BO STENDAL SØRENSEN

PERSONAL VISUALISATION OF SPATIOTEMPORAL AND SOCIAL DATA.

KONGENS LYNGBY 2012

The front and back are visualisations of
location data from the Sensible DTU project.
Each color represent a participant in the
project during June and July 2012.

Copyright © 2012 Bo Stendal Sørensen

TECHNICAL UNIVERSITY OF DENMARK
Anker engelundsvej 1
DK-2800 Kongens Lyngby
Denmark

INFORMATICS AND MATHEMATICAL MODELLING
Phone +45 45253351
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-MSC-2012-77

This thesis is dedicated to my parents.

Abstract

THIS THESIS IS A CASE STUDY in the analysis, design and implementation of a personal informatics system that enable users to gain self-knowledge and self-reflection. The system, Sensible You, is based on data collected in the Sensible DTU project. The Sensible DTU project collect data using mobile phones from students and staff participating in a trial.

Privacy issues was analysed and discussed as the data collected and visualised could be regarded as very personal.

Using location and device proximity to infer nearby people, the user is presented with personal visualisations of their data. One visualisation present the user with a map, where they are able to explore how their location changes during the day. Furthermore they are able to playback the data. While analysing the location data it was found that clustering was required. Both a K-Means and a DBSCAN algorithm was implemented and evaluated for this task. This resulted in the release of an open source JavaScript clustering library. To visualise the infered social connections, different visualisation options was analysed and evaluated. The result is a chord graph, showing how much time users spent with each other.

The design and implementation of the visualisations was based on theories by Bertin and Tufte, as well as other related work. An evaluation consisting of qualitative user feedback and performance tests was conducted. While the user feedback was limited, it showed good results and participants was positive towards the system and project as a whole.

Perspectives for further work include basic improvements to the project, but also the usage of spatio temporal analysis in a personal informatics context. Furthermore we argue that the privacy issues could be solved by decentralising some of the personal informatics systems.

Preface

This thesis was prepared at Department of Informatics and Mathematical Modelling (IMM), at the Technical University of Denmark in partial fulfillment of the requirements for acquiring the degree of Master of Science in Engineering (Digital Media Engineering).

The thesis supervisors are Jakob Eg Larsen, Michael Kai Petersen and Sune Lehmann Jørgensen, Department of Informatics and Mathematical Modelling, Technical University of Denmark.

Lyngby, 31-July-2012

Bo Stendal Sørensen

Acknowledgements

First of all I would like to thank the participants in the pre-alpha test for their feedback on the system.

Furthermore, I would like to thank all the people working on Sensible DTU: Arkadiusz Stopczynski, Søren Ulrikkeholm, Vedran Sekara and Andrea Cuttone. You all gave valuable advice and feedback during meetings.

A special thanks to Niklas Quarfot Nielsen for keeping my head up all the way from Livermore, California, as well as for comments on this thesis. Also my deepest gratitude to Michael Lunøe for the countless discussions on visualisations and personal informatics. I also owe a big thank you to my brother for reviewing this thesis.

Finally I would like to thank my advisors Jakob Eg Larsen, Michael Kai Petersen and Sune Lehmann Jørgensen. Your support and advices have been invaluable. I am truly grateful for the opportunity to work on the Sensible DTU project.

Thank you.

Contents

From caves to big data

Humans have been drawing pictures for thousands of years. Some of the oldest known are from the Chauvet caves, dated 32000 BP[1]. It was however not until much later we began using drawings to visualise information - the earliest being maps.

[1] J Clottes. Chauvet Cave: the art of earliest times. 2003

In the 17th century the first visualisations of scientific data appeared. As the prominent science of the time was astronomy, it was mostly illustrations of celestial bodies. Some of the first were made by Galileo Galilei. Instead of using text, Galilei showed the placement of Jupiters moons by simply drawing how he saw them in the telescope. Today this might seem trivial, but compared to contemporary texts this was a radical change. Even though drawing-techniques were unpar with todays standards, Galilei and his contemporaries were able to communicate the information in their visualisations very clearly.

While the work of Galilei seems unfinishied and note-like by todays standard, the almost contemporary work by Chrstiaan Huygens seem crisp and not unlike something we would see today. This is seen in Huygens' *Systema Santurnium*, which among other things show the ring inclination of Saturns ring.

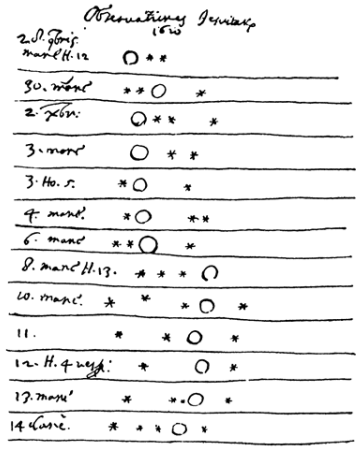

Figure 1: The placement of Jupiters moons by Galileo Galilei, from *Sidereus Nuncius* (1610).

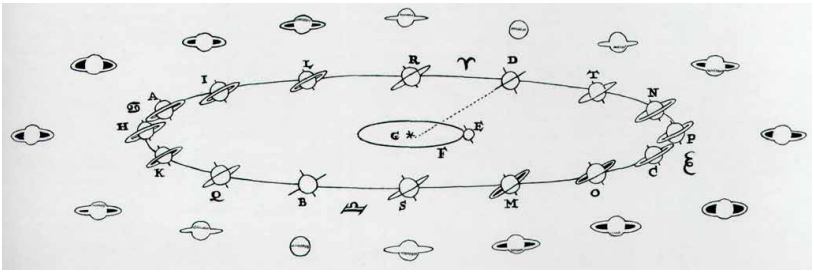

Figure 2: Saturns ring inclination as illustrated by Huygens in *Systema Saturnium* (1659).

During the enlightenment, William Playfair made tremendeous progress in the field of information graphics.

Playfair invented many of the most important graphs and charts, still widely in use today. Playfairs inventions count bar graphs, line graphs, histograms, area graphs and pie charts (circle graphs)[2].

Later, in the 19th century, Charles Minard, a french engineer and cartographer, refined the work of Playfair to make incredible graphic visualisations. His flow map of Napoleons march to and

[2] William Playfair. *Commercial and political atlas*. 1786; and William Playfair. *The statistical breviary*. 1801

from Moscow is even considered the best visualisation ever made[3]. The map combines a geographic map with the size of Napoleons army, as well as a timeline and the temperature during the march. The simplicity of the map together with the comlexity of the data is what makes this map stand out.

[3] H Wainer. How to Display Data Badly. *American Statistician*, 38(2):137–147, 1984; and Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, Conn., 1983

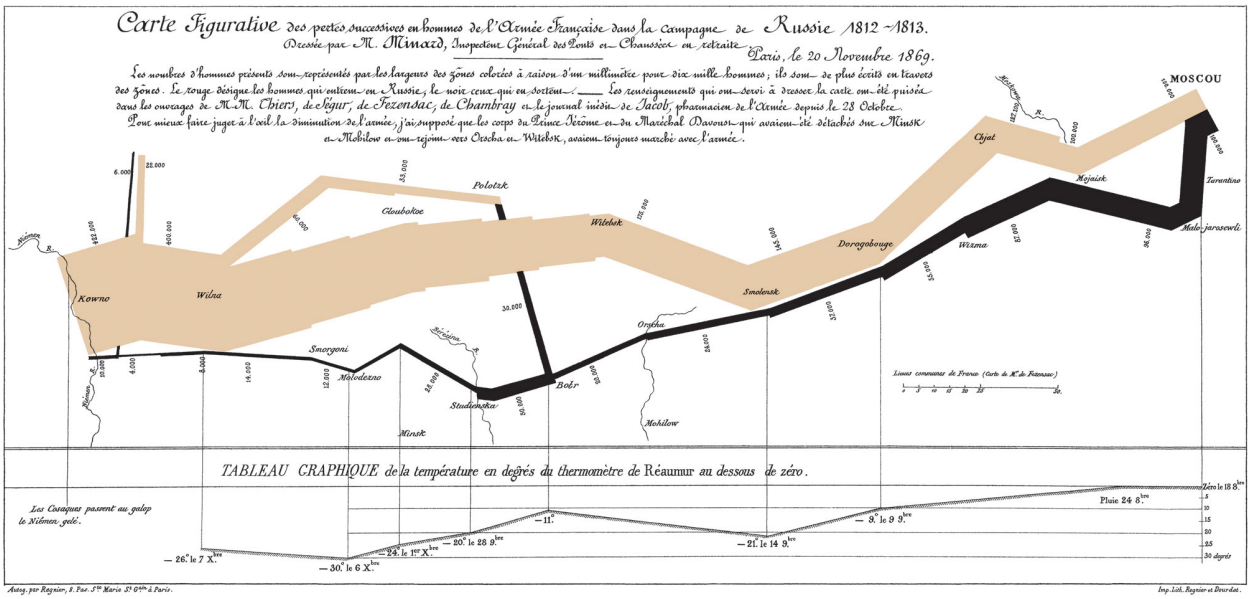

Figure 3: Napoleons march to and from Moscow in 1812 as illustrated by Minard in *Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813* (1869).

IT WAS NOT UNTIL the 20th century people actually began to study what good information visualisation is. Some of the initial work was made by Jacques Bertin in *Semiologie Graphique (Semiology of Graphics)*[4].

He looked at information graphics as a language consisting of symbols. In his study of these symbols, he found that the brain is only able to handle a finite amount of visual variables (*e.g.* position, size, color, texture, orientation). Furthermore he found that the visual variables has a heirachy of which kind of information (nominal, ordered or quantitative) they are able to convey. To make it easier to compare different forms of visualisations, Bertin defined the notion of graphical schemas. The schemas show how different information types are used in the visualisation.

[4] Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps (translation from French 1967 edition)*. ESRI Press, 1st edition, 2011. ISBN 9781589482616

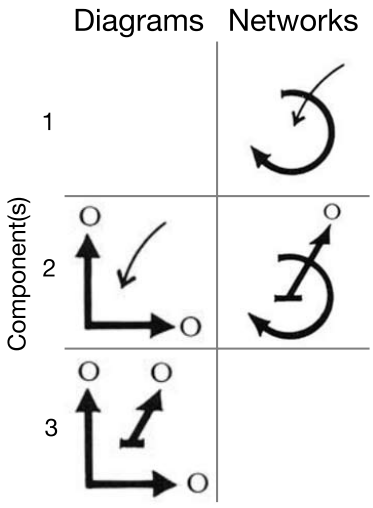

Figure 4: Examples of graphical schemas and the number of data components they contain.

INSPIRED BY BERTIN, Tufte started working with information visualisation. In *The Visual Display of Quantitative Information* he defines five principles of good data visualisation: Graphical integrity, data-ink, chartjunk, data density and small multiples[5].

Graphical integrity is the notion of truth in the visualisation. He describes how the representation of numbers (*e.g.* a bar in a bar graph) must be directly proportional to the number it represent.

Data ink is the ink on a graph that represents data. All other ink is non-data ink. He defines a good graph as having as high a ratio of data ink to non-data ink as possible.

Chartjunk is unnecessary graphical effects. Examples of this

[5] Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, Conn., 1983

include unnecessary use of multiple colors and unnecessary use of 3d perspective to graphs, as often seen in powerpoint presentations.

Data density describes how much of the total graph area is dedicated to actual data. He prefers graphs with a high data density.

The concept of small multiples are small graphs repeated many times with different data. It enables the author to visualise large quantities of information in a relatively small amount of space.

THE FOUNDATION OF INFORMATION VISUALISATION has not changed since Bertin and Tufte defined it. The latest advances in the field has been within new types of visualisations, new tools and frameworks, and how to visualise large amounts of data.

The need to visualise larger and larger amounts of data has increased along with processing power, processing tools and cloud technology. Previously, a mainframe was required to process large amount of data. Now, anybody can rent computing power at a low hourly rate. Furthermore frameworks such as Hadoop[6] has minimised the engineering effort required to run large computations on clusters of computers.

[6] http://hadoop.apache.org/

As computing power has become almost ubiquitous, the need for good visualisations of the, sometimes large amounts of, data has become aparrent. The new forms of visualisation makes it possible to comprehend much larger amounts of information than previously.

Introduction

People have been collecting information about themselves, to reflect and gain self-knowledge, for centuries. From a common personal diary to Benjamin Franklins tracking of his 13 virtues[7]. In recent years, tracking various information about oneself have become easier due to advances in computing power and sensor technology. Furthermore computers are capable of visualising datasets that could not easily be visualised by pen and paper.

In recent years we have seen people tracking everything from sleep[8] to pill intake[9]. Some people use self-tracking to gain self-knowledge and reflection. Others use it to drive behavioral change.

To help people gain self-knowledge the collected data must be presented in a meaningful way. While this might be trivial with simple datastreams, as we gain access to more datastreams and those datastreams gain complexity, presenting the data become non-trivial.

[7] Benjamin Franklin, John Woolman, and William Penn. *The Autobiography of Benjamin Franklin*. P. F. Collier & Son Company, 1909

[8] Zeo is a headband that allow users to track their sleep patterns (www.myzeo.com).

[9] Nancy Dougherty used pills with sensors to track pill-intake (quantifiedself.com/2011/08/nancy-dougherty-on-mindfulness-pills).

Sensible DTU

A research project, Sensible DTU, at The Technical University of Denmark (DTU) aims to gain knowledge about social systems by utilising the capabilities of modern smartphones. The smartphones, given to students and staff, collect information about their location, who is nearby, call history, communication on online social networks etc. An artificial limitation enforced in the system restrict all data gathered to the DTU campus area.

One objective of the project is to collect and analyze the overall dynamics of the spatiotemporal networks, i.e. on a macro-scale. Another aspect is to investigate if the behavior of participants changes when they have access to personal informatics and if changes in the interface affect behavior.

The vast amount of data collected for each user makes this a non-trivial task, both in terms of handling the data and showing it to the user.

In June and July 2012, a pre-alpha test was carried out to test the data-collection app used in the project. The test consisted of 21 participants, all working at the department of informatics and mathematical modelling at DTU. During the test, participants had

access to the personal informatics interface described in this thesis. In september 2012, approximately 200 phones will be given to new students at DTU. They will also have access to the personal informatics interface.

Goals

THIS THESIS IS A CASE STUDY in the analysis, design and implementation of a personal informatics system, Sensible You, that enable users to gain self-knowledge and self-reflection. The system is based on the data collected in the Sensible DTU project.

The learning objectives of the project is to

- Analyse and evaluate privacy concerns in regards to collecting and visualising sensitive data.
- Analyse current research regarding visualisations and graphing.
- Design a personal informatics system capable of displaying a large and complex dataset.
- Implement a system based on the proposed design.
- Evaluate the devised system based on parameters such as usability and performance.

The outcome is a working prototype able to visualise the data gathered by participants in the Sensible DTU project.

Related work

The work in this thesis is related to three distinct research areas: Personal informatics, visualisations and spatiotemporal data.

In this chapter, the related work in those areas are described.

Personal informatics

Personal informatics is a rather new research field. Li et. al. defined a stage-based model of personal informatics systems[10]. Five stages of personal informatics, distinct properties of each stage and recommendations for each stage is defined. Furthermore it is defined that a stage can be user- or system-driven, meaning that the task in the stage is either done manually by the user or automatically by the system. Another interesting point in the paper is about uni-faceted vs. multi-faceted systems. They argue that most personal informatics systems are uni-faceted, but that multi-faceted systems offers a greater potential to users.

In another paper Li et. al. found that six different kinds of questions were asked in personal informatics and that two phases exist in reflection: Discovery and Maintenance. Each kind of question require a different answer and thereby a different kind of visualisation.

Multiple research projects uses personal informatics to directly drive behavioral change. UbiFit[11] and RecipeBox[12] works with health issues encouraging people to live healthy by being active and eating right. UbiGreen collects transit data to encourage users to live "greener". Both UbiFit and UbiGreen[13] utilises a wearable sensor for a semi-automatic approach to data-collection and a mobile phone to give users qualified feedback on their actions.

In a study, Lee and Dey worked with old adults, tracking when the participants took their pills and how they used their phones[14]. The research showed that participants were "more consistent and aware of their pill taking and phone use to safeguard their independence".

Based on personal informatics interfaces for energy consumption, He et. al. found that different users require different kind of feedback but that most personal informatics systems generalise this. A pschycology framework that addresses the different stages of

[10] Ian Li, Anind Dey, and Jodi Forlizzi. A stage-based model of personal informatics systems. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, page 557, 2010

[11] Sunny Consolvo, James a. Landay, and David W. McDonald. Designing for Behavior Change in Everyday Life. *Computer*, 42(6):86–89, June 2009

[12] Noreen Kamal, Sidney Fels, and Kendall Ho. Online social networks for personal informatics to promote positive health behavior. *Proceedings of second ACM SIGMM workshop on Social media - WSM '10*, page 47, 2010

[13] Jon Froehlich, Tawanna Dillahunt, Predrag Klasnja, Jennifer Mankoff, Sunny Consolvo, Beverly Harrison, and James A. Landay. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. *CHI Proceedings of the 27th*, 2009

[14] Matthew L Lee and Anind K Dey. Reflecting on Pills and Phone Use : Supporting Awareness of Functional Abilities for Older Adults. *CHI '11*, pages 2095–2104, 2011

behavioral change is presented.

Peesapati et. al. utilised e-mails[15] to encourage participants to reminisce about their past. In one trial, participants received e-mails containing old data from social networks or common life experiences. The interface furthermore allowed participants to respond to the memory-triggers by writing about their thoughts on a particular e-mail. Another paper described the use of a map interface to drive reminiscing[16]. Participants would manually enter memories tied to a specific location. The study showed that some paticipants left out bad or boring memories, resulting in a positive bias in the data. The research also suggest that there is real value in using places and maps as a basis for reminiscing.

Rivera-Pelayo et. al. created a framework for reflective learning[17]. Describing how the reflection process is affected by personal informatics systems. Furthermore it is argued that multi-faceted system could improve the reflection process.

In a study, Hsieh et. al. found that in trials using experience sampling[18], compliance increased by 23% for participants that had access to visualisations of the reported data.[19]

[15] ST Peesapati, Victoria Schwanda, and Johnathon Schultz. Pensieve: supporting everyday reminiscence. *Proceedings of the 28th*, pages 2027–2036, 2010b; and Dan Cosley, Victoria Schwanda Sosik, Johnathon Schultz, S. Tejaswi Peesapati, and Soyoung Lee. Experiences With Designing Tools for Everyday Reminiscing Experiences With Designing Tools for Everyday Reminiscing. (July):37–41, 2012

[16] S. Tejaswi Peesapati, Victoria Schwanda, Johnathon Schultz, and Dan Cosley. Triggering memories with online maps. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4, November 2010a

[17] Verónica Rivera-Pelayo, Valentin Zacharias, Lars Müller, and Simone Braun. Applying quantified self approaches to support reflective learning. *Conference on Learning*, pages 111–114, 2012

[18] Experience sampling is a method for collecting data about user behavior in research. Participants give feedback to the researchers at given intervals.

[19] Gary Hsieh, Ian Li, Anind Dey, Jodi Forlizzi, and Scott E. Hudson. Using visualizations to increase compliance in experience sampling. *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, page 164, 2008

Visualisations

MUCH RESEARCH has gone into the topic of visualisation. This section covers part of the research area, specifically general information visualisation, geovisualisation and visualisation of networks.

While many books cover the topic of information visualisation, as mentioned in the chapter "From caves to big data", Bertin and Tufte codified a lot of the knowledge about visualisations[20].

Many of the more recent books fall into two categories, practical books or "portfolio books". Both "Designing Data Visualizations" and "Visualizing Data" fall into the former category[21]. They take the reader through the process of creating a good visualisations. An example of the later is the book "Beautiful Visualization"[22]. It covers fourteen different visualisations, the thoughts behind and how they were made.

Common for both types of books, is that they communicate the same theory and techniques as Bertin and Tufte.

[20] Edward R. Tufte. *Envisioning Information*, volume 40. February 1991; Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, Conn., 1983; and Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps (translation from French 1967 edition)*. ESRI Press, 1st edition, 2011. ISBN 9781589482616

[21] Noah Iliinsky and Julie Steele. *Data Visualizations*. 2011. ISBN 9781449312282; and Ben Fry. *Visualizing Data*. O'Reilly Media, Inc., 2008. ISBN 9780596514556

[22] Julie Steele and Noah Iliinsky. *Beautiful Visualization*. 2010. ISBN 9781449379872

IN THE BEGINNING OF THE 1990S thematic maps evolved into geovisualisations as computing power made it possible to craft new, sometimes animated, visualisations.

Menno-Jan Kraak applied geovisualisation techniques to the historic map, of Napoleon march to Moscow, by Minard[23]. The spatiotemporal data from Minards map is visualised in a space-time cube instead of using a traditional map.

Kwan and Lee used 3d geovisualisation to show movement patterns of people in Portland, Oregon[24]. It is argued that "GIS-based 3D geovisualization ... are useful for the exploratory analysis

[23] Menno-Jan Kraak. Geovisualization illustrated. *ISPRS Journal of Photogrammetry and Remote Sensing*, 57(5-6): 390–399, April 2003

[24] Mei-Po Kwan and Jiyeong Lee. Geovisualization of Human Activity Patterns Using 3D GIS : A Time-Geographic Approach. *Spatially integrated social*, 2004

of activity-travel patterns.". However, the visualisations presented in the paper could have been made in 2d and it is unsure if the 3d effect adds any value.

Rae visualised the 2001 UK census data, showing migration patterns[25]. A technique of changing the visual display of data based on the distance of the migration is used. Another notable quality of the examples in the paper is a very high data-density for all visualisations.

[25] Alasdair Rae. From spatial interaction data to spatial interaction information? Geovisualisation and spatial structures of migration from the 2001 UK census. *Computers, Environment and Urban Systems*, 33(3): 161–178, May 2009

Visualising networks become a non-trivial task as the networks grow. While traditional node-link representation is sufficient for small networks, it become cluttered when you have dense networks or many nodes. Since clutter reduction is display dependant, two separate directions within the research-field exist: Taxonomies for clutter reduction[26] and novel ways of visualising the information.

Heer and boyd created an interactive system for visualising friendships in the online social network Friendster[27]. Key concepts include connectivity highlight, linkage view and inferred community structures.

The Orion visualisation system uses many different types of visualisations[28]. It enable analysts to do exploratory research of data using different visualisations: adjacency matrices, node-link diagrams and the centrality distribution of networks. Essentially letting the user find the best visualisation to answer their question.

GraphPrism use a regular node-link visualisation to show the network.[29] It is furthermore augmented by a set of histograms showing various properties such as connectivity, transitivity and density of the network. Their initial prototype showed promising results in a user test.

NodeTrix utilises Tuftes micro/macro principle by visualising large networks using both node-link graphs and adjacency matrices[30]. Global structure is shown using a node-link network. Adjacency matrices are used to display local communities within the graph.

[26] Wei Peng, M.O. Ward, and E.a. Rundensteiner. Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering. *IEEE Symposium on Information Visualization*, pages 89–96, 2004; and Geoffrey Ellis and Alan Dix. A taxonomy of clutter reduction for information visualisation. *IEEE transactions on visualization and computer graphics*, 13 (6):1216–23, 2007

[27] Jeffrey Heer and Danah Boyd. Vizster: Visualizing online social networks. *Visualization, 2005. INFOVIS 2005. IEEE*, 2005

[28] Jeffrey Heer and Adam Perer. Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 51–60, October 2011

[29] Sanjay Kairam, D MacLean, and Manolis Savva. GraphPrism: compact visualization of network structure. *Proceedings of the*, 2012

[30] Nathalie Henry, Jean-Daniel Fekete, and Michael J McGuffin. NodeTrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*, 13(6):1302–9, 2007

Spatiotemporal networks

Spatiotemporal networks can be found in many places. Both the location data and the social data in this thesis can be considered as spatiotemporal networks. The work described here consists of work on spatiotemporal networks that are interesting in a personal informatics context.

In two studies, Madan et. al. worked with data from sensors in mobile phones, to predict user behavior[31]. In both studies they used proximity of other users as a measure for whom they had social ties to. This was used to predict diseases and political opinions.

Using Bluetooth proximity, call hustory and cell-tower data from mobile phones, Eagle and Pentland was able to infer social network

[31] Anmol Madan, Manuel Cebrian, David Lazer, and Alex Pentland. Social sensing for epidemiological behavior change. *Proceedings of the 12th ACM international conference on Ubiquitous computing - Ubicomp '10*, page 291, 2010; and Anmol Madan, Katayoun Farrahi, Daniel Gatica-perez, and Alex Sandy Pentland. Pervasive Sensing to Model Political Opinions in Face-to-Face Networks. *Lecture Notes in Computer Science*, 6696/2011:214–231, 2011

structures and recognise patterns in the daily activity of users[32]. Furthermore, based on half a day of data, they were able to predict user behavior for the rest of the day with 79% accuracy.[33]

[32] Nathan Eagle, Alex Sandy Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of ScienceEagle, N., Pentland, A. S., & Lazer, D. (2009). Inferring friendship network structure by using mobile phone data. Proceedings of the National Academy of Sciences of the United States of America, 106(36), 1527*, 106(36):15274–8, September 2009

[33] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, April 2009

Analysis

The objective of this project is to anayse, design and implement a personal informatics system. Li et. al. defines such systems as "those that help people collect personally relevant information for the purpose of self-reflection and gaining self-knowledge"[34].

In this chapter, user needs are defined and developed into use cases. The privacy implications and the data available is furthermore analysed.

[34] Ian Li, Anind Dey, and Jodi Forlizzi. A stage-based model of personal informatics systems. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, page 557, 2010

Scenarios, needs and use cases

In a user scenario, we look at a hypothetical DTU student. Camilla Jensen studies to become a bachelor in mathematics and technology. Camilla is curious by nature and volunteered in the Sensible DTU research program mostly to get a much needed upgrade of her aging phone. When she entered the project she was told the catch about the phone - it would track her whereabouts at DTU and who she was nearby.

Camilla had always wanted to know where she spend most of her time at DTU. She had a rough idea but was not sure. Also, how many of the lectures did she actually attend? An interface to the collected data would help her answer those questions. Now at her third semester at DTU, she was also wondering if the students she talked to at introduction-week was still talking to eachother but she could not really see it from Facebook. Often studying late at night at the campus, she would also like to be able to see if some of her friends are at campus as well.

Christian Hansen, is also enrolled in the research program. Being a bit introvert, Christian is curious about how many or few people he is actually talking to throughout the day. Suspecting it is low he would like it to increase. Christian studies for a masters in computer science and engineering with a specialisation in computer security. After political initiatives like ACTA and SOPA, Christian has become very aware of his own privacy. It is important to him that privacy is taken seriously.

Based on the above scenarios we define the following general user needs.

- Personal visualisation of how the user moved around historically.
- Personal visualisation of whom the user was nearby historically.
- Statistics with short facts about each day.
- Playback of the personal visualisations.
- Visualisation of where friends are located.
- A strong privacy model.

Based on the user needs above, we are able to create a use case diagram and use cases. A few select use cases are presented here, the rest is available in appendix A on page 69.

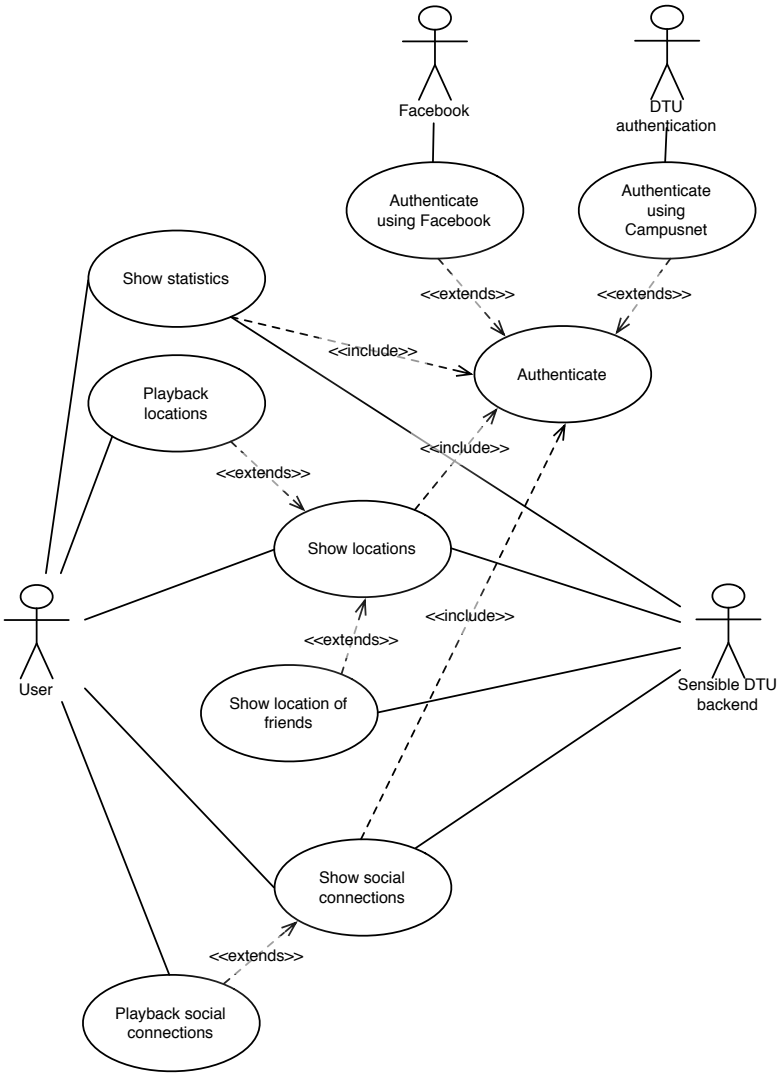

Figure 5: A use case diagram for the visualisation application.

Use case	#1: Authentication
Description	The user is able to log into the application.
Actors	User, Facebook, DTU Authentication.
Main scenario	1. User starts up application. 2. Login dialog opens. 3. User is presented with posibility to log in using CampusNet or Facebook. 4. User authenticates.
Extensions	4a. User authenticates using CampusNet. 4b. User authenticates using Facebook.

Table 1: Use case #2: Authentication

Use case	#2: Show locations
Description	Show the historic locations of the user.
Actors	User, Sensible DTU backend.
Main scenario	Include use case #1 "authentication". 1. User clicks on "Location". 2. The last locations of the user is shown.
Extensions	3a. User playbacks the last locations on the screen (the changes over time is shown on the screen).

Table 2: Use case #3: Show locations

Privacy concerns

THE DATA COLLECTED in the Sensible DTU project can be considered as sensitive data. Due to this fact, we must take into account privacy concerns. Both for the Sensible DTU project as a whole and for the visualisations presented here.

THE DISCUSSION ON PRIVACY STRECHES as far back as 1890, when the Harvard Law Review published an article on the subject by Samuel Warren and Louis Brandeis[35]. Back then, the topic of the discussion was the many new newspapers and photographs. Now the discussion revolves around surveillance.

In 2006 the European Union adopted the *data retention directive* that require membership countries to store telecommunication data for at least 6 months and up to 2 years. The German Green party politician Malte Spitz tried numerous times to gain access to the data his phone company logged about him. When he finally did gain access, he decided to let people see the consequences of the directive by visualising the data logged about him[36]. Most people who see the visualisation are surprised by the detail. It depicts Malte Spitz moving around in Germany, how he took the train from Berlin to Munich and so forth. Furthermore it show how much he used his phone for calls, text messages and internet access. The German Constitutional Court later ruled the EU directive as unconstitutional in 2010.

Most of these discussions concern data logged by third parties. But with the increased usage of online social networks, more people have become aware of the various privacy implications with using such services. The difference between social network data and the data collected by the EU telecommunication companies becomes blurry when social network data about a person is not only the data uploaded by herself, but also contains data uploaded by friends and familly.

Of course, the privacy issue is no issue at all if the data uploaded to the social network is only available to the particular user it is about, however that would work against the whole idea of the social networks. Some networks are even allowing users to track their whereabouts by using check-ins[37]. As of now the majority of those check-ins happen manually by the user, however when they begin to happen automatically the privacy implications become bigger. As mentioned in related work, research have shown it is possible to accurately infer social ties by using location data. This mean it would be possible to re-construct social networks if a person gained access to all the check-in data.

In relation to Sensible DTU, it is very important to take the privacy of the users seriously. The system does not only contain the specific location of the user at all times, but also information about whom they were nearby and who they talked on the phone with.

[35] Samuel D. Warren and Louis D. Brandeis. The Right to Privacy. *Harvard law review*, 4(5):193–220, 1890

[36] The visualisation was made in cooperation with die Zeit and is available at `www.zeit.de/datenschutz/malte-spitz-data-retention`.

[37] Foursquare (`foursquare.com`) and Facebook (`facebook.com`)

Two of the goals for Sensible DTU is to allow participants to access all their information stored, but also to allow researchers to access the full or parts of the dataset anonymised.

The last goal requires consideration due to the fact that even if the data is stored in an anonymised way[38], it is possible to de-anonymise according to the latest research[39]. One method of solving this issue is to not release the actual data, but instead to release data with the same properties, *e.g.* a graph with the same amount of edges etc. The first goal implies that data cannot be stored in an anonymised way, at least not without also making it possible to de-anonymise it on a per-user basis.

[38] For example by changing personal information like name to a unique number.
[39] Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. 2009

Another potential privacy problem with the first goal is sharing of data. While it is very interesting for a user to have access to information about their friends, it does raise certain concerns. A key principle in that regard is to always make data-sharing opt-in for the users and to give users fine-grained control over how much data is available to each one of their friends.

Available data

With the use cases as the starting point, we now look at the available data.

The dataset we are considering are being collected on smartphones. Some of the data is able to tell us the physical location of the users at various granularities:

- WIFI access points nearby.
- Cell towers nearby.
- Physical location of phone (based on a combination of GPS, WIFI and cell towers).

Other tell us how the users communicate:

- Proximity of other devices (using Bluetooth).
- Contact list on the device.
- Call history.
- SMS history.

Furthermore we have access to data from access points from the campus-wide WIFI network. The access point data tell us which access points a specific user has been connected to and thereby their location.

All of the above data is being sampled and stored at predefined intervals, essentially giving us all the changes in the above data over time.

Before we are able to create a useful interface for users to explore data, we must first ourselves try to comprehend the data. This

enable us to create a better experience for the users, since we already have an idea of the visual properties of the data. This could be achieved by calculating statistical properties for the data, such as distribution. Another method is to show the properties by visualising the data.

Measurements show that if a user is located at a single location in an extented period of time a variation in the location exist. This is expected as the technology used for location (GPS, WIFI, Cell towers) have inaccuracies.

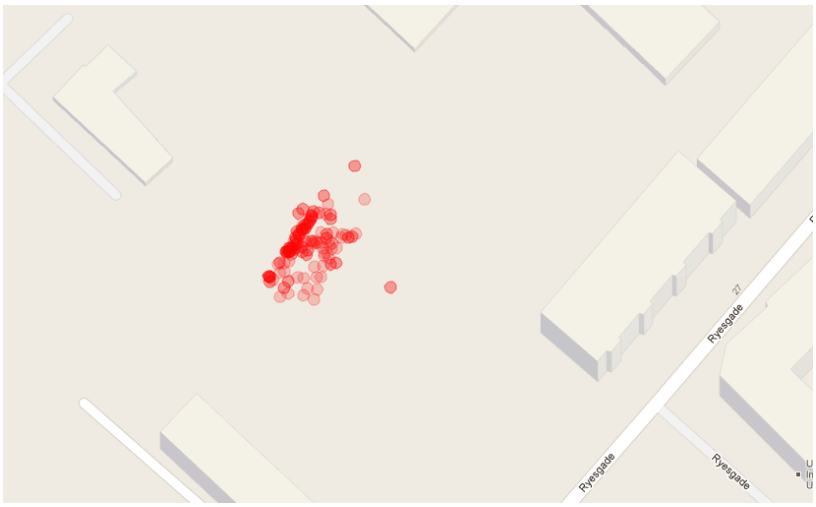

Figure 6: Variation in data on a single location (specifically the phone lying on a table in my apartment). The data shown is from a single day (June 11th, 2012).

When users move around, the actual route between two locations is not recorded, as the sample rate is too low. Instead we see beeline movements between each sample.

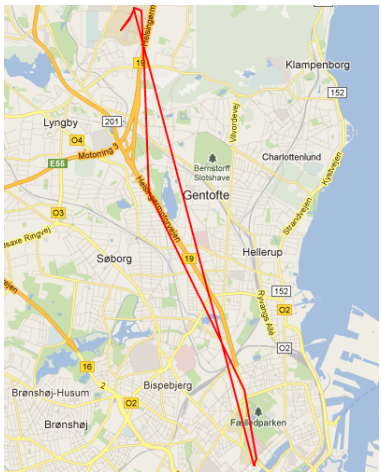

Figure 7: Reported movements on a single day (June 11th, 2012).

THE LOCATION DATA PROVIDED directly contains information about where the user is located. So does the data about WIFI access point, albeit more indirectly. While the location data tell us a quiet precise location of the user, the WIFI data have by nature a lower granularity. One could argue the WIFI data is of little use since we already have location data at a higher granularity. But from the WIFI data we are able to infer other information very easily. For example, if you want to know if a user is at DTU at a certain time, one could setup location boundaries based on longitude and latitude and check if the user is within the boundary. But if the boundary is not square, as in the case of DTU, we would run into trouble and the seemingly trivial problem would suddenly become non-trivial. Instead one could look at the WIFI data and from that infer whether or not a user is at DTU. This is possible since three distinct access points, DTU, eduroam and eksamen, are broadcastet all over the campus area.

Cell tower data can be used much like WIFI data at an even lower granularity.

MUCH OF THE SENSIBLE DTU PROJECT revolves around how people communicate when they are "offline" so to speak. A very im-

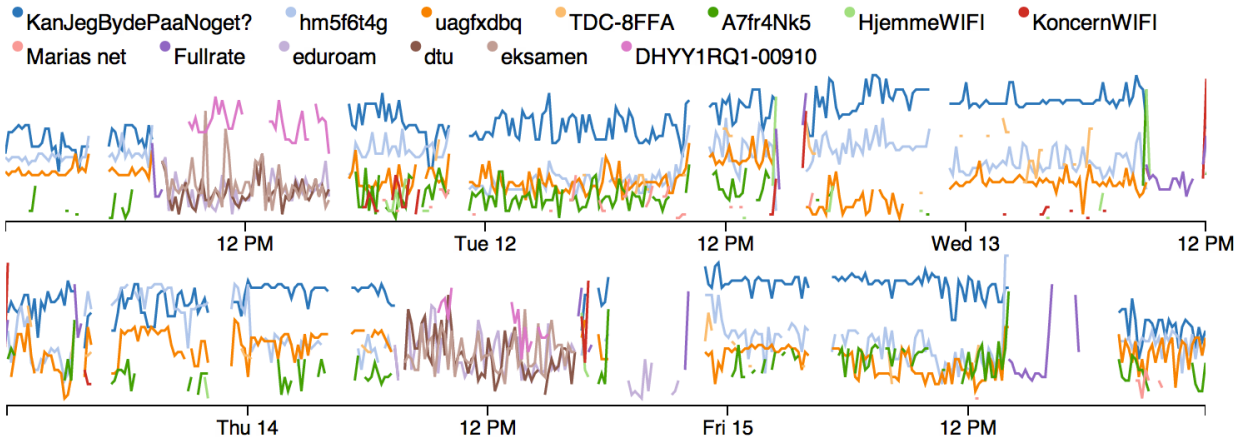

Figure 8: The access points a user encountered from Tuesday 12th of June 2012 through Friday the 15th of June 2012. Notice the location patterns in the data.

portant factor of this, is who they are around at a given time. The data from the Bluetooth hardware is able to tell us about the proximity of other phones. Since we know the identity of the person who carries each phone, we are able to infer who is physically close to each other at a given time. Figure 9 depicts whom, a specific user spent time with during some days in may and june.

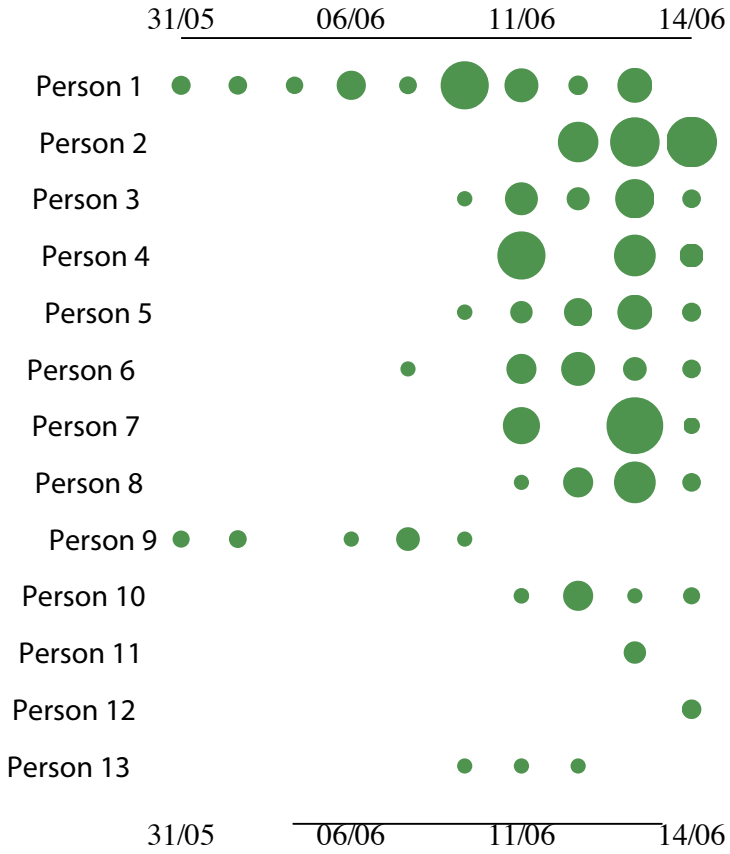

Figure 9: The Bluetooth activity for a user. It is clearly seen who the user spent the most time with those days.

An interessting question answered with the social data could be "How much time did person A and B spent together yesterday.". Each Bluetooth scanning contain all the Bluetooth devices the phone is nearby at the given time. Since the data does not directly

answer the question if two persons where together at a specific time, we must infer that information from the raw data. The data is sampled on each phone at intervals, but even though the interval should be constant, the actual interval could be more or less frequent depending on other scannings etc.

Furthermore even though two phones are only 5 feet away from each other during multiple scans, it is possible that one of the phones detects the other, but not the opposite. This can be caused by Bluetooth antenna variations or varations in surroundings, *e.g.* the building structure.

Pre-processing of the Bluetooth data must be done to infer social relations based on the raw data.

BASED ON THE USE CASES, the system must show the location and social connections of the user. As for the location, four possible data sources exist. The WIFI access points detected from the phone is limited to areas where WIFI coverage is good and well-known. While the location of access points are well-known at the DTU campus, the coverage is only good inside buildings. Cell towers cannot be used to cover small areas like the DTU campus, since only a few cell towers cover the whole area. The data collected from the access points lack the same capabilities as when the acess points are detected from the phones. The last possible data source uses both GPS, WIFI and Cell towers to detect the location of the phone resulting in a general higher accuracy. Due to the higher accuracy of the location, the last provider is chosen.

Multiple data sources exist containing social data, however only one is able to tell us about nearby devices and thereby people. The Bluetooth data is chosen as the base from which social connections are infered.

Data pre-processing

THE DATA ANALYSIS ABOVE, show that pre-processing must be done to increase the quality of the data.

Location data

THE LOCATION DATA should be pre-processed by clustering multiple datapoints at the same location into a single point.

Several clustering algorithms exist. The simplest and most common methods are hierarchical clustering, partitioning-based clustering and density-based clustering.[40]

Hierarchical clustering works by connecting points nearby to form a cluster. Unfortunately the most general algorithm is rather naive and thereby slow. Even though hierarchical clustering has some very nice clustering properties, with a general runtime of

[40] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Morgan Kaufmann Publishers, San francisco, 3rd edition, 2001. ISBN 9780123814791

$\mathcal{O}(n^3)$ it is very slow, considering large datasets.

The partitioning-based clustering algorithm, `K-Means`, works by choosing a number of clusters on beforehand. A centroid is defined for each cluster. The centroids are then used to define which cluster a given point is in.

Density-based clustering uses the density of an area to determine if a cluster exist or not. This implies that a density drop at the border of a cluster must exist. This is however not always the case, *e. g.* when two clusters are located close to each other. The usage of density also means that it works very well for detecting outliers.

Due to the slow runtime of the hierarchical clustering methods, only centroid- and density-based clustering was implemented and evaluated. The chosen algorithms was `K-Means` (centroid-based) and `DBSCAN` (density-based).

An important property of the `DBSCAN` algorithm is the distance function it uses. It is used to calculate the distance between two points. Together with the epsilon variable it is an important factor in determining if a point is part of a cluster or not. In a cartesian coordinate system, the distance between two points could be calculated as the Euclidean or Manhattan distance. On a map, where the curvature of earth must be taken into account, the distance can be meassured using the haversine distance or the spherical law of cosines.

Due to time-aspect or our data, we cannot simply cluster the data using a regular distance function. If a person is located at location l_1 at time t_1 to t_{10}, then moves to location l_2 from t_{11} to t_{12} and then back to l_1 from t_{13} to t_{20} we want to cluster the individual data points as 3 clusters. We do this by using the time in the distance function. The epsilon variable of the algorithm determines how far away a point can be from the other points in a cluster and still be included in the cluster (*i.e.* it determines how dense the cluster must be). To add time as a factor we simply add a cut-off time, *e.g.* 30 minutes, and if 2 points differ more than 30 minutes we define a distance of infinity between them. That ensures we will be able to take into account the time difference as well as the actual difference in distance.

Social data

As described earlier, we must infer social relations based on the raw Bluetooth data. This can be done using various methods. The most naive being that two persons are infered as being close to eachother from the begining that one phone detects the other and until none of the phones detect the other within a certain timeframe.

One of the problems with the naive approach, is that the detection does not have to reciprocal. But since it is very important that the detection is mutual in the visualisation, meassures would have

to be taken to ensure this.

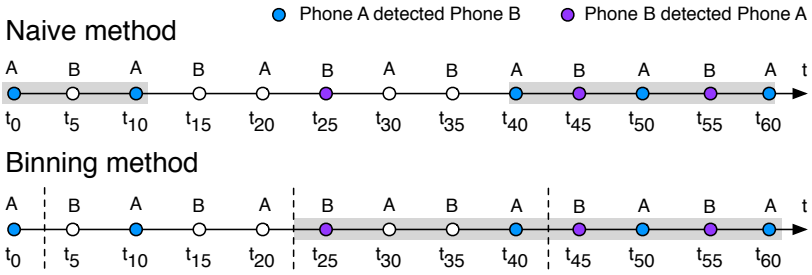

Figure 10: Two methods of infering social relations from Bluetooth data. The naive algorithm is shown to have a time-limit of 10 minutes. The grey bars show when each algorithm would infer a relationship.

Another method is to bin the data by an interval (*e.g.* an hour) and define that if both phones detect the other within that timeframe, they have been near each other during that timeframe. Since the naive method does not take into account that both phones must have detected each other, the binning method is chosen for the design and implementation.

Summary

IN THIS CHAPTER we found a set of use cases that will be used for the design and implementation of the personal informatics system. Furthermore we analysed potential privacy concerns in regards to the system. We chose the data feeds used in the system and analysed them further. Last we found that the data chosen must be preprocessed by clustering and binning.

Design

This chapter describes the general technical architecture of the sytem and the design of the graphical user interface.

In the project roll-out in 2012, more than 100 users will gain access to the personal informatics system devised in this thesis. Developing desktop applications that run on the computers of a hundred people is non-trivial. On the other hand, the internet and the web browser is ubiquitous. All use cases can be implemented using web-technologies. Therefore a browser-based implementation is chosen.

In parallel with the development of Sensible You, the Sensible DTU backend was still being designed and built by another group of researchers and students. As of this writing, it consist of a simple RESTful API where clients are able to retrieve data about users. No authentication or authorisation had yet been implemented.

Due to the sensitive information stored, it was decided that users had to log in to see their visualisations. To achieve this, a provisional backend was designed and built. The backend was designed to be sliced in between the Sensible You frontend and the Sensible DTU backend. An additional gain, is that it is possible to preprocess data on the backend if necessary, thus reducing bandwidth.

Based on the use cases in the analysis, the following requirements of functionality can be devised.

- Authentication.
- Visualisation of location.
- Visualisation of social relations.
- Statistics about the user.

It must be possible to playback the location and social relations visualisations. Furthermore the location visualisation should be able to show the location of friends.

From the required functionality above, five distinct UI components can be devised.

- Authentication panel.
- Location visualisation.

- Social relations visualisation.
- Timeline with playback.
- Statistics panel.

A mockup of the UI components, put together in a general user interface, is show in figure 11.

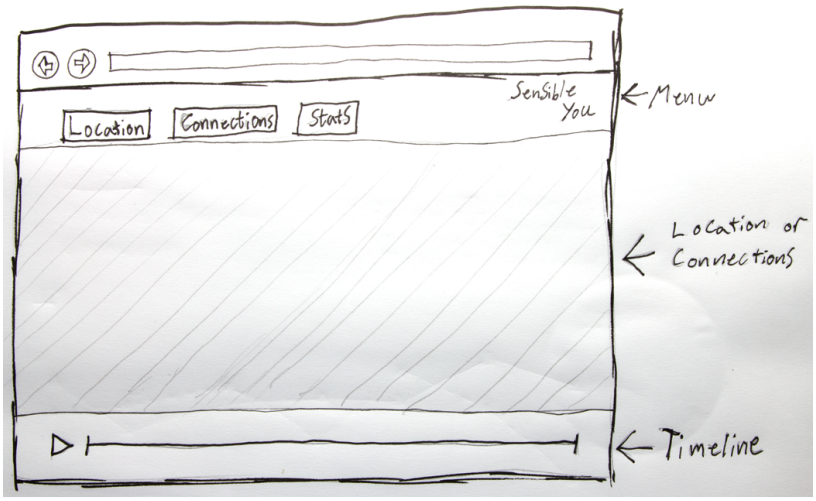

Figure 11: Mockup of general user interface.

Location

THE OBJECTIVE OF THE LOCATION element is to show location changes of users over time. Based on that criteria, mockups of design options was created.

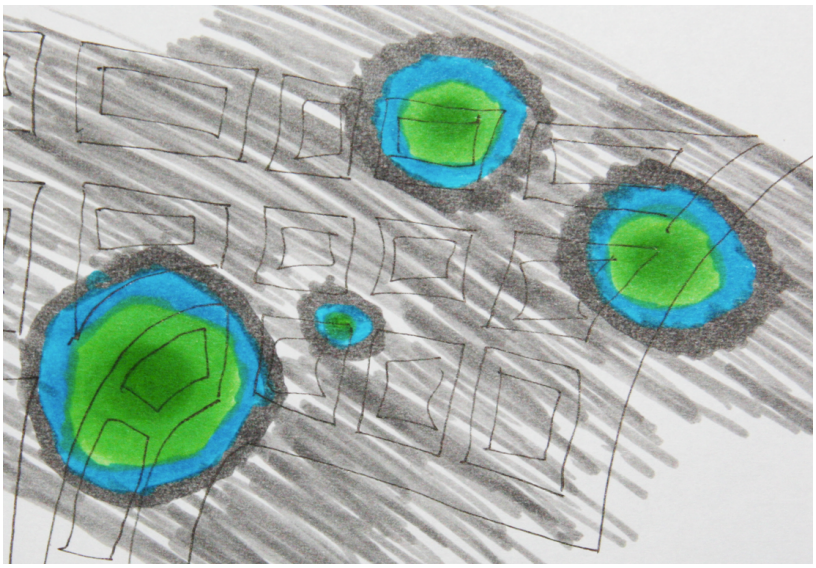

Figure 12: Mockup of a location element. The heatmap depicts where the user spent the most time.

One way to do this is by using a map. It is very well-known to most users from map services like Google Maps. A heatmap-style map, show a very nice overview of the places the user has spent

the most time. Unfortunately it does not depict the relationship between locations or how the user moved between them. To display the relationship between the locations, lines could be added between the points, letting the line to the latest location be the most opaque and the other lines be more transparent. The heatmap makes it harder to see other artifacts on the map and can make the data look "muddy". Instead, circles could be used to depict locations, varying in size based on the duration of the users stay at that location.

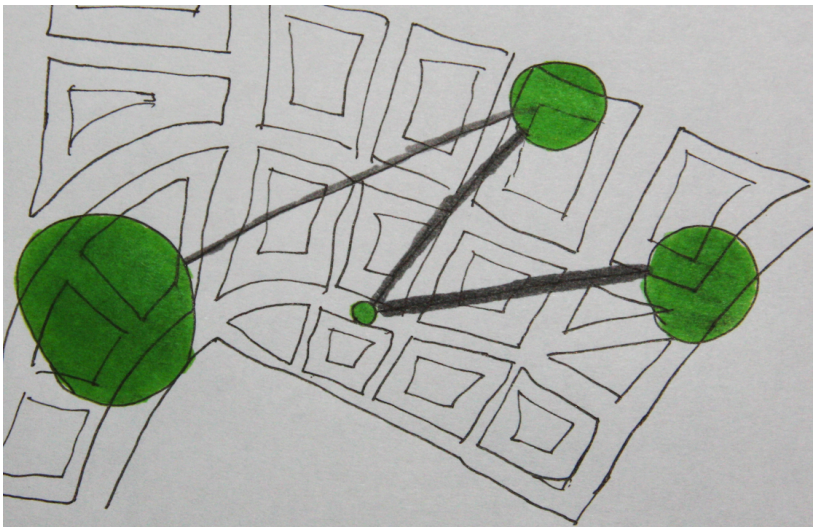

Figure 13: Mockup of a location element. The connected circles show how a users location changed through time.

Both of these designs lack the ability to quickly pinpoint where a location is, unless you know the map and the sourroundings very well. An alternative design could be, to not show the data ontop a map, but instead show it in a network structure. Growing circles and trailing lines could still be used to visualise duration at locations and time. Instead of pinpointing the circles to a specific location ontop a map, the name of the location would be shown in the circle.

The graph enable the user to quickly see movement-patterns. Unfortunately the visualisation would no longer have the same spaciousness - New York would look to be as far a way as the super market down the street. Furthermore the design would require that we know the name of every reported location, which is not the case.

Based on the different properties of maps described above, a map where the locations are shown as cirlces are chosen. With that design as the base for our implementation, we are able to analyse the data in that context. Bertin introduced the notion of data components as a way of looking at the data in a visualisation context. The components of the location data, that should be visualised are:

- $\neq$ The user.
- **GEO** The location of the user.
- **O** Time of the stay at the location.
- **Q** Duration of stay at location.

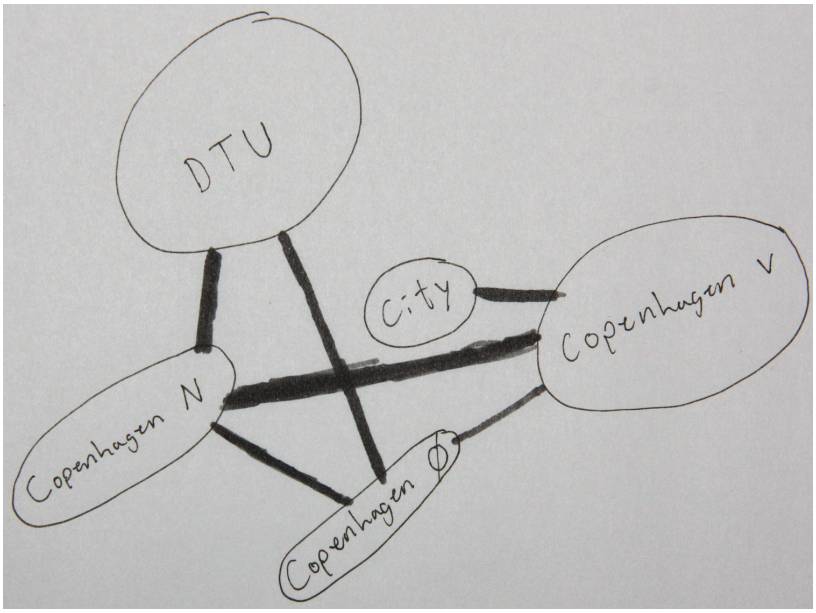

Figure 14: Mockup of a location element. The networked circles let's the user see connections between named locations.

Since the data consist of more than 2 components, a simple map cannot be used. Instead either an *inventory map*, a *processing map* or a *cartographic message/synthetic schema* must be used.

Given the fact that 4 components must be showed as well as the nature of the components, an inventory map is the best choice. A graphic construction of the map can be defined as:

	Information	Graphic construction.
$\neq$	The user.	Variation of colours.
GEO	The location of the user.	Variation in position.
O	Time of the stay at the location.	Line opacity/saturation.
Q	Duration of stay at location.	Circle size.

An important fact to keep in mind, the design is to be implemented, is the definition of the circles size. According to Bertin and Tufte, it must be directly proportional to the actual component it depicts. Therefore, it is important that it is the circle area, not the radius, that differ proportionally to the component.

Social connections

THE SOCIAL ELEMENT should show how social connections changes over time. A natural way to show this is using a graph. A mockup of such is shown in figure 15. Notice how the line width differs in the graph. This is to show that some nodes, or people, have a closer relation than others.

The mockup does not depict another information we have about each user, their occupation. However, including this in the design adds information to the graph, thus making it easier to distinguish

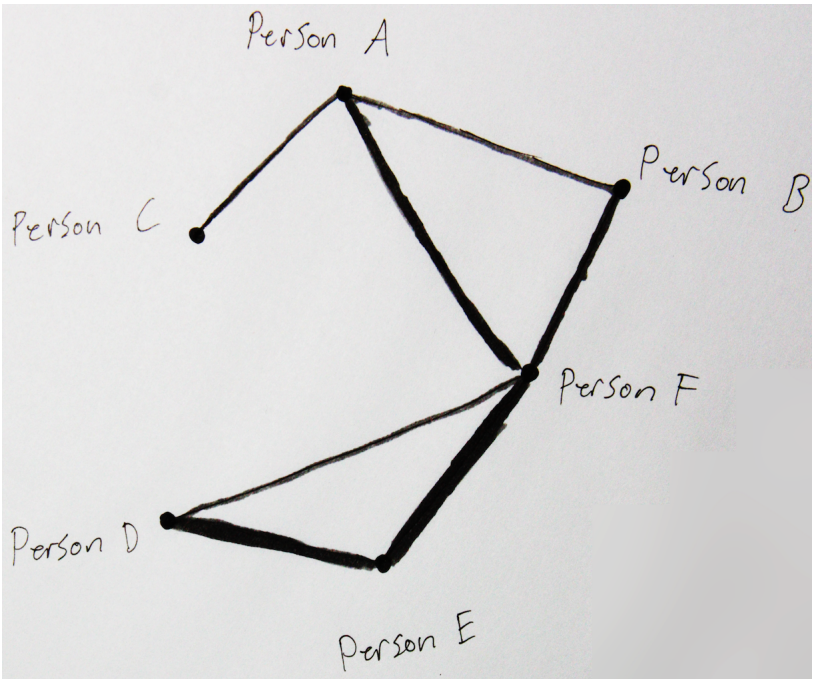

Figure 15: Mockup of a social element. The graph enables the user to see how strong a relation is between two users.

on user from the other. In summary we have the following components in the graph.

- $\neq$ The user.
- **Q** How strong the social relation is between users.
- **O** Occupation of the user.

Even though, only a mockup of a node-link graph was made, additional representations was implemented. All implemented representations are described in the chapter "Implementation".

Statistics

THE PURPOSE OF THE STATISTICS element is to give a short quantifiable overview of the data.

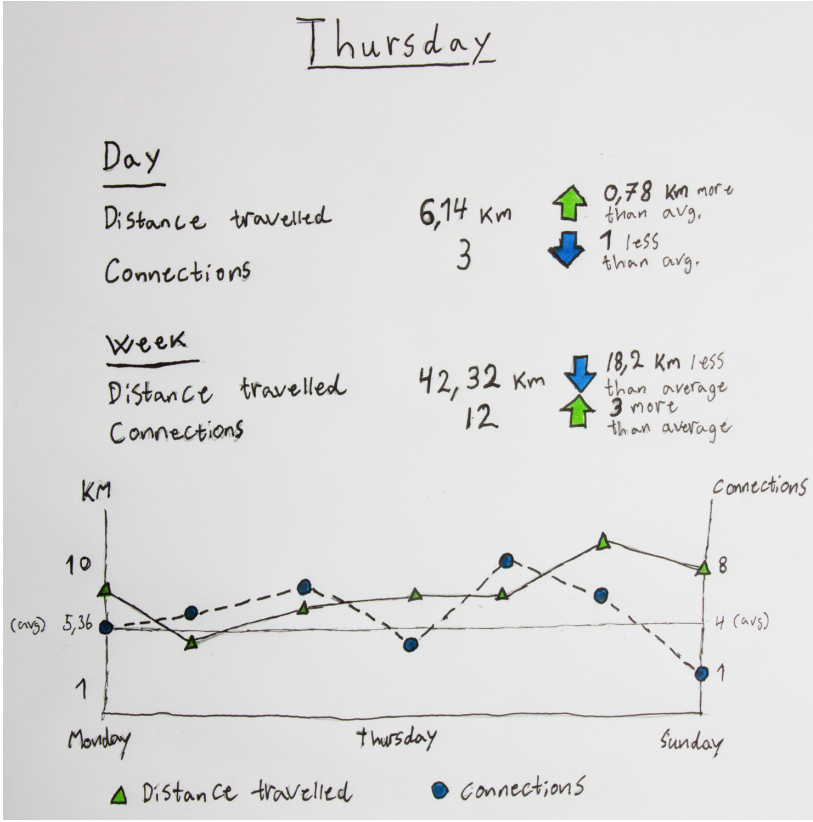

Figure 16: Mockup of a quantifiable overview of the data. The graph and comments for each item enable the user to look at her data at a glance.

Average values for each day, makes it possible to encourage the user to work harder, by showing them how far away they are from an average day. To encourage them even further the average day could be exchanged with a user-defined goal.

Timeline

THE PURPOSE OF THE TIMELINE is to let the user change the timeframe of the location and social connections visualisations. Even though a structure like the map naturally lets the user zoom in and pan out on the geographical data, the timeline should allow the user to do it on the time-dimension of the data.

A traditional timeline, seen in video-players etc. consist of a play button and the actual timeline that displays how much time has elapsed. This kind of timeline is very familiar to most users. Unfortunately, the timeframe of such timeline must be predefined, *e.g.* to a single day. Furthermore it doesn't allow the user to reframe the content (*e.g.* only showing content from 12.00 to 14.00).

To solve the last issue, two sliders could be incorporate into the

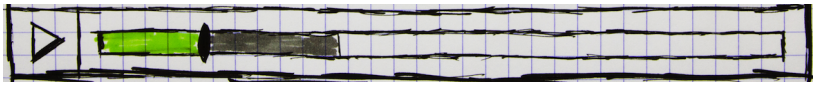

Figure 17: A traditional timeline enables the user to playback content from a predefined timeframe.

timeline. This would enable the user to reframe the content and view only the most interesting part of the data. Even if we added playback to this kind of timeline, the full timespan of the timeline would still have to be predefined. This would make it difficult to move the sliders precisely if the timeline contains a year worth of data.

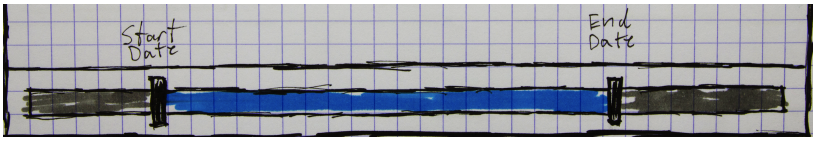

Figure 18: The scaling timeline let's the user reframe content based on their needs.

One way to solve the issue of precisely moving the sliders, would be to create a zooming timeline. The timeline could highlight interesting days in the timeline. This kind of timeline would work very well, if only a single point would have to be selected. But if the user must be able to reframe the content, the interface would become cluttered, as it would have to zoom in on two days.

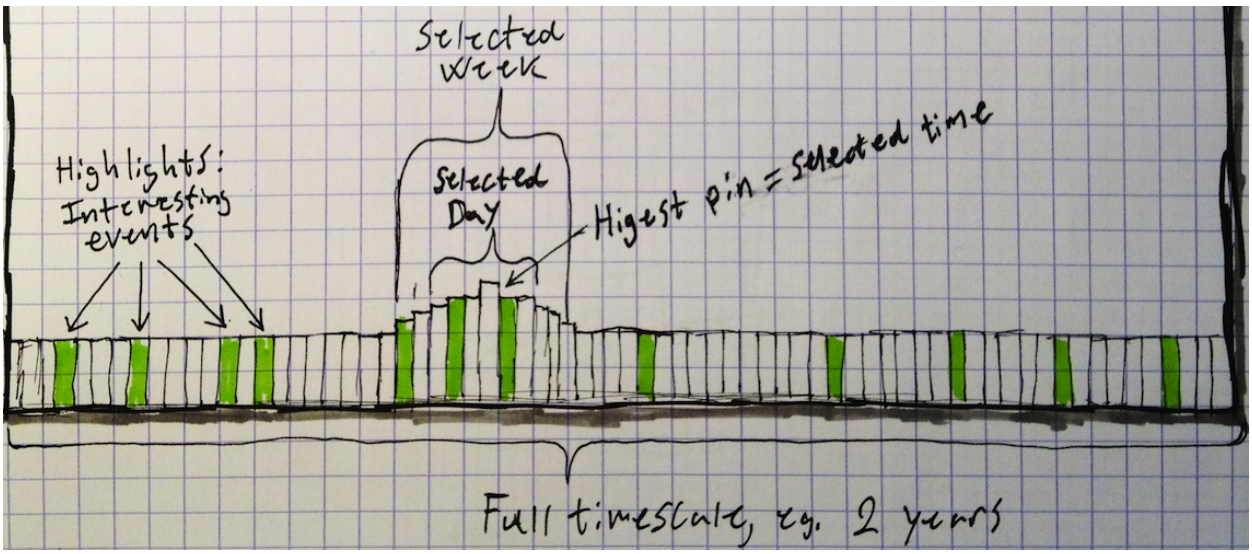

Figure 19: The zooming timeline lets the user pick a point very precisely by zooming in on a date when the mouse hovers the timeline.

Instead of showing all days directly on the timeline, we could move the selection of a day into a separate day-chooser, and let the timeline span a single day. This way we would be able to combine the playback functionality of the traditional timeline with the framing abilities of the scaling timeline. The downside of this timeline is it's inability to compare data from different days.

Based on the ability to reframe the content in the timeline and change date, the timeline based on a calendar is the good compromise. Therefore it is chosen as the starting point for the first iteration of the implementation.

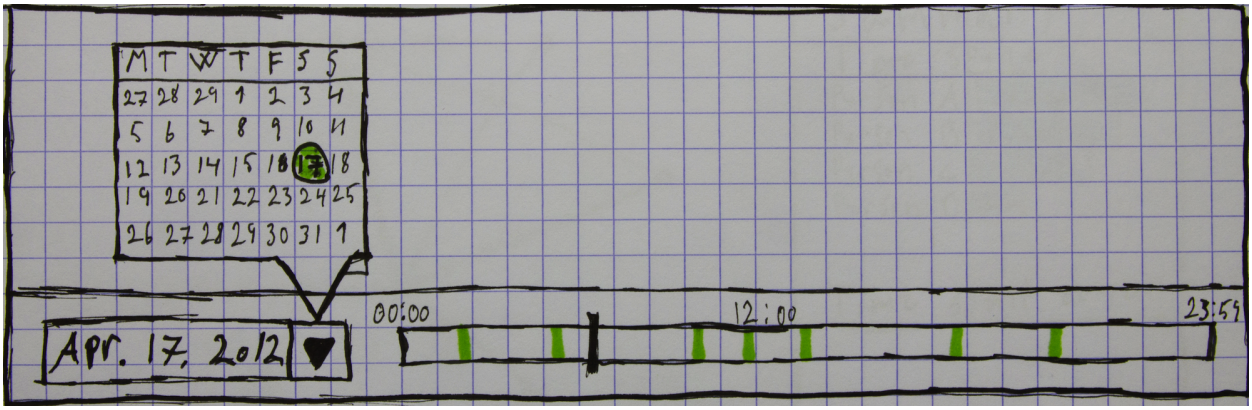

Figure 20: The calendar let's the user select a specific day to "zoom" in on.

Implementation

The rough sketches and overall design from the previous chapter only provided a guideline for the development of the system. At first the system resembled the mockups made, but after numerous improvements the system looks quiet different from the first sketches. This is the result of a long development process with constant feedback from people related to the project.

This chapter describes the significant changes in the iterations of the system.

As described in the previous chapter, the system is being developed as a web-application. This reduces the choices of programming languages/platforms. Posibilities include Java (Applet), Adobe Flash, Microsoft Silverlight and JavaScript. Due to no prior experience developing on the three former platforms, JavaScript was chosen. Furthermore a lot of development has gone into JavaScript in recent years, advancing the capabilities of the language in browsers.

With the emerging support of SVG and faster computers, many client-side JavaScript visualisation libraries have appeared. For example Processing.js[41], protovis[42] and data-driven documents (D^3)[43] to name a few.

In most of the frameworks, standard visualisation types like bar- or linegraphs are easily created, while it is more difficult to create more advanced visualisations. The philosphy behind D^3 is to make standard markup (*e.g.* HTML, SVG etc.) data driven. To do this, in D^3, you describe how the markup should look like given some data, then the data is applied and the markup generated. This makes it very easy to use CSS for styling and advanced animations. Furthermore D^3 contains many handy helper functions, making development a lot easier compared to vanilla JavaScript.

D^3 is chosen as the visualisation framework based on the philosophy of using standard markup. It could however just as well have been Processing.js or Protovis.

[41] Processing.js. URL `http://processingjs.org/`

[42] Michael Bostock and Jeffrey Heer. Protovis: a graphical toolkit for visualization. *IEEE transactions on visualization and computer graphics*, 15(6): 1121–1128, 2009; and Jeffrey Heer and Michael Bostock. Declarative language design for interactive visualization. *IEEE transactions on visualization and computer graphics*, 16(6):1149–56, 2010

[43] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-Driven Documents. *IEEE transactions on visualization and computer graphics*, 17 (12):2301–9, December 2011

Overview

THE GENERAL ARCHITECTURE of the system is a common client-server architecture. The collector application on the phones send data to the Sensible DTU backend. The visualisation web application retrieves location and Bluetooth data from the backend. In this case, an extra backend was pushed in between the web application and the Sensible DTU backend, having all data passing through the extra backend.

This introduces an extra delay each time the user fetches data. To limit this delay the extra backend caches the data from the Sensible DTU backend for a day, thus limiting the delay to the first request every day.

Authentication

AT FIRST EACH PARTICIPANT was given a specific username and password to access their data. Subsequently a central single-signon service at the university was used. The authentication service implemented the JASIG CAS protocol[44].

The authentication flow of the client is very simple. The backend redirects the user to the authentication service with a callback url as a parameter (A). After successful login (B), the user is redirect to the callback url together with a ticket parameter (C). The backend is now able to verify the validity of the ticket at a webservice endpoint of the authentication service (D). In the verification process, properties such as username and realname is also retrieved (E).

A LOT OF IRRELEVANT BLUETOOTH DATA is collected for each user. The irrelevant data mostly consist of scans where no friends are in Bluetooth proximity. Since sending all data to the client would require a lot of bandwidth, the data is processed at the backend before sending it to the client. The processing consist of collecting Bluetooth data from all friends[45] and removing all scans which does not contain data about friends.

Frontend

THE FRONTEND CONSIST of a single page where all navigation happen within. A controller takes care of all views and the navigation in the application. Each view controls an area on the screen. Data are loaded from the backend, pre-processed and inserted into two data stores, one for location data and one for Bluetooth data.

[44] http://www.jasig.org/cas/protocol/

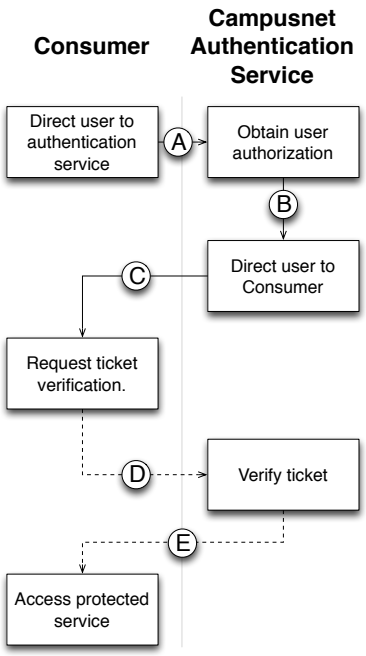

Figure 21: Flow of the university-wide user authentication service (Campusnet Authentication Service).

[45] In this initial trial, a friend is anybody who have a phone collecting data.

The map

THE SYSTEM UTILISES the capabilities of CloudMade[46] to fully customise the look and feel of the map. The amount of distinct colors on the map was kept at a minimum. The focus should be on the data on top of the map.

In the analysis and design chapters, the basis of the map is to show location data from friends as well as the users own data. Due to privacy concerns, only the users own data is shown at the moment. If data from multiple users had to be shown, the colors of the markers should be as distinct as possible. This would allow the user to easily see the differences between users on the map. But since only, the users own data is displayed a single high-contrast color can be chosen to help it stand out from the map. The color chosen is a deep red.

[46] http://cloudmade.com

DURING THE DEVELOPMENT of the map a need for more information about a single geolocation point became apparent.

To solve this issue a dialog is showed when the user clicks at a specific point on the map. In the dialog information about when and how long the user was at that location is showed. Furthermore a visualisation of the users connections at that point is showed.

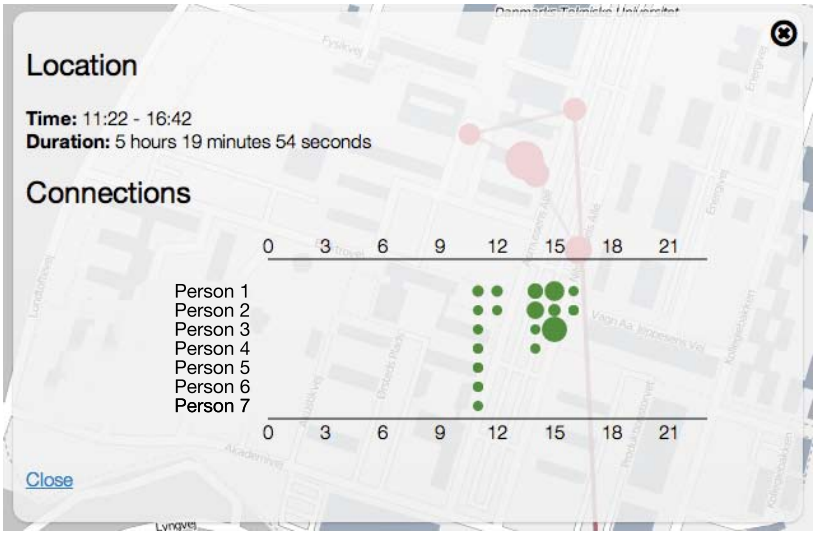

Figure 22: Popup dialog shown on specific location-points. Social connections are shown for each hour of the day.

The visualisation show how much time the user spent with each person - the size of the dot - in each hour.

ANOTHER DISCOVERY during development was the need for an additional map-overlay. The default map overlay show landmarks and roads in a very unobtrusive way, thus letting the user focus on their data. But sometimes, users would like to see exactly where they were located at a certain point in time. In cities a roadmap is able to do that to some extent. But if a user is mountainbiking in a forrest, the roadmap of the area would only consist of a large square. A

satellite map on the other hand give a much more detailed view on the sourroundings of the location.

Both Google Maps, Microsoft Bing Maps and MapQuest provide satellite tiles for maps. Googles terms of service does not allow the usage of their tiles outside their own mapping implementation. Both Microsoft Bing Maps and MapQuest allow their satellite tiles to be used with other mapping implementations. Since it is possible to zoom in further on the Microsoft tiles and the general image quality was better, they were chosen.

As previously noted, the geolocation points on the map must be clustered. Unfortunately no clustering library for JavaScript was found during research. Therefore the clustering algorithms chosen, `K-Means` and `DBSCAN`, was implemented. The implementation was done using pseudocode from Data mining essentials[47] by Jiawei et.al. A library containing the implementation was subsequently released as open source[48].

`K-Means` is a partition-based clustering algorithm that requires the developer to select a number of clusters before start. In the most naive algorithms this amount of clusters are then initialised with a random value from the dataset. The algorithm works by running through all points in the dataset, if a point is close enough to the center of the cluster, it is inserted into the cluster and the mean value of the points in the cluster is calculated as a new center. The algorithm runs for a predefined number of iterations or until it converges against an equilibrium.

The `DBSCAN` algorithm is density-based. It works by "growing" the clusters one by one. The first point in the dataset is chosen as a cluster. Now the algorithm walks through the dataset recursively looking for datapoints within distance ϵ. If a point is found, it is added to the cluster and the search now works from that point looking for nearby points. When the algorithm reaches a point where no neighbours are found, the growth continues from the previously selected point.

While `K-Means` is only able to form spherical clusters, `DBSCAN` is able to form clusters that are non-spherical. This could prove useful if a person is located in a non-square building. Furthermore the fact that you have to predefine a number of clusters with `K-Means` makes it difficult to work with as one user could have datapoints scattered in a much larger area than another, thereby having more distinct locations.

Both algorithms were tested on live data, a qualitative analysis of the clustering revealed that `DBSCAN` performed better than `K-Means`.

[47] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Morgan Kaufmann Publishers, San francisco, 3rd edition, 2001. ISBN 9780123814791

[48] The clustering library is available at `http://github.com/bss/clustering.js`.

Timeline

The first version of the timeline, shown opposite, looked very similar to the first mockups made. Interesting points are de-

picted as lines on the timeline. However, this added clutter to the design and was subsequently removed.

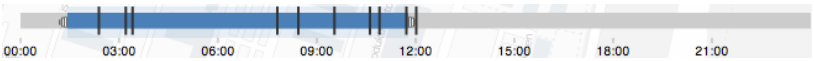

Figure 23: The first version of the timeline. Notice the scale which let the user reframe the content.

The second iteration included playback of the data. This allowed the user to playback the data shown, giving a much more dynamic feel to the visualisations above. In addition to playback, the ability to repeat playback was also added. This made it possible for the user to loop the playback of her day.

If we look at the data ink-ratio of the timeline it is very low, especially after the markers were removed. To increase the data ink-ratio, we let the timeline depict the activity happening throughout the day. The activity is shown as a line graph in the timeline.

Figure 24: Timeline showing the activity through the day. For the map the activity depicts how much the user changes location. For social connections it depicts how many social encounters happen though the day.

The addition of the line graph enable the user to easily see when she moved around through the day. However, the data/ink-ratio can still be improved by removing the part of the line graph where no data is present.

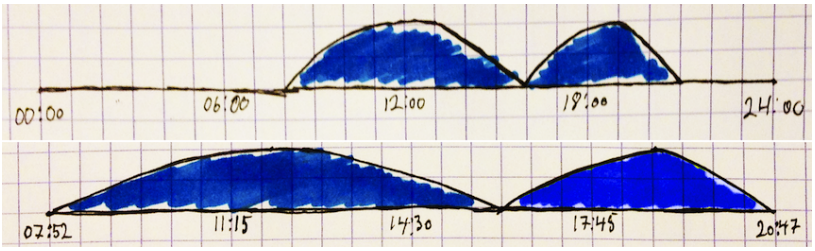

Figure 25: Rough sketches of the timeline iterations. From showing a full day even though data is not present, to "zoom" in on the actual activity of the day.

During development it became apparent that, at certain occasions, it would be benneficial to be able to control the playback speed. A slider was added to the interface, allowing the user to do so.

Even though the timeline is limited to the timeframe, of the day, in which data existed, a similar pattern occured for most users. Nothing happened in the timeline (datapoints were still reported) at night, at 8-9 AM the user started commuting to work, not much movement occured until 16-17 when the user commuted back home. Now if the user decided to playback the movements throughout the day, she would have to wait for the playback to reach 8-9 AM before anything of interest happened. She could of course have reframed the timeline manually, but it would work much better if the system was able to do that for her.

A naive approach was followed to solve this problem. Through trial and error it prooved to work fairly well. We find the interesting area of the graph by calculating the 0.95 quantile of the data.

Then the timeline is reframed to the first and last occurence of a datapoint outside the 0.95 quantile. An additional buffer of 10 minutes is added to each side of the reframed timeline to ensure that we include at least some of the slope leading up to the highest points on the graph.

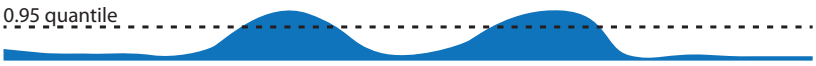

Figure 26: The timeline is automatically reframed based on the properties of the data.

This solution does have some limitations. If we have a graph with a "bump" in the beginning of the day and one at the end, we would reframe the content to include both "bumps". This is not always the optimal solution. But since most data have had most movement occur in the middle of the day, the approach have been sufficient so far.

The social-graph

THE DESIGN FOR THE SOCIAL GRAPH presented previously is a very common node-link graph. It has the advantage of being very widely recognisable.

However, it does have some limitations in the amount of data is is able to convey clearly. As described in related work, clutter reduction can be done in various ways. The most common approach is to either add interaction to a common type of graph or to introduce new types of visualisation. To find the best method to visualise the social network, three types of visualisation was implemented.

The nature of the data resulted in a fairly dense graph as shown opposite. A chord graph of the same data is shown in figure 28. In the chord graph the connections are more distinct. However if the graph contains more nodes and edges, even the chord graph is insufficient.

For larger graphs, an adjacency matrix is a possibility (see figure 29). It conveys connections very easily between two persons. However, it is difficult for the user see how connections lead from one person to the next in the same way a node-link or chord graph allows.

To reduce clutter in the node-link graph, a fisheye distortion could be used. It works as a lens on the graph, by expanding the area where the mouse hovers, thus making it easier to distinguish connections. Another possible clutter reduction method is to highlight links connected to a specific node, when you hover that node.

Similar clutter reduction techniques can be used for the chord graph. When you hover a persons name at the perimiter of the circle, all other links fades away. An example of this is shown in figure 30.

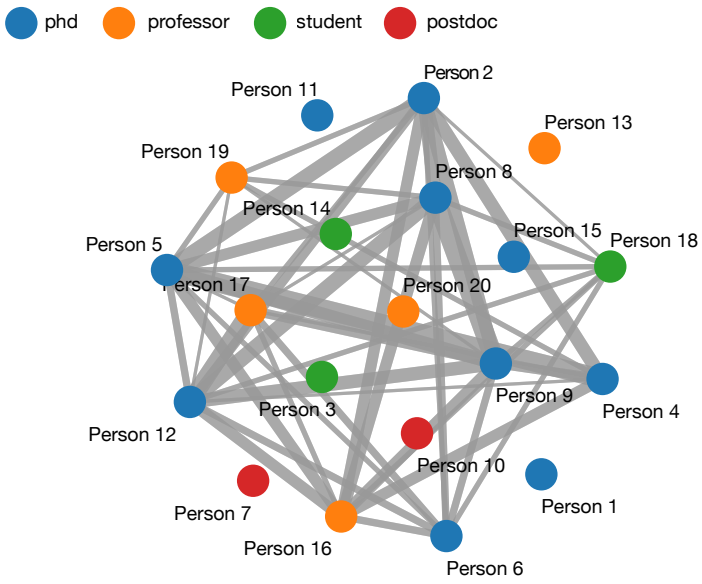

Figure 27: Node-link graph of the social connections.

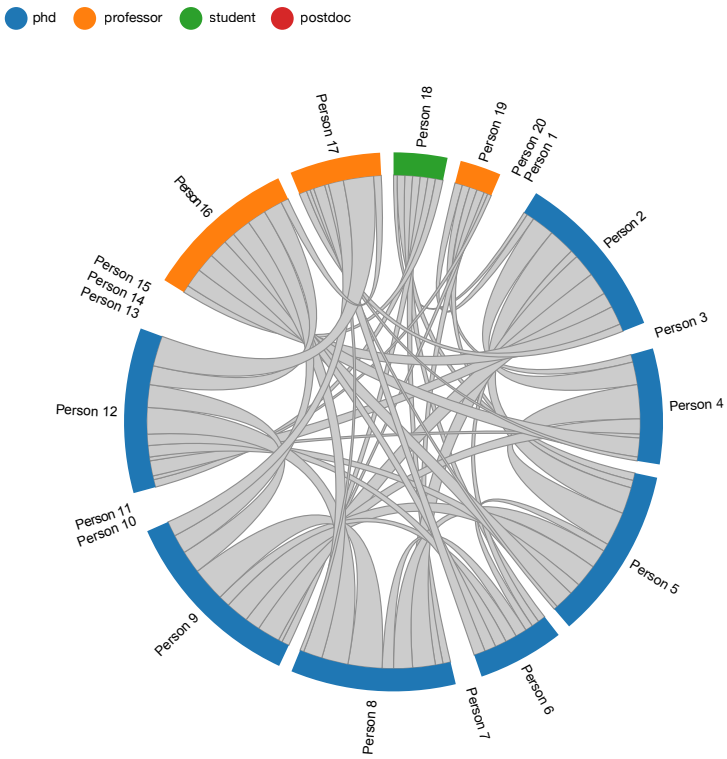

Figure 28: Chord graph (or radial table) of the social connections.

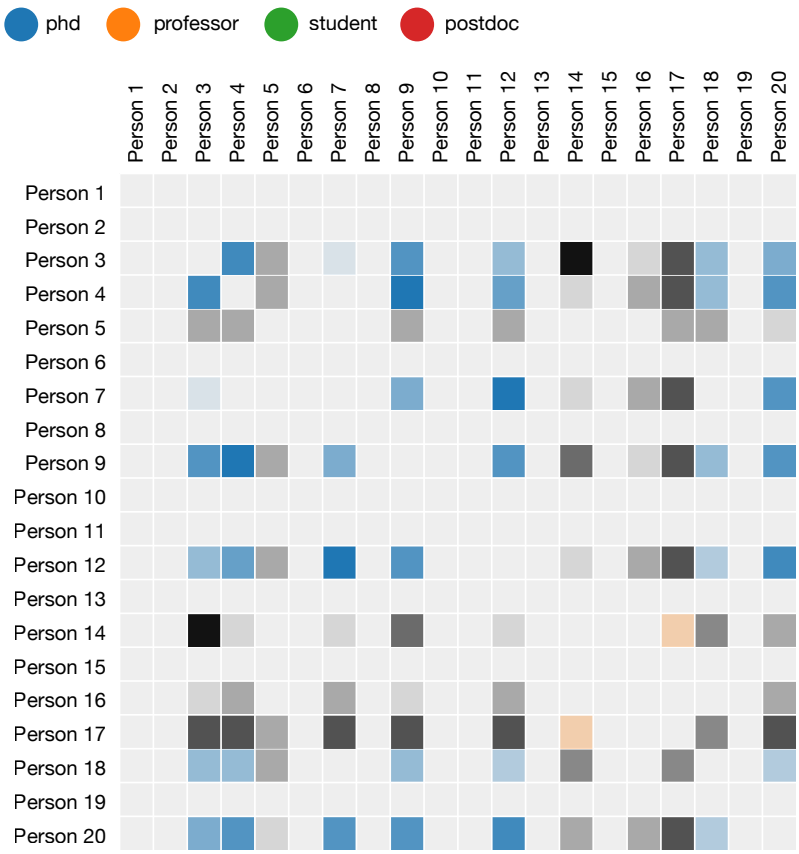

Figure 29: Adjacency matrix of the social connections.

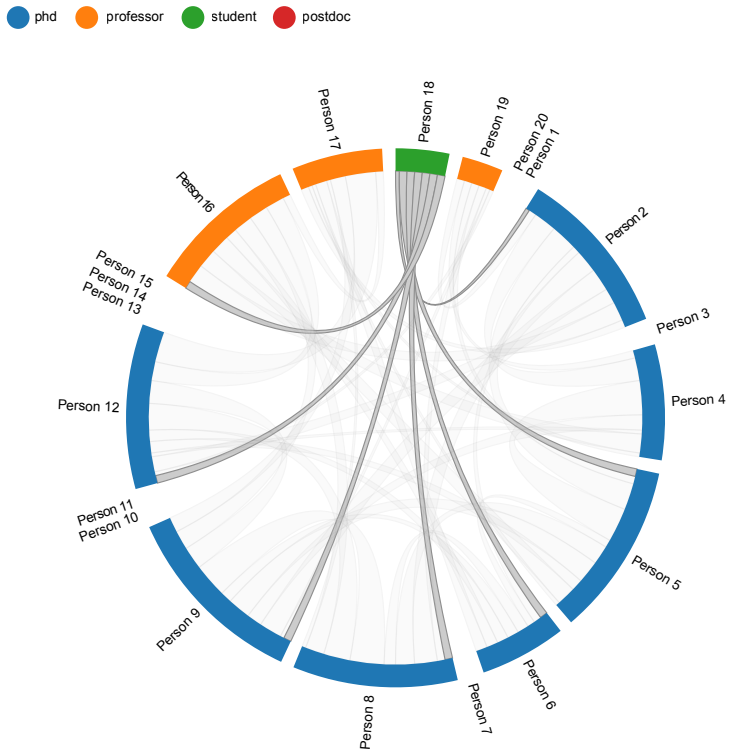

Figure 30: The display of the chord graph when the user hovers a name on the graph. Notice that most links fades away.

Even though the node-link and chord graphs share similar properties and the node-link graph is more known by users than the chord graph, the chord graph was chosen for the final implementation. The location of the nodes in the node-link graph is implemented using approximation of physical forces. This results in graphs where a specific node are located at different locations whenever you refresh the visualisation. In the chord graph on the other hand, a user is always shown close to the same persons around the circle.

The steady location of users makes the chord graph behave more calm when animated.

Stats

The stats panel is used to give a short overview of my movement and social connections by the numbers. Basically this is very concise overview for personal informatics use. While the map and social connections let the user explore their past behavior and maybe even reminisce about it. The stats overview present a very concise and comparable version of the data.

Due to time constraints, not as much of the stats overview as desired was implemented. Much of the interface could be improved, to better answer questions such as:

- At what time of the day do I move the most?

- Do I create most social bonds in specific locations?
- Where are those locations?
- Does those locations change over time?
- Do the connections I spend the most time with, change over time (*e.g.* a year)?

The final version of the stats panel is shown in figure 34 on page 53.

Limitations

THE FINAL SYSTEM contain various limitations.

All user data is loaded from the backend when the application start up. This does not pose a problem with small amounts of data, but as more data become available for each user, this will eventually become an issue.

A possible solution is to only load the last couple of weeks worth of data and then lazy load the rest of the data when needed.

Currently, the social graph in the application show the relationship between all users. When additional users are added to the system, this will obviosuly become an issue. A solution to this, is to only show actual friends in the visualisation.

Performance improvements

DURING DEVELOPMENT, the performance of the implementation was optimised in different ways. This was done to ensure a responsive user interface.

Clustering

ONE OF THE BOTTLENECKS in the application is the clustering algorithm. Clustering is recalculated every time the user reframes the timeline and at every tick at playback of the data.

The running time of the naivly implemented clustering algorithm is $O(n^2)$. A general optimisation of the algorithm exist that have a running time of $O(n \cdot \log n)$. However due, to the properties of our visualisation we are able to optimise the algorithm even further.

The reason we are clustering the data is to reduce the number of points we have to show on the map. If we bin the data into bins of 300 datapoints each and cluster each bin, we get a running time of $O(n \cdot (x \cdot \log x))$ where the worst-case value of x is 300. By reducing the expression, the second part become constant and we are thus gaining a linear runtime. The impact of this on our visualisation

is that we gain an extra point on the map for each bin. This has proven tolerable in the visualisation.

The performance of the clustering optimisation is evaluated in the next chapter.

Timeline

In the social connections view, the timeline rendering was not performing as well as for the map data. This was due to additional processing required by the data format of the social data.

The timeline is made by splitting the data into a number of slots. Now the algorithm runs through the slots, taking all data in a specific slot, binning it and counting the data in each slot.

Previously the data for a specific slot was found using the built-in JavaScript filter function on the data array. Since the filter function must work on unsorted arrays the running time of it is $\Omega(n)$. Since the data is already sorted, we are able to optimise the problem by using a binary search instead of the filter function. This reduces the average-case running time to $O(\log n)$.

Final implementation

When a user opens Sensible You, the first thing she sees is the login panel (figure 31 on the following page). It allow the user to log into the application using either a previously defined username and password or the DTU Campusnet Login. After successful authentication, the map is shown (figure 32 on the next page). In the top menu, the user is able to change the date shown. In the bottom the data can be reframed or played back using the timeline.

Clicking on "Connections" takes the user to the social connections part (figure 33 on page 53). In here the user is able to see how much time she spent with other people, as well as how much time they spent with each other. The timeline is also available in this view. Last, the user is able to click on "Stats". This show the personal informatics panel, that allow the user to see statistics about herself at a quick glance (figure 34 on page 53).

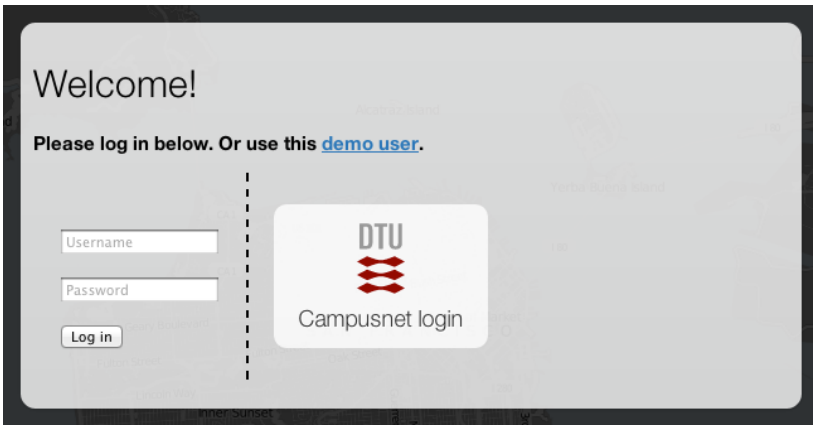

Figure 31: The login panel in the final implementation.

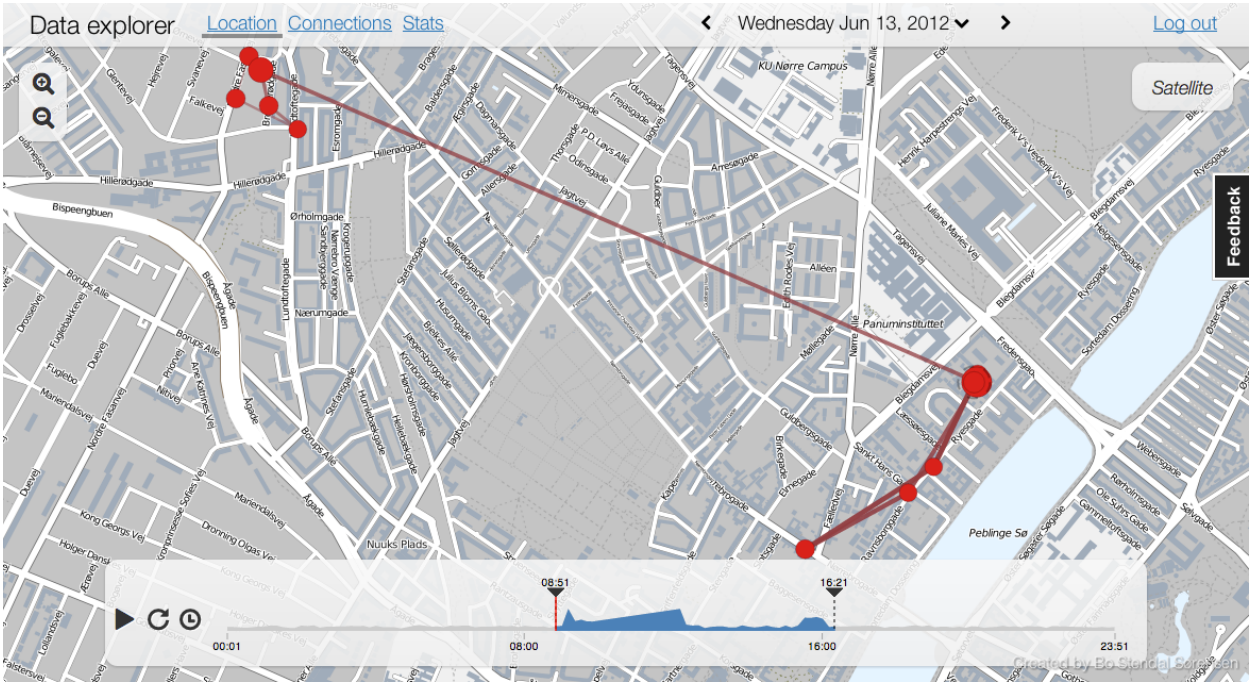

Figure 32: The final implementation of the map panel.

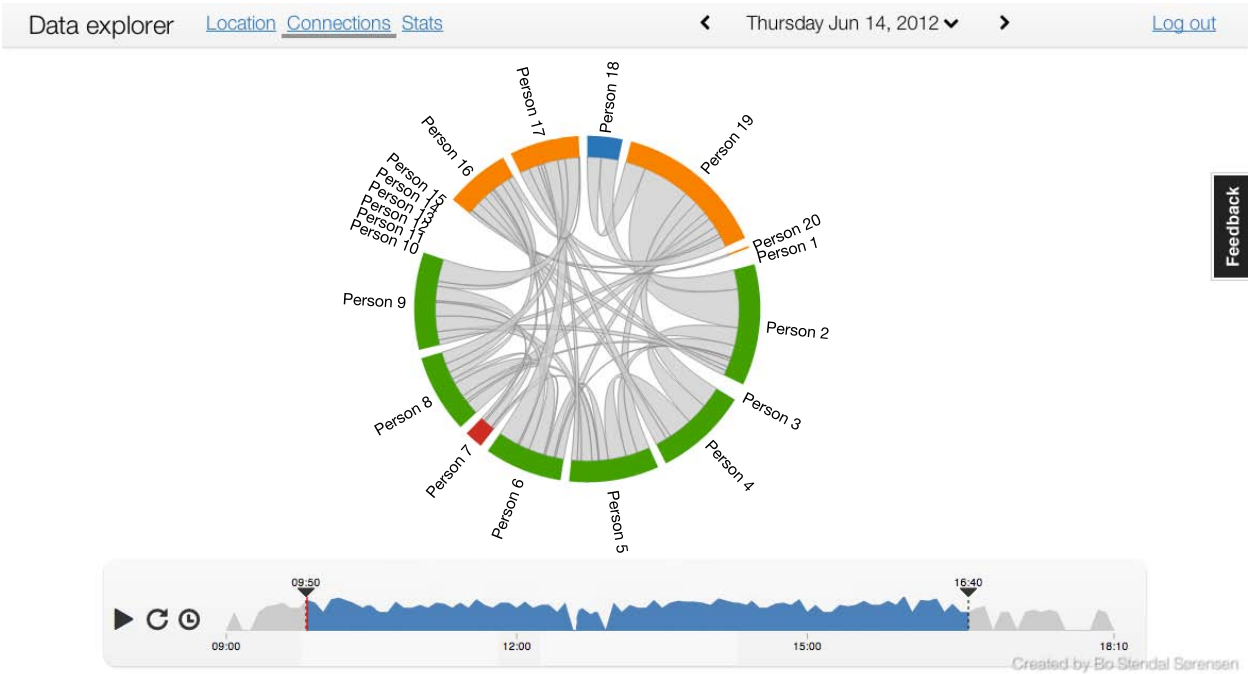

Figure 33: The final implementation of the social connections panel.

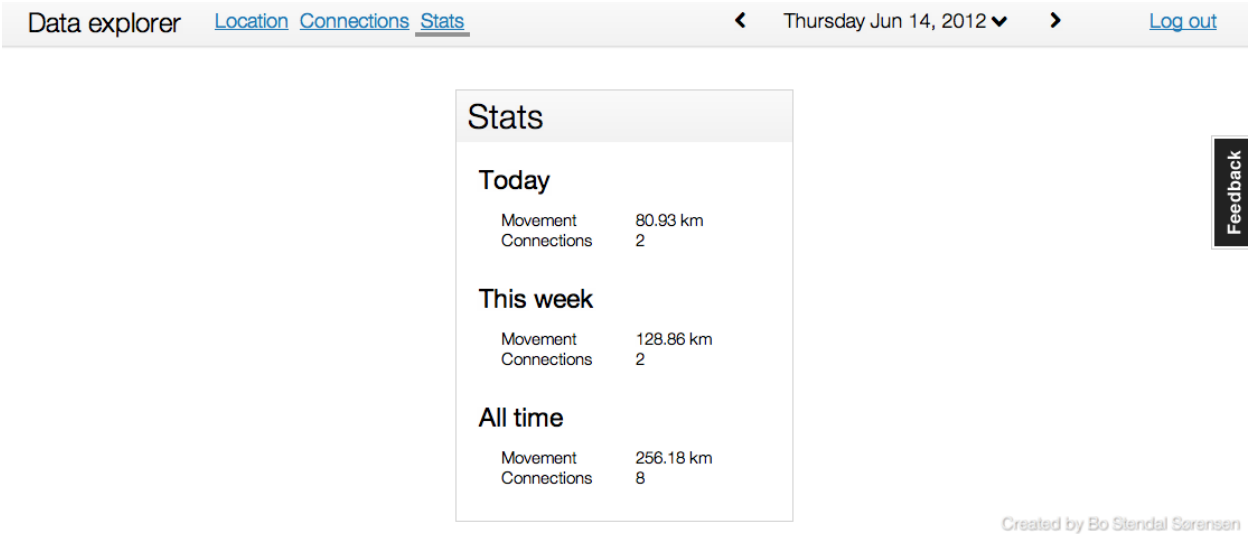

Figure 34: The final implementation of the stats panel.

Evaluation

The devised system is evaluated using feedback from the pre-alpha test participants. Furthermore the performance of the system is evaluated to ensure responsiveness when more data has been collected and more users have access to the system.

User feedback

A thoughrough testing framework was not setup for the system. Instead participants in the pre-alpha test of the experiment were asked to send feedback on the prototype. Unfortunately, only two participants gave feedback. The feedback is thus by no means statistically significant, we are however able to use the feedback qualitatively as general pointers towards a better system.

Generally the feedback fell into four categories:

- Out of scope
- Interface comprehension
- Data collection
- Improvements

One of the participants pointed out that it is almost impossible to use the visualisation on a smartphone. It was however never the intention that the system should be used on smartphones. The remark really emphasise how ubiquitous smartphones have become in recent years and how webbrowsing is no longer an activity reserved for a desktop computer. While it was not the intention to support smartphones, it should be taken into account when future developments are made to the system.

The user would have liked the visualisation to span more than a day at a time. A very valid point, that should be taken into consideration.

Asked if she would use such system on a regular basis, the user directly said that if the system contained more information and worked better on her mobile phone, she would use such system.

Furthermore, the user experienced errors in the collection of location data. She said it at some points was as much as 1km off

target. While this is very unfortunate, it is out of the scope of this system. The data collected does however include the accuracy of the location data. This could be used in the visualisation to notify the user of the inacurracies for each datapoint.

Both of the participants mentioned that they did not fully understand what the function of different parts of the UI was. One of them, even suggested adding tooltips to the buttons. Another approach would be to add text beneath each icon explaining the button. The user also suggested to add a quick tutorial the first time the system opens. While a quick tutorial could explain what different parts of the UI does, the deeper issue could be that the user do not know what the idea behind the system is. An introduction video is a good idea. It should however explain the objectives of the system, not the details of the interface.

Even though the feedback was short, it highlighted some aspects of the system that must be considered. If time had permitted it, a more in-depth user test would have been very interesting.

Performance

THE PERFORMANCE of a web-application is very important. According to Kohavi and Longbotham experiments at Amazon showed that every 100 ms increase in response time decreased sales by 1 percent.[49].

[49] Ron Kohavi and Roger Longbotham. Online Experiments: Lessons Learned. *Computer*, (September):103–105, 2007

As mentioned previosly, clustering is one of the bottle necks in the application.

The implementation chapter describes how the running time of the clustering algorithm can be improved by slicing the data into manageable chunks and then subsequently cluster each chunk.

A meassurement of different slicing values was done on to compare the running time[50]. Each clustering method was run 100 times and the median was taken. The result show the clear tendency of the original algorithm being quadratic.

[50] Benchmarking was run using JavaScript web workers on a Macbook Pro with a 2 Ghz Quad-core Intel Core i7 processor and 8 Gb RAM

It is important to maintain a responsive interface even though clustering is being run.

The data on the map imply that we will never have entirely smooth playback as a lot of movement on the map could potentially happen within a few miliseconds, i.e. when the locations tracked are far from eachother. However, the timeline in the bottom contains a playback-indicator that must be moving smoothly. Trial and error showed that the lowest framerate where the indicator was still running smootly was at 15 FPS. If the indicator must run at 15 FPS, the update time of the playback must run at least every 66 ms. Since clustering is done at every tick, it cannot take any longer than 66 ms. From the benchmark data we can extract table 3. It shows how many points we are able to cluster within the 66ms timeframe. A good estimate for a day worth of data is 300-400 datapoints. Using slices of 100 points each we could show up to around a weeks

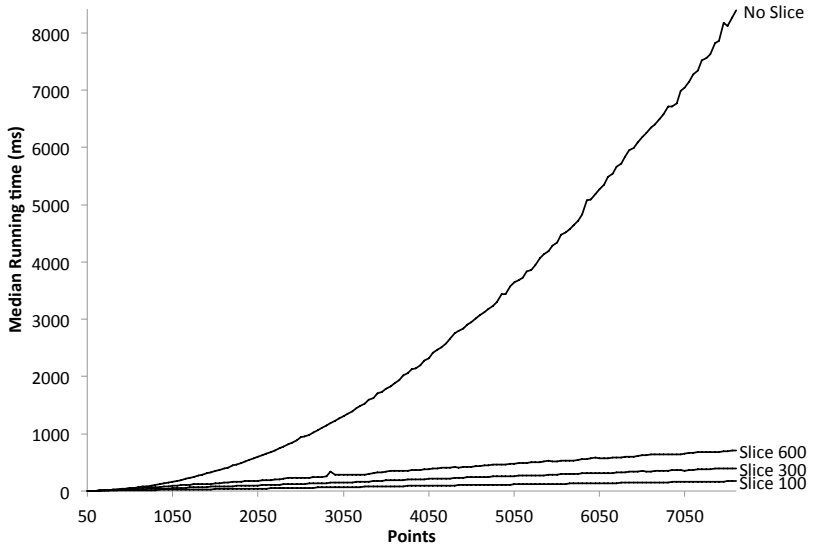

Figure 35: The median time it takes to cluster points using different slicing values. Notice the quadratic runtime, when slicing is not used and the linear runtime when slicing is used.

worth of data and still have realiable playback.

Slicing	Running time	Points
-	63.57 ms	650
600	62.84 ms	800
300	65.08 ms	1400
100	64.73 ms	3000

Table 3: The amount of points we are able to successfully cluster within a timelimit of 66 ms (corresponding to an update frequency of 15 FPS) given different slicing values.

Clustering is done on the main thread. Performance could be improved further by using web workers instead of relying on the main thread. This would introduce the possibility of making the playback animation run more smoothly. This could be achieved by having a fast framerate on the playback indicator in the timeline and do the actual update of the points more infrequently. We would gain seemingly smooth playback from a users perspective and the ability to process more points at each tick.

Another performance problem in the implementation occured in the rendering of the social connections timeline. We found that it was possible to improve the algorithm by using a binary search instead of the linear search. Initial performance meassurements showed that it had a running time of 16712ms before the optimisation. Afterwards the running time was 288ms. Both meassurements were made with a month of data (100028 datapoints).

Discussion

The objective of the thesis is to analyse, design and implement at personal informatics system for the Sensible DTU project.

The market for electronic devices that help users maintain a quantified self is flourishing. In recent years we have seen companies like Fitbit, Withings, Zeo and Nike market devices that let us collect data about our health. Furthermore, there is no reason to believe we will have fewer of these kind of devices in the future.

One problem with all of these devices is that most of them only allow the user to upload data to a single data platform, owned by the manufacturer of the device. This lock the data into data-silos, making mutli-faceted analysis of the data difficult. The same limitations with data-portability apply to Sensible DTU.

Instead, open platforms for data collection should be created, essentially giving back the data to the users. Allowing them to decide which company has the best device and who makes the best data analysis tools.

Open platforms could also open up the posibilities of a visualisation marketplace, where the best analysis tools and visualisations could be sold to people on a subscription basis.

Feedback

One of the important questions of this thesis, is if it makes sense for the users to reflect on the data collected about them. Even though the feedback recieved was very sparse, at least one person acknowledged the usefulness of reflecting upon the information gathered. Furthermore, related work tell the same story.

When Sensible DTU has been running for some time and more people are participating, the foundation for user feedback is much more solid. A more detailed user test should take place then.

Privacy

Services like CureTogether and PatientsLikeMe allow users to gain knowledge about diseases they have. Users are able to create communities and gather knowledge about very rare diseases. This could potentially help researchers find cures for those diseases.

However, if privacy is not being taking serious, patient-doctor confidentiality could essentially become a relic of the past.

Some potential privacy issues for Sensible You exist. One of the visualisations show how much a users friends are connected. While some users might not care that this data is available to their friends, others would feel it is an invasion of their privacy. To limit these kinds of concerns, users must be able to set privacy levels on their data. This in turn require that the visualisations take this into account, as they would otherwise give the wrong picture of the data.

A privacy issue in Sensible DTU is the fact that all data is stored centrally. This opens up for central attacks that could result in data leaks. Furthermore, with this model, users must trust the central authority to keep their data safe.

This could be solved by creating a decentralised system, allowing users to collect and store the data themselves or let trusted 3rd parties do it. A secure protocol for data access would need to be devised.

Future work

THE POSIBILITIES OF FUTURE WORK within personal informatics and spatiotemporal networks are many. In this section we propose future work that could be done in regard to this thesis.

Basic improvements

THE GENERAL ARCHITECTURE of the application consist of a single controller-class. The general manageability and maintainability could be improved a lot by refactoring this code. Also, the algorithm that find the optimal frame for the timeline could be improved.

One of the pre-alpha test participants said in the feedback that she would have liked to see more data than from a single day. This could be solved by changing the resolution to a week. However, some users might rather have a resolution of two weeks or keep the resolution as it is.

Instead, the interface should be changed, allowing users to seemlesly change the resolution themselves. A possible solution inspired by Google stocks is depicted below.

So far, the above solution is only a mockup. Further testing is required, to see if users understand the more complicated interface.

In an age of communication on online social networks, another obvious improvement is to gather data from the online social networks. The most prominent, at least in the western world, is Facebook and Twitter.

An integration would allow us to compare different forms of

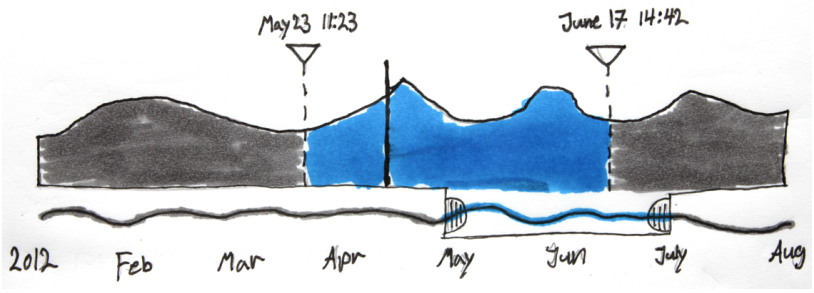

Figure 36: Posible solution to the resolution problem. The bottom handles let the user set the resolution for the upper timeline. The upper handles reframe the content of the visualisation. The vertical solid line in the upper timeline is the playback indicator. Notice the use of Tuftes micro/macro principle.

communication. Furthermore in a personal informatics context, more data sources means the user is able to better self-reflect.

Spatiotemporal analysis

AN INTERESTING IDEA, is to analyse the full spatiotemporal network to gain information that can be used in a personal informatics context.

In graph theory, the concept of centrality tell different things, depending on the data in the graph. For social network structures, Linton Freeman, introduced the concept of betweenness to describe centrality[51]. The betweeness tell us how influential a person in the network is compared to the other people in the network. This information could be used as a personal informatics metric, letting users know how influential they are are in their network.

Multiple studies show that it is possible to predict peoples behavior based on their previous actions. It would be interessting to let users know how predictable they are and subsequently study if their predictability changes when they are aware of their predictability.

[51] LC Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 1977

Driving behavioral change

IN REGARD TO THIS THESIS, only a very small user survey was conducted. A larger survey with more participants might give us more answers to some of the fundamental questions about such an interface.

One of the interesting questions is if the interface drives behavioral change among the participants. Furthermore it would be intersting to investigate how changes in the interface affect the motivation for behavioral change.

Adding gamification concepts to the application could be one of such changes. Gamification has previously showed to affect behavior of users[52].

[52] Henriette Cramer, Mattias Rost, and Lars Erik Holmquist. Performing a Check-in : Emerging Practices , Norms and âĂŸ Conflicts âĂŹ in Location-Sharing Using Foursquare. 2011

Conclusion

THIS THESIS IS A CASE STUDY in the analysis, design and implementation of a personal informatics system that enable users to gain self-knowledge and self-reflection. The system is based on the data collected in the Sensible DTU project.

The evaluation of the test result, as well as the analysis of related work, indicate that a personal informatics system does add value to users. It allows them to gain self-knowledge, reflect and reminisce about their past. Furthermore based on the data presented to the user, behavioral change could occur as well. A practical by-product of the work done in this thesis is an open source JavaScript clustering library.

The system build during the work of this thesis is a solid foundation for further work with personal informatics. Future work include using spatio temporal analysis on the social network in a personal informatics context. Studies in using personal informatics systems to drive behavioral change is another obvious direction for further study.

Bibliography

Processing.js. URL http://processingjs.org/.

Jacques Bertin. *Semiology of Graphics: Diagrams, Networks, Maps (translation from French 1967 edition)*. ESRI Press, 1st edition, 2011. ISBN 9781589482616.

Michael Bostock and Jeffrey Heer. Protovis: a graphical toolkit for visualization. *IEEE transactions on visualization and computer graphics*, 15(6):1121–1128, 2009.

Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-Driven Documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–9, December 2011.

J Clottes. Chauvet Cave: the art of earliest times. 2003.

Sunny Consolvo, James a. Landay, and David W. McDonald. Designing for Behavior Change in Everyday Life. *Computer*, 42(6): 86–89, June 2009.

Dan Cosley, Victoria Schwanda Sosik, Johnathon Schultz, S. Tejaswi Peesapati, and Soyoung Lee. Experiences With Designing Tools for Everyday Reminiscing Experiences With Designing Tools for Everyday Reminiscing. (July):37–41, 2012.

Henriette Cramer, Mattias Rost, and Lars Erik Holmquist. Performing a Check-in : Emerging Practices , Norms and ' Conflicts ' in Location-Sharing Using Foursquare. 2011.

Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7): 1057–1066, April 2009.

Nathan Eagle, Alex Sandy Pentland, and David Lazer. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of ScienceEagle, N., Pentland, A. S., & Lazer, D. (2009). Inferring friendship network structure by using mobile phone data. Proceedings of the National Academy of Sciences of the United States of America, 106(36), 1527*, 106(36):15274–8, September 2009.

Geoffrey Ellis and Alan Dix. A taxonomy of clutter reduction for information visualisation. *IEEE transactions on visualization and computer graphics*, 13(6):1216–23, 2007.

Benjamin Franklin, John Woolman, and William Penn. *The Autobiography of Benjamin Franklin*. P. F. Collier & Son Company, 1909.

LC Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 1977.

Jon Froehlich, Tawanna Dillahunt, Predrag Klasnja, Jennifer Mankoff, Sunny Consolvo, Beverly Harrison, and James A. Landay. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. *CHI Proceedings of the 27th*, 2009.

Ben Fry. *Visualizing Data*. O'Reilly Media, Inc., 2008. ISBN 9780596514556.

Galileo Galilei. *Sidereus nuncius*. 1610.

Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques*. Morgan Kaufmann Publishers, San francisco, 3rd edition, 2001. ISBN 9780123814791.

Jeffrey Heer and Michael Bostock. Declarative language design for interactive visualization. *IEEE transactions on visualization and computer graphics*, 16(6):1149–56, 2010.

Jeffrey Heer and Danah Boyd. Vizster: Visualizing online social networks. *Visualization, 2005. INFOVIS 2005. IEEE*, 2005.

Jeffrey Heer and Adam Perer. Orion: A system for modeling, transformation and visualization of multidimensional heterogeneous networks. *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 51–60, October 2011.

Nathalie Henry, Jean-Daniel Fekete, and Michael J McGuffin. NodeTrix: a hybrid visualization of social networks. *IEEE transactions on visualization and computer graphics*, 13(6):1302–9, 2007.

Gary Hsieh, Ian Li, Anind Dey, Jodi Forlizzi, and Scott E. Hudson. Using visualizations to increase compliance in experience sampling. *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, page 164, 2008.

Christiaan Huygens. *Systema Saturnium*. 1659.

Noah Iliinsky and Julie Steele. *Data Visualizations*. 2011. ISBN 9781449312282.

Sanjay Kairam, D MacLean, and Manolis Savva. GraphPrism: compact visualization of network structure. *Proceedings of the*, 2012.

Noreen Kamal, Sidney Fels, and Kendall Ho. Online social networks for personal informatics to promote positive health behavior. *Proceedings of second ACM SIGMM workshop on Social media - WSM '10*, page 47, 2010.

Ron Kohavi and Roger Longbotham. Online Experiments: Lessons Learned. *Computer*, (September):103–105, 2007.

Menno-Jan Kraak. Geovisualization illustrated. *ISPRS Journal of Photogrammetry and Remote Sensing*, 57(5-6):390–399, April 2003.

Mei-Po Kwan and Jiyeong Lee. Geovisualization of Human Activity Patterns Using 3D GIS : A Time-Geographic Approach. *Spatially integrated social*, 2004.

Matthew L Lee and Anind K Dey. Reflecting on Pills and Phone Use : Supporting Awareness of Functional Abilities for Older Adults. *CHI '11*, pages 2095–2104, 2011.

Ian Li, Anind Dey, and Jodi Forlizzi. A stage-based model of personal informatics systems. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, page 557, 2010.

Anmol Madan, Manuel Cebrian, David Lazer, and Alex Pentland. Social sensing for epidemiological behavior change. *Proceedings of the 12th ACM international conference on Ubiquitous computing - Ubicomp '10*, page 291, 2010.

Anmol Madan, Katayoun Farrahi, Daniel Gatica-perez, and Alex Sandy Pentland. Pervasive Sensing to Model Political Opinions in Face-to-Face Networks. *Lecture Notes in Computer Science*, 6696/2011:214–231, 2011.

Charles Joseph Minard. *Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812–1813*. Regnier et Dourdet, Paris, 1869.

Paul Ohm. Broken promises of privacy: Responding to the surprising failure of anonymization. 2009.

S. Tejaswi Peesapati, Victoria Schwanda, Johnathon Schultz, and Dan Cosley. Triggering memories with online maps. *Proceedings of the American Society for Information Science and Technology*, 47(1): 1–4, November 2010a.

ST Peesapati, Victoria Schwanda, and Johnathon Schultz. Pensieve: supporting everyday reminiscence. *Proceedings of the 28th*, pages 2027–2036, 2010b.

Wei Peng, M.O. Ward, and E.a. Rundensteiner. Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering. *IEEE Symposium on Information Visualization*, pages 89–96, 2004.

William Playfair. *Commercial and political atlas*. 1786.

William Playfair. *The statistical breviary*. 1801.

Alasdair Rae. From spatial interaction data to spatial interaction information? Geovisualisation and spatial structures of migration from the 2001 UK census. *Computers, Environment and Urban Systems*, 33(3):161–178, May 2009.

Verónica Rivera-Pelayo, Valentin Zacharias, Lars Müller, and Simone Braun. Applying quantified self approaches to support reflective learning. *Conference on Learning*, pages 111–114, 2012.

Julie Steele and Noah Iliinsky. *Beautiful Visualization*. 2010. ISBN 9781449379872.

Edward R. Tufte. *The visual display of quantitative information*. Graphics Press, Cheshire, Conn., 1983.

Edward R. Tufte. *Envisioning Information*, volume 40. February 1991.

H Wainer. How to Display Data Badly. *American Statistician*, 38(2): 137–147, 1984.

Samuel D. Warren and Louis D. Brandeis. The Right to Privacy. *Harvard law review*, 4(5):193–220, 1890.

Appendix A
Use cases

This appendix contains the remaining use cases for the project.

Use case	#3: Show statistics
Description	Show statistics about the user.
Actors	User, Sensible DTU backend.
Main scenario	Include use case #1 "authentication". 1. User clicks on "Statistics". 2. Statistics about the user is shown.

Table 4: Use case #4: Show statistics

Use case	#4: Show social connections
Description	Show the historic social connections of the user.
Actors	User, Sensible DTU backend.
Main scenario	Include use case #1 "authentication". 1. User clicks on "Connections". 2. The users social connections are shown.
Extensions	3a. User playbacks the social connections on the screen (it is seen how they change over time).

Table 5: Use case #5: Show social connections

Use case	#5: Show location of friends
Description	Show the location of the users friends.
Actors	User, Sensible DTU backend.
Main scenario	Include use case #1 “authentication”. 1. User clicks on “Location”. 2. User clicks “Show friends”. 3. The last locations of the users friends are shown together with the location of the user.

Table 6: Use case #6: Show location of friends