1.1 System-Wide Functional Requirements

- The system must be fed with the data collected by the sensor devices
- The system must require an authentication system for the sensor devices and the web application users
- The system must provide software tracing

System Qualities

Usability

- The web application must be user friendly and easy to use by means of a GUI
- The sensor devices must be easy to set up by end users
- The sensor device must collect and deliver data automatically
- All the command line interfaces must be configurable

Reliability

- The system must be reliable in low bandwidth and high network latency
- The system must be reliable in high concurrency scenarios

Performance

- The sensor devices must be able to send [Y] request per hour
- The system must be able to process observations received from [X] sensor devices
- The system must able to display a sample from a certain sensor before its next sample is received

Supportability

- The system must have an event logging system
- The system must conform to the OGC Sensor Observation Service (SOS)

- The system must be horizontally scalable
- The codebase must be maintainable and extendible

3.1.2 System Interfaces

Interfaces to External Systems

- The system must integrate 52 North SOS implementation to provide a standard interoperability layer

3.1.3 System Constraints

- The sensor devices must send 6 power observations per hour
- The system must be able to run in commodity servers
- The whole codebase must be portable in order to be deployed in any platform including PaaS and IaaS

SOS

3.1.4 System Compliance

- The system and all its components must adhere to Apache license

3.2 Technology Research

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Specification

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Design

In this chapter we argue all design desitions taken regarding the building of Redch.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Implementation

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Testing

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 8

Deployment

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 9

System execution and maintenance

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 10

Project Management

10.1 PaaS and IaaS cost comparison

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 11

Results

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 12

Conclusions

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 13

Bibliography and references

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix A

Tables

Table A.1: Armadillos

Armadillos	are
our	friends

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Figures

Figure B-1: Armadillo slaying lawyer.

Figure B-2: Armadillo eradicating national debt.

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] Tim Berners-Lee. Html tags. *CERN*, 1991.