



NTNU – Trondheim
Norwegian University of
Science and Technology

Improving an Existing Natural Language Bus Route Information System

Adding Support for Geographic Expansion
and Visualization in Maps

Marius Qvam Wollamo

Master of Science in Computer Science

Submission date: July 2013

Supervisor: Bjørn Gambäck, IDI

Co-supervisor: Rune Sætre, IDI

Norwegian University of Science and Technology
Department of Computer and Information Science

Abstract

Real-time itinerary information is increasingly used and the functionality of travel planners is becoming more and more advanced. This gives the users the opportunity to tailor their demands and receive simple and fast answers, either through a mobile app, website or voice mail. This thesis explores how the BusTUC system can be improved to take the needs of the users into account. To investigate this, the system is expanded to return answers also for tram departures, and support for visualizing routes in a map is added. 14 persons have tested the new system on a website. Then, the same persons have given feedback through a survey. These answers have been analyzed to discover problems in the web client, and to detect how the experience on the website is, in terms of response time. It seems that correspondence between different means of transportation is important to the users, but this functionality is not present in the current system. Several clients want the system to take all of Norway into account, not just Trondheim. The support for tram schedules and the map functionality seem to increase the credibility of the BusTUC answers, which again lead to more satisfied users. The report also describes how BusTUC can be used in another domain to return intelligent answers. This is an aspect that should be further explored in the future.



Sammendrag

Bruken av rutetidsinformasjon i sanntid blir stadig vanligere og funksjonaliteten til reiseplanleggere blir stadig mer avansert. Dette bidrar til at brukerne av reiseplanleggere kan skreddersy spørsmål og få det svaret de er ute etter på enklest mulig måte, enten gjennom en mobil app, nettside eller telefonsvarer. Denne masteroppgaven forsøker å finne ut hvordan systemet BusTUC kan forbedres for å ta hensyn til brukernes behov. Til å undersøke dette er BusTUC utvidet til å returnere svar for trikk og det er laget støtte for å vise rutene i kart. Det nye systemet er blitt testet av 14 personer via en nettside. Deretter har de samme personene gitt tilbakemeldinger i et spørreskjema. Tilbakemeldingene har blitt analysert for å finne hovedproblemene til brukerne, og hvordan den nye nettsiden blir oppfattet med tanke på responstid. Analysen viser at manglende korrespondanse mellom forskjellige ruteselskaper er viktig. Samtidig ønsker flere at systemet tar hele Norge i betraktning, og at all tilgjengelig kollektivtransport bør være inkludert. Støtten for trikk og kartfunksjonen øker troverdigheten til BusTUC-svarene, og det fører til flere fornøyde brukere. Masteroppgaven beskriver også hvordan BusTUC kan bli brukt i et annet domene for å returnere intelligente svar. Dette er aspekter som bør undersøkes videre senere.

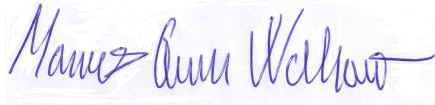


Preface

This Master's thesis was written by Marius Qvam Wollamo during the spring semester of 2013 and is a contribution to the *Futures Ultimate Intelligent Route-Organizing System* (FUIROS) project. The work is the final report for the Master of Science (MSc) Degree in Computer Science at the Department of Computer and Information Science at the Norwegian University of Science and Technology (NTNU).

I would like to thank my supervisors Björn Gambäck and Rune Sætre for their guidance and support. Rune has been a valuable resource for his great knowledge of the BusTUC system. I would also like to thank the persons that participated in the survey and filled out the questionnaires.

Trondheim, 2013-07-05



Marius Qvam Wollamo



Problem Description

The assignment was given by Rune Sætre at IDI. The FUIROS-project was initiated by Tore Amble and aims towards being the ultimate route information system for the future:

FUIROS - The Future's Intelligent Route Information System

BusTUC is a natural language bus route system for Trondheim. It gives information about bus schedules, and has some information about the real passing times. AtB has installed GPS tracking on all the buses, giving access to real passing times and delays, and not only for Trondheim. With new smart phones arriving rapidly on the market, there are possibilities for GPS localizations, connections to maps and voice input. The project will be based on an existing advanced smart phone application called TABuss, developed by four MSc students. The current Bus Oracle system should be extended from Trondheim to cover a greater part of Norway, and it should support automatic updates whenever AtB change their regular schedules around holidays, and between winter and summer. The system could also be extended to include information about trams, boats, trains and airplanes.



Contents

Abstract	i
Sammendrag	iii
Preface	v
Problem Description	vii
Table of Contents	ix
List of Figures	xi
List of Tables	xiii
Acronyms	xvi
1 Introduction	1
1.1 Motivation	1
1.2 Research Questions and Goals	2
1.3 Research Methods	3
1.3.1 Literature Review	3
1.3.2 Survey	4
1.3.3 Observation	4
1.3.4 Questionnaires	5
1.4 Validation Models	5
1.5 Thesis Structure	6
2 Theory and Background	7
2.1 State-of-the-Art	7
2.1.1 Existing Travel Planners for Norway	8
2.1.2 Comparing Travel Planners	10
2.1.3 Travel Planners with NLP Support	10
2.1.4 State of the Art Conclusion	13
2.2 Background	14
2.2.1 BusTUC	14
2.2.2 BusTUC Interfaces	16
2.3 Transit Data Exchange Standards	17

2.3.1	SIRI	17
2.3.2	General Transit Feed Specification	18
2.3.3	REGTOPP Format	18
3	Contributions	19
3.1	MapTUC	19
3.1.1	Technology	19
3.1.2	Implementation	22
3.1.3	Testing	24
3.2	Geographical Expansion	25
3.2.1	Process of Importing new Dataset	25
3.2.2	Testing New Data	26
3.3	Tram Support	27
4	Results	29
4.1	MapTUC	29
4.1.1	Response Time	31
4.1.2	Route Suggestion	31
4.2	Questionnaires	34
4.2.1	BusTUC's Answers	35
4.2.2	Map Functionality	36
4.2.3	Evaluations	37
4.3	Tram Support	38
5	Discussion and Conclusion	39
5.1	Discussion	39
5.1.1	MapTUC	39
5.1.2	Geographical Expansion	39
5.1.3	Questionnaires	40
5.2	Answers to Research Questions and Progress towards the Goals	42
5.2.1	Research Question 1	42
5.2.2	Research Question 2	42
5.2.3	Research Question 3	43
5.2.4	Goal 1	43
5.2.5	Goal 2	43
5.2.6	Goal 3	43
5.2.7	Goal 4	43
5.3	Conclusion	44
6	Future Work	45
6.1	Integrating Real-Time Information into BusTUC's Answers	45
6.2	MapTUC	45
6.3	Logging	46
6.4	Expansion	46

List of Figures

2.1	Google Maps Directions	11
2.2	A search result from the Travel Planner at ixigo.com	12
3.1	The communication process for returning an answer in MapTUC	24
4.1	MapTUC design	30
4.2	The new map response for the question “Malvik to Hommelvik station”	30
4.3	The real-time information in MapTUC	31
4.4	Displaying the route for the query: “From Strindveien 37 to Valentinlyst station”	32
4.5	Displaying the route for the query: “From Magnus Blindes vei to Valentinlyst station”	33
4.6	The average time (in minutes) spent on solving the scenarios for test subject with and without previous experience of BusTUC.	34
4.7	The test subjects’ perception of BusTUC’s answers.	35
4.8	The test subjects perception of the new map functionality.	36
4.9	The test subjects evaluation of BusTUC’s future	37
4.10	Tram Route in MapTUC	38

List of Tables

2.1	Comparison of travel planners' support for public transportation . .	10
2.2	Examples of questions and answers from BusTUC	14
2.3	REGTOPP version 1.2	18
3.1	Requirements MapTUC	19
3.2	Technology Used in MapTUC	20
3.3	Server setup	25
3.4	REGTOPP extraction	26
3.5	UTM notation in the REGTOPP format	26
4.1	MapTUC: Browser Test	29

Acronyms

- AJAX** Asynchronous JavaScript and XML
- API** Application programming interface
- CSS** Cascading Style Sheets
- DOM** Document Object Model
- IDI** Department of Computer and Information Science
- IME** Faculty of Information Technology, Mathematics and Electrical Engineering
- FUIROS** The Future Ultimate Intelligent Route Information System. In Norwegian: Fremtidens Ultimate Intelligente RuteOpplysningsSystem
- GTFS** General Transit Feed Specification
- GPS** Global Positioning System
- HTML** HyperText Markup Language
- HTTP** Hypertext Transfer Protocol
- JSON** JavaScript Object Notation
- MultiBRIS** Multiple-platform approach to the Ultimate Bus Route Information System
- NLP** Natural Language Processing
- NTNU** Norwegian University of Science and Technology
- REGTOPP** Regional Transit Information format
- SIRI** The Service Interface for Real Time Information
- SMS** Short Message Service. Allows sending short text messages between phones and other handheld devices
- SQL** Structured Query Language
- XML** eXtensible Markup Language

Chapter 1

Introduction

The focus of this thesis has been to present the answers from BusTUC, which is a natural language bus route system, in a map and to test how the system could be expanded into other domains. This is broken down into several research question and goals. BusTUC is further described in Section 2.2.1.

The datasets for the whole of Norway were published at <http://labs.trafikanten.no/archives.aspx>, January of 2013, and allow travel planners like BusTUC to redistribute the information. However, these datasets were not discovered until the start of May 2013, and therefore only one dataset was imported and tested in order to highlight the challenges using data outside BusTUC's current knowledge domain.

1.1 Motivation

The FUIROS project has been around for several years and aims to be the ultimate intelligent route information system for the future. It started with BusTUC (Bratseth, 1997), which is a natural language system that returns answers to questions about the bus schedules in Trondheim. Later, a spoken dialog information system was implemented to deal with speech recognition and text-to-speech (Johnsen et al., 2003). Years later, both a cross-platform application MultiBRIS (Andersstuen et al., 2012) and an Android application TABuss (Marcussen et al., 2012) were developed to utilize the real-time information provided by the public transportation agency in Trondheim, AtB.

Today, the number of travel planners is increasing and the distribution of information has made the everyday life of travelers easier. When BusTUC was first released, it was one of a few travel planners for bus transportation in Trondheim. A lot has changed since then and other travel planners have now reached the market.

Generally speaking, everyone using the public transport wants a service that is reliable and without delays. Especially for a meeting or a lunch date, the traveler wants to be there on time.

Most of the travel planners today have limitations regarding what kind of transportation they support. Some only have information for buses, like BusTUC, while others have support for boats, trams, trains, and more. The trend of

gathering all transportation information in one travel planner, like the collaboration project dit.no¹, could be one of many competitors to BusTUC.

What separates BusTUC from the other travel planners is its natural language support. This is what makes BusTUC so unique. This project aims to enhance the functionality to cover a greater part of Norway. By making a method to support questions for tram, this could be repeated to cover other transportation modes, such as ferries, trains, etc. Another goal is to add a map for presenting BusTUC's answers.

1.2 Research Questions and Goals

Based on the problem description the following research questions should be answered:

Research Question 1 (RQ1) How do users perceive the answers given by a natural language travel planner?

Research Question 2 (RQ2) What could help increase the natural language travel planner's credibility?

Research Question 3 (RQ3) How can the local natural language travel planner cover the entire world?

To answer the research questions, four goals have been specified.

Goal 1 Create a new web page that is able to present BusTUC answers in a map. This involves adding a new JSON response to all BusTUC answers.

Goal 2 Add support for tram route information in BusTUC.

Goal 3 Import a new dataset to see if BusTUC is able to provide the same intelligent answers, in that domain. This approach should enlighten the challenges in adding new information into BusTUC.

Goal 4 Conduct a test to see if the users find BusTUC's answers credible. The test subjects should be presented with different scenarios to solve on the new web page, created in order to measure the perception of both the answer and the map feature.

Goal 1 is the system that should be used when conducting the survey of Goal 4. Goal 2 should be implemented to see if it is possible to add support for trains

¹m.dit.no

and ferries in the same manner as for tram. The new version of BusTUC should be finished before conducting the survey, since one of the scenarios includes the tram. Goal 3 should investigate whether other datasets will also provide the same intelligent answers as the current system does in Trondheim. Goal 3 will also determine the challenges of making BusTUC more scalable.

1.3 Research Methods

A methodology is, according to Pressman and Ince (1992), a set of activities, methods, techniques and practices that together form a tool to simplify the software development process. A combination of methodologies was used in this thesis. The next section describes the application of these methodologies extending BusTUC.

In the field of research many different approaches and techniques are used to collect data. The techniques used in this thesis are:

- Literature review
- Survey
- Observation
- Questionnaire

The literature review was used to get up to date with the current research and systems in the field of travel planners with and without natural language support. A survey was chosen both to collect opinions about the new web page and because this method is low cost. The number of test subjects was 14, due to the time constraints of this project. Thus, the sample size is not as high as wanted. Observation was used in order to see how the test subjects used the new web page, and to discover how they reacted on the different scenarios. A questionnaire was used to gather comparable data to be analyzed.

These methods are described further in the next sections.

1.3.1 Literature Review

In all types of research, literature review is one of the most common research methods used. This method is an approach to get up-to-date with the research that has already been conducted. By reviewing the literature on other similar systems, the researcher can discover new ideas and approaches for his own research. The information extracted should come from respected sources and be useful for the ongoing research. For example, the knowledge can be obtained by reading books, articles, journals, newspapers and web pages. The search and extraction of information should follow some predefined sets of requirements. In this thesis, the literature study was performed to gather information about the state-of-the-art in the field of trip planners and to gain more knowledge about technologies used during the development phase. The search phase consisted of looking for publications in electronic libraries as Google Scholar², Cite Seer X³ and DAIM⁴. The goal was to

²<http://scholar.google.com>

³<http://citeseerx.ist.pdu.edu>

⁴<http://daim.idi.ntnu.no>

discover other related travel planning systems. The web is another source that was used in order to review related systems to analyze their features and functionalities. By searching for keywords related to the research questions, relevant information was obtained. Many of the travel planners reviewed were found by searching the web. Related systems were explored in order to determine similar features and functionalities. The topics for the review were:

Transit data formats: Used whenever a data provider shares their information to be used in a trip planner. By studying some of the transit formats that exist, ideas to how the BusTUC system could become more scalable and utilize more data sources could be found.

Travel planners: Were studied in order to investigate how they present their information, and what features and functionalities they provide. This is an interesting field, because the datasets containing the public transit information in many counties in Norway have recently been made available to the public. Some of the new travel planners were not discovered until the very end of this thesis. Trip planners that support natural language processing refer to the services and prototypes that allow the users to query the system with the use of natural text or speech, like BusTUC. These systems are investigated in order to detect what the current state-of-the-art within the field of natural language travel planning is.

These steps constitute the implementation phase.

1.3.2 Survey

In this project, a set of test subjects was gathered. They evaluated the web page, which combines text and map answers. The main goal of the survey was to check if the overall perception of BusTUC's answers is considered more reliable in combination with a map. The survey consisted of four different scenarios, which the users should try to solve by querying BusTUC through the new web page. These scenarios encompass cases where the users need to find information about bus, tram and night bus, and real-time information, in the map. Other scenarios specify that the users should look for a way to get home from the cinema, thus, making the users ask different questions to BusTUC. After working through the scenarios, the users were asked to fill out an evaluation form and to write down any comments. Both experienced and inexperienced BusTUC users were asked to contribute, and the time they spent completing the scenarios was measured.

1.3.3 Observation

During the problem solving, some of the test subjects were observed in order to examine how they approached the tasks and how they interpreted the presented information. This method is used to get useful information, when a questionnaire

falls short, and will also provide feedback on whether the new web page⁵ is more attractive than the old one⁶ or not.

1.3.4 Questionnaires

A questionnaire is a set of predefined questions that gathers qualitative data. This method is used in order to get the test subjects' opinions after performing the survey. The questionnaire is a mixture of multiple-choice and scale questions. The test subjects give their answers by choosing on a scale from 1 to 4, where 1 represents that they strongly agree and 4 that they strongly disagree. The scale is chosen deliberately so that the test subjects cannot choose a neutral response. The questions are divided into four categories:

1. First a general section asking for some background data such as gender and whether they had any previous experience with BusTUC.
2. The second section asks for the users' perception of BusTUC and if they felt that the information returned was reasonable.
3. The third section asked if they thought the map added to the system was useful.
4. The last section asks about what they think about possible future BusTUC improvements.

The results from this survey are presented in Chapter 4.

1.4 Validation Models

In order to validate the results of the new web page and the extension of functionality in BusTUC, the accuracy must be measured. "Observational models" is a method that collects relevant data as a project develops, and is divided in four different types (Zelkowitz and Wallace, 1998). The models are:

- **Project Monitoring** describes the collection of data under the development phase. Data is generated during the development, and this model has no clear goals for how the collected data is gathered.
- **Case Study** is like project monitoring, but with a specified goal of how and what parts of the system are evaluated. This approach is unique for each project and is therefore hard to reproduce.
- **Assertion** is, according to Zelkowitz and Wallace (1998), a model used in approximately 33 % of published papers. The reason for this is that the researcher acts as both the experimenter and the subject of study. By using ad hoc techniques, fatal defects can be found quickly. The intention is not to document the steps, rather to show that a technology or system is superior.

⁵<http://busstuc.idi.ntnu.no/maptuc/>

⁶<http://busstuc.idi.ntnu.no/bustuc/>

- “**Field Study**” examines several projects simultaneously and is often performed by an external group. This model is typically used for analyzing completed products.

In this research, the assertion and literature review methods are used in order to validate the data obtained from the survey, questionnaire and the implementation phase.

1.5 Thesis Structure

The next chapters in this thesis are structured like this:

1. **Theory and Background** introduces first the state-of-the-art in the field of travel planners and then underlying systems and theory are described.
2. **Contributions** describes contributions developed for the FUIROS project.
3. **Results** presents the results from the user test and the new web page.
4. **Discussion** discusses the results and concludes on how the research questions have been answered.
5. **Future Work** describes the suggestions for future work.

Chapter 2

Theory and Background

This chapter will first look further into the state-of-the-art in the field of trip planners and give a description of what a travel planner and natural language processing is, before the state-of-the-art is explained. Next the BusTUC interfaces are explained and last some transit data exchange standards are described.

2.1 State-of-the-Art

Travel Planner: A travel planner is a search engine that provides information about public transport trips. A typical travel planner prompts the user to specify an origin and a destination before it calculates the possible routes and return an answer for the available public transport services. Some travel planners allow the users to specify which public transport mode they want, before sending the query. Today, travel planners are common as web applications which utilize maps, graphs and other presentation methods in order to present route information.

Natural Language Processing: Natural language processing deals with the challenge of making natural language understandable for computers.

Within the field of travel planning, maps are commonly used to present the route suggestion. A route suggestion could be the fastest available public transport from point A to B and further advanced parameters could be given for a more advanced search. Today, travel information seems to be found everywhere. Today's mobility has made the need for instantaneous information easier, and for some people, it is especially useful to know if the next public transport departure is delayed or not. New services are constantly emerging and some of them are mentioned and described next.

In recent years, Google has initiated a research project that ended in the service Google Transit. Now it is well integrated in Google Maps and allows both basically natural language queries and more advanced search options. Another open source trip itinerary is OpenTripPlanner that utilizes open standards to provide a free travel planner that every agency with a GTFS dataset could deploy on their web page. In Norway, the datasets for entire country have been recently been made

available. This was an initiative to let third party developers create and improve the selection of travel planners. This resulted in the release of new travel planners like Gule Sider and dit.no. There are also several approaches, like BusTUC, that utilize natural language for getting a route suggestion. Let's Go is an example of a spoken dialog system that aims to improve the pronouncement for the elderly and non-native speakers in order to provide bus information to the Pittsburgh area. A spoken dialog system has also been addressed by the system StopMan and its descendant TravelMan used in several Finnish cities. The search engine ixigo.com is a trip planner that tries to understand a natural language questions and returns graphics as response.

The following sections will look further into each of the systems that are similar to BusTUC, concerning natural language support, and the systems that constitute the most popular travel planners in Trondheim and Norway.

2.1.1 Existing Travel Planners for Norway

Being able to get from one place to another might not be an easy task. The motivation for agencies to release their dataset allows travelers to easier find the information they are looking for. In some parts of Norway, like Oslo and in the county Rogaland, the public transport information is made available in Google Maps. This trend allows third parties to create their own travel planner and makes the access for timetables more available. In the next sections, travel planners that cover Trondheim and Norway are presented.

Bartebuss

Bartebuss is probably Trondheim's most popular travel information service with up to 50.000 downloads on Google Play.⁷ The application is a cross-platform application build with HTML5, JavaScript and CSS, and is available for Android, iPhone, iPad and on the web.⁸ The service is free and developed on a hobby basis by Rune Martin Andersen. The primary functionality is telling the user when a bus would depart from a specific stop, based on the real-time information supported by AtB. Bartebuss also has a map view showing the nearest stops. It also supports queries to BusTUC.

Gule Sider

Gule Sider has recently released a trip planner that combines route information about public transportation such as tram, train, boats, ferries, planes, underground, and buses for Norway. They have integrated this solution in their map service. This is one of few services that has information for the entire Norway and can give you a route suggestion from north to south. For example, if a traveler from *Kirkenes* to *Kristiansand* were to query this service, an answer to take a plane, airport express, train, etc., could be the response. Gule Sider has also, like many other travel planners, today the ability to detect the current position of the user by either HTML5 or a GPS from a mobile phone. This functionality is needed

⁷<https://play.google.com>

⁸<http://bartebuss.no/>

in cases where the user, for example, wants to go to the nearest shop from his current location. The travel planner is linked to the existing search functionality. This means that when searching for a person, company or a place, no matter where in Norway, an overview of how to get there by public transport is suggested.

Norsk Reiseinformasjon

Norsk Reiseinformasjon (NRI) was until the spring of 2013 through the web page `rutebok.no` the only trip itinerary that provided routes across the country of Norway and abroad. This service covers every public transportation in Norway and can give route suggestions anywhere in Norway.

Dit.no

Dit.no is a web page designed to be used in mobile phones. It is a service that collects route and traffic information for Norway. Their goal is to be the most complete traffic information service in Norway. They currently have a web-application `m.dit.no` which could tell the user if there is any deviation in the traffic, list nearby departure points, and plan a journey by car or public transportation. This service has just been published and is still in Beta-version, and does not have transit support for entire Norway yet. Nonetheless, it has information about buses, trams and trains.

TravelMagic

One of the most used travel planners by counties in Norway is *TravelMagic*, which is developed by DataGrafikk.⁹ Their first travel planner was launched in 1997 and when they combined the use of Google Maps the popularity increased. This travel planner is used by several counties in Norway and its domain is, therefore, different for each county. So a search between counties is usually not supported. AtB¹⁰ is using TravelMagic as the travel planner together with BusTUC.

This system allows the user to specify an origin and destination. Both local and regional bus routes with an origin in Trondheim are supported. The travel route is presented as a list with detailed information of what kind of transport is possible and the departure times. In addition to a list, a feature displays the selected route on a map.

⁹<http://www.datagrafikk.no/?tag=travelmagic>

¹⁰<https://www.atb.no/atbreiseplanlegger/>

2.1.2 Comparing Travel Planners

At the beginning of 2013 many new travel planners were introduced in Norway. Gule Sider launched a service that could return answers for every public transportation in Norway. Another example of this is the service at dit.no.

In Table 2.1 the support of public transportations are compared. The result shows that BusTUC is the only service that only has support for one transportation type.

Service	Bus	Tram	Train	Ferries	Flight	Underground
BusTUC	X	-	-	-	-	-
Bartebuss	X	X	-	-	-	-
Dit.no	X	X	X	X	-	X
Gule Sider	X	X	X	X	X	X
NRI	X	X	X	X	X	X
TravelMagic	X	X	-	X	-	-

Table 2.1: Comparison of travel planners' support for public transportation

2.1.3 Travel Planners with NLP Support

This section will address some of the popular NLP travel planners in the world. User-friendly systems, which allow the users to type a query in natural language, have been the research for many years. Some of these systems are presented next.

Google Transit

Google Transit¹¹ is a public transportation planning tool that uses real-time and timetable data to provide route suggestions to the users. It is a part of the popular map service Google Maps.¹² The service started as an idea in 2005 and was a part of Google Labs until 2007 when it was fully integrated in Google Maps. Just after the Google I/O conference¹³ in 2013, Google announced a new version of Google Maps. One of the most interesting new features is the support for natural language queries to present public transportation options. For example, if the user types *“from Nationaltheateret to Stortinget”*, Google Maps would suggest a trip with the underground. This support of natural language queries is also available in Norwegian. The search: *“fra hovseter til oslo s”* presents a route suggestion by car or underground as depicted in Figure 2.1. This support only considers queries that have the keywords “from” and “to” or just “to” between two places. Thus, a user cannot ask the question: “the next train from Manchester to London”. However, this is quite a good service that works well in the places where the local public transportation agencies have released their datasets to Google. The coverage in Norway is currently in Oslo, the county Rogaland and the National Railways (NSB) for entire Norway.

¹¹<http://www.google.com/intl/com/landing/transit/>

¹²<http://maps.google.no>

¹³http://en.wikipedia.org/wiki/Google_I/O

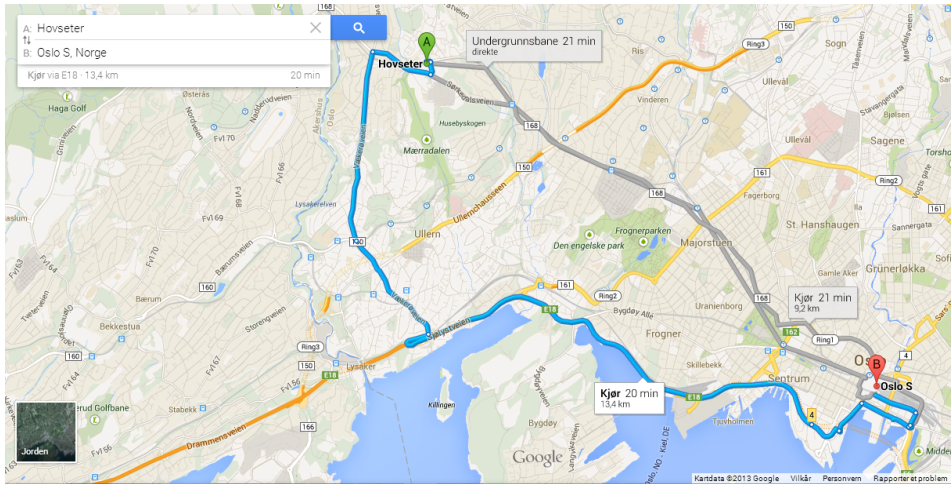


Figure 2.1: Google Maps Directions

A limitation is, therefore, when a user asks for “*from Trondheim to Oslo*” and really wants to take a bus instead of a train. This is probably only a limitation today, but as time goes by, more and more information could become available. One of the great functionalities is that Google Maps also has enough information about street names and places, so the result could guide the user from his current position to the destination by car, bus, tram, some walking, and etc. Google Maps will most likely be one of the most used travel planners in the future.

ixigo.com

ixigo.com claims to be the leading travel search engine in India. Like BusTUC it supports natural language input. The domain for this search is more like a tour guide than just a dedicated travel planner for public transportation. However, the search “from London to Mumbai” will bring a presentation of transportation modes, direction and destination suggestions. The first suggestion and cheapest is to take a flight from London to Mumbai in 8 hours and 50 minutes. This seems like a reasonable result. A search within Norway like *Trondheim to Oslo* suggests a flight from Trondheim to Copenhagen and from Copenhagen to Oslo. A total of 23 hours 55 minutes is far away from a reasonable and wanted trip. On the other hand, the reason for the suggestion is likely that the travel search engine works best in its current domain and does not have local transportation information in Norway other than domestic flights. A search within India “Mumbai to New Delhi by train” presents a list of the current departure times nearby Mumbai to New Delhi as depicted in Figure 2.2. This is a more reasonable result and really utilizes a presentation of an info graphic for the answer.

The screenshot shows the ixigo.com website interface. At the top, there is a navigation bar with icons for trip planner, flights, hotels, trains, buses, and more. A search bar contains the text "mumbai to new delhi by train". Below the search bar, there are options to like, add to trip, and view all modes. The main content area displays a table of train results for the route from Mumbai to New Delhi. The table includes columns for Train Name, Depart, Arrival, Duration, Running Days, Class, and ₹ Price. Each row also has a "Check Availability" button.

Train Name	Depart	Arrival	Duration	Running Days	Class	₹ Price	
Mumbai Rajdhani 12951	04:40 PM	08:30 AM	15h 50m	Sun ,Mon, Tue, Wed, Thu, Fri, Sat	1A 2A 3A	3694 2152 1483	Check Availability
Bdts Nzm Yuva E 12247	04:55 PM	09:40 AM	16h 45m	Fri	CC	707	Check Availability
Nzm Garib Rath 12909	04:55 PM	09:40 AM	16h 45m	Tue, Thu, Sat	3A CC	815 707	Check Availability
Aug Kr Raj Exp 12953	05:40 PM	10:55 AM	17h 15m	Sun ,Mon, Tue, Wed, Thu, Fri, Sat	1A 2A 3A	3694 2152 1483	Check Availability
Ndis Duronto Ex 22209	11:15 PM	04:55 PM	17h 40m	Mon, Fri	1A 2A 3A	3670 2068 1369	Check Availability
Rajdhani Exp 12431	06:10 PM	12:40 PM	18h 30m	Wed, Fri, Sat	1A 2A 3A	3694 2152 1483	Check Availability
Swaraj Express 12471	07:55 AM	04:11 AM	20h 16m	Sun ,Mon, Thu, Fri	2A 3A SL	1866 1277 493	Check Availability

Figure 2.2: A search result from the Travel Planner at ixigo.com

OneBusAway

OneBusAway is a transit traveler information system (Ferris et al., 2010). It has provided real-time arrival information to transit travelers in the Seattle area since 2008. This started first as a web interface and SMS services before making applications for iPhone, Android and Windows Phone. OneBusAway has a focus on providing real-time information for the users. One of the varieties of tools is the phone interface. This service allows the user to type a bus stop number and get the real-time arrival information spoken back.

Let's Go

Let's Go is a spoken dialog system which has been released to the public to provide bus schedule information to the Pittsburgh area (Raux et al., 2005). This system has also been tuned to improve the accuracy for elderly and non-native user.

TravelMan

A multimodal mobile application for public transport information called TravelMan is described in a paper by Turunen et al. (2007). It provides many inhabitants in Finland with information and route guidance for transports as metro, tram and bus traffic. The main functionality of TravelMan is to plan a trip from a departure to destination stop and to give interactive guidance when the trip starts.

OpenTripPlanner (OPT)

OpenTripPlanner is according to their web page¹⁴ the leading open source platform for trip planning and network analysis. Since it is an open source application no proprietary data formats are used. The only prerequisite for an agency to get an instance of OTP up and running is use of a General Transit Feed Specification (GTFS) dataset, which is Google's open standard for exchanging public transit information. Together with publicly available map data an agency could deploy its own version of the OPT and be the owner of all its data. The OPT has been deployed in 10 different countries and has currently around 20 active deployments in total. Optitrans¹⁵ is an example of a deployed OPT.

2.1.4 State of the Art Conclusion

A state-of-the-art survey shows that there exists several travel planners with a variety of functionality spread throughout the web. Also, new travel planners have recently been published that cover a great part of Norway. The correlation of travel planners today is that they are using maps to present their route suggestions. Only a few travel planners have the ability to understand natural language speech or text to interpret the users' needs. This field is continuously in change and any travel planner should keep up with this development to maintain a good and useful service.

¹⁴<http://opentripplanner.com/>

¹⁵<http://london.optitrans.net/>

	Question	Answer
1	When does the next bus from Tiller to Nardo depart?	The station nearest to Tiller is Torvtaket. The station nearest to Nardo is Nardosenteret. Bus 46 goes from Torvtaket at 2:44 pm to Tempe kirke at 2:59 pm and bus 8 goes from Lerkendal stadion at 3:07 pm to Nardosenteret at 3:12 pm. The hours indicate the earliest passing times.
2	night bus from centrum to lade saturday	Bus 1301 goes from Munkegata M4 at 1:00 am to Nova kinosenter at 1:02 am and Night bus 106 goes from Nova kinosenter at 1:31 am to Lade allé 80 at 1:42 am .The hours indicate the earliest passing times.
3	from Trondheim to Oslo	I have only routes for buses in Trondheim .

Table 2.2: Examples of questions and answers from BusTUC

2.2 Background

This section describes the underlying technologies that are used and previously created in the FUIROS project and some theory of transit data exchange formats. There have been several contributions in the FUIROS project like the natural language system BusTUC, the location aware Android application TABuss and the cross-platform application MultiBRIS. A description of BusTUC and its interfaces are described next.

2.2.1 BusTUC

BusTUC is a natural language bus route oracle (Amble, 2000) (Bratseth, 1997). It aims to answer any question about the bus domain in Trondheim. Since the initial startup in 1996, BusTUC has become more complex and gotten more intelligent (Amble, 2009). Table 2.2 presents some common question and answers given by BusTUC.

Since BusTUC is dependent on different factors, as the date and the time, the answers given to identical questions will not necessarily be the same. BusTUC consists of several components, where each has its own responsibility.

- **Parsing the input text:** This component consists of a parser that uses English and Norwegian dictionary and grammar to build a representation of the input in a parse tree. This parse tree is further evaluated to provide a semantic meaning.
- **Semantic knowledge base:** This database consists of a semantics network that maps a word to a meaning in a specific domain.

- **Logical knowledge base:** This database consists of several rules that need to be fulfilled in order to give an answer to a given question.

Whenever a question is asked, BusTUC must perform an analysis and a reasoning of this question, before drawing a conclusion of what to return as an answer. BusTUC's answers rely on the interpretation of the first order expression, which is the result of an analysis of the question without context. The context is provided by the semantic knowledge base in BusTUC.

First, a lexical analyzer splits the question into tokens that form a syntactic parse tree if and only if the question consists of words that are recognized by the analyzer. Each leaf node in the tree represents words such as nouns or verb phrases. For example, the question *"Who is superman?"* returns *"incomprehensible words: superman"*. This is because the word *"superman"* is not in BusTUC's knowledge base and therefore the strict parser cannot translate the question into first order logic. This is to preserve the problem domain of BusTUC. However, if a user asks BusTUC the question *"What is a man?"* the returned answer is *"I don't know"*. Since the noun *"man"* is stored in the semantic knowledge base of BusTUC, the system can give an answer.

Second, BusTUC translates from the tree representation to a first order logic expression called TUC Query Logic (TQL). TQL expressions consist of predicates, functions, constants and variables. They give an interpretation of a question as understood by BusTUC. The TQL expression is the input for doing reasoning before BusTUC is able to provide the answer.

The semantic knowledge base in BusTUC is composed of rules and facts written in Prolog. For example, the question *"When is the next bus from downtown to Tiller leaving?"* contains the bus stop name *"Tiller"*. This is recognized by BusTUC as a bus stop since this is a fact declared in the semantic knowledge base. This approach also applies to most of the street addresses because they are mapped to the nearest bus stop.

The system is created to understand some misspelled words, and therefore, these are mapped to the correct word in the knowledge base.

BusTUC supports both Norwegian and English and some Swedish, which is treated as misspelled Norwegian. The language detection is performed for each question by counting the number of words unrelated to both languages. The system will answer in the language with the least unrelated words.

The reasoning in BusTUC is designed so that if a user does not specify two bus stops, the system assumes that the user wants to leave from downtown area of Trondheim. This assumption enables BusTUC to give a reasonable answer to an incomplete question.

If a user asks *"From Tiller to Samfundet"* and does not specify the time, an answer with the next departures from Tiller is returned. The time plays a very important role in the reasoning in BusTUC. When asking for a time that has passed, the system will return routes for tomorrow. This awareness is one of the features contributing to returning a reasonable answer to the user.

2.2.2 BusTUC Interfaces

There exist several tools to get information from BusTUC. Some only renders BusTUC's answers, and others use real-time information as an answer. The tools currently available are listed below.

1. **Web Interface**¹⁶ - A web interface that returns an answer from BusTUC.
2. **Telephone Interface**¹⁷ - A natural language interface for BusTUC answers.
3. **Android Interface**¹⁸ - A native location-aware client for Android phones for real-time information and BusTUC answers.
4. **SMS Interface**¹⁹ - A Short Message Service interface for BusTUC information.
5. **Mobile Web**²⁰ - A location-aware mobile web application for real-time departure information

Web Interface

The web interface²¹ is basically a text input field which sends the request to a Java Servlet that sanitizes the text input and performs a direct query to BusTUC. The answer is written as an HTTP Response back to the web interface before it is displayed to the user. The typical waiting time before an answer is presented is under one second. The user could choose between a regular text- or JSON response. This web interface is also available in a commercial version hosted by the current bus agency in Trondheim, AtB²². The web interface at IDI is used for continuous development and testing purposes before a new BusTUC version for commercial use is deployed.

Short Message Service

BusTUC is also available through the Short Message Service (SMS). BusTUC will then answer a question in the same way as the web interface. The difference is that the answer is shortened to fit into one SMS because of the restriction in the SMS message size.

Over the Telephone

BusTUC could also be accessed over the telephone. Here the user meets the voice of "Nora", a Norwegian speaking lady that asks kindly how she might assist you. This system is based on speech recognition and speech synthesis system called Buster (Hartvigsen et al., 2007) and remains as an experimental version. The answer is

¹⁶<http://busstuc.idi.ntnu.no>

¹⁷Telephone service is available at +4773521290

¹⁸TABuss <https://play.google.com/store/apps/details?id=test.BusTUC>

¹⁹prefix "rute" and your question to 2027

²⁰MultiBRIS <http://busstuc.idi.ntnu.no/MultiBRIS/>

²¹<http://busstuc.idi.ntnu.no>

²²<http://www.atb.no>

then translated into a speech output provided by Microsoft Speech API (SAPI). This has been also implemented into the Android application (Section 2.2.2).

Android Interface

The Android application TABuss, developed by Marcussen and Eliassen (2011), is a travel planner that detects the user's position and from this provides a route suggestion based on which stop the user wants to go to. One of the functionalities is the real-time information for each stop. Another functionality added is to recognize a bus stop name based on speech input.

Mobile Web

MultiBRIS, made by Andersstuen and Engell (2011), is a mobile web application that collects real-time information from AtB and allows adding other parameters as the current GPS position, the number of bus stops to search for nearby, and destination of the wanted trip. A response will return the arrival times for all the related buses.

2.3 Transit Data Exchange Standards

A standard is used whenever systems or computers must communicate and exchange information. Essentially it is an agreed way of doing something. In the same way as a language sets rules for speech and writing to ensure good communication, a standard set, and the rules for the communication. By using the same standard, communication between two parties is made easier. This naturally makes exchanging information more efficient, and potentially harder if they speak different languages.

An open standard is a standard that is published and made freely available to be used by everyone. This is standards like HyperText Markup Language (HTML), Cascading Style Sheets (CSS), Structured Query Language (SQL) and eXtensible Markup Language (XML).

Proprietary standards are typically made by companies that hope their standard will be the new and dominant standard in the market. For example Microsoft's Word format.

The most popular format for exchanging timetable information in Norway today is the REGTOPP format. Many agencies throughout Norway are publishing their data in this format. The Norwegian State Railways(NSB)²³ and the county agency in Rogaland, Kolumbus²⁴, are some agencies that also publish their data in the GTFS format.

2.3.1 SIRI

The Service Interface for Real Time Information (SIRI) is a European interface standard for exchanging real-time information about public transport services and

²³<http://www.nsb.no>

²⁴<http://www.kolumbus.no/>

vehicles. SIRI is using the XML protocol for exchanging the information distributed between computers. The information gathered by SIRI could be used for different purposes like providing departure times for each stop to be displayed on the actual stop.

2.3.2 General Transit Feed Specification

GTFS is Google’s format for exchanging public transportation timetables. When Google Transit was published, two formats were created in order to collect timetable- and real-time data, GTFS²⁵ and GTFS-realtime²⁶ for real-time information respectively. The whole idea is to allow operators to exchange their data and in return get a fully functional trip planner service that uses their data. New data that is fed to Google Transit is updated once a week.

Files	Used by BusTUC
FORMPAR.FRM	no
TURIX.TIX	yes
TURMSTR.TMS	yes
HPL.HPL	yes
DAGKODE.DKO	yes
DESTNAVN.DST	no
MERKNAD.MRK	no
GANGVEI.GAV	no
SAMTFK.SAM	no
SONE.SON	no
LINJE.LIN	no
VOGNLØP.VLP	no
TABVER.TAB	no
PERIODE.PER	no
RUTEPKT.RUT	no

Table 2.3: REGTOPP version 1.2

2.3.3 REGTOPP Format

The “Regional Trafikkopplysning (REGTOPP)²⁷ format” is a Norwegian standard for exchanging timetable information. It came as a result of the project “REGTOPP” which was running in the 1990s as described in Trafikanten (1996). This format is used for exchanging transit information between public transportation agencies and travel planners, like AtB and BusTUC, and is the “de facto standard” in Norway.

Together, all the files in Table 2.3.2 make up the version 1.2 of the REGTOPP format and define information such as the bus stops, what stops a line passes, and what numbers the buses have.

²⁵<https://developers.google.com/transit/gtfs/?hl=no>

²⁶<https://developers.google.com/transit/gtfs-realtime/?hl=no>

²⁷<http://labs.trafikanten.no/2011/4/13/dokumentasjon-av-regtopp-formatet.aspx>

Chapter 3

Contributions

This Chapter describes the contributions to the FUIROS project and the technologies used during the development.

3.1 MapTUC ²⁸

In order to test the expansion, new functionality, a survey and a new web page was created. It should present BusTUC answers in a map in combination with the text answer. By using technologies like CSS, HTML, JavaScript, AJAX and Google Maps API, the new web page could become more attractive for the users.

The next sections describe the development process of MapTUC.

3.1.1 Technology

Requirements

Before MapTUC was created, a set of requirements was defined:

ID	Requirement
1	Present a route in a map
2	Display stops as markers in the map
3	Keep the user interface minimalistic
4	Present real-time information in the map for each stop
5	Center the map to the user's current position
6	Interaction with the web page should trigger a response in less than one second

Table 3.1: Requirements MapTUC

Selection

After the requirements were defined the technology used to create MapTUC was chosen. Table 3.2 summarizes the technologies used in the development of

²⁸MapTUC is the web page <http://busstuc.idi.ntnu.no/maptuc/>

MapTUC.

Technology	Purpose
JSON	Data communication between server and client
HTML5	Supports a geolocation API
CSS	Easy to make a consistent layout for all of the popular browsers
JavaScript	Dynamic interaction on web page
jQuery	Used for making AJAX calls
Google Maps	Provides ways to add markers and overlays in the map
Google Direction Web Service	Used to draw a bus route suggestion

Table 3.2: Technology Used in MapTUC

The technologies in Table 3.2 are described further in the next sections.

JavaScript Object Notation (JSON)²⁹ is a data representation format that is used for interchanging information between different systems. Its format is easy to read and write. The data is also easy to parse and generated by computers. Altogether this makes the JSON format a popular choice when interchanging data from BusTUC through the web as a HTTP request.

HTML5³⁰ is the new hypertext markup language for web pages from the World Wide Web Consortium. HTML5 will replace not only the previous version of HTML, HTML 4.01, but also XHTML and the Document Object Model (DOM). Since HTML4 became a World Wide Web Consortium Recommendation in late 1999, the web has changed a lot. Some of the most important goals were to add support for the latest multimedia, keep it easy to read, consistency interpretation by the web browsers. The support for HTML5 is now covered in the most popular browsers like Google Chrome, Mozilla Firefox, Safari, and Internet Explorer, thus making it a popular choice for cross-platform applications. As of 17th of December 2012 HTML5 became a W3C Candidate Recommendation. The development phase of making HTML5 was based on the following standards:

- HTML, CSS, DOM and JavaScript should be the base for new features.
- The use of external plugins should be reduced.
- Improve the error handling.
- Markup should replace scripting.
- HTML5 should be device independent.
- The development process should be open for the public.

²⁹<http://json.org/>

³⁰<http://www.w3.org/html/wg/drafts/html/master/>

HTML5 is used in MapTUC to utilize the users' current positions. This is achieved by using the HTML5 Geolocation API that gets the geographical position of the user.

CSS³¹ is a language that is used to define the semantic presentation of a markup language like HTML. The design helps to separate the content and layout presented on a web page which makes it easy to style fonts, colors or layouts in different parts of the web page.

JavaScript³² is a programming language that primarily is used in a web browser to allow dynamic interaction on a web page. Using script-tags will allow the code to run locally in the browser. A common use of JavaScript is to change, add or remove text, or to open pop-up windows. JavaScript is used to center the map on the user's position when a user opens the web page, and other manipulations in the map and text responses.

jQuery³³ is a JavaScript library that simplifies the scripting of HTML on the client-side. It is an open source software and works across many browsers. This is what makes jQuery the preferred library when making the Asynchronous JavaScript and XML (AJAX) calls.

Google Maps The map service Google Maps³⁴ was chosen as the best solution as it provides an easy JavaScript API with a lot of functionality.

Using Google Maps allows the use of several overlays, plotting markers, adding pop-up windows in the map and showing directions. BusTUC already has the localization for each stop, which allows this information to be presented in the map, showing where a trip starts and ends.

The Google Directions API³⁵ is a web service that through an HTTP request calculates the route between a starting point(position or place name) and end, and potential waypoints in between. The result could either be in an XML or JSON format.

A trade-off by using Google's Directions API and the driving travel mode is that a bus route would in some cases not follow the same direction as a car would, as some of the roads are closed for traffic that is not public transportation. Only the bus operator actually knows where the routes go, and therefore the route shown on the map can be misleading.

In some cases, where a bus uses the ferry or boats to get from shore to shore, driving mode will be very confusing. For routes following a straight road, which is open for all traffic, the map directions will work correctly.

³¹<http://www.w3.org/Style/CSS/>

³²<http://www.w3schools.com/js/>

³³<http://jquery.com/>

³⁴<https://developers.google.com/maps/documentation/javascript/>

³⁵<https://developers.google.com/maps/documentation/directions/>

3.1.2 Implementation

An extension of the current server communication was done in order to allow MapTUC to present the BusTUC answers in the map. A new JSON response was made to return the information needed.

New JSON Response

BusTUC has standardized its answers from a textual response to also provide a JSON format (Engell, 2012). However, this data is not sufficient when used in combination with a map. By altering the Prolog code a new JSON response was achieved, as the result from the question “*From Vollabakken to Strindveien 37*” in Listing 3.1 is an example of.

```
{
route: [
{
name: "null",
type: "streetaddress",
role: "start"
},
{
name: "Vollabakken",
type: "busstop",
nr: 16010550,
role: "enter",
xcoord: 63.42174725228497,
ycoord: 10.400479207754223
},
{
name: "Høyskoleringen",
type: "busstop",
nr: 16010197,
role: "pass",
xcoord: 63.42032463199806,
ycoord: 10.404951905348156
},
{
name: "Jonsvannsveien",
type: "busstop",
nr: 16010206,
role: "exit",
xcoord: 63.42261365345412,
ycoord: 10.413402768241413
},
{
name: "Strindvegen 37",
type: "streetaddress",
role: "stop"
}
],
busstuc: "Holdeplassen nærmest Strindvegen 37 er Jonsvannsveien.
Buss 63 passerer Vollabakken kl. 1530 , kl. 1545 og kl. 1600 og
kommer til Jonsvannsveien , 0-14 minutter senere .Tidene angir
tidligste passeringer av holdeplassene. "
}
```

Listing 3.1: BusTUC’s map answers in the JSON format.

For each stop some attributes are specified:

- **name** The name of the street address or stop
- **type** Specified as a street address or a bus stop
- **nr** The stop number- used for real-time information
- **role:**
 - Start - starting from street address
 - Stop - destination to street address
 - Enter - Stop to boarder the bus
 - Exit - Stop to disembark the bus
- **xcoord** - Describes the latitude coordinate (-90,90)
- **ycoord** - Describes the longitude coordinate (-180,180)

Extending the Java Servlet

By extending the MultiBRIS server with a new Java Servlet³⁶, called Coord, information about each bus stops latitude and longitude is available in one easy HTTP request that only requires a bus stop identification number. The reason for this is that the existing coordinates in BusTUC are not entirely up to date, and therefore some bus stops might miss their corresponding coordinates. This solution potentially requires more HTTP requests, which could increase the total response time. However, the time spent making the new Servlet is less than the time spent writing a new program that will create the coordinates from scratch to Prolog-code or manually type in the missing coordinates. In cases were even the Coord Servlet cannot find the requested bus stops location, the bus stop has to be skipped when plotting the route and markers on the map. Otherwise the direction request will fail and no directions will be displayed. The process, see Figure 3.1, from when the user enters a question until the answer is readable in both text and on the map is described next.

1. An HTTP request with the query for BusTUC activates the map flag.
2. A JSON response is generated by BusTUC and sent back. (See Listing 3.1)
3. The response is parsed and if some of the stops do not have an x- or y coordinate, a new HTTP request to the MultiBRIS server with the bus number ID is sent.
4. A JSON response with the latitude and longitude to the bus number requested.
5. An HTTP request for calculating the directions between all the bus stops.
6. An answer with the calculated route.

³⁶<http://www.oracle.com/technetwork/java/javaee/servlet/index.html>

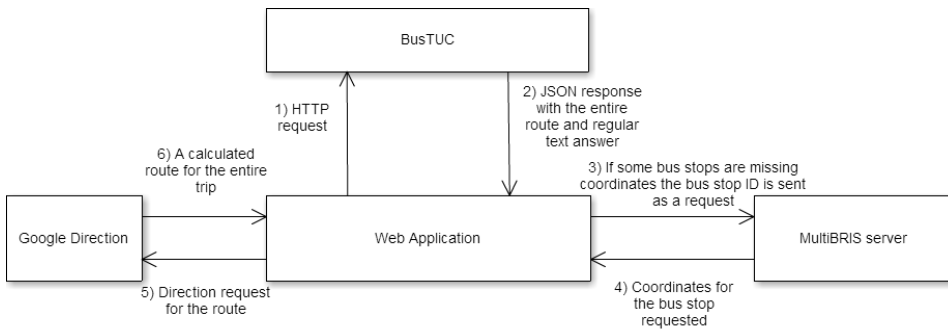


Figure 3.1: The communication process for returning an answer in MapTUC

Issues

One of the problems with using Google’s Directions API is the limitation of waypoints. There is a restriction of eight intermediate waypoints per HTTP request (23 for business customers). A typical bus passes quite a few stops on its route. Therefore, a trip that is equal to or less than ten stops (origin, destination and eight waypoints) will only need one request. For every trip that exceeds this, additional requests have to be sent. The results from these requests are put together by adding the overlays on the same map.

Another problem occurs whenever the GPS location for a stop is lacking from the dataset. Such stops must be omitted from the map, which will lead to a misunderstanding if the user wants to arrive or depart from such a stop.

3.1.3 Testing

A batch test is used as a benchmark to check whether the answer from BusTUC remains correct and reasonable after refactoring the code. The batch test consists of many different questions that are executed in a particular order, and the developer validates the answers returned manually. A modified version of the batch test was created in order to check the answers about tram departures.

Server Setup

During the development of this project the server in Table 3.3 has been used. It provides the web services for BusTUC³⁷ and MultiBRIS³⁸. It also provides the web interfaces for BusTUC³⁹. The MapTUC web page is also hosted by this server.

³⁷<http://busstjener.idi.ntnu.no/busstuc/oracle?q=samfundet>

³⁸<http://busstjener.idi.ntnu.no/MultiBRISserver/RealTime?bID=16010495>

³⁹<http://busstuc.idi.ntnu.no>

Attribute	Value
CPU	2x 5.2GHz, VMware shared pool
Memory	4GB dedicated
OS	Ubuntu 12.04.1 LTS

Table 3.3: Server information for busstjener.idi.ntnu.no and busstuc.idi.ntnu.no

3.2 Geographical Expansion

A geographical expansion of BusTUC should rely on common standards in order to have a streamlined import process that ensures the quality. For instance, the option of scraping data from the agencies web pages is foreclosed. This process will be too time consuming and lead to a lot of maintenance as well as processing and parsing time. Therefore, BusTUC should aim to provide an import procedure for standards such as GTFS and REGTOPP.

The current procedure of importing new datasets into BusTUC is partly automated, but some steps remain to be solved by an administrator. For each new dataset, the exact same procedure has to be followed in order to use the new information. An expansion will therefore require the import procedure to be executed each time and without too much manual effort.

BusTUC has since its release been using datasets from the bus operator in Trondheim, AtB. The information about street addresses, place names, etc., has been added manually over the years. So, whenever new datasets are imported, this information is unaltered. This procedure does not pose any major challenges when using data from AtB, because the existing knowledge is tailored to the area around Trondheim. However, this approach is not suitable if the knowledge is to be extended in the future, since there are too many lines of code to be maintained. Therefore, attempts to create automatic updates of the knowledge should also be considered. Recently the datasets for most of Norway have been released, which gives an opportunity to create a version of BusTUC outside its current knowledge domain.

3.2.1 Process of Importing new Dataset

New datasets are stored in a REGTOPP format. First the data must be processed. This involves both a Java program and a Prolog predicate in order to generate the necessary files, which together defines a route module in BusTUC. A Java program is made for parsing REGTOPP formats, both the newest and older versions, and will be used in order to import a new dataset.

The extraction of the dataset is done by a Java program that reads the four files in Table 3.4 line by line, to generate information for BusTUC.

File type	Information
DKO	Start date for the dataset and information on whether the route is driven that day
HPL	Each stop
TDA(v1.1)/TMS(v1.2)	Link each departure to a set of stops
TIX	Information of all departures

Table 3.4: REGTOPP files used to extract information

Quality of Data

An important step in gathering new information is to perform a validation check to check the quality of the data. The new map functionality is dependent on the location of each stop. The location for each stop is given by an x- and y coordinate. Both are declared in ten digits and should be specified in a UTM-grid notation. This information is not declared as a mandatory field in the REGTOPP format and might not be specified for some stops or in an entire dataset. It is a problem that the notations for the coordinates are not consistent. Table 3.5 describes the various notations used and the total percentage. For example, the x-coordinate is switched with the y-coordinate in a few datasets. This is unfortunate because the import procedure must be implemented to handle the data differently. 8.8% of the stops in the datasets did not have any coordinates specified. However, as this is not a mandatory field, this is to be expected.

X-coordinate	Y-coordinate	% of total stops
N 61o20,92	0E 00501,0	<0,1%
0000273592	0006806514	~ 84.5%
6.64363e+0	0000601291	~ 6.6% (X- and Y-coordinate is switched)
No coordinates		~ 8.8%

Table 3.5: Examples of the UTM notation in 32 datasets with a total of 65017 stops. X-coordinate is the easting position and Y-coordinate is the northing position in UTM.

After handling the different cases the converting from UTM to longitude and latitude notation is performed. However, 2.5% of these new coordinates are outside the boundaries of Norway. A reason could be that some routes go outside of Norway, but in most cases the information is incorrect.

3.2.2 Testing New Data

The first attempt considered the possibility to import every available dataset from Norway. The approach considered a solution of providing many datasets and only one output. A total of 37 datasets were tested to see if the import procedure was scalable. However, this was causing the execution time of the Java program to scale exponentially for each dataset. Therefore, the attempt was decreased to ten datasets. After the import process was completed, another problem was discovered. When querying BusTUC for new information, the returned answer

presented Sunday routes when it should have presented Monday routes. The reason for this behavior is that every dataset uses a Boolean string to specify if a specific route is driven this day or not. This is valid for a date specified in each dataset, and therefore when combining several datasets a common start date was not decided. The time restrictions did not allow a refactoring of the code, therefore the approach of combining every dataset into one output was discarded.

Still, an approach of only using one dataset was tested. A dataset from Nordland was chosen to make a test version of BusTUC. Listing 3.2 shows an example of querying the new version.

```
Question:
When does the bus from Oksebakken to Skavdalslia go?
Answer:
Bus 509 passes by Oksebakken at 4:50 pm , at 5:50 pm and at 6:50
pm and arrives at Skavdalslia , 45 minutes later .The hours
indicate the earliest passing times.
```

Listing 3.2: BusTUC's answer from dataset from Nordland.

The bus number used to identify a route in the REGTOPP format is not necessarily the same as the number displayed on the front of the bus. This is the case when using the data from AtB, and therefore the import procedure did not use the correct number when the information from Nordland was imported. The result in Listing 3.2 is an example of this. It tells the user to take bus number 509, although it is number 2, which is displayed when this route is driven. As BusTUC did not have any information about Nordland, except the stop names, it became harder to get reasonable answers.

3.3 Tram Support

Throughout the years the question and answers from BusTUC have been logged. They provide valuable information that could help enhance the current system. Reviewing the log for the past two months indicates an average of ten questions per day asking for tram information. Also questions for the airport express bus represent a fair amount of the questions that BusTUC currently has no information for. Therefore, by enabling BusTUC to provide answers about the tram, the same approach could be taken later in order to incorporate other transportation like trains, ferries and underground.

The goal is to enable queries about tram in the current domain in Trondheim, where there is only one tram route and easy to test. Since BusTUC previously had the functionality to answer question about the tram, a lot of the logic was in place when the review of code started. First, a review of the code was performed to get the overall picture of why the system fails to answer the questions about the tram.

By using a SICStus plug-in for Eclipse, called Spider, debugging can be done effectively. The debugger allows a breakpoint to be set in the code and then back-traces the predicate calls with the help of Eclipse. This was one of the most time consuming parts of extending the functionality of BusTUC.

The goal is that questions containing the word “tram” should be interpreted as a request for tram routes. This means that a translation from a first order logic to

the route database query (buslog) must be performed correctly. So, when a query contains the word “tram”, it must be expressed in the first order logic and then a translation to buslog must search for tram routes and not bus routes as before.

The challenge is to make the right assumptions to return the correct answer. In situations where a user queries for stops that both have a bus and a tram route without words such as “bus” or “tram”, the system must take a decision. Another problem is that some of the tram stops could be declared as street names or neighborhoods. To deal with this problem, it is assumed to return tram routes only when the word “tram” is a part of the query.

Chapter 4

Results

4.1 MapTUC

This section presents images of the new web page⁴⁰ made to utilize the map functionality. The overall thought was to keep the web page as simple as possible. Table 4.1 shows the browsers the web page was tested on.

Browser	Version
Opera	12.14
Internet Explorer	10
Google Chrome	27.0.1453.110 m
Mozilla Firefox	21.0

Table 4.1: The Browser used to test MapTUC

The layout was consistent throughout all the browsers tested. This was also true for the functionality and the geolocation service. When the web page is loaded for the first time, the map is centered at the user's location. This is presented in Figure 4.1.

⁴⁰www.busstuc.idi.ntnu.no/maptuc

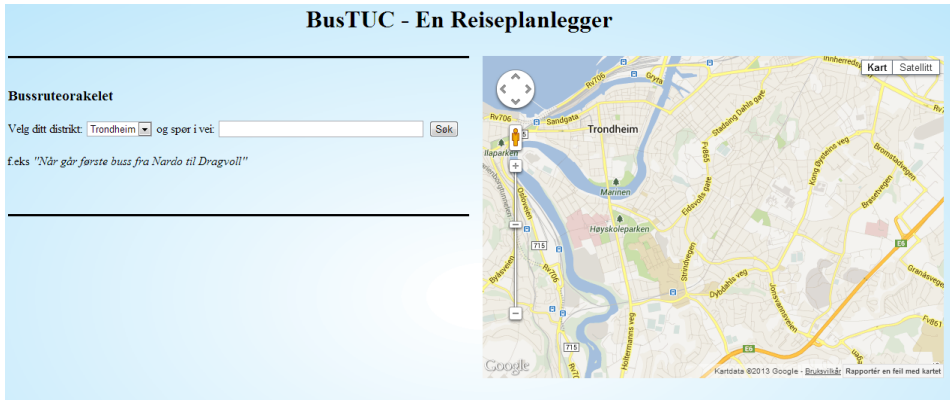


Figure 4.1: MapTUC design

An example of how MapTUC presents a solution is shown in Figure 4.2. Figure 4.3 present the pop-up dialog that describes the next five real-time departures within the map.

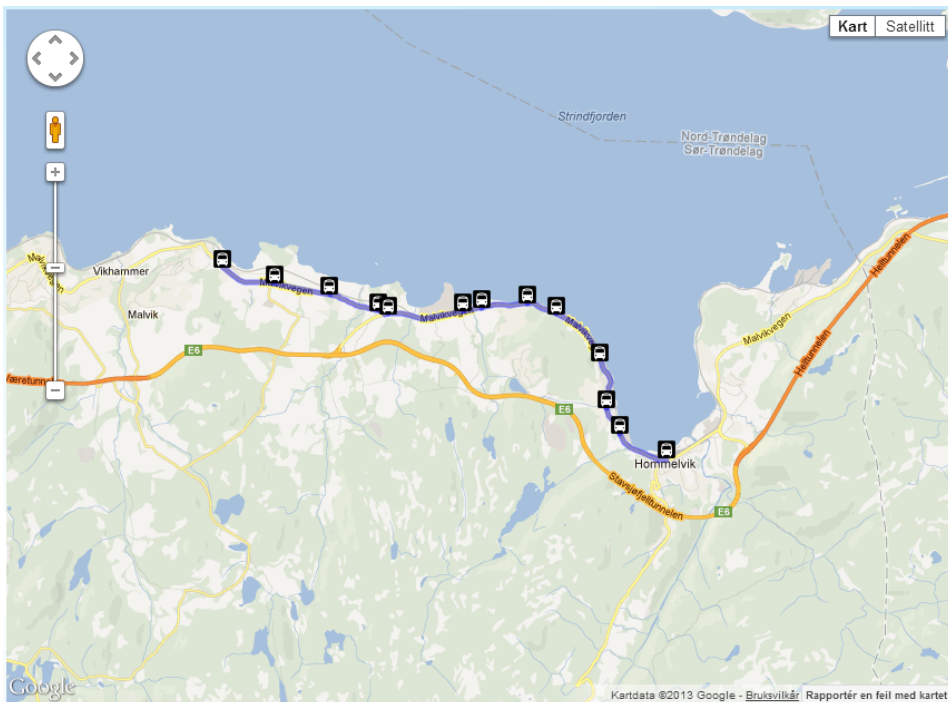


Figure 4.2: The new map response for the question “Malvik to Hommelvik station”

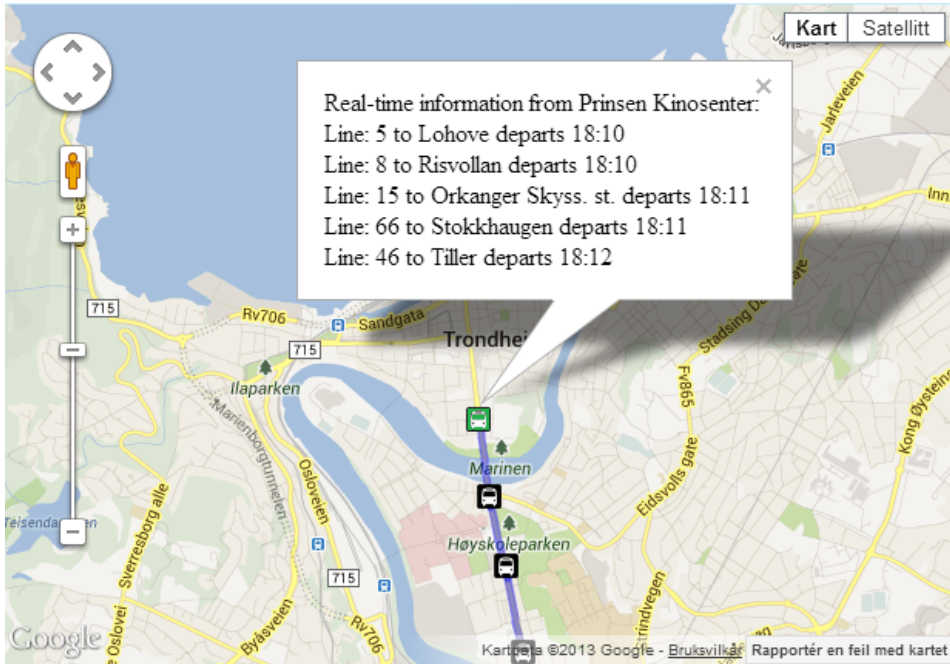


Figure 4.3: The real-time information in MapTUC

4.1.1 Response Time

This section presents the result from the response time for each BusTUC question returned and the rendering of the map with markers and overlays. During the solving of the scenarios each question was logged. A total of 144 questions were asked. A batch test that consisted of these questions was created. This test was performed by using a Java program that reads each question in the batch test and sends a HTTP request to the web service for BusTUC. The total executing time was divided by the total of questions.

The result was an average response time of 650 milliseconds.

The map is rendered for each question and the time for the markers and direction to be drawn is equally fast or faster as the BusTUC response.

4.1.2 Route Suggestion

The way BusTUC currently is implemented is that the first departure is taken as the best suggestion. This means that the first bus leaving to your destination would be considered as the wanted trip. However, there might be other buses that will depart later, but arrive earlier than the first choice. So users wanting to get from one place to another as fast as possible would not be happy with such an answer. In the cases where there is only one bus line nearby, this logic would perform as optimal. The trip from Malvik to Hommelvik in the Figure 4.2 shows an optimal route suggestion.

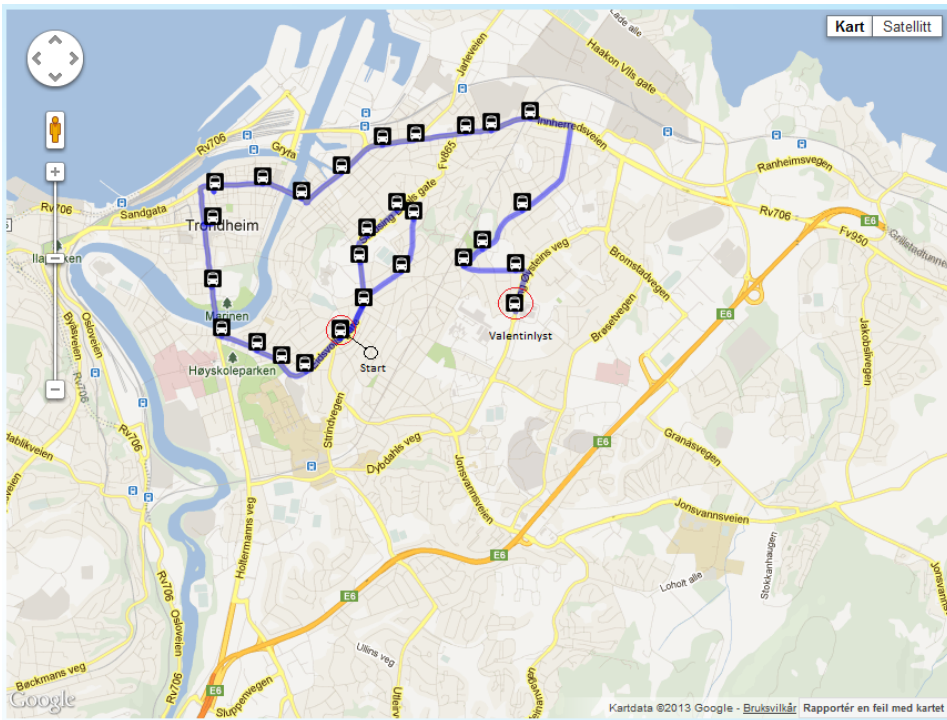


Figure 4.4: Displaying the route for the query: “From Strindveien 37 to Valentinlyst station”

BusTUC is also able to map a street address to the nearest stop. In the situation where you want to go from the street address “Strindveien 37” to the stop “Valentinlyst”, BusTUC would suggest that you take a trip to downtown before taking the bus back again. This suggestion would take approximately 26 minutes. This case is displayed in the Figure 4.4.

As we can see in the Figure 4.5 the fastest trip would be to take the bus from the station “Magnus Blindes Vei” to Valentinlyst. This will take approximately 4 minutes. This is 22 minutes faster than the solution suggested when typing in the street address Strindveien 37.

These two cases are good examples of how the logic is handled. It will try to present the best solution it could get from its database, and certainly it presents the best solution some times. However, based on total travel time it might not.

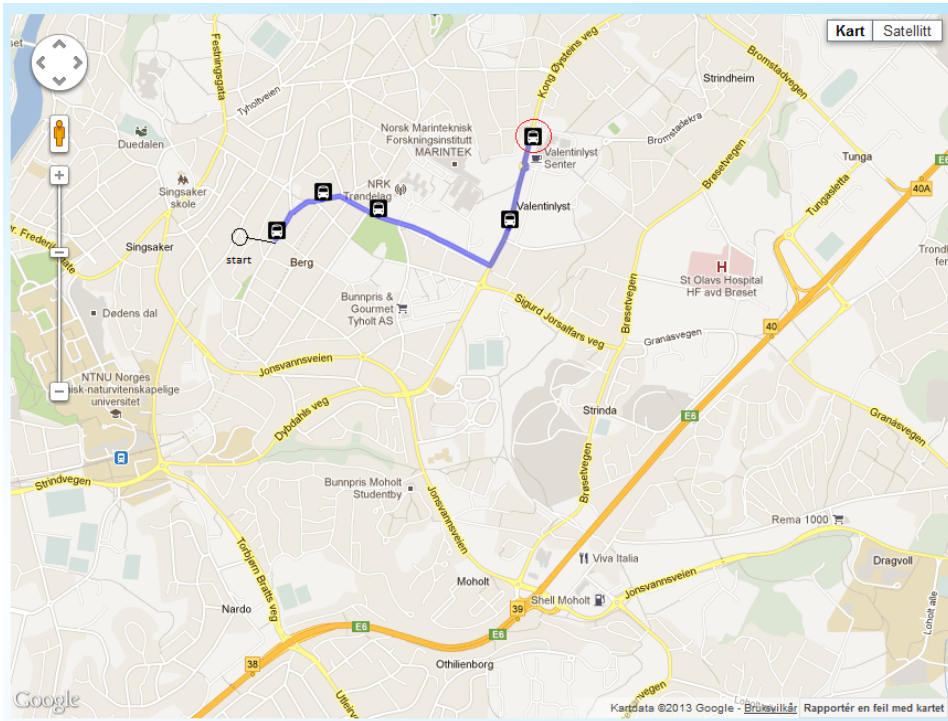


Figure 4.5: Displaying the route for the query: “From Magnus Blindes vei to Valentinlyst station”

4.2 Questionnaires

This section is going to present the results from the questionnaires. A sample size of 14 test subjects answered this form. A variety of test subjects from 10 to over 61 years took the survey. The youngest and the oldest participants used the longest time to answer the scenarios. Figure 4.6 shows the average time the test subjects used depending on whether they had any previous experience with BusTUC or not.

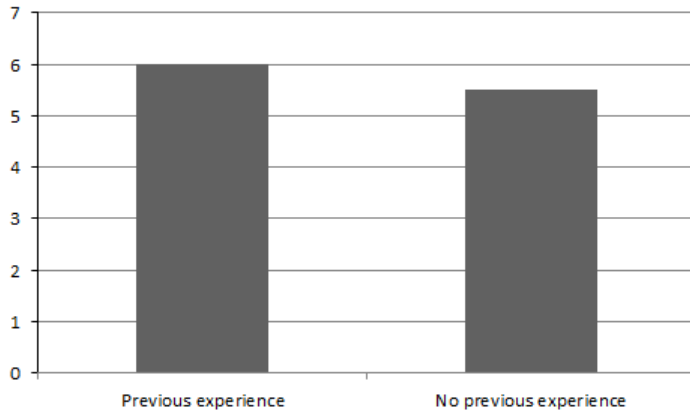


Figure 4.6: The average time (in minutes) spent on solving the scenarios for test subject with and without previous experience of BusTUC.

4.2.1 BusTUC's Answers

This section presents how the test subjects perceived the answers returned by BusTUC. The questions were asked to try to figure out what the test subjects thought of the answers returned from BusTUC. The results are given in Figure 4.7.

B1 I find BusTUC easy to use.

B2 I think that answers were correct and reasonable.

B3 I did not need to reformulate my questions to get a reasonable answer.

B4 I think the answer was returned fast enough.

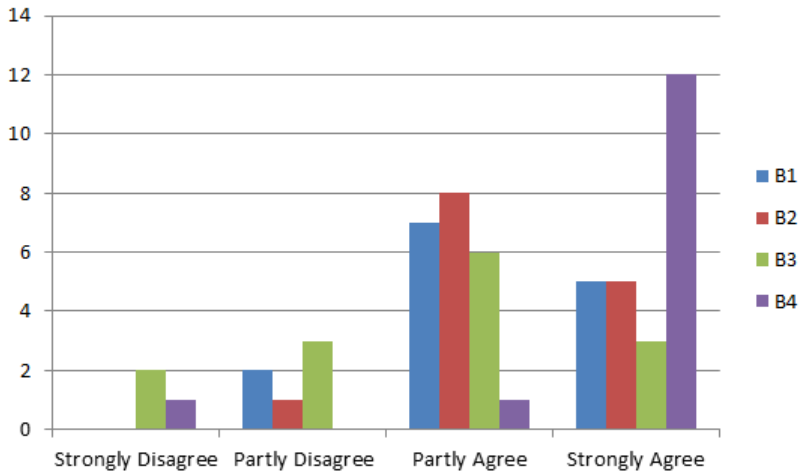


Figure 4.7: The test subjects' perception of BusTUC's answers.

4.2.2 Map Functionality

This section describes the test subjects' perception of the map in combination with the text answer. These questions were asked to determine if the new map functionality was a useful addition to the text answer. The results are presented in Figure 4.8.

M1 I find the use of a map in combination to the text answer useful.

M2 I think that the answer in the combination with the map gave the answer more credibility.

M3 I think that the travel route drawn in the map was useful.

M4 I think that the real-time information should be integrated into the answers given from BusTUC.

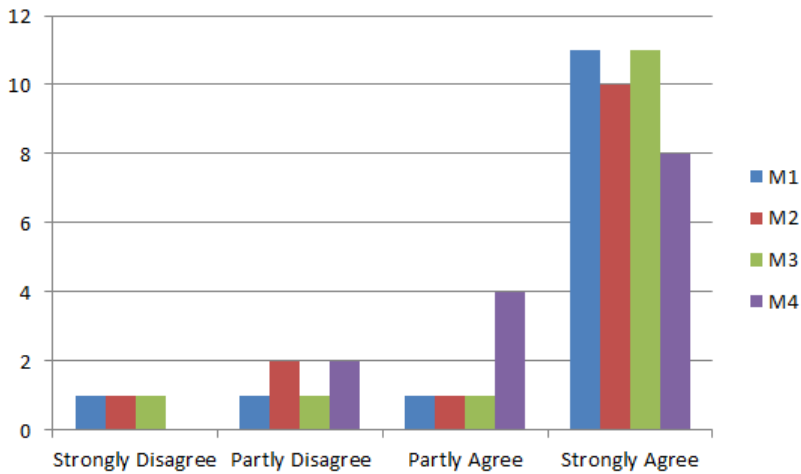


Figure 4.8: The test subjects perception of the new map functionality.

4.2.3 Evaluations

This section presents the results from the questionnaires of what the test subjects thought about adding more information into BusTUC. Figure 4.9 present the results.

E1 I would use BusTUC more if it had information all types of every public transportation.

E2 I think that BusTUC would become popular if it had support for all of Norway.

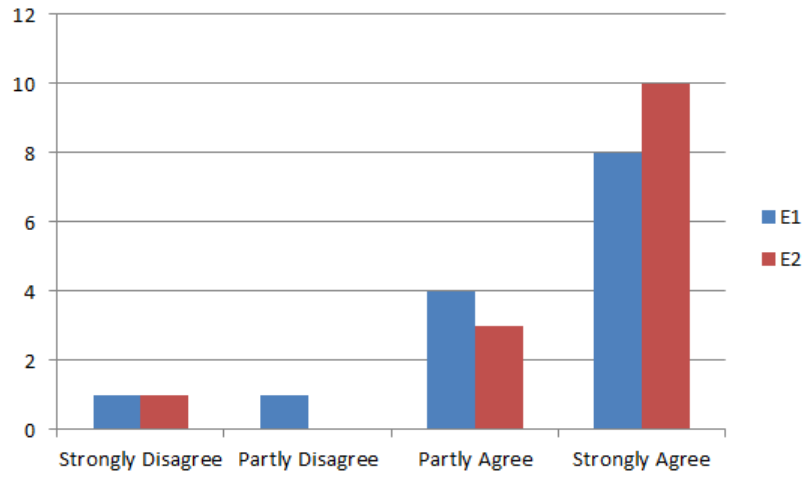


Figure 4.9: The test subjects evaluation of BusTUC's future



Figure 4.10: Tram Route in MapTUC

4.3 Tram Support

The Listing 4.1 shows the results after the tram information was added.

Question:

When does the tram from belvedere to skansen go?

Answer:

Tram 1 passes by Belvedere at 8:26 pm , at 8:56 pm and at 9:26 pm and arrives at Skansen , 4 minutes later .The hours indicate the earliest passing times.

Listing 4.1: BusTUC's tram answer

However, this new support does not work correctly for all the tram stops. The reason for this is the classification of words: some of the stop names for the tram are also classified as street names and bus stops. This prevents BusTUC from checking the tram routes.

The Figure 4.10 shows a route presented in the map.

Chapter 5

Discussion and Conclusion

This chapter will first discuss the results from the questionnaires and the new web page, before the research questions and goals are answered. At last a conclusion is provided.

5.1 Discussion

This section will discuss the results from Chapter 4. Both the questionnaires and the observation of the scenarios lay the base for some of the conclusions.

5.1.1 MapTUC

Based on the answers from the questionnaires, the new web page is determined to be a better solution than the old one. Almost all the test subjects thought that the map functionality was useful and that it contributed to increase the credibility of the answers returned from BusTUC. The reason for this is that the users can visualize the text answer in a map, and also that the map functionality provides means for validating the answer. For example, if the user asks for a street name and the answer returned is interpreted as another street name, this could easily be discovered in the map.

5.1.2 Geographical Expansion

Will an expansion of BusTUC, which can cover a greater part of Norway, preserve its accurate and intelligent answers? An expansion of BusTUC to cover all of Norway is within reach. However, by adding more bus stops and information into BusTUC, the accuracy of the answers could decrease, since the additional place names, street addresses and bus stops would also increase the ambiguity. A common place name like “*solbakken*” could be located all over Norway. Another problem that could occur is when the origin or destination is left out, in which case BusTUC will assume that the user wants to go from or to downtown. An expansion would therefore also struggle to know where downtown is, that is, in which city.

Will the total amount of correct answers decrease? The prototype of BusTUC that covers Nordland returned correct answers as long as the query contained two stop names in the dataset. This makes it much harder to query BusTUC and get useful answers in return. Therefore, a knowledge base for Nordland must be provided to ensure better suggestions. Like the travel planner TravelMagic (section 2.1.1), BusTUC could create a domain specific system for each county. This solution could help increase the correctness in the answers, but it could not be used as a travel planner for the entire Norway in one application. This approach will cause several individual versions of BusTUC and can be hard to maintain if the logic must be updated on all of them, instead of from one centralized place.

By adding the ability to answer questions about the tram, it should be easy to generalize the system to also incorporate other public transportation such as boats, ferries, airport express bus, train, and other solutions. However, this will cause a lot of planning and refactoring of the code and may cause additional problems. The test subjects did not comment on why they only were able to get route information about bus and tram, and not the airport express. Just by reading the name BusTUC, the user can think that this application could only provide bus information. This was true when this system first started in the late 90s, and the experienced users might think that this system only has bus information if nothing else is explicitly stated on the web page, for example.

5.1.3 Questionnaires

An interesting result shows that the test subjects without any prior experience with BusTUC used half a minute less to solve the scenarios than the ones with previous experience. This is an indication that natural language is something that every user feels comfortable with, and it is therefore easy to query the system, regardless of previous experience.

Around 57% of the test subjects use the official timetables from AtB to check the schedules and while only 21% use a mobile app. This is an indication that the test subjects without previous experience with travel planner applications think the official paper version given by the bus operator, AtB, are more reliable than other services. This confirms the necessity of providing the correct answer when creating a travel planner.

One of the comments highlights the potential of increasing BusTUC's route planner to consider more than one bus stop per street address. This was the case when querying for the night bus from downtown to a home address, where the answer given stated that there were no departures to the bust stop closest to the home address. However, the second closest stop is operated by the night bus. This indicates that the logic in BusTUC has a potential of evaluating more than one stop, in order to achieve better answers. Sometimes a user may consider leaving the bus at the second best option, rather than having no alternative for transport at all. Therefore an answer that tells the user that there are no possible routes, when there actually exists one close to the enquired address, will decrease the credibility of BusTUC. This is a functionality that SL⁴¹, which administrates the public transportation services in Stockholm, already support. The user can check

⁴¹<http://reseplanerare.sl.se/bin/query.exe/sn>

a radio button to search for a quicker route from other nearby stops, than the specified origin and destination stops.

Another comment was that BusTUC should have the ability to suggest the tram, for instance, in the cases where a bus route does not exist, even though the user asks explicitly for bus, since some users might not know that there exists any tram route alternative to the bus. Adding such information can help increase the overall intelligence and credibility of BusTUC when the users do not think that the transportation mode is important.

Many of the test subjects got confused by the real-time information displayed in the map. For some it made no sense that real-time information for other routes than the one given in the text answer was presented. These feedbacks emphasize the importance of providing consistent data, and to avoid misunderstandings by only presenting the same information (bus number). The map should be a tool that enhances the user experience and not a confusing addition to the web page. However, 79% of the test subjects strongly agreed that the map was useful in combination with the text answer. Because most travel planners today use a map, this is something that the test subjects might have previous experience with.

The result from question M2 in Figure 4.8 shows that 72% of the test subjects thought the answer in combination with the map gave a more credible answer. This indicates that a map might help the users confirm that the text answer retrieved is actually the route drawn in the map.

One comment was that the map is more useful in the cases where the user is unfamiliar with the surroundings than the regular commuter. This is correct for users taking one route back and forth, for example to their job or school each day. However, for users that want to know where a route goes, the map will be a valuable feature on the web page. Even though the use of Google Maps and the directions drawn was sometimes not perfect, at least the stop markers were correctly plotted.

Some test subjects thought the answers should have integrated the real-time information if there were any delays in the route. The answers to question M4 in Figure 4.8 shows that $\sim 57\%$ strongly agreed and $\sim 29\%$ partly agreed with this. Moving the real-time information from a clickable marker into the text answer can return a more reliable and faster answer. Then only one question is needed to get the route drawn, the real-time information and the text answer.

36% of the test subjects strongly agreed that the system was easy in use and 50% partly agreed. Since the web page consists of only one input field and a map, the risk of mistakes is small.

An interesting result shown in Table 4.7 is that most of the test subjects had to reformulate their questions in order to achieve a reasonable answer. When solving the scenarios, a minimum of 5 questions would be needed; however, a total of 144 questions divided by 14 test subjects gives around 10 questions on average. This means that two questions were asked on average in order to get one correct answer. One of the reasons for this is that the test subjects tried queries containing street names or place names that BusTUC has no information for. Another reason for this could be misspellings caused by the users trying to type too fast.

86% of the test subjects strongly agreed that the time before an answer was presented was short enough. This is an indication that the back-end service is returning the responses at a sufficient speed. Every question asked in the survey

was stored and later used to query BusTUC in order to determine the average response time per question. As stated in Chapter 4, the system has an average response time of 650ms. This indicates, according to Miller (1968), that the system lies under the acceptable limit that allows the user to stay focused on the task.

71% of the test subjects strongly agreed that BusTUC would have become more popular if it had had information about the whole of Norway. Also, the test subjects agreed that they would start using BusTUC if it had information about all types of public transport. These results indicate that by extending BusTUC's functionality and domain, users could be satisfied with the results returned and not choose other travel planners in order to find an answer.

All test subjects answered that they would recommend BusTUC to a friend or colleague. Since 57% of the test subjects did not have previous knowledge of BusTUC, this might indicate that even though this system has existed for many years, not all travellers in Trondheim have used or even heard about it. Hence, there is a potential of attracting more users if they will recommend BusTUC to other people unfamiliar with this travel planner.

One of the observed differences in the test subjects' questions was the verbosity. Especially, both the youngest and oldest test subjects tended to be more verbose when querying BusTUC. Nonetheless, these questions returned the same reasonable answers as shorter and more incomplete questions. This indicates that natural language processing has an advantage by including a variety of users.

Another observation was that the youngest and oldest subject in this test used most time to write their questions. This is most likely to be explained by their keyboard skills. By implementing autocomplete-suggestions when typing, the time writing questions can be decreased.

5.2 Answers to Research Questions and Progress towards the Goals

BusTUC's strength is how it understands questions. A regular question about the bus time tables should give a reasonable answer. The next sections answers the research questions as defined in the introduction:

5.2.1 Research Question 1

How do users perceive the answers given by a natural language travel planner?

The test results indicate that the overall perception of the answers is accurate. This applies to test subjects both familiar and unfamiliar with BusTUC prior to the test.

5.2.2 Research Question 2

What could help increase the natural language travel planner's credibility?

Combining a presentation with a map could help increase the credibility of BusTUC. As the test results show, the use of map was a factor that increased the users' perception of the accuracy. Also 71 % of the test subjects meant that

by extending the functionality to cover all types of public transportation, BusTUC would become more popular. By increasing the popularity an increase in credibility will also be achieved.

5.2.3 Research Question 3

How can the local natural language travel planner cover the entire world?

The intelligence accumulated in Trondheim could not easily be adopted to a new domain without making changes to the knowledge base of the system. The result by testing the timetables from Nordland in MapTUC shows that there is a potential, but the knowledge has to be adopted. An approach to generalize a support for entire Norway, for example by adding support for other data formats, could be the key to expanding the system to cover the rest of the world.

5.2.4 Goal 1

Create a new web page that is able to present BusTUC answers in a map. This involves adding a new JSON response to all BusTUC answers.

The new web page was created and worked properly in the most popular browsers. The answers were displayed in the map despite the fact that they did not always return the correct route. A JSON response was successfully integrated into BusTUC.

5.2.5 Goal 2

Add support for tram route information in BusTUC.

A tram support was enabled in BusTUC, but because of time shortage it was not possible to enable the support for all tram stops. Therefore, the approach for enabling the tram could not be used in order to activate information about trains, ferries, etc. Hence this goal has only been partially achieved.

5.2.6 Goal 3

Import a new dataset to see if BusTUC is able to provide the same intelligent answers, in that domain. This approach should enlighten the challenges in adding new information into BusTUC.

A dataset was tested outside of Trondheim, but BusTUC was able to only answer questions that contained bus stop names; therefore the intelligence of BusTUC was not as good as in Trondheim.

5.2.7 Goal 4

Conduct a test to see if the users find BusTUC's answers credible. The test subjects should be presented with different scenarios to solve on the new web page, created in order to measure the perception of both the answer and the map feature.

The new web page was used in order to solve the scenarios and all the users were able to finish, without any big issues. Both the answer from BusTUC and the map feature was evaluated.

5.3 Conclusion

BusTUC could be expanded to cover the entire world, if the data is available and standardized. The result from using a dataset from Nordland shows that it is possible to expand the current system to another region. However, keeping up with major companies such as Google with their Google Map service will be hard. Therefore, BusTUC must continue to acquire domain specific information that will keep answering questions in an intelligent manner. This means gathering information that is not given in the datasets from the public transportation agencies.

The support for tram is the first step of making BusTUC able to answer more than just the bus schedules. An extension from bus to other means of transportation will increase the complexity of BusTUC, but it can attract more potential users.

The new web page, MapTUC, is a contribution that emphasizes the use of a map to present the results. The map functionality solution was a factor that increased the credibility to the BusTUC answers. This solution depends on the quality in the datasets, such as making sure that the bus stop markers are in the right places on the map.

MapTUC can also be adapted to mobile phones that have smaller screens to improve the accessibility. The map feature can furthermore be used as a verification tool to check if the stop locations are marked correctly on the map.

The reason for BusTUC's popularity has been its support of understanding natural language queries and together with a telephone service this is a universal design that can simplify the communication between BusTUC and its users. Instead of trying to beat other travel planners that have better route suggestions, a solution can be to let BusTUC provide a standard output that could be implemented and used in other travel planners to enable normal text as a search input. This could be an efficient and user friendly functionality for both regular and experienced users.

Chapter 6

Future Work

This chapter describes the enhancement that should be considered in the future.

6.1 Integrating Real-Time Information into BusTUC's Answers

The challenge of integrating real-time information for each departure in BusTUC should be carefully thought through. The functionality and services are already available, it is only a matter of how this information could be tied together without decreasing the performance of BusTUC. For instance, an answer containing more than one suggestion could imply several calls to the MultiBRIS server, one for each stop. On the other hand, the answers may differ from the actual passing time if the real-time information is not working properly. However, access to real-time information is what the everyday commuter might want. In contrast, the people reading the official timetables from their agency might want a more reliable answer such as the scheduled arrival time. This potential conflict of interests could be solved by returning both the scheduled time and the real-time in case of delay. Regardless, this solution should not be a tradeoff for the performance. The overall goal should be to return an answer within a second, or else the users might find this service too slow.

6.2 MapTUC

Other ways of drawing the routes in the map should be considered. This could give a better drawn route when, for example, displaying a tram route that does not follow a road. Instead of using Google Maps Direction API, the marked direction could be drawn as a straight line from one stop to the next. This would, however, just give an indication of which roads the transportation uses, and not the actual one.

6.3 Logging

A system that utilizes the information gathered from the logs could be helpful for improving the current knowledge of BusTUC. Such a system can extract and analyze the question asked and the answers returned, and not keep information that can be connected to a specific user to preserve the users' privacy. Analyzing the most used words and bus stops could give an overview of where the users travel. This information could be helpful for both further enhancements in BusTUC and AtB. This system could also detect if the users ask for other information such as the train, tram, airport bus, and others. This system should also log the response times to determine if they have increased since the last update or not.

6.4 Expansion

The challenge of including information about boats, ferries, train, etc. is to decide which means of transportation the user is asking for. For instance, if the user asks explicitly for tram routes, an answer should present tram suggestions. However, in cases where a user only specifies the start and stop destinations, a more advanced logic must be in place. If these destinations only exist in the bus domain, BusTUC should present a bus suggestion. Otherwise, the answer should return other transportation means as well. Also place names and street addresses have the same issue, as they could be localized in different cities. This ambiguous input is one of several issues that need to be addressed. In order to incorporate the information from other datasets than from AtB, an abstraction of the logic and the facts need to be addressed. An expansion must consider splitting BusTUC into more modules. This would allow an easier scalability and maintenance in the future. For instance, a module that only has information for Trøndelag could be exchangeable by a module that has information for Nordland while still allowing BusTUC to return intelligent and reliable answers. This means that the logic should not be directly tied to the information modules and to a general module that tries to reflect the real world to abstract redundant information in each domain specific module. Until a more scalable solution exists, the expansion could only evolve gradually. However, the import procedure that converts the datasets to useable information for BusTUC is working perfectly for other parts of Norway. This is not the hardest part, as the datasets use the same REGTOPP format, but how to use the new information to provide intelligent answer, as BusTUC has done many years in Trondheim, is the brainteaser.

Bibliography

- Tore Amble (2000). BusTUC: A Natural Language Bus Route Oracle. In *Proceedings of the sixth conference on Applied natural language processing*, pages 1–6. Association for Computational Linguistics.
- Tore Amble (2009). BusTUC–A Savant Level Intelligent Bus Oracle. In *Norwegian Artificial Intelligens Symposium (NAIS)*. Tapir Akademisk Forlag.
- Runar Andersstuen, Trond Engell, Rune Sætre, and Björn Gambäck (2012). A Multiple Platform Approach to Building a Bus Route Information System for Mobile Devices. In *12th International Conference on Innovative Internet Community Systems*.
- Runar Andersstuen and Trond Bøe Engell (2011). MultiBRIS: A Multiple-platform approach to the Ultimate Bus Route Information System for Mobile Devices. Technical report, Department of Computer and Information Science, NTNU.
- Jon S Bratseth (1997). BusTUC - A Natural Language Bus Traffic Information System. Master’s thesis, NTNU.
- Trond Bøe Engell (2012). TaleTUC: Text-to-Speech and Other Enhancements to Existing Bus Route Information Systems. Master’s thesis, Norwegian University of Science and Technology.
- Brian Ferris, Kari Watkins, and Alan Borning (2010). OneBusAway: A Transit Traveler Information System. In *Mobile Computing, Applications, and Services*, pages 92–106. Springer.
- Ole Hartvigsen, Erik Harborg, Tore Amble, and Magne Hallstein Johnsen (2007). Marvina–A Norwegian Speech-Centric, Multimodal Visitors’ Guide. In *NODALIDA 2007 Proceedings (The 16th Nordic Conference of Computational Linguistics)*.
- Magne Hallstein Johnsen, Tore Amble, and Erik Harborg (2003). A Norwegian Spoken Dialogue System for Bus Travel Information. *Spoken Language Technology in Telecommunications*, pages 125–131.
- Christoffer Marcussen, Lars Eliassen, Rune Sætre, and Björn Gambäck (2012). Context-Awareness and Real-Time Information in an Intelligent Smartphone Application. In *12th International Conference on Innovative Internet Community Systems*.

- Christoffer Marcussen and Lars Moland Eliassen (2011). TABuss: An Intelligent Smartphone Application. Technical report, Department of Computer and Information Science, NTNU.
- Robert B Miller (1968). Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, pages 267–277. ACM.
- Roger S Pressman and Darrel Ince (1992). *Software engineering: a practitioner’s approach*, volume 5. McGraw-hill New York.
- Antoine Raux, Brian Langner, Dan Bohus, Alan W Black, and Maxine Eskenazi (2005). Let’s Go Public! Taking a Spoken Dialog System to the Real World. In *in Proc. of Interspeech 2005*.
- Trafikanten (1996). *The REGTOPP-format (version 1.2)*. <http://labs.trafikanten.no/media/12753/RF1-2-1-1.pdf>.
- Markku Turunen, Jaakko Hakulinen, Anssi Kainulainen, Aleksi Melto, and Topi Hurtig (2007). Design of a rich multimodal interface for mobile spoken route guidance. In *Proceedings of interspeech*.
- Marvin V Zelkowitz and Dolores R. Wallace (1998). Experimental models for validating technology. *Computer*, 31(5):23–31.