

Chao Jiang's Bachelor Graduation Project

INDOOR POSITIONING SYSTEM



Chao Jiang
Fontys University of Applied Sciences
June 8 2012

GRADUATION REPORT
FONTYS UNIVERSITY OF APPLIED SCIENCES
HBO-ICT: English Stream

Data Student:	
Family name, initials:	Jiang, C.
Student number:	2184760
Project Period:	Feb 1st 2012 - June 30th 2012
Data company:	
Name company:	Neotrack Systems B.V.
Department:	Mobile Development
Address:	Twinning Center Eindhoven, TU/e De Zaale 11 5612 AJ Eindhoven
Company tutor:	MSc. Yongxin Xu
Family name. initials:	Xu, Y.
Position:	Technical Director
University tutor:	Peter Boots
Family name, initials:	Boots, P.J.H.M.
Final report:	
Title:	Indoor Positioning System
Date:	June 8 2012

Approved and signed by the company tutor:

Preface

The report mainly presents the process of the project, *Indoor Positioning System*, which was conducted in Neotrack Systems BV. Neotrack is a start-up company mainly focusing on social network systems based on Location-Based Service (LBS) technology.

I am honored to execute this project from the beginning of February to the end of June as the graduation project of my Bachelor Degree. I am the only student involved in this graduation project.

The objective of this project was to revive and improve the legacy indoor positioning system. The current system was constructed during last three years, which can detect users' position inside a building in a relative high accuracy. The goal of the assignment is to first recover the system from a hibernate status and then find some possibilities to improve it, for instance, make it more convenient to use and more flexible to integrate into other systems.

During this project, I recovered the entire system from non-maintained state and achieved the recovery, analysis, improvement and test work by following the Software Archaeology methodology with quite sufficient documenting work. MSc. Yongxin Xu, the company tutor of me, helps me a lot on schedule the progress of this project and tackle with the stage plan.

I want to express my gratitude to both the school tutor Peter Boots and the company tutor MSc. Yongxin Xu, for always reminding me to keep tracking on schedule of the project and timely giving me critical comments and feedback on my work in a patient and precise way.

Finally, I would really like to thank Remon van der Heide, the director of Neotrack Systems. In the past few months, Remon not only taught me how to become an excellent staff with good communication skills, but also gave me enthusiastic encouragement.

Contents

PREFACE	1
SUMMARY	5
 I A Challenging Assignment: Indoor Positioning System	 6
Chapter 1 INTRODUCTION	7
1.1 The Problem	7
1.2 About the Company	7
1.3 The Assignment	8
1.4 The Strategy	8
1.5 Structure of This Report	8
Chapter 2 ABOUT THE COMPANY	10
2.1 The History of Neotrack Systems	10
2.2 Location	10
2.3 Market Position and Products	10
2.4 Team	11
2.5 Current Clients	11
2.6 My position	11
Chapter 3 ASSIGNMENT OVERVIEW	12
3.1 Initial Situation	12
3.2 The Purpose of the Project	14
3.3 The Expectation of the Company	14
3.4 Assignment Description	14
Chapter 4 RESEARCH	16
4.1 Information Gathering	16
4.2 Used Literature	16
4.2.1 Books	16
4.2.2 Papers, Reference from Internet	16
4.3 Techniques	17
4.3.1 Java	17
4.3.2 Android	18

4.3.3	RMI	18
4.3.4	Webservice	19
4.4	Software Tools	20
4.4.1	IDE	20
4.4.2	Code Analysis Tool	20
4.4.3	Web Container	20
4.4.4	Version Control System	20
4.4.5	Project Management System	21
4.5	Development Environment	21
4.5.1	Operating System	21
4.5.2	Hardware	21
4.5.3	Testing Device	22
 II An Adventure of Software Archaeology		23
Chapter 5 METHODOLOGY		24
5.1	A Short Introduction of Software Archaeology	25
5.2	Tools And Techniques	25
5.3	Practice This Methodology in This Project	26
Chapter 6 PROJECT PHASES		27
6.1	Phase 1: Preparation	27
6.2	Phase 2: Complete and Improve	27
6.3	Phase 3 (optional): Integrate More Possibilities	28
Chapter 7 PHASE 1.1: SYSTEM RECOVERY		29
7.1	The Task And The Process	29
7.1.1	Get The Executable Software	29
7.1.2	The Recovery of Source Code	29
7.1.3	Access The Database	29
7.1.4	The Recovery of Document Work	30
7.2	Tackle With The Problems	31
7.2.1	Make A Short Plan	31
7.2.2	Discover The Legacy System	31
7.3	The Result	31
Chapter 8 PHASE 1.2: UNDERSTAND THE SYSTEM		32
8.1	The Task And The Process	32
8.1.1	The User Requirement	32
8.1.2	New techniques for me	32
8.1.3	The source code projects	32
8.1.4	The Server Side	33
8.1.5	The Client Side	33
8.1.6	API Documentation	34
8.2	Tackle With The Problems	34

8.2.1	Figure Out The Code	34
8.2.2	Fast Learn	36
8.2.3	The Architecture	36
8.3	The Result	37
Chapter 9	PHASE 2.1: SYSTEM REPARATION	39
9.1	The Task And The Process	39
9.1.1	Wipe Out Errors	39
9.1.2	Related Documentation	40
9.2	Tackle With The Problems	40
9.2.1	Adjust the Plan	40
9.2.2	Communication in The Company	40
9.3	The Result	40
Chapter 10	PHASE 2.2: IMPROVEMENT	41
10.1	Tasks and Process	41
10.1.1	Analysis	41
10.1.2	The Improvement of The System	42
10.2	The Result	42
	CONCLUSION	43
	RECOMMENDATION	44
	EVALUATION	45
	BIBLIOGRAPHY	45
	Appendix A: Project Plan	47

Summary

The Company Neotrack Systems is a start-up company located in Eindhoven University of Technology, the Netherlands. Open, Trendy and Innovative is the slogan of Neotrack. Although it is a small and starting company, Neotrack owns the top talented people on innovation design, computer science, sales&marketing. It is an extremely passionate and smart company.

The Initial Situation of this project is quite a challenge. Since there is no document work and full version of the source code, no extra information to help me recover and understand the system. So the initial situation that I face to is very tough. But fortunately, the quality of the source code, as I can understand, is highly standard.

The Assignment of this project is to first recover the whole system includes the mobile application on smart phones and the backend (a large set of server side component). What the company expected is, first, reviving the system to an active status. Secondly, I need to find possibilities to improve the current system. Thus, the assignment is kind of open assignment.

The Approach, or in academic way of speaking, the methodology that I try to follow, is software archaeology. According to my experience, there is no more suitable approach to tackle with this kind of assignment than software archaeology, since what I supposed to do is to recover and improve a system that nobody in the company knows at present. Software archaeology is the methodology to guide engineers to address this kind of problems.

The Result of this graduation project, in my opinion, is more or less satisfying. After devoting a lot of efforts, the indoor positioning system switches to an active status and I achieve a lot of documentation work to guide people continuously improve it in the future.

In Conclusions, by participating in this project, I acquired quite sufficient skills and experience to confront with difficult situations the future career of me. Also, I was influenced by the way of communication and corporation inside the company.

Part I

A Challenging Assignment: Indoor Positioning System

Chapter 1

Introduction

1.1 The Problem

Now GPS and LBS applications are ubiquitous, Google Maps, Foursquares, TomTom, etc. But one of the bottlenecks of GPS technique and other applications that rely on GPS technology is that those applications are not suitable indoor, in part because of weak or non-existent received satellite signals, but also because the channel between the GPS receiver and the satellites is severely and unpredictably affected by building structures. Therefore, when you are located in a very huge indoor environment, you easily lose your way. The purpose of Neotrack System of building the Indoor Positioning System is to guide people find their own positions when they are in such big venues, conference centers, shopping malls, stadiums, etc.

Neotrack took a lot of efforts on realizing this system. Finally it came true. But because of some internal issue, one year ago, Neotrack met its trough in a short period. During that period, most of the software engineers left the company. Since that moment, this system was in lack-of-maintain state.

But as everybody can see, this system has an extremely huge usage in the real world. And now Neotrack passed through the lowest point and started getting better and better. Indeed, Neotrack desired to revive this project from the hard disk of the workstations. Now, inside the company, nobody knows this system, no document, just one copy of source code.

1.2 About the Company

Neotrack Systems BV focuses on social network based on physical meetings assisted by the indoor localization. One of goals of the system is to navigate the users toward each other or the target locations.

About two years ago, Remon van der Heide, the director of Neotrack Systems decided to build an indoor position system to help people find their current positions in indoor environment. Also, Neotrack has some other social network systems.

1.3 The Assignment

As we can see in the first part of this chapter, the problem was described. There might be some chances to improve the software, but the first thing that need to be addressed is getting the clear picture of the whole system, to find out the correct version of the source code. Finally, achieve the goal of making this system active again and using this technology to help people in the real world.

1.4 The Strategy

The strategy is first to make clear that what they (programmers once in the company) have, then solve some fatal bugs, etc. Generally, I will follow the software archaeology methodology to tackle with the tasks, but I will also adjust my plan to adapt the changing situation.

In the second part of this report, I will give a detailed introduction of the software archaeology methodology which I use during this project.

1.5 Structure of This Report

This report is divided into two parts. In the first part, it describes the assignment and other information related to the assignment, like company information, the goal of the project, etc. In the second part I will mainly introduce the process of the project. In the end is the evaluation of this project. Also, preface, glossary, bibliography, appendix are all included in this report.

There are four chapters written in the first part. The chapter of Introduction is an overview of the project. The followed chapter is About the Company, which gives an introduction of Neotrack Systems. The third chapter is the assignment overview. The fourth chapter is the research information towards this project.

The second part of this report consisted by seven chapters. Here I put the project phases information as the first chapter of this part. The Methodology chapter gives out the concept of Software Archaeology. After the

methodology part, I use several chapters to describe the three phases (actually it is two phases happened, because according to the timetable, I have no time to start phase 3, the optional phase) of the project. First phase consisted by *system recovery* and *understand the system*. Second phase includes *reparation* and *improvement*. In the end of this part, is the conclusion.

Chapter 2

About the Company

2.1 The History of Neotrack Systems

Neotrack Systems was founded by Remon van der Heide three years ago. This company mainly focuses on social network system based on indoor positioning technology. In the beginning, Neotrack attracted a lot of top talent people working together. They built fantastic indoor positioning system and achieved a lot of excellent code work. After a long run, Neotrack restructured the team, a more flexible, simplified team was established to tackle with future challenges.

Last year, Neotrack established a sub firm called Qbengo BV, which produces mobile indoor navigation systems for big locations. Qbengo inherits Neotrack's innovation spirit. By using the technology from Neotrack, Qbengo built high standard software in another total different application domain.

2.2 Location

Neotrack located in the campus of Eindhoven Technical University. Eindhoven, which has the title of "The Smartest Region in the World", created thousands of innovation companies. Neotack is one of them. In Eindhoven, smart people are everywhere, and smart companies always find collaboration opportunities to create more and more success stories. Neotrack is also a member of Brainport Eindhoven.

2.3 Market Position and Products

The main branch of Neotrack is social network systems based on indoor positioning system. The goal of the company is to build an interactive platform for other people to get information about each other in indoor environment

like parties, exhibitions, job events, etc. Guiding users to meet each other physically according to the information they get from the system, which is called Match&Meet. By now, this system is still in developing process.

The sub branch, Qbengo, holds the technology of indoor navigation. The product that Qbengo BV built, is called Qbengo app, is a tailor-made mobile application for exhibition world. By using this app, visitors of exhibitions can easily get information about the stands and stands holders. It is not only valuable for visitors but also gains add value for stand holders in business perspective. Now the application is available both on App Store and Google Play Market.

It is obvious that, such kinds of indoor positioning and indoor navigation technology have a huge potential in the real life. The vision of Neotrack is not limited on exhibition world. Now we are also interested in build positioning and navigation software for museums, campuses, shopping malls.

2.4 Team

Remon van der Heide, the director of Neotrack, is a top talent sales&marketing people who take charge of communicating with potential clients and convincing them of our services. He did do remarkable work in business field. He can also explain the general overview of our service in technical perspective.

Neotrack now has multi approaches to implement software. We have three developers, mainly focus on project management and R&D part. We also have strategic partners in Netherlands which build software for us. Recently, we built a collaboration laboratory in China, which composed by three lecturers and several master students, to complete both development and research work.

2.5 Current Clients

Our current clients include Eindhoven Technical University, Evoluon, TEDx Brainport, nH Koningshof, etc.

2.6 My position

My position in is project, is the project leader. MSc. Yongxin Xu, technical director of Neotrack, will play the role of both formal client and supervisor of this project. During the graduation project, MSc. Yongxin Xu has the right to dynamically adjust the requirement of this project according to the feedback from top level of the company.

Chapter 3

Assignment Overview

3.1 Initial Situation

When I start this project, I noticed that what I can get, no matter the source code or the document work, was quite a challenge for me. There is only one copy of the source code, which cannot be compiled because of some errors. At the beginning there is no document handing in to me. I began to aware that the information which can help me understand the system is insufficient. There are some risks existing to take over this kind of project. Nevertheless, there should be a right approach to such kinds of situation, for instance, reverse engineering.

After roughly looking inside the source code, I found that the comments

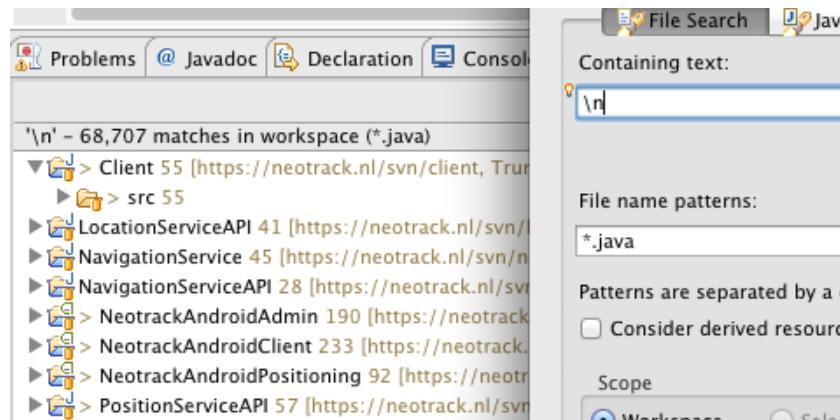


Figure 3.1: Source code statistic of client side

between the source code can explain more or less of some functionalities. According to my three years work experience in Java platform, in my opinion the quality of the source code is high standard.

As you can see in the figure, only for the client side, there are almost 70 thousand lines of source code (*see Figure 3.1*). For a single purpose software, I think it is quite a lot source code involved according to my knowledge. The client side is developed based on Android platform and Java, and the server side is based on Java and RMI technology.

The theoretical background of this system is based on Yongxin Xu's Master thesis (10). The system uses WiFi signals to determine the position of a client. There are several approaches of achieving indoor positioning based on WiFi signals(7). This system implements one of these methods, which called location fingerprinting(6). In simple terms, location fingerprinting-based positioning is by dividing the space into small regions and characterizing the signal properties in each one of these regions. Eventually, the system will match the fingerprinting that your device just captured to a small region in this location. (*see Figure 3.2*)

The server side program of this system provides Web service interface to

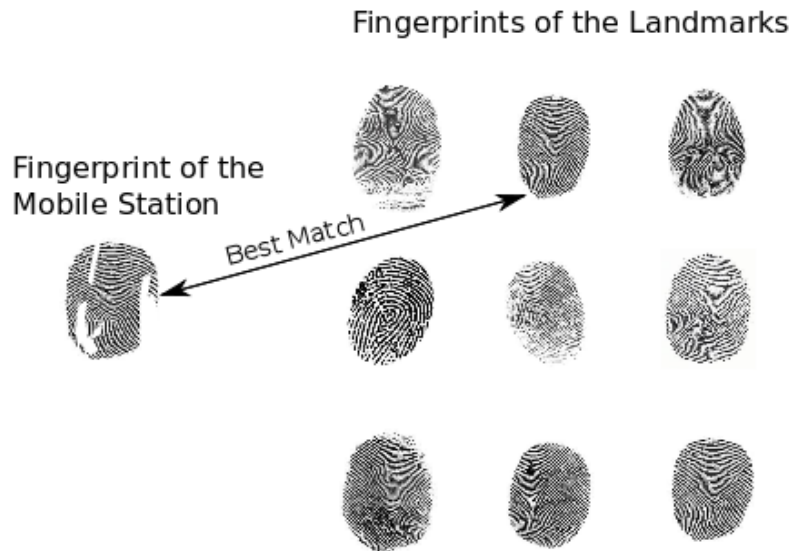


Figure 3.2: A metaphor: How the WiFi fingerprinting works

communicate with clients. It can accept the WiFi fingerprinting from device and calculates and return the position information to the device. Functions of the client side seems simpler: collecting WiFi signals and sending to server. When server return the position information, the client side display it inside a digital map on the screen.

3.2 The Purpose of the Project

The purpose of this project is to re-active the indoor positioning system. It is obvious that the system has a huge potential in real world application. Imagining that if this indoor positioning system becomes true in the real life, everyone will benefit from it. No matter how big and complex a venue or a building is, you can always find your current location and eventually you will be in the right route.

3.3 The Expectation of the Company

The general expectation of the company about this project is quite simple: Revive the indoor positioning system. Simply this task can be separated to three phases.

The first phase of course is to totally recover every part of the system, figure out a functional version of the source code and achieve sufficient document work on this system. Besides the indoor positioning application in Android platform, this system also contains a complex database, a backend system on the server side which calculates the position of a client, and the server side backend also contains several parts. Finally, all of these parts of the indoor positioning system are supposed to be in an active state and any programmer who getting sufficient knowledge can work with it.

The second phase is to complete and improve the current system we have. According to the initial situation, the improvement of the system mainly depends on the work of recovery. If there are some bugs existing, then the improvement can be fixing bugs. If after recovering, the system works quite well, then we can find possibilities to improve functions and add new features. So after the whole project, there should be some improvements in some aspects.

The third phase is to find possibilities on add new features and integrated the system to other systems. This part is optional, will start only when there is enough time left.

3.4 Assignment Description

- **Achieve Compilable and Maintainable Source Code**

It is a common sense that a compilable and maintainable source code is one of the most important part of a software project. According to the initial situation, this part still need to be fixed. After that, a ver-

sion control system for this project can also be established to maintain the source code versions.

- **Achieve Sufficient Document Work**

Document work is also a significant part of a software project. An insufficient document work can lead to big problems according to the software development history in last 40 years. In this assignment, I need to work out enough document work to explain this indoor positioning system. For future extension or improvement of this system, document work will be very helpful.

In this case, I had to add class diagrams, function descriptions, block diagram of the architecture, etc.

- **Achieve Improvement**

If the two previous points can be achieved within an acceptable time, then the improvement step should start. In this case, improvement means improve current system in one or more aspects, add new features, improve user interactive design, etc. There are a lot of possibilities, the plan is first finishing first two points, then executing the improvement part.

Chapter 4

Research

4.1 Information Gathering

During this project, I use several ways to gather information.

The main approach I use is searching for sources of information on Google. Search engines are an excellent tool to identify relevant sources of information. It creates the possibility of focusing on specific categories of information, such as from books, thesis or other scholarly sources.

I also go online and locate local library locations. Sometimes it is not really necessary to read the entire volume. Information relevant to my project can be always found in a specific chapter or by using the index located at the end of the books.

4.2 Used Literature

4.2.1 Books

- *Hello, Android: introducing Google's mobile development platform*(1) An excellent book about basic knowledge for developing Android applications.
- *Professional Android 2 application development* (8) An advanced guide-book for Android development.

4.2.2 Papers, Reference from Internet

See REFERENCES part of this report.

4.3 Techniques

4.3.1 Java

Generally speaking, the indoor positioning system contains two main parts: The server side program and the client. The server side program is implemented in Java technology. The client which is running on Android smart phone OS, is also developed in Java language.

There's no doubt that Java nowadays is the biggest part of programming market. But the reason that Neotrack chose Java as the developing platform is not only because of the market share but also Java itself is a very flexible and powerful tool to build software. So what is Java? What are the advantages of Java?

JAVA is an object-oriented programming language and it was intended to serve as a new approach to manage software complexity. Java refers to a number of computer software products and specifications from Oracle that together provide a system for developing application software and deploying it in a cross-platform environment. Java is used in a variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end. Java is nearly everywhere in mobile phones, Web servers and enterprise applications, and while less common on desktop computers.

The well known advantages of Java platform include:

- It is an open source, so users do not have to struggle with heavy license fees each year
- Platform independent
- Java API's can easily be accessed by developers
- Java perform supports garbage collection, so memory management is automatic
- Java always allocates objects on the stack
- Java embraced the concept of exception specifications
- Multi-platform support language and support for web-services
- Using JAVA we can develop dynamic web applications
- It allows you to create modular programs and reusable codes

4.3.2 Android

The mobile client of indoor positioning system is based on Android SDK version 2.2. Actually, the market of smart phone market is occupied by Android and iOS. Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs to begin developing applications on the Android platform using the Java programming language.

Features of Android platform:

- Application framework enabling reuse and replacement of components
- Dalvik virtual machine optimized for mobile devices
- Integrated browser based on the open source WebKit engine
- Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- SQLite for structured data storage
- Media support for common audio, video, and still image formats
- GSM Telephony (hardware dependent)
- Bluetooth, EDGE, 3G, and WiFi (hardware dependent)
- Camera, GPS, compass, and accelerometer (hardware dependent)
- Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

4.3.3 RMI

The server side program of this indoor positioning system has a very flexible structure. There is a central server program which uses Web service interface to represent services behind. And all of those services (like calculation position) are designed as a component or so-called plug-in that connected to the server through RMI protocol.

RMI (Java Remote Method Invocation)(2) enables the programmer to create distributed Java technology-based to Java technology-based applications, in which the methods of remote Java objects can be invoked from other Java virtual machines, possibly on different hosts. RMI uses object serialization

to marshal and unmarshal parameters and does not truncate types, supporting true object-oriented polymorphism.

Advantages of RMI technology include:

- Handles threads for you
- Handles Sockets for you
- Nice GC of lost clients. ie Unreferenced
- Marshalls objects for you
- Dynamic loading of classes are available
- Can also make changes on the server end, that might not mean you need to change anything on the client side

4.3.4 Webservice

Although RMI is a very advanced and high efficient way to connect to program through network, there are also some drawbacks existing. For instance, RMI strictly attached on Java Technology, and it is not generally supported as much as Web service (which normally based on HTTP).

According to those disadvantages, Neotrack chose to construct a front-end interface to represent services that the central server provides. And certainly, this front-end interface works as a Web service interface. Thus, no matter what kind of technology the clients use, they can always communicate with the server through Web service interface.

Web Services offer many benefits over other types of distributed computing architectures.

- Interoperability - This is the most important benefit of Web Services. Web Services typically work outside of private networks, offering developers a non-proprietary route to their solutions. Services developed are likely, therefore, to have a longer life-span, offering better return on investment of the developed service. Web Services also let developers use their preferred programming languages.
- Usability - Web Services allow the business logic of many different systems to be exposed over the Web. This allows you to develop services and/or client-side code using the languages and tools that you want.

- Reusability - Web Services provide not a component-based model of application development, but the closest thing possible to zero-coding deployment of such services.
- Deployability - Web Services are deployed over standard Internet technologies.

4.4 Software Tools

4.4.1 IDE

- Since the server side program of the indoor positioning is based on J2EE, I chose **Eclipse JEE Edition** as the integrated development environment. The server side program refers JEE libraries, like Servlet library, etc. Eclipse JEE edition provides those libraries originally. Also, I can configure the web container Tomcat to integrate into this IDE.
- **Eclipse Standard Edition with Android ADT** is a conventional tool to develop Android application. It is a platform-independent environment for developers. Google provides Android Development Tool plugin for Eclipse. Android simulator is also included in this toolkit.

4.4.2 Code Analysis Tool

- **AgileJ StructureViews** is a Java Eclipse plugin supporting configurable reverse engineered UML class diagrams. In this project, such kinds of reverse engineering tools will be used very frequently according to the undocumented source code.

4.4.3 Web Container

- **Apache Tomcat** is an open source software implementation of the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed under the Java Community Process. In the indoor positioning system, Tomcat is deployed as the web container of the server front-end Eeb service.

4.4.4 Version Control System

- Version Control System is the management of changes to documents, computer programs, large web sites, and other collections of information. **Apache Subversion** (often abbreviated SVN, after the command name svn) is a software versioning and revision control system distributed under an open source license. Developers use Subversion

to maintain current and historical versions of files such as source code, web pages, and documentation.

Apache Subversion is a full-featured version control system originally designed to be a better CVS. It contains powerful features, and got a lot of documentations support. Better still, for some IDE, there are SVN plugins integrated inside.

4.4.5 Project Management System

- Project Management System is a term covering many types of software, including estimation and planning, scheduling, cost control and budget management, resource allocation, collaboration software, communication, quality management and documentation or administration systems, which are used to deal with the complexity of large projects. **Redmine** is a free and open source, web-based project management and bug-tracking tool. It includes a calendar and Gantt charts to aid visual representation of projects and their deadlines. It handles multiple projects. Redmine provides integrated project management features, issue tracking, and for multiple version control options.

The reason that we chose Redmine is the seamless integration of Redmine and SVN. At the same time we can manage the project and the source code repositories only through Redmine.

4.5 Development Environment

4.5.1 Operating System

- **Ubuntu** is one of the best Linux distributions in the world. It provides enough efficiency on document work and developing work to software engineers. It is totally free. Also, it is very user-friendly. The main asset of Ubuntu is that the creators and developers made this distribution easy to install, maintain and use.

4.5.2 Hardware

- We use **Dell Workstation** as the developing workstation for this project. Hardware inside is high standard, stable and secure.
- We deployed our server side program on a **Dedicated Debian Server in Clouds**, to make sure the calculation part of the system can work

out. Neotrack use this server for almost three years. There are several systems running on this server. The performance of the server is sufficient to run the server side program.

4.5.3 Testing Device

- **HTC Desire** provides sufficient performance for testing our system. The Android version of this smart phone is 2.3, and during the testing phase, it works in a high accurate when captures WiFi signals.
- **Linksys Wireless Router** is the WiFi router we used during the test phase. It provides stable and powerful WiFi signal for wireless devices. We use a set of routers to construct a simulation indoor WiFi environment.

Part II

An Adventure of Software Archaeology

Chapter 5

Software Archaeology Methodology

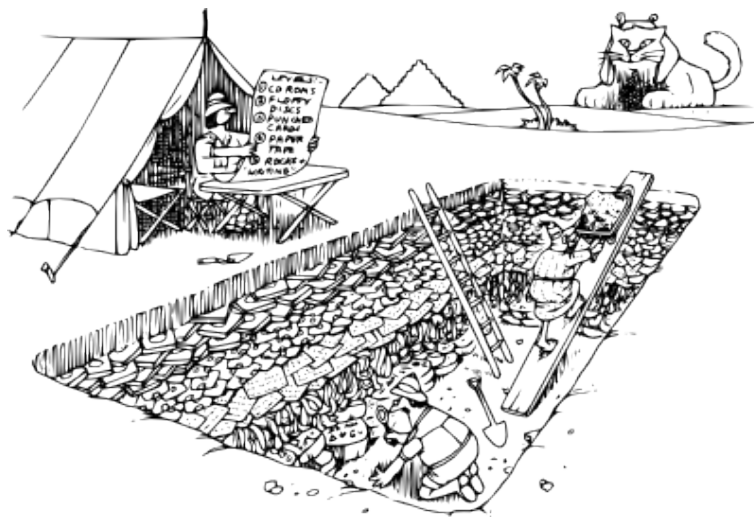


Figure 5.1: Software Archaeology: Digging the Legacy System

“This isn’t programming, this is archaeology.” the programmer complained, wading through the ancient rubble of some particularly crafty pieces of code. It’s a pretty good analogy, actually. In real archaeology, you’re investigating some situation, trying to understand what you’re looking at and how it all fits together. To do this, you must be careful to preserve the artifacts you find and respect and understand the cultural forces that produced them. But we don’t have to wait a thousand years to try to comprehend unfathomable artifacts. Code becomes legacy code just about as soon as it’s written, and suddenly we have exactly the same issues as the archaeologists. What are we looking at? How does it fit in with the rest of the world? And what were they thinking? It seems we’re always in the position of reading

someone else's code: either as part of a code review, or trying to customize a piece of open source software, or fixing a bug in code that we've inherited.(4)

5.1 A Short Introduction of Software Archaeology

“software archaeology” (4) or software archeology is the study of poorly documented or undocumented legacy software implementations, as part of software maintenance. Software archaeology, named by analogy with archaeology, includes the reverse engineering(5) of software modules, and the application of a variety of tools and processes for extracting and understanding program structure and recovering design information. Software archaeology may reveal dysfunctional team processes which have produced poorly designed or even unusable software modules. The term has been in use for several decades, and reflects a fairly natural metaphor: a programmer reading legacy code may feel that he or she is in the same situation as an archaeologist exploring the rubble of an ancient civilization.

Software archaeology has been generally used for large old (legacy) systems, but it is valid for any type of software with independence of its age and size. While maintaining a given piece of software, developers have to understand source code that has usually changed many times in the past, producing a result which is the addition of all those changes. If the code is stored in a version control system, its complete history is available and can be analyzed with appropriate tools.

5.2 Tools And Techniques

There are several tools and techniques that frequently used in software archaeology:

- Scripting languages for filtering diagnostic output
- Ongoing documentation in basic HTML pages or Wikis
- Synoptic signature analysis, statistical analysis, and visualization tools
- Reverse-engineering tools such as Together's ControlCenter
- Operating-system-level tracing via truss and strace
- Web search engines and tools to search for keywords in source files
- IDE file browsing to flatten out deep directory hierarchies of the source

- Test harnesses such as Junit and CPPUnit
- API documentation generation using Javadoc, doxygen, and so on
- Debugger

5.3 Practice This Methodology in This Project

The first problem that I met is to choose an appropriate methodology for this project. In the beginning, I thought a traditional approach of system development might be suitable for this project. Soon after, I found that the initial situation that I am facing is not so optimistic. I need to find an alternative solution to tackle with this legacy system. After quite a while searching of related information, I found an article (9) that describing a way of working with old undocumented legacy software systems, which is called Software Archaeology. In this article, the authors explained how to handle with legacy systems by using archaeology as a metaphor. Also they gave some suggestions when practice software archaeology. After thoroughly reading this article, my interest has been aroused.

After that, I gathered more information about software archaeology. Because that this is not a typical approach of software engineering, there is almost no book related to this topic. Most of them are papers and articles, but a lot of valuable information included. The most used tools and approaches in software archaeology were listed in these materials. There are also some phases mentioned in some papers which show me an overview of the process. It might not be exactly suitable for every legacy system since there is a variety of situations existing in legacy systems. However, at least the project phases that I made was following the philosophy of software archaeology.

Software archaeology provides an interesting framework for digging in the past of a project, so that we can learn patterns and information relevant to infer its future. In this case, the system is undocumented and costs a very long time to build, so there is huge information for me to dig. In my opinion, this is a perfect methodology that I can follow when I tackle with this project.

Chapter 6

Project Phases

This chapter shows a clear picture of the different stages of the project.

6.1 Phase 1: Preparation

Prepare the development software and hardware environment.

- **Phase 1.1: Recover the system**

The first task that I need to do is to get the complete source code of the software which was built by the software engineers before. The initial situation is that there is no version control system running or known by the current employees in the company. And also no document work and database hand to me. So I separate this work as an independent part of this project.

- **Phase 1.2: Understand the system**

After getting the undocumented source code, I need to get familiar with it. Multi approaches to understand the source code will be applied, such like debug, reverse engineering, search engines. The software archaeology might be helpful to assist me to understand the source code.

Deliverable: User requirement, descriptions of the functions, and other related document work.

6.2 Phase 2: Complete and Improve

-

- **Phase 2.1: Reparation**

After getting the source code, I need to compile it to executable file and test it. There is a big chance that the source code I got do not match the executable file. So I need to check.

I was also informed that in this system there are some bugs already existing. For my understand, firstly, I need to fix those bugs. The output of this task is supposed to be a compilable version of the source code. There are several independent source code projects which are considered as independent API for the system. The API should also be stable with the description of these API.

- **Phase 2.2: Improvement**

If any part of this system needs to be improved, I will investigate the possibility of improving it. If the time is affordable, I will take the responsibility to improve it.

Deliverable: Valid source code with sufficient documentation.

6.3 Phase 3 (optional): Integrate More Possibilities

This is an optional phase. This phase will start only when there is sufficient time left. The possibilities can be combining the positioning system and the routing function based on Dijkstra's algorithm, or integrating the system into the company's product which is already on the market.

Deliverable: System integration document and new feature-update document

Chapter 7

Phase 1.1: System Recovery

7.1 The Task And The Process

7.1.1 Get The Executable Software

When this project started formally, I got the install file of the positioning application in Android platform. I successfully installed it to the test device, a HTC Desire smart phone. I launched the application and tried out for a while. There is a version of server side program running on our server which seems remained there for a long time according to the timestamp. I tried to connect the mobile application to the test server in our office, which the server side program running on it. The positioning function performed very well in the 5th floor of Twining Center. (*see Figure 7.1*) And I more or less got the idea of the architecture of this system.

7.1.2 The Recovery of Source Code

After getting the source code, I found some reference errors in the source code project. The source code project cannot be compiled. A lot of information missed. Some of the missed libraries are open source library, so I can find them on the Internet. Some seems tightly related to this project. Fortunately, when I was checking the server of Neotrack, I found a SVN system installed on our server which contains all the source code repositories of this system.

7.1.3 Access The Database

Soon after, I realized that I need to take a look at the database. But first I need to find out where the database is deployed. Also what kind of database system they used? Fortunately, according to my experience on J2EE development, I found out the database configuration information

Before the project started, I was informed that there is no documentation work left for this system. But now at least I found something that can assist me to understand this system.

7.2 Tackle With The Problems

7.2.1 Make A Short Plan

Now the initial situation is clear for me and seems not so optimistic situation. According to my experience, when you work in real software industry, this kind of situation happens a lot. To tackle with this kind of unpleasant situation, I need to make a short plan to give a rough estimation on time. After the short plan was made, I transferred the plan to my agenda (learning from my colleague).

7.2.2 Discover The Legacy System

When my supervisor gave me the access of our dedicated server, I thought there might be some valuable information on it. Finally I found a SVN system installed on this server, and there are 26 source code projects inside, including the indoor positioning. Because what I got in the beginning is just one unknown version of the source code, but now it seems that I can browser the version control system and check the update logs of the source code. By using the same way, I also found some piece of the requirement and design document on the workstation.

In my opinion, when we face to this kind of legacy system, we need to try multi approaches to gather information.

7.3 The Result

- Runnable application. It is just installation file for Android. This version is from the legacy system, not compiled from the source code that I got.
- SVN system with all source code projects. This repository should be the one once the company used for the development.
- Some incomplete document. For example the specification of some function, some Use Case diagrams, etc.

Chapter 8

Phase 1.2: Understand of the System

8.1 The Task And The Process

8.1.1 The User Requirement

There is no complete user requirement specification. What I found is some use case diagrams, which can more or less give me some feeling about this indoor positioning system. But it is still not enough as the user requirement. According to the school tutor, Peter Boots' advice, the requirement document is unnecessary to be found out when the system was already been built. I agree with that point.

8.1.2 New techniques for me

After looking inside the source code, I found that my brain need to be "charged" for some new techniques appeared in this system. Time is very short for me to get familiar with those terms, like RMI, WebSocket(3), etc. Although I did some preparation on Android development, it is not enough for me to handle with a complex application like this. So I need to figure them out faster.

8.1.3 The source code projects

There are more than twenty source code projects related to this system. (*see Figure 8.1*) I can understand the general function of some of these source code projects by the name, but it is very hard to get an idea of their exact responsibilities and roles in this complex system.

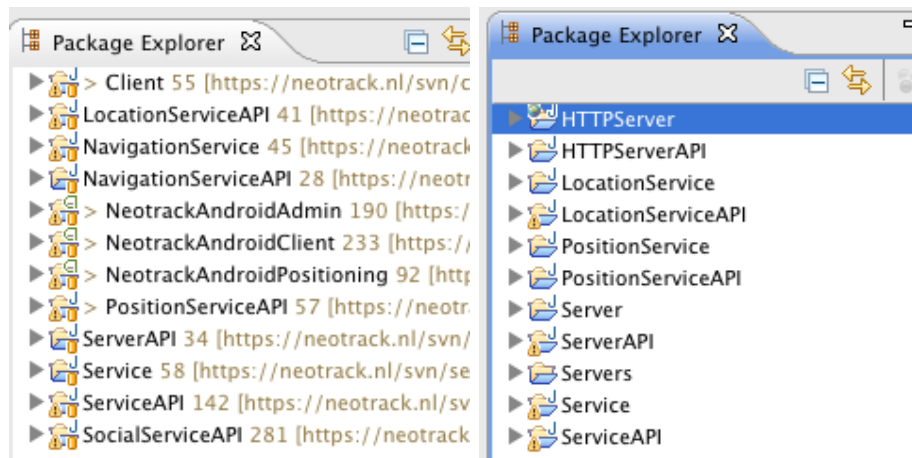


Figure 8.1: Client side and server side source code projects

8.1.4 The Server Side

According to the deploy plan and my work experience, I tried to understand the entire architecture of this system. I draw some architecture diagram, and then verify my idea by browsing the source code. Although I didn't thoroughly understand all of the code, but at least a clear picture of the system architecture showed up in my mind. I didn't forget to record all the information that I understand to a concrete document.

By figuring out the function of different service component, I started to analyze the code of those service components, for instance the service of calculating user's position. All of these component are supposed to docking to the central server. So I started from the external interface of some components. The components based on RMI technique, so new knowledge simultaneously came to me as well.

Database of this system is also poorly documented. I generated the structure of the database (*see Figure 8.2*) by using reverse engineering tools (The structure diagram generate function in MySQL Workbench (*see Figure 8.5*)) and adding some useful comments for each table based on my understanding.

8.1.5 The Client Side

By using the similar way, I also figured out the structure of the client side by analyzing the source code in different perspective. I also made sure related document work is done in the same time. I thought what I am doing is kind of reverse engineering, but more abstract.

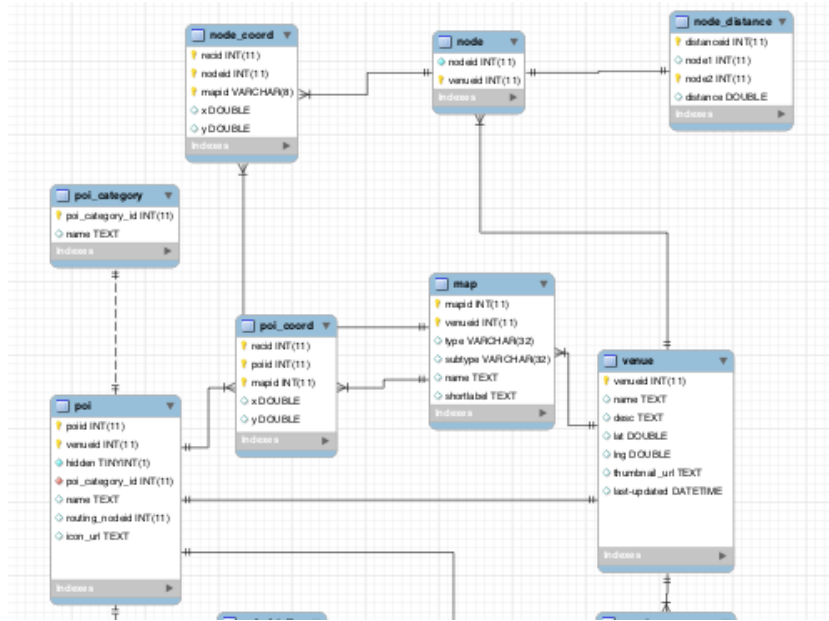


Figure 8.2: Database structure

8.1.6 API Documentation

It is easy for Java developers to use Javadoc to generate all the API comments to a API Documentation. Because of the high standard code work and some useful comments in between of the code, I decided to use Javadoc to generate the API Documentation of all the source code projects involved in this system.

8.2 Tackle With The Problems

8.2.1 Figure Out The Code

For detail implementation, comments in source code can always help us to understand. (see *Figure 8.3*) But the difficult part is, to get the idea of how classes corporate with each other. Normally, before implementation phase, we always have a design phase. The deliverable of design phase is some kind of object-oriented design. For a legacy system, the only way that we can get class diagram is reverse engineering. In this project, I faced to the huge amount of source code. So I tried to use reverse engineering tools to generate some class diagrams (see *Figure 8.4*). It helped a lot when I am trying to understand how those classes corporate in this complex system.

According to the timeline, I didn't have enough time to thoroughly read and understand all of the source code. So I mainly focus on some high-priority

```

import nl.neotrack.communication.message.Message,

/**
 * Abstract class a message serializer.
 *
 * @author Chris Adriaensen (Neotrack Systems)
 */
public abstract class MessageSerializer {

    /**
     * CLASS LOADERS
     */

    /**
     * Add <code>loader</code>.
     *
     * @param loader
     *     The class loader to add.
     * @throws IllegalArgumentException
     *     If <code>loader</code> is invalid.
     */
    public abstract void addLoader(ClassLoader loader)

```

Figure 8.3: Comments between source code

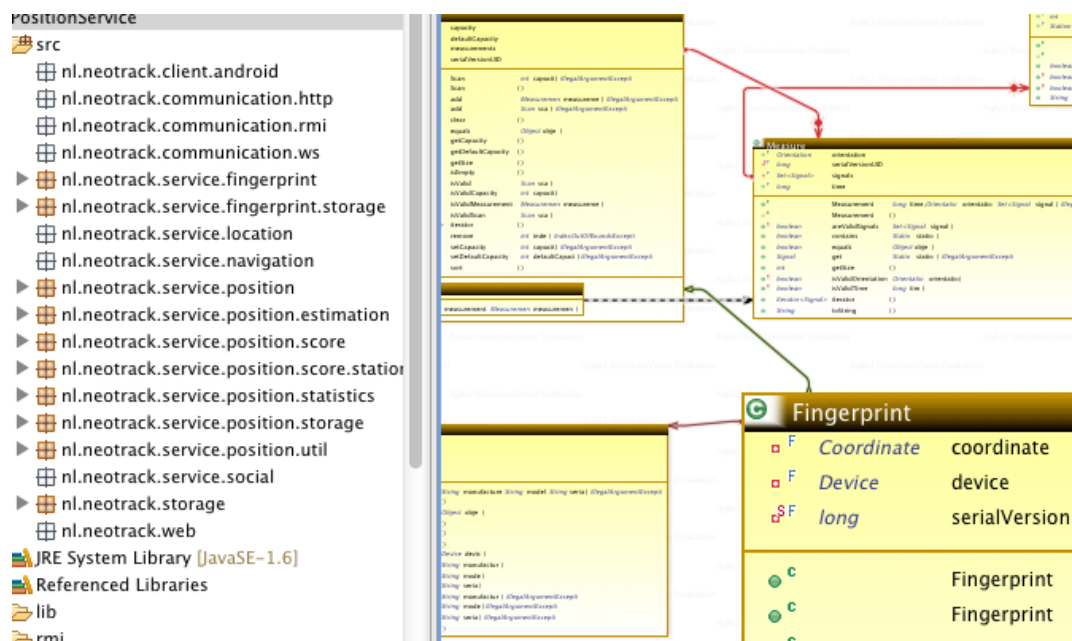


Figure 8.4: Generate class diagrams from source code by using reverse engineering tools

parts, for instance, the central server, the RMI interface, the map screen of the mobile client. Although some part I still need time to investigate, but in total I already got a big picture of this system in my mind, which in my

opinion was enough for me to continue the process.

For some complex classes and methods, I setup some breakpoints and used debug tool to analyze how they work.

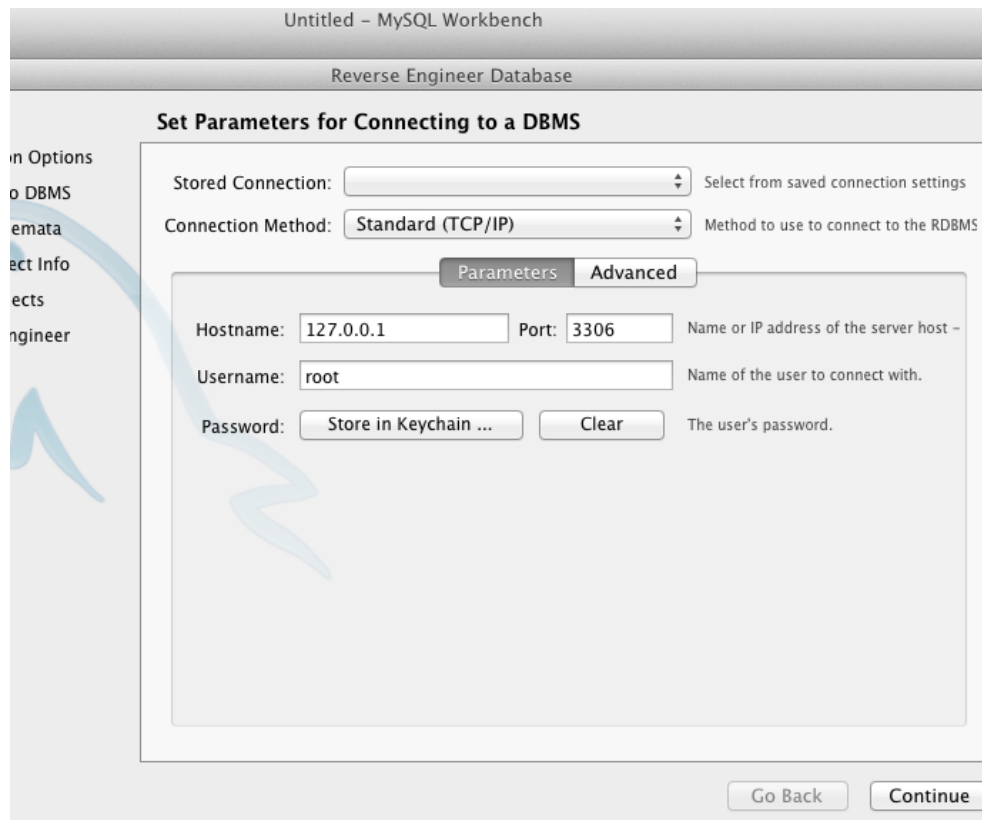


Figure 8.5: Reverse Engineer Tools in MySQL Workbench

8.2.2 Fast Learn

For programmers, I think we should adapt to the “life-long learning” habit. New techniques come when we are sleeping, it is hard to catch up the trend if we say no to these new terms. And especially when you are tackling with a system that based on some techniques that you never heard about it, you will learn faster than just reading a book. Because the motivation of learning is so clear to you, you have no choice on it.

8.2.3 The Architecture

When I analyze the architecture of this indoor positioning system, the work experience becomes helpful. During my three years work experience on

J2EE, I saw a lot of different architectures. Sometimes I even participated in system architecture. That made me acquire the sense of system architecture. When I faced to this kind of complex system, I can rapidly get the big picture of the entire system. The knowledge that I accumulated in past several years also an important key when I analyze this system.

8.3 The Result

- API documentation. This is generated from the source code by myself, by using Javadoc tools.
- Architecture diagrams and descriptions. According to my understanding of this system, I created architecture diagrams (*see Figure 8.6*) of this system.

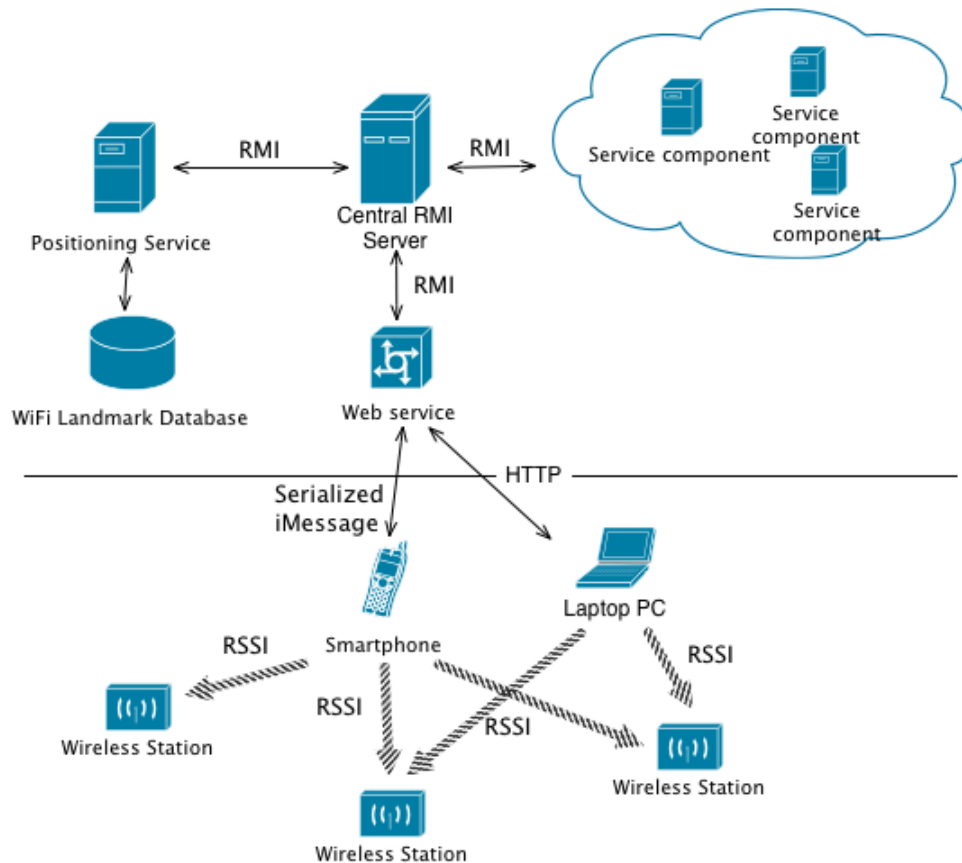


Figure 8.6: One of the architecture diagrams: System Overview

- Figure out some new terms: RMI, WebSocket, etc. Mainly through Internet.
- Function description of each part. I wrote a function description of main functions in the system based on my own understanding.
- Database structure with my comments. By using reverse engineering tools, I generated the structure of the database.

Chapter 9

Phase 2.1: System Reparation

Describe what kind of bugs/problems exist in current system and what kind of decision I made above those bugs/problems.

9.1 The Task And The Process

9.1.1 Wipe Out Errors

After looking in-depth of these source code projects, I started to address some concrete bugs and problems that still existing. In the beginning, I thought the latest version from SVN is adequate for me to start with. Soon I found that some new but unfinished features were added to the latest version. So firstly I need to find the right version of the source code which can match functions of the runnable software I got. Before I can successfully compile the source code to software, all of the red crosses in my IDE should be removed (in some Integrated development environment software, if there is an error in the project, a red cross will be always show there). There are quite a lot errors in the latest version of the source code. For instance, the third party packages that most of the projects referred are missed, and absolute paths were used instead of relative paths.

During this period, I solved all of the reference errors, path errors, the conflicts of versions, etc. I practiced a variety of approaches to wipe out those errors and problems. Some of the approaches were following the way mentioned in papers and articles of software archaeology. For instance, analyzing logs and output, analyzing the output of decompiling or other reverse engineering tools.

9.1.2 Related Documentation

Although it is hard to recover all the document work, I still tried to add supplements on documentation. Afterwards, the documentation work will be helpful for the further development of this system.

9.2 Tackle With The Problems

9.2.1 Adjust the Plan

Normally, there are always some changes during software project because it is very difficult to give a right time estimation on unpredictably development tasks. When I started to tackle with the errors and bugs in this system, I found that the situation is beyond my estimation. It is not possible to fix all of them within the timeline. As a software engineer, I need to handle such situation professionally. I arranged a small meeting to report the progress and the situation to my company tutor. We both agree to adjust the plan to merge more time for these tasks.

9.2.2 Communication in The Company

As a member of the team, no matter what happened, we should keep a sufficient communicate with colleagues and the supervisor. Obviously, now most of the software projects require a very tight teamwork. Above the teamwork, communication between team members is the most important element. When I faced to unpleasant situation on my task, I chose to discuss it with my supervisor. As desired, the team will solve the unpleasant problem together with limited capacities and resources.

9.3 The Result

- Bug free and compilable source code with some comments and documentation.
- Some adjustment on the plan.

Chapter 10

Phase 2.2: Improvement of the System

10.1 Tasks and Process

10.1.1 Analysis

I tried to analyze the system and figure out which functions should be improved and the estimation of time and work to improve it. In my opinion, there are several aspects can be improved.

First, I noticed that when I deal with data of this system, some labor work can be replaced by the machine. For instance, the map tiles used by the client application for displaying floor plans is compressed and cut manually. It is possible to produce a software or a script to handle with this task more efficient.

Second, the architecture of this system is too complex. In the server side, several programs and services need to be launched only for the positioning function. After thoroughly reading the code of positioning service for a long time, I thought a Web service application can meet the demand of positioning function in the server side. I discussed this point with my supervisor, I try to find possibilities to improve it. The supervisor told me that my consideration was very valuable. However, they planned to construct a flexible platform instead of just the positioning function when this system was designed. By using this architecture it is possible to add more service components to the server side, and deploy them as clusters. After this conversation, I realized that this architecture also got its advantages and it is better to keep like this way.

10.1.2 The Improvement of The System

I discussed with my supervisor about the proposal of building utilities for some labor-work task. He fully agreed with me on that. I built a tool for generating and compressing map tiles based on Linux Shell programming and ImageMagick. It can automatically produce the map tiles we want in seconds instead of several hours by manually doing it. These compressed map tiles were used by the client application, to display the floor plan. Instead of using one picture as a floor map, these map tiles were used to optimize the performance of the client side. Also, I implemented a GUI software which can generate the nodes data of the map within few clicks. The nodes data is used in navigation component of the system. This tool can load the floor plan of buildings. By clicking on the map and other simple operations, the coordinates of the nodes and the weight (relative distance between two nodes) can be generated to CSV format. Before they did this work manually.

For the architecture aspect, I agreed with the supervisor's consideration. For legacy software systems, we need time to discover more information about it. There could be some reason for the formal team leader when he chose specified techniques or structures in the legacy system. We need to handle this carefully.

10.2 The Result

- A utility software for generating nodes data for maps.
- A utility software for generating and compressing map tiles.

So far, the goal of this project is more or less achieved. The phases mentioned in original plan were executed. I will give a conclusion in CONCLUSION part of this report.

Conclusion

The indoor positioning system is an excellent system. It is a good interpretation of the spirit of innovation. A relatively high standard coding work has been reached. Also the architecture of the entire system is quite remarkable.

However, in another aspect, the documentation is insufficient. Under such conditions, it is hard for other people to improve or even understand the system. Moreover, the version management work did not receive adequate attention, so versions of the source code is more or less in chaos.

The real situation of this project was actually tougher than I thought. In my opinion, to recover a legacy system sometimes is more difficult than develop a new one. A good methodology in this case becomes concretely significant. By following software archaeology, I realized what kind of tools and approaches should I take when facing undocumented legacy system. Looking back to this project, I essentially benefited a lot from following the right methodology.

During this project, I think that I kept on schedule. Although the stage plan was adjusted and some optional parts were skipped, the final result meets the expectation in beginning. Planning, communication and being patient are what I always insist. I benefited a lot from these elements.

As the result, the indoor positioning system was recovered to a considerably optimistic situation. The latest version of the source code is stable and compilable. The whole system has been tested and performed as well. Documentation work of the system has been partly added. Also some utilities have been built to meet the demand of generating data automatically and efficiently.

Recommandation

During this project, I found that this indoor positioning technique can be applied in several scenarios. Especially when users are located in some big conference centers or exhibition centers. One of my recommendations is to integrate this indoor positioning technique into the new exhibition mobile app, Qbengo app. Now Qbengo app contains indoor maps and navigation function. If the indoor positioning feature can be added to this app, it will be a significant upgrade.

Similarly, we can imagine that there is also a huge usage of indoor technique in museums. By combining users' indoor position information with the information of the artworks or culture relics in the museum, the old fashioned "audio guiding device" will be taken place very soon.

Furthermore, besides the usages of this technique, in my opinion, the way of maintaining the system should be improved. I suggest that Neotrack can strengthen the documentation work of software projects. In traditional software development, documentation work is one of the most important aspects in a project. Even now, the agile way of development suggests reducing unnecessary documentation work, the core documentation still plays an important role in software projects.

Evaluation

Overall, I believe that I achieved number of valuable accomplishments during this project. I practiced the software archaeology on recovering this legacy system. Although I have quite a long time work experience on programming, this is the first time that I face this kind of situation. By participating in this project, I acquired some basic knowledge of how to dig information from legacy system, how to practice reverse engineering, how to analyze large-scale undocumented source code, etc.

However, looking back to my original planning, it took slightly more time than I expected. I think the main reason is that this is the first time that I practice recovering legacy system, there are a lot of uncertainties. The scale of this system was also beyond my expectation. In my opinion, an essential investigation in-depth of a legacy system is very necessary before starting to recover it. It is more desirable to give a time estimation on recovering such system after a thoroughly looking inside.

It is also the first time that I work in a western company. There are big differences between Chinese companies and western companies. Besides the languages, ways of communication, solving problems and handling things are also different. After arriving Europe six months later, I was in the position to adapt the culture differences. Open, innovation, interactive enterprise culture is what I rarely saw in China. I benefit a lot from it.

Bibliography

- [1] E. Burnette. *Hello, Android: introducing Google's mobile development platform*. Pragmatic Bookshelf, The, 2009.
- [2] T.B. Downing. *Java RMI: remote method invocation*. IDG Books Worldwide, Inc., 1998.
- [3] I. Fette and A. Melnikov. The websocket protocol. 2011.
- [4] A. Hunt and D. Thomas. Software archaeology. *Software, IEEE*, 19(2):20–22, 2002.
- [5] R.E. Johnson. Reverse engineering and software archaeology. *The DoD SoftwareTech News*, 8(3):4–8, 2005.
- [6] K. Kaemarungsi. *Design of indoor positioning systems based on location fingerprinting*. PhD thesis, University of Pittsburgh, 2005.
- [7] B. Li, J. Salter, A.G. Dempster, C. Rizos, et al. Indoor positioning techniques based on wireless lan. In *First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications, Sydney, Australia*, pages 13–16, 2006.
- [8] R. Meier. *Professional Android 2 application development*. wrox, 2010.
- [9] A. Schneider and P. Windle. Software archaeology. *The DoD SoftwareTech News*, 8(3):9–13, 2005.
- [10] Yongxin Xu. A study on indoor positioning system. Master's thesis, Technische Universiteit Eindhoven, 2011.

Appendix A: Project Plan

Neotrack System B.V.

Graduation Project Plan

Indoor Positioning System

Chao Jiang
02-03-2012

Graduation Project Plan: Indoor Positioning System

Table of Content

Introduction.....	2
Project Statement	2
• Formal client	2
• Project leader	2
• Current situation.....	2
• Project product.....	2
• Project deliverables and non-deliverables	3
• Project constraints.....	3
• Project risks.....	4
Project Phasing.....	4
• Phase 1: Preparation.....	4
• Phase 2: Complete & Improve.....	4
• Phase 3: Integrate & More possibilities	5
Management plan.....	5
• Money	5
• Skills	5
• Quality	6
• Information	6
• Time	7
• Organisation.....	8

Graduation Project Plan: Indoor Positioning System

Introduction

This is the project plan of Chao Jiang's graduation project. In this document, you can get the information about this project, include the project statement, project phasing, management plan.

The project goal is to improve a series of software prototype of indoor navigation which can track the location of the smart phone over time. The complete assignment can be divided into the following parts:

1. Optimize the Android client prototype for indoor positioning.
2. Design and implement a quality Android APP for indoor navigation.
3. Investigate routing implementations for JAVA.
4. Integrate the routing implementation and indoor positioning prototype into an Android prototype.

Project Statement

- **Formal client**

The formal client of this project is my supervisor MSc. Yongxin Xu. This project is not considered as a complete product to the market. So the formal client of this project is the one who will integrate my project to an entire product.

- **Project leader**

As I can see, I will be the one who will act as the project leader and is responsible for all communication between the project participants and the external parties.

- **Current situation**

Neotrack System B. V. focuses on indoor positioning and navigation and its application. The indoor positioning prototype application which based on WIFI signal strength has been finished with some limited functions.

- **Project product**

Neotrack Positioning System, which consist by 4 parts:

Graduation Project Plan: Indoor Positioning System

- Positioning mobile application (for user) based on Android platform.
- Fingerprinting collection mobile application (for administrator) based on Android platform.
- Web service program which provides information to mobile app.
- Server side Database.

- **Project deliverables and non-deliverables**

Deliverables

- New requirement document and a project schedule
- Document and design document for new features
- Source code and complete released program package
- System integration document and new feature-update document

Non-deliverables

- Manual of the system

As we can see the product of this software is not for terminal users. It will be integrated to some other systems as a component. So there is no manual but some description text for future integration and modification.

- Complete design document of the whole system

Because this system already have a version of complete design document for current situation, so in the end of the project a partly design document for the part which was improved in this project and new features will be deliver instead of a complete one.

- **Project constraints**

- Improvement and fix work should base on the current system
- The development work of the mobile application should base on Android platform v2.2
- The development work of the server side program should base on Java language and Servlet 2.0 standard
- The database should base on MYSQL 5.0 database system

Graduation Project Plan: Indoor Positioning System

- **Project risks**

- Risk1: The current function of this system is not like what it described in current documents that I get.

Alternative scenario: Adjust the plan to first implement functions which is the dependence of the task in this project plan.

Cannot-Cope: The current situation of this system(especially for the part which is the dependence of this project) does not match the description in the document and cannot be fixed in short time.

- Risk2: The feature of dynamic-position-display and smooth-movement is too difficult to implement in time.

Alternative scenario: Adjust the timetable of the project to give reasonable extra time on such tasks

Cannot-Cope: These task takes too long time even cannot be finished within the whole project period.

Project Phasing

- *Phase 1: Preparation*

- Prepare the development software and hardware environment
- Take over and make clear of the current source code , database and document of this system
- Run and make clear the current situation of the app and the server side program
- Point out which part should be fix/complete/improve and how

Deliverable: A new requirement document and a project schedule

- *Phase 2: Complete & Improve*

According to the new requirement and new schedule

- Fix the problems and bugs in current system
- Complete the unfinished API of the system

Graduation Project Plan: Indoor Positioning System

- Add dynamic-position-display feature and smooth-movement feature
- Improve the finished parts in mobile app and server side program

Deliverable: A complete system and design document

- *Phase 3: Integrate & More possibilities*

- Find the possibility of combine the positioning system and the routing function based on Dijkstra's algorithm.
- Find the possibility of integrate the system to the company's product which is already on the market.
- Add routing features to the system.

Deliverable: System integration document and new feature-update document

Management plan

- **Money**

For this project we need PC for development and some android phones for testing. There are some android phones already arranged by the company. And the laptop that been arranged to this project is capable to run the IDE(eclipse and ADT) and the simulator program of android platform. So there is no money cost on hardware.

Because this project is not a complete product for the real world market, so there is no profit that we can get from this project.

- **Skills**

- Making project plan
- Requirement analysis
- Software design(UML diagrams, for instance class diagram, sequence diagram)
- Programming in Java and Android platform
- Software Testing (functional and performance)
- Document writing

Graduation Project Plan: Indoor Positioning System

- Mathematical skills, especially statistic and modeling

- **Quality**

In software engineering, a high quality project not only means a functional software product, but also means high quality documenting work, high quality communication with formal clients and always keep going on the right direction of the project.

For my understanding, a high quality project should have such features below:

- Complete and accurate technical document
- High quality coding work and complete code comments
- Fast response for the change of requirement
- Keep balance between requirement change and time & money

To make sure the quality of the project, few items below should be respected

- Report the project process to the formal client, supervisor and tutor with a suitable frequent
- Always add necessary comments within the code
- Arrange enough time on coding work and document
- Tackle with requirement change and new requirement carefully
- Afterwards a thorough test(include performance test and functionality test)

- **Information**

	Project plan	Project report	Analysis & Design Doc	Test Report
formal client	A, Di	Di		Dr
tech leader	A, Di	Di	Di, R, A	R

Graduation Project Plan: Indoor Positioning System

project leader	Dr, S	Dr	Di, Dr	Dr
tutor	A, R	A,R		

• Time

Phase / Activity name	Man-hour	Plan
Phase 1: Preparation		
- Prepare the development software and hardware environment	8	week 1
- Take over and make clear of the current source code , database and document of this system	100	week 1 – week 3
- Run and make clear the current situation of the app and the server side program	40	week 4
- Point out which part should be fix/complete/improve and how	24	week 5
Phase 2: Complete & Improve		
- Fix the problems and bugs in current system	160	week 6 – week 9
- Complete the unfinished API of the system	120	week 10 – week 12
- Add dynamic-position-display feature and smooth-movement feature	200	week 13 - week 17
- Improve the finished parts in mobile app and server side program	80	week 18 - week 19
Phase 3: Integrate & More possibilities		
- Find the possibility of combine the positioning system and the routing function based on Dijkstra's algorithm.	40	Week 20

Graduation Project Plan: Indoor Positioning System

- Find the possibility of integrate the system to the company's product which is already on the market.	flexible	flexible
- Add routing features to the system.	flexible	flexible

- **Organisation**

