

4th year internship
Computer Science and Management
Academic year 2015-2016



Strengthening of analysis systems of SOA applications' quality

Internship report

Carried out by
Gaëtan FRANÇOIS

Under the supervision of
Yann-Gaël GUÉHÉNEUC and **Anne-Laure de LAUZUN**

At



From the 6th of June 2016 to the 29th of July 2016



Table of contents

- Introduction..... 4
- I. Work environment 5
 - 1. Polytechnique Montréal..... 5
 - 2. The research lab 5
 - A. Hours of work..... 5
 - B. Equipment at disposal..... 5
 - 3. People I worked with..... 5
- II. The project 7
 - 1. Concept 7
 - 2. Functioning..... 7
 - 3. The mission..... 8
 - 4. Results 8
- III. Balance sheet 12
- Conclusion 13

Introduction

As a fourth year student in Computer Science and Management at Polytech Montpellier, I have to make an internship and decided to make it at Polytechnique Montréal in a research lab.

The title of my work placement was “Strengthening of analysis systems of SOA applications’ quality” which means I had to work on a tool which goal is to help people improving their application using services-oriented architecture.

This report describes this internship by introducing my work environment, presenting the project and making a balance sheet of this one.

This report is written in English as my tutor asked me.

I. Work environment

1. Polytechnique Montréal

Polytechnique Montréal is an engineering school affiliated with the Université de Montréal, often called UdeM, in Montréal, Québec, Canada. Previously called Ecole Polytechnique de Montréal, it has been founded in 1873 and its motto in latin is “Ut tension sic vis”, meaning “As the extension, so the force”. 12 undergraduate programs are offered by Polytechnique which are managed by 7 departments such as Electrical Engineering Department or Computer and Software Engineering Department within I made my internship.

2. The research lab

The Computer and Software Engineering Department’s research lab is made of several rooms which are for some assigned to either working teams or PhD students. Some other rooms seem not to be assigned to anyone in particular. Each room of the laboratory is closed and accessible only thanks to an identity card delivered by Polytechnique which grant access to the rooms you need based on your identity. As a “guest”, I had access to rooms chosen by my tutor for the length of my work placement.

A. Hours of work

The rooms look like open spaces where people can come and go whenever they want during the open hours as long as the work is done. Actually, only they have to get the requested results on time and answer the emails, it doesn’t matter the time they spent in the lab. That was the same for me during my internship.

B. Equipment at disposal

In the lab, everyone seems to have his usual office. However, that was not the case for me. Actually, no office has been assigned to me. So what I used to do was to choose the most convenient place among the empty ones when I arrived at work. Thanks to the holidays, most of people did not come to work, but I sometimes had to move from one place to another.

Concerning the computers, I worked in BYOD, Bring Your Own Device. Most of researchers here do that and as I did not have any stable office to work, it was easier for me.

3. People I worked with

During my internship, I worked with the work of, on the first hand, Francis PALMA and, on the other hand, Moustapha BOULGOUDAN.

Francis is currently working as a post-doctoral fellow at the Concordia University in the Department of Electrical & Computer Engineering. He completed a PhD in August 2015 from Polytechnique Montreal; his main research interests included Service-Oriented Architecture Antipatterns and Quality of Service for Service-Based Systems. He was the one in charge of giving me the tasks I had to perform.

Moustapha is a Master student whose subjects of interest are web development and web services currently making a thesis called “Development of a tool in the form of a Web Service for detection of Patterns and Antipatterns REST in APIs”.

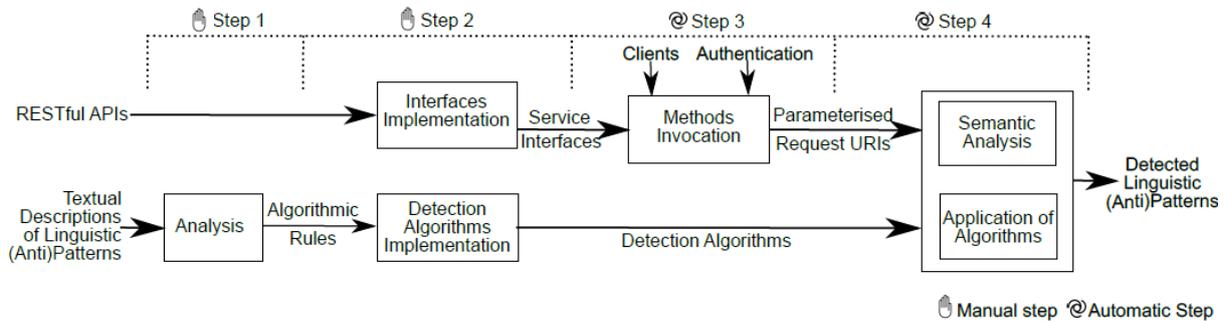
My work was to handle theirs to integrate Francis’ work into Moustapha’s.

II. The project

Named Restpad, the project aims to help developers to improve their web services' quality by analysing and detecting linguistic patterns and antipatterns in RESTful APIs.

1. Concept

The process consists of four steps as shown in the figure below.



1 : Project concept

Step 1. Analysis of Linguistic Patterns and Antipatterns:

This manual step consists in analysing the description of REST linguistic patterns and antipatterns from the literature to identify the properties relevant to their detection. We use these relevant properties to define algorithmic rules for patterns and antipatterns.

Step 2. Implementation of Interfaces and Detection Algorithms:

This manual step involves the implementation of detection algorithms for patterns and antipatterns based on the rules defined in Step 1 and the service interfaces for RESTful APIs, which include the list of methods to be invoked.

Step 3. REST Methods Invocation:

This automatic step deals with the consumption of RESTful APIs by calling their methods to access their underlying methods automatically from the interfaces defined in Step 2.

Step 4. Detection of Linguistic Patterns and Antipatterns:

The last automatic step deals with the semantic analysis of resource URIs and API documentations by applying automatically the detection algorithms (implemented in Step 2) on resource URIs and documentations of RESTful APIs obtained in Step 3 for the detection of linguistic patterns and antipatterns

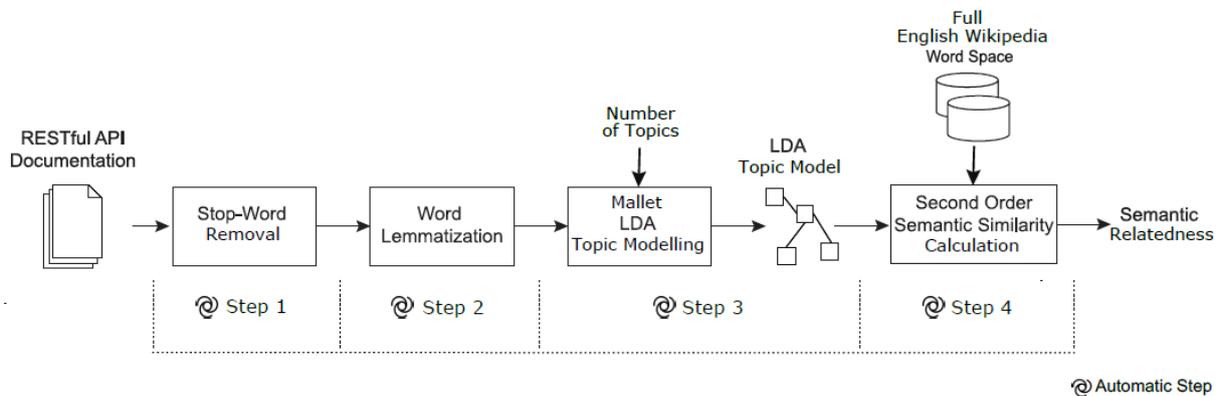
2. Functioning

The semantic analysis of REST URIs and documentation involves four automatic steps:

1. collection of APIs documentation corpus and performing some pre-processing, e.g., exclusion of stop words. Each language has its stop words which are mainly usual words such as "a", "is" or "sorry" for example.
2. truncating URI nodes to their base form, i.e., lemmatisation, that is to say grouping words of a same family, using Stanford Core NLP which is an integrated framework

which goal is to make very easy to apply a bunch of linguistic analysis tools to a piece of text

3. based on the collected corpora, performing topic model extraction
4. measuring the second order similarity between the acquired topic model and nodes in a URI



2 : Semantic analysis

3. The mission

Restpad will be deployed online so my mission was to propose an interface for this tool for the static detection in URIs.

Developed in Java, I worked on a dynamic web project using the tool provided for the detection. Actually, the web project shared Restpad dependencies, so that we can call methods from this last one in the first one.

First of all, I started designing the interface in order to handle all the possible errors and make it easy to use for the user. That is to say that by simply clicking and filling text boxes, the user will be able to have his results.

Secondly, I refactored some methods from Restpad to make it useable in the web project.

Finally, I used the refactored methods in the interface to make this work.

I explain it in more details in the next part called "Results".

4. Results

My work result is a web interface using Restpad tool to detect patterns or antipatterns in a URI given by the user.

The first step is to give the URI you want to analyse and choose the patterns or antipatterns you would like to detect. The list of antipatterns and patterns has been established when restpad was developed.

URI:	<input type="text"/>
Static REST Patterns:	<input type="checkbox"/> All <input type="checkbox"/> Versionned URI
Static REST Anti-Patterns:	<input type="checkbox"/> All <input type="checkbox"/> Amorphous URI <input type="checkbox"/> CRUDy URI <input type="checkbox"/> Contextless Resource Names <input type="checkbox"/> Non Hierarchical Nodes <input type="checkbox"/> Singularised Node <input type="checkbox"/> Non-pertinent Documentation

3 : Overall view

Here the user has to fill in the form. This form is made in HTML and all the information are retrieved from a MySQL database where a pattern is known by an id, its name, its type (antipattern or pattern), its description, and its structure (dynamic or static). Here, all the patterns or antipatterns are static.

URI:	<input type="text"/>
Static REST Patterns:	<input type="checkbox"/> All <input type="checkbox"/> Versionned URI
Static REST Anti-Patterns:	<input type="checkbox"/> All <input type="checkbox"/> Amorphous URI <input type="checkbox"/> CRUDy URI <input type="checkbox"/> Contextless Resource Names <input type="checkbox"/> Non Hierarchical Nodes <input type="checkbox"/> Singularised Node <input type="checkbox"/> Non-pertinent Documentation

4 : Static detection

On the figure below, we can see that some antipatterns need information that have to be filled before processing to the analysis. Indeed, the Contextless Resource Names antipattern needs a file gathering all the URIs of an API to analyze them one by one, and the Non-Pertinent Documentation need a description of the URI to check if it describes well this one.

URI:	<input type="text" value="https://content.dropboxapi.com/1/commit_chunked_upload/auto/"/>
Static REST Patterns:	<input type="checkbox"/> All <input type="checkbox"/> Versionned URI
Static REST Anti-Patterns:	<input type="checkbox"/> All <input type="checkbox"/> Amorphous URI <input type="checkbox"/> CRUDy URI <input checked="" type="checkbox"/> Contextless Resource Names <input type="checkbox"/> Non Hierarchical Nodes <input type="checkbox"/> Singularised Node <input checked="" type="checkbox"/> Non-pertinent Documentation
Description of all URIs for Contextless Resource Names:	<input type="button" value="Choisissez un fichier"/> <input type="text" value="Aucun fichier choisi (txt file)"/>
Description of URI for Non-Pertinent Documentation:	<input type="text"/>

5 : Static detection needs

Once the user has chosen the patterns and antipatterns he wants to detect, he has to click on process to get the result which is displayed in table gathering the chosen patterns and antipatterns' names, their type and the result of the analysis.

URI:	<input type="text" value="https://content.dropboxapi.com/1/commit_chunked_upload/auto/"/>
Static REST Patterns:	<input checked="" type="checkbox"/> All <input checked="" type="checkbox"/> Versionned URI
Static REST Anti-Patterns:	<input type="checkbox"/> All <input checked="" type="checkbox"/> Amorphous URI <input checked="" type="checkbox"/> CRUDy URI <input type="checkbox"/> Contextless Resource Names <input type="checkbox"/> Non Hierarchical Nodes <input type="checkbox"/> Singularised Node <input type="checkbox"/> Non-pertinent Documentation

6 : Static detection choice

The result of the analysis is stored in a HashMap with the pattern as the key and the result as the value. We are able to get each pattern thanks to its id. To do so, we use the Data Access Object development pattern which allow us to make a link directly between the model and the logics of the code using the database content.

Static Patterns Detection Results Page.

Analyzed URI: https://content.dropboxapi.com/1/commit_chunked_upload/auto/

Name	Type	Result
Versionned URI	P	Version 1 found.
Amorphous URI	AP	Poor URI: underscore detected. Poor URI: trailing slash detected. Good URI: no extension detected. Good URI: 0 upper case letter(s) detected.
CRUDy URI	AP	Good URI: no CRUDy word detected.

7 : Static detection results

Here we can see the results for the chosen patterns and antipatterns. If we analyse it, we can see that the URI uses the Versioned URI pattern, which is a good thing. There are also no extension such as « .pdf » or « html » in the URI, nor upper case in the middle of words. It is ok to have one at the beginning of the word but not after. No CRUDy word such as « delete » or create are detected either. However, there is an undescore detected and a trailing slash, which are not good practices. Thanks to thes results, the developer is able to modify its URI to respect all the good practices.

III. Balance sheet

Unfortunately, this internship didn't meet my expectations. Indeed, I was willing to improve my skills and particularly the development ones during this work placement. However, this didn't happen.

First of all, when I arrived, the project was not ready. I mean that the things I had to work on were not ready, so I had to wait before starting working. I had to ask for work several times by sending emails. I wasted time doing nothing during my internship so eventually I feel like I did not bring anything to the lab during my internship and it did not bring me anything either.

Secondly, there were no methods of work. I mean, speaking about development, there were no UML diagrams to help me understand how works the project, the code was barely commented and no documentation such as Javadoc was available. Concerning management methods, there were no check of my work; I had to ask for appointments for this. No agile methods were used either. I tried to use the Kaban method for me, but as the demands about what I had to do were not clear and changed so often, I abandoned it.

Finally, as a consequence, I did not find my place in the project. I did not understand what my role was and had no motivation to go working. I did not find any excitement in this experience whereas I have always found some in my previous projects.

Conclusion

This report is being the image of my internship with no organization and no fun. There are not much things to say as I did not much but I don't think that's my fault as, as I wrote in the previous part, I asked for job. I am not proud of my job either because I know I can do so much better. There may have a lot of negativity in this report, but that's because this experience was the worst one I ever had, professionally speaking. I know now that I don't want to work in a research lab later.

I hope I will find a real work placement next year and enjoy myself working.

Table des figures

- 1 : Project concept..... 7
- 2 : Semantic analysis 8
- 3 : Overall view 9
- 4 : Static detection 9
- 5 : Static detection needs..... 10
- 6 : Static detection choice 10
- 7 : Static detection results..... 11