Växjö University

School of Mathematics and Systems Engineering

Reports from MSI - Rapporter från MSI

Content Visualization of GeoAudio Notes

Ilir Jusufi
Lulzim Junuzi

Jul
2008

MSI
Växjö University
SE-351 95 VÄXJÖ

Report 08076
ISSN 1650-2647
ISRN VXU/MSI/DA/E/--08076/--SE

Abstract

The total population of GPS-enabled location-based services (LBS) subscribers is constantly increasing [1]. This fact implies new research possibilities for visualizing geospatial data produced by these mobile devices. The aim of this thesis is to explore novel techniques and methods to visualize the content of voice notes (messages recorded by users on GPS-enabled devices) that will be placed in maps using GPS coordinates, and visualize the semantical, temporal, and spatial relations between the notes. Our research is part of the Geovisualization field which deals with geospatial data.

Based on our research and analyzes of this problem, we combined different visualization and interaction techniques, thus providing a novel approach to achieve the research aim. We have built a prototype application, called GNV System (GeoAudio Notes Visualization System), that demonstrates our achievements.

Keywords: GPS, GeoVisualization, Visualization Techniques, Interaction Techniques.

Acknowledgments

We would like to express our gratitude to both of our supervisors Dr. Andreas Kerren and Dr. Marcelo Milrad. They encouraged and stimulated us in each step of our work. Moreover, we thank Arianit Kurti who helped us in understanding a lot of issues considering research related to a master thesis. Furthermore, our greatest gratitude goes to our families and our beloved that made our studies possible, for their constant moral support even though thousands of kilometers separated us. Finally, we would thank all of our friends for being there with us all the way through.

Table of Contents

List of Figures

List of Tables

1 Introduction

In 2011, the total population of GPS-enabled location-based services (LBS) subscribers will reach 315 million, up from 12 million in 2006, according to ABI Research [1]. This estimation stresses the necessity to research new possibilities of visualizing geospatial data produced by these mobile devices.

If we go back and analyze the field of geographic visualization, we see that it has been used long time ago in history. The earliest examples of geographic visualization date back to the stone age with map-like wall paintings depicting the surroundings of our ancestors [14]. In 1854, John Snow, a physician from London, mapped cholera cases in a map. Figure 1 illustrates the connection of people infected by cholera in relation to the water pumps that they used. It was reported that an anomaly of that pattern finally prompted his insight, namely the case of a workhouse with very few infections in the center of the cholera outbreak: it had an independent water source [24].

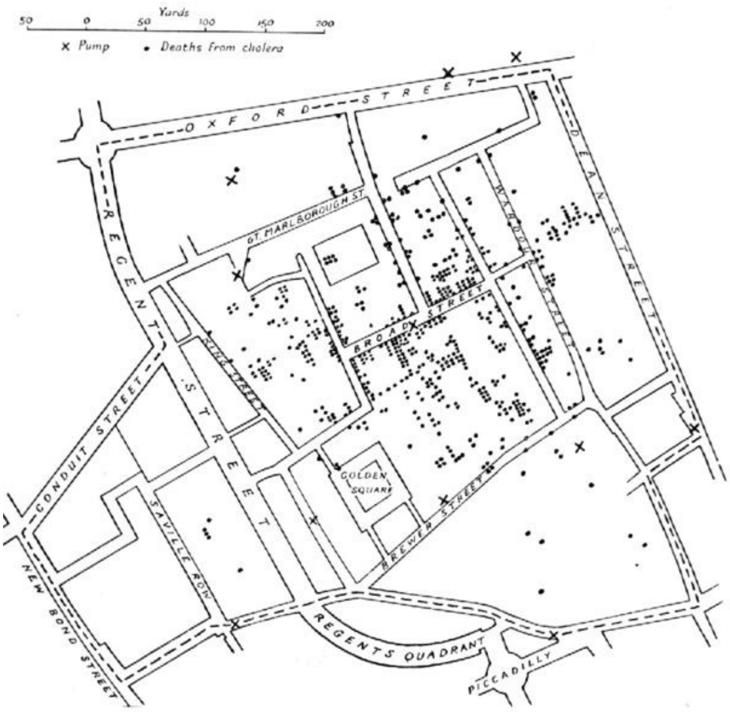


Figure 1: John's Snow map of cholera death, London 1854 [24]

The increasing amount of geographic data available in recent years are visualized on a wide range of display sizes for variable applications [2]. The basic challenge for designing compelling applications within the field of geovisualization (GeoVis) is the avoidance of too complex, too detailed and too dense visualizations [2]. The objective is simply to allow users to quickly locate and easily decode relevant geographic information to make inferences [2].

"Geovisualization integrates approaches from scientific computing (ViSC), cartography, image analysis, information visualization, exploratory data analysis (EDA), and geographic information systems (GISystems) to provide theory, methods and tools for visual exploration, analysis, synthesis, and presentation of geospatial data" [3]. Other authors describe geovisualization as "the creation and use of visual representations to facilitate thinking, understanding and knowledge construction about geospatial data" [4], or "the use of visual geospatial displays to explore data and through that exploration to generate hypotheses, develop problem solutions and constract knowledge" [22].

As mentioned in the beginning of this section, the number of devices that produce geospatial data is increasing [1]. This fact, introduces new challenges for exploring and constructing knowledge from the increasing amount of these data through visualization. Much of these data, such as recorded voice notes are unstructured, thus we face another challenge for showing the content and possible semantic relations.

1.1 Purpose and Scope

The purpose of this thesis is to explore and develop novel techniques and methods to visualize the content of voice notes that will be placed in maps according to the GPS coordinates where the notes have been recorded (e.g., Mobile phone, PDA, etc.). By visualizing these notes we may gain a number of advantages as described by Ware [18]:

1. The ability to comprehend huge amount of data.
2. The perception of properties that are otherwise not anticipated.
3. The extraction of problems with data itself, i.e., detecting outliers or anomalies.
4. The understanding of both large-scale as well as small-scale features of data.
5. The creation of various hypotheses related to the data.

The scope of our research will fall under the space of Geovisualization which deals with geospatial data. According to MacEachren and Kraak [3], geospatial data and information are fundamentally different from other kinds of data since they are inherently spatially structured in two or three dimensions (latitude, longitude, and possibly altitude). Visualization of geospatial data introduces some of the restrains which may/will lead to usage of pure InfoVis techniques in combination with Geovisualization. InfoVis is an abbreviation of Information Visualization which is "the use of computer-supported, interactive, visual representation of abstract data to amplify cognition" [25].

The system to be presented in this thesis successfully visualizes voice notes geospatially in a map and provides additional visualization of their content. Our research focuses on finding, analyzing and proposing novel interaction techniques to query the data. We will introduce visualization highlighting techniques to better extract information/knowledge about the content of the notes. The term "highlighting" is used to describe the transient visual effects that are applied on data items across views when a mouse cursor or other input device has moved overhead [5]. We will research the means to combine and interactively change many of the visual properties of the display, and also provide new techniques to show ever-increasing amounts of data [6].

From the design and implementation point of view, our work covers mainly the visualization part of the problem. Since our research focus is to find suitable techniques for visualizing the content of geoaudio notes, we need to convert audio notes into text firstly. There are different commercial and open source applications that converts voice messages into text [37]. In order to visualize the content of the voice note, we could apply similar approaches like tag clouds [38], which visualizes word frequencies. We will research the possibility of adding interaction and highlighting techniques to improve the tag cloud approach. Thus, the main questions that we will explore in this thesis can be formulated as follows:

1. *Which are suitable techniques for visualizing the content of recorded audio notes in Digital Maps such as Google Maps?*

2. *How can different visualization and interaction techniques be combined to explore novel ways of showing semantic relations and patterns between the content of different audio notes in space and time?*

Note that these questions will be answered only in the context of combinations of interaction and visualization techniques, and the problem we presented above. A *suitable technique* in our context is a visual mapping from our data set to a visualization that is *expressive* and *effective*. The term *expressive* means that all and only the actual data set is represented. A mapping is "more *effective* if it is faster to interpret, can convey more distinctions, or leads to fewer errors than some other mapping" [42]. We will not try to explore and find novel ways for data mining and information retrieval. Furthermore, a review of current literature and work (Chapter 2) will show that at our best knowledge no complete solution to the problem has been presented yet.

1.2 Thesis Outline

Chapter 2 presents the results of the literature review, where we have been exploring relevant research efforts in the fields of Information and Geo Visualization in connection to our work. In Chapter 3 we explain the methodology we have used for our research. In Chapter 4, we describe possible scenarios to identify the features of the envisioned system. In the next chapter, Analysis and Design, we have determined the requirements, use cases, and we propose the architecture for our application. Chapter 6 consists of the initial ideas and interactions we had for our GUI. Chapter 7 is about the technical approach of our application. In that chapter, we describe the visualization techniques, the tag clouds, the clustering algorithm that we have chosen, and the interaction techniques that we have used for having a robust and flexible geovisualization system for visualizing notes and their content. In Chapter 8, we give some explanation about technologies that we have used, and explained the classes that we implemented. In the next chapter (Case Study), we used the first imagined scenario based on what we presented in Chapter 4 and tested it with the application we have implemented. Finally, in Chapter 10 we give our conclusions and we present future work that can help our system to deliver better results.

2 Literature Review

In this chapter we will discuss relevant research related to our problem. First, we will present some information about the literature covering Information Visualization and Geovisualization in general. In the next section, we will review some of the current research efforts that are more closely related to the focus of our research.

2.1 Information Visualization and Geovisualization

Nowadays, we are witnesses of mass production of data. It is very hard to understand, analyze and extract knowledge from these data by reading them one by one without any help of visualization. We might say that visualization is “The use of computer-supported, interactive, visual representation of abstract data to amplify cognition” [25]. Though visualization has not been always related to computer technology, it existed long time before the existence of the computer. Nowadays, visualization without the help of a computer is not possible, especially if we want to make it interactive. Information visualization deals with abstract information without a need to make any physical or geometric correspondence to the data being visualized, and usually it is not possible to map information into the physical world [26]. One classic example of information visualization was developed by Sir Edward Playfair's Circles (see Figure 2) in order to show that the British Empire was far too much taxed in comparison to other empires of the time [39]. The left line above a circle shows the gross national product and the right shows the tax income.

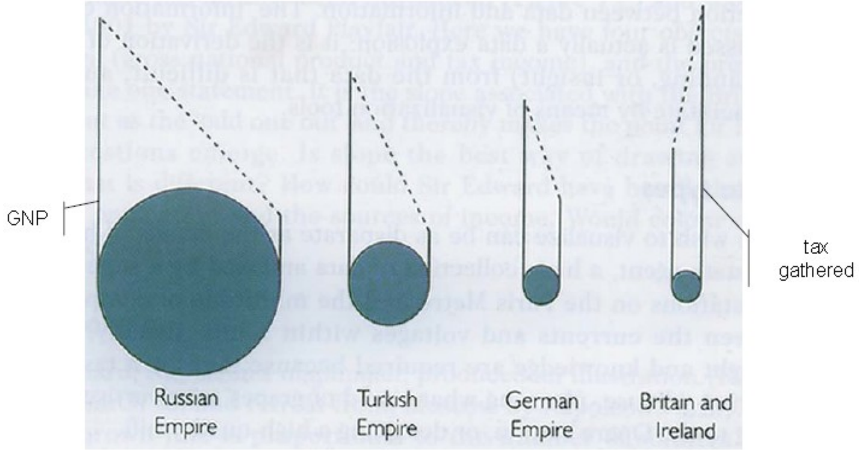


Figure 2: Sir Edward Playfaire's Circles [40]

The geographical position is irrelevant for most data, but there exist a considerable increasing number of geospatial data which can be visualized in the map as the number of GPS enabled devices is growing [1]. The Geovisualization discipline deals with these kind of data. We presented some of the definitions for geovisualization and an example of it in Figure 1.

2.2 Current Work

We have reviewed a significant amount of papers and book articles which helped us a lot while defining the problem and proposing a solution. We noticed there is a vast amount of books and papers [2, 3, 5, 6, 9, 13, 14, 16, 17, 18, 19, 23, 26, 27, 28, 29, 30] that are directly or indirectly related to our problem, but so fare none give a full explanation or answers to our questions. We will review some of latest papers and book chapters that inspired us in our work. We will do this by choosing an issue we faced and, present possibly the latest works in solving those issues and explain if and why we chose (or chose not) to implement the answers proposed in those papers.

First of all, we had to find out about the most recent work in field of geovisualization. Our main source of information about the subject was the chapter of Geographic Visualization written by M. Nöllenburg [14]. The underlying book (A. Kerren et al.) offers also a lot of information about visualization in general. It was a good point of start for understanding some of the key concepts. We learned about issues concerning visualization and interaction techniques.

Later on in our work, we needed to focus on more concrete problems of our thesis. The first step was the way to visualize the content of a single voice note. Since our notes will be transformed into text, we had to deal with unstructured text visualization. There are articles about visualization of text through the use of color (to represent grammatical structures of the text, such as nouns, verbs, adjectives, etc.) in order to get an insight of the style of writing, sentence structure, and to provide some hint about whether the text is fictional or scientific [27]. The authors of this paper claim that they found a way to visualize the genre of the text, but it actually shows the difference of the style writing (narrative vs. scientific). Considering the fact that most of the notes will belong to one of the writing styles (probably scientific field report), we found that this method had a minor impact for our work. A more sophisticated approach towards this issue can be found in Daniel Keim's and Daniela Olke's paper about literature fingerprinting where a sequence of feature values per text are calculated on different hierarchical levels and presented to the user as characteristic fingerprint [28].

One other option to visualize text is the so called Tag-Cloud approach. “A Tag-Cloud is a list of the most popular tags, usually displayed in alphabetical order, and visually weighted by font size“ [29]. There exist a number of tutorials on Tag-Cloud approaches, and they basically give the same common knowledge about them. However, there is a way to improve the approach by using clustering algorithms for visual layout [29]. Unfortunately, we did not have enough time to experiment with this approach, but we consider this to be part of the future work.

Other issues are related to visualizing the semantic relation of notes and algorithms, to cluster the notes for those relations, and the issue of visualizing the note in the map. A very good paper [30] on the issue of visualizing text in maps has been published recently. It offers a lot of interesting options which should be considered to be implemented in our prototype in the future. The paper proposes an architecture for visualizing text documents. The authors developed a toolkit called FEMARepViz, which is used to extract and visualize temporal, geospatial, and summarized information from FEMA National Update Reports. The application visualizes the documents using Google Maps or Google

Earth. Documents are categorized into predefined number categories, and the authors claim a high classification accuracy. For uncategorized documents, clustering based topic detection and segmentation algorithm can be employed [30]. This application offers a very good solution by answering the question of visualizing text in the map (which could be done better by providing some meta data to the text), and categorizing the text into specific topics (thus providing a sort of content relation of documents). However, it lacks the content visualization of the document.

2.3 Summary

We have shown that visualization is not new, but with the introduction of computer technology new possibilities were created. A lot of work has been conducted to improve the existing and find new ways to visualize data. We presented a review of current papers and books which are directly or indirectly related to our work. Some of these papers offer interesting approaches that we used in our work, while some can be considered as improvement of our approach in future. None of the papers and books offers a complete answer to our question. The literature review was an important phase in our research methodology, which we will describe in the next chapter.

3 Methodology

In this chapter, we will describe the methodology we used throughout our research. In order to have good results, we combined two models for our research. The first model is based on the Jenkin's Model [15] and the second one is UPEDU (which shortly means Unified Process for EDUcation) model [8].

3.1 Jenkin's Model

Jenkin's model consist of 8 sequential steps [15] as it is shown in Figure 3.

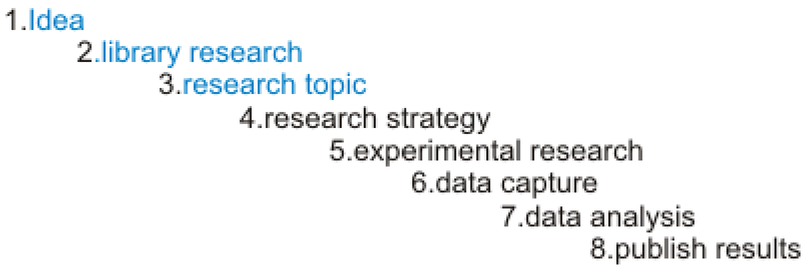


Figure 3: Jenkin's model[15]

The initial three steps of our research were based on Jenkin's model (steps with blue color in Figure 3.

- The first step is the main idea. Having the idea, we can proceed with next steps. Before the idea there is a certain state of affairs occupying our mind, thus a problem or a question we want to find out an answer [15].
- In the second step, library research, Jenkins himself wants to emphasize that there is no substitute for library research in refining the initial idea to enable the next step [15]. In theory-testing studies Jenkins wants to encourage a researcher firstly to know what the others have written about that research idea, and then to elaborate the research idea [15]. Even thought it is hard to find a research that is similar to ours, we will try to gather as much as possible information that are related to our research.
- The third step, research topic, can be broken into three main components: originating question(s), i.e., what you want to know; research rationale, i.e., why you want to know it; and specifying questions, i.e., which particular questions you need to investigate in order to provide the answers to the originating question(s) [15].

3.2 UPEDU Model

The second model that we used is the UPEDU software engineering process. It is decomposed into four sequential phases: Inception, Elaboration, Construction, and Transition [8].

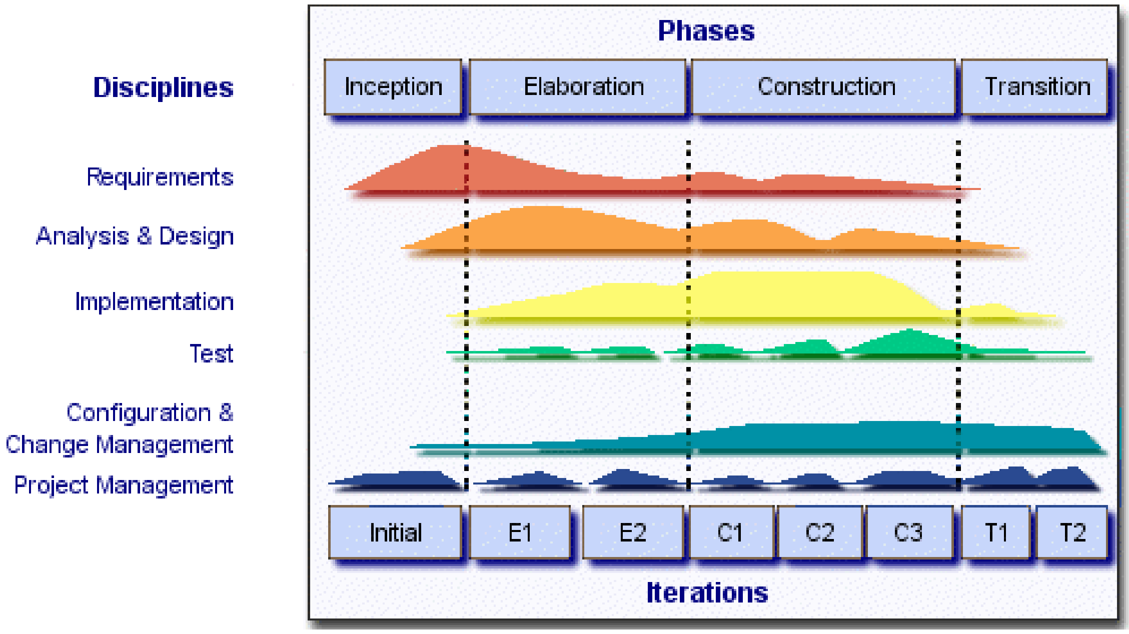


Figure 4: UPEDU model [8]

- *Inception phase*: In this phase, it is very important to get a good knowledge for the work that we will have to deal with. In the inception phase, the aim is to formulate the scope of the project [8] and prepare the environment for the project [8]. In our case, this phase will comprise the first three steps of Jenkin's model discussed above.
- *Elaboration phase*: The goal of this phase is to build a stable architecture which will provide a stable basis for the future work in implementation. In this phase, our work will be mainly concentrated on analysis (use cases, sequence diagrams, etc.) and designing (class diagrams, architecture, etc.). We will analyze and decide on some of the key visualization techniques. We will analyze possibilities of finding optimal filtering methods in order to display only relevant data to avoid the overloading of the visualization. Then, we will examine the use of different clustering techniques for better knowledge extraction.
- *Construction phase*: The goal of this phase will be to start implementing the project based on the analysis and design that we will make in elaboration phase. In this phase, the main work will be on implementing prototype version by using the most suitable visualization techniques for our project. Also, it is possible to work on analysis and design if there is a need for changes. As a result, it is important to achieve a useful version and to start testing.
- *Transition phase*: By the end of transition phase, lifecycle objectives should have been met and the project should be in position to be closed out [8].

Figure 4 shows the amount of time spent on each activity regarding to the phase we have been working on. Each phase is divided into one ore more iterations. For instance, the *Elaboration* phase is divided into two iterations *E1* and *E2*. In each iteration, a minor milestone is planed and should be completed and documented [8].

3.3 Summary

As presented above, we combined two methodologies although we mostly focused on UPEDU methodology. But the first phase (Inception phase) of the UPEDU methodology was combined with the first three steps (idea,library research, topic research) of the Jenkin's model. The elaboration phase, construction phase and transition phase was used without a combination with Jenkin's model or other models. It is worth to mention that the initial phases (of Jenkin's model) were spent towards overall research such as literature and topic exploration. The remaining phases, respectively the UPEDU model were aimed more towards the design and implementation of the prototype. In order to design good prototype, we should define requirements we need to implement. In the next chapter we will present some scenarios that will help us in defining those requirements.

4 Application Scenarios

In this chapter, we describe three scenarios that are suitable for using our application.

4.1 Scenario: Forensic Investigation

A murder has occurred and the forensic agents are in a crime scene. There can be more than one forensic agent taking notes about the crime scene. These notes will be uploaded in the repository and will hold the GPS location where the audio note is taken. Later on, all agents can compare the notes between them by visualizing them with the support of the system. When the agent logs in from the work, he will see the map of the city and he can go to the location of the crime. Beside his notes, will find other notes from his colleagues visualized on the map. When he clicks on some of the notes from that location, he will see the note's visualization on the predefined area of the display. In this way, he can get an insight of the content of each note he selected. He can change the visualization view and switch to other tab to see the semantic relations of the notes. Here, he can find all the notes semantically related to the selected note (he can further add clustering criteria such as space and time). When he sees the clusters, he will find out that there are notes from different location that match with selected note. This might show some patterns between two or more crimes. This way, forensic agents can possibly discover some correlation between their findings.

Imagine that a few days latter, another murder has occurred, but in some other part of the city. This other murder falls into some other police district, and both of them do not seam to be correlated. However, when they upload their notes in the system, the visualization will show possible pattern that connect these two cases. The agent will see through the help of visualization, the content of other cases that are in the same cluster with his case. He will find that one case has some interesting similarities and a certain pattern exists between them. These patterns will be discovered really fast by the agent since the visualization will use *preattentively processed* features. Certain features of the visual image are identified easily after a very brief exposure of the image. This process, which logically occurs before the attention of the user is concentrated on the details of the visual image, is called *preattentive processing* [13].

So, we might get the possibility that the murders are committed by the same person. This way, we will help agents to gain more knowledge of the case which will hopefully lead to solving it.

4.2 Scenario: Ecological Field Work

Imagine the case where a lot of ecological researchers are conducting field work. They explore some region and take audio notes which are later uploaded to GNV System. After they have done the work on the field they upload their notes on the repository. The system will visualize their note on the map using GPS coordinates pointing to the place where the note has been taken. The researcher can see other notes in the proximity to his working area. He can select one of the notes and see the visualized content of it. With the help of this visualization he will have an overview of the content of the note, and if interested, he can view the entire text or listen the note.

Later on he might want to see if other researchers have come to the similar findings as himself, or if they discus about similar subjects in their notes. He can choose the clustering option to automatically group messages with similar semantic content. The visualization will help him find those notes. For example, if he discovered that certain flower grows in one area, he might find that other researchers in his proximity speak about the same plant. He can change the clustering parameters by adding time as criteria. Moreover, he can find out that there are notes talking about that plant only during certain period of time. Thus, researchers can conclude during which season the plant grows. He can also discover what kind of terrain is more suitable for the plant by looking at the map.

4.3 Scenario: Solutions for Unknown Diseases

A Swedish care center has obliged all doctors which had complex cases and than found a solution to them to make a report about that disease and to upload the report in GNV System. In addition to the report which will describe all the symptoms of the disease, doctors need to give also the solution about that disease, thus explaining how they cured it. If the doctors will encounter a patient with a strange disease, they will use GNV System to see if somewhere else in Sweden existed a patient with same symptoms. If it existed and the patient was cured successfully, then in the GNV System will exist a solution for that disease uploaded by the doctor who cured that patient.

Doctors of the Växjö Hospital have a difficult emergent case where one patient is suffering a disease that doctors do not know exactly what kind of disease. After the patient went through different tests for finding from which disease is suffering and what is the cause of that, still the doctors were without conclusions. The patient is in critical condition, and if doctors do not react soon, the patient life may be in big danger.

One of the doctors takes a note about all the symptoms that the patient has and after that uploads the note in the repository of GNV System. The note of her patient appeared on the map, exactly at the position where Växjö Hospital is. After the doctor has concluded with the first step, she selects all the map of Sweden, and clusters all the notes that are uploaded by other doctors across the country trying to find if some other doctors had the same problem in the past.

The doctor understands from the system that her note has some relations with other 5 notes in Sweden. After the doctor reads other notes that are related with her note, she finds that one of the note contains identical details with the note that she uploaded about her patient. The doctor can see that the same case has occurred in Gothenburg 2 years ago. After reading the solution report that was issued by the doctor that had the same case in Gothenburg Hospital, she analyzes that with other colleagues and they give the exact cure to the patient which after some hours shows delighted improvements.

4.4 Summary

From the scenarios described above, we came to a phase that defines our problem better. By reading these scenarios, we can understand what a user would expect from the system. We can agree that users would expect to visualize the notes in a specified position in the map, get a fast overview of the content of the note and, show semantical relation between notes through various interaction techniques among other things. Based on these three scenarios, in the next chapter we will try to define our requirements and use cases that will help us later, to design and implement our application.

5 Analysis and Design

In this chapter, we will present the requirements of our prototype. We will define all use cases that will support our implementation. We will give more details on some of the most important use cases. This chapter will end with a section describing the proposed architecture of our prototype.

5.1 Requirements

The purpose of the requirements is to define the boundaries [1] of the application we will have to build. From the requirements, system developers will have a better understanding of the system requirements. There are two types of requirements: functional requirements and non-functional requirements. Based on the scenarios presented earlier we have made a list of requirements that our system should fulfill.

5.1.1 Functional Requirements

Functional requirements specify actions that a system must be able to perform, without taking physical constraints into consideration, thus specify the input and output behavior of a system [1].

FR0	**Setup connection with Google Maps**
Importance	Essential
Description	The browser makes a simple HTTP request on a web server (Apache Server, IIS, etc.) which returns an HTML page with JavaScript. JavaScript will call the Google Map server, and than the page will be rendered.
FR1	**Transform voice to text**
Importance	Essential
Description	After the voice message is uploaded in the web service, the voice message will be converted to text.
FR2	**Extract most frequent tags**
Importance	Essential
Description	Most frequently used tags will be extracted from the text document.
FR3	**Store the data into database**
Importance	Essential
Description	After text processing (converting voice into text and extraction

	of most frequent words), processed data will be stored into database.
FR4	**Cluster the data**
Importance	Essential
Description	Clustering techniques will be used in order to separate a set of notes in groups by similarity of their content. Certain frameworks exists in the way a set of clusters are defined in relation to each other.
FR5	**Visualize notes in a map**
Importance	Essential
Description	Using GPS coordinates the note will be visualized in the map in the exact location where the note has been created. Each note in the map will be generated as a bitmap. From the visualization of the note, users will get some information about the note: time created, number of words, and duration.
FR6	**Visualize content of the note**
Importance	Essential
Description	The content of the note will be visualized in the Notes Visualization Panel by displaying most frequently used tags on the note.
FR7	**Select a space**
Importance	The user should have the possibility to select a desired space on the map allowing the user to have the opportunity to manipulate the notes inside the selected space.
Description	
FR8	**Visualize the notes with semantic relations**
Importance	Essential
Description	The system will visualize the semantic relations of the notes by placing them to the corresponding clusters. Notes in one cluster should have the semantic relation between them.
FR9	**Visualize the notes with semantic relations in time**
Importance	Essential

Description	The system will visualize all the notes that are semantically related in the required time by the user.
FR10	**Visualize the notes with semantic relations in space**
Importance	Essential
Description	The system will visualize all the notes that are semantically related in the required space by the user.
FR11	**Display the text document of the note**
Importance	Essential
Description	When the user will decide to view the whole document of the note, the text document of the note will be displayed.
FR12	**Search for the note**
Importance	Essential
Description	When the user will search for a specific note, the system will visualize the notes that are related with the user search.
FR13	**Select the note**
Importance	Essential
Description	When the note will be shown on a map, the user can select it. After the note has been selected, the application will display more information about that note (e.g. More tags, relations with other notes, etc.).
FR14	**On mousover**
Importance	Desirable
Description	If the user decides to know more information about the notes in a map, it just needs to go over that note with the mouse, and the application will display more tags about that note .
FR15	**Interaction between Map Panel and other panels**
Importance	Essential
Description	When the note has been selected, the tags of the chosen note in the Note Visualization tab should be displayed. Also. after the selection the note should be selected in the Clustering tab. This should have also an effect to the Tool Box Panel, where the information of the selected note will be displayed.

FR16	**Interaction between Notes Visualization Panel and other panels**
Importance	Essential
Description	This panel contains the Note Visualization tab and Semantic Relation (Clustering) tab. When the user selects the note on the Clustering tab, that note should be highlighted on the Map Panel, and the information about that note should be displayed on the Tool Box Panel. When the user will click on the Note Visualization tab, on the Tool Box Panel should be shown the options about the Note Visualization Panel. The same should happen when the user will click on the Clustering tab.
FR17	**Interaction between Tool Box Panel and other panels**
Importance	Essential
Description	The Tool Box Panel should interact with the Map Panel when the user conducts a search. The Tool Box Panel will interact with both tabs on the Notes Visualization Panel. The user will use the tag slider if he/she will want to change the number of tags displayed on the Note Visualization tab. Moreover, if the user wants to change options about clustering, it will need to interact with the cluster sliders.
FR18	**Compare notes**
Importance	Desirable
Description	The user can compare the notes and decide if there are similarities between content of the selected notes.
FR19	**Play the recorded note**
Importance	Desirable
Description	The user can play the recorded voice note.

Table 1: Functional Requirements

5.1.2 Non-functional Requirements

Non-functional requirements are often categorized as usability, reliability, performance, and substitutability-requirements [8]. They are often requirements that specify need of compliance with any legal and regulatory requirements [8].

N-FR0	**The system should work fast**
Importance	Essential
Description	The system should work fast enough so the user will not wait too much for response from the application.
N-FR0	**The system should work in different browsers**
Importance	Desirable
Description	The application should work with most popular browsers (e.g. Internet Explorer, Mozzila Firefox, Opera, etc.).
N-FR0	**The application should be scalable**
Importance	Desirable
Description	The system should be built in a way that makes possible the adding new features and operations in the future.

Table 2: Non-Functional Requirements

5.2 Use Cases

Based on the requirements we have designed use cases. Use cases will serve later to implement the system. Use cases will be used to understand the required system behavior and to refine the system [1]. Figure 5 shows use cases for our system.

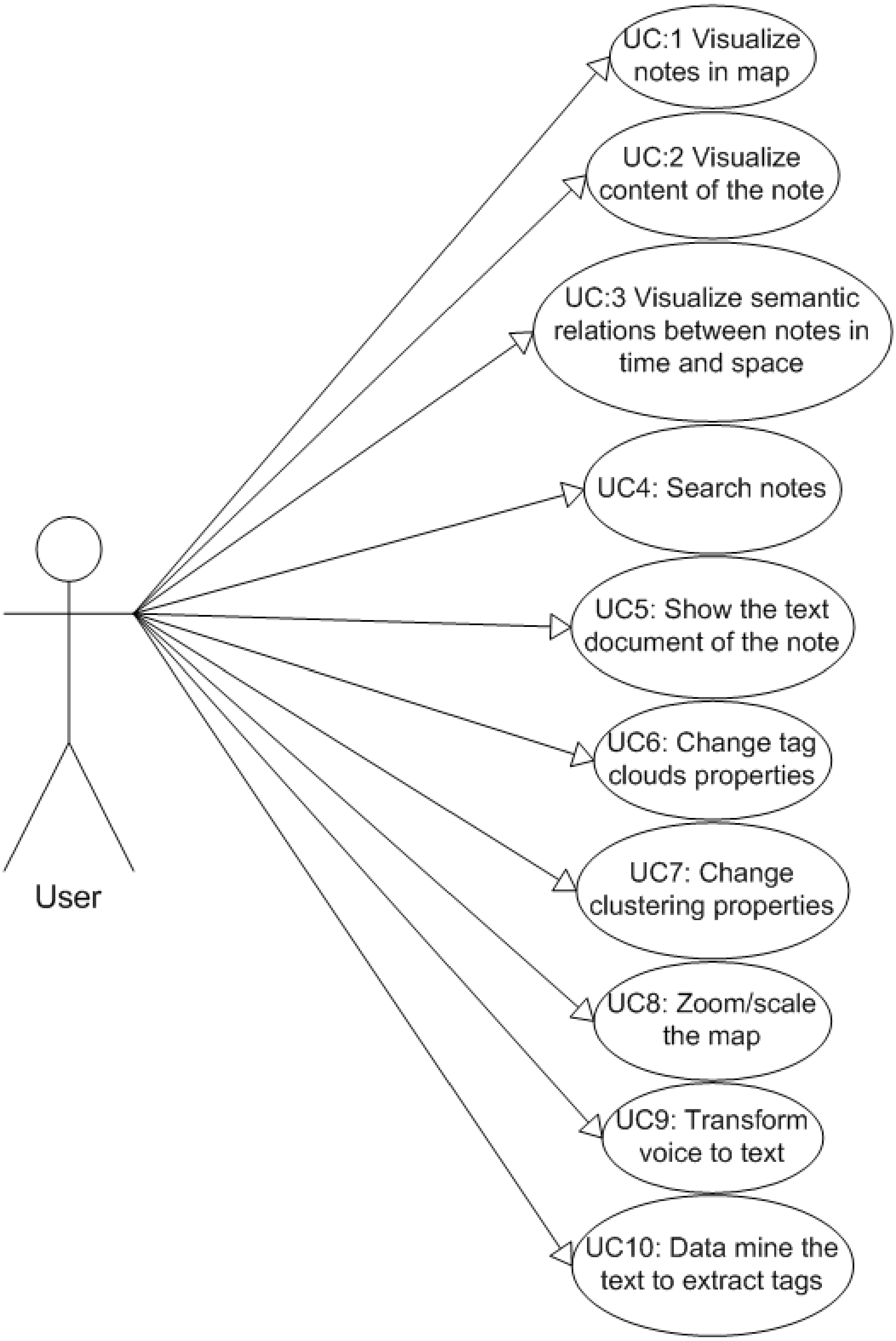

UC:1 Visualize notes in map
UC:2 Visualize content of the note
UC:3 Visualize semantic relations between notes in time and space
UC4: Search notes
UC5: Show the text document of the note
UC6: Change tag clouds properties
UC7: Change clustering properties
UC8: Zoom/scale the map
UC9: Transform voice to text
UC10: Data mine the text to extract tags
User

Figure 5: Use Case diagram

UC:1	**Visualize notes in map**
Description	After the application starts, it shows the initial view of the map. All the notes from that view are visualized in the map in places defined by the GPS data each note has. Notes are presented in the map as a bitmaps. The bitmap will contain different colors. Each color will present some information about the note (e.g., time created, number of words, duration of the note, etc.). After the notes are visualized in the map, user can perform various interactions with them
Flow of events	Application starts. The notes are displayed in the map as a bitmaps. Each note bitmaps have colors dependently on the meta data of the note. Selecting the note, most frequently used tags of that note will be shown in Note Visualization Panel. Selecting the note, the information about that note will be displayed in the Tool Box. Clicking twice on the note, the information window will appear in which will be displayed some text from the note. Also, in the information window users will have the option to view the entire text, or to listen the note.
Preconditions	Notes should exist in corresponding view (area) where the application starts.
Postcondition	Depending on what user has searched for or the position of the view in the map, the notes will be shown on the map. The user can view the details of each note, by selecting the note, or by clicking twice on the note. Each note will be presented on the map with different colors dependently on the meta data of the notes.
Sequence diagram	Figure 6

Table 3: Use Case 1

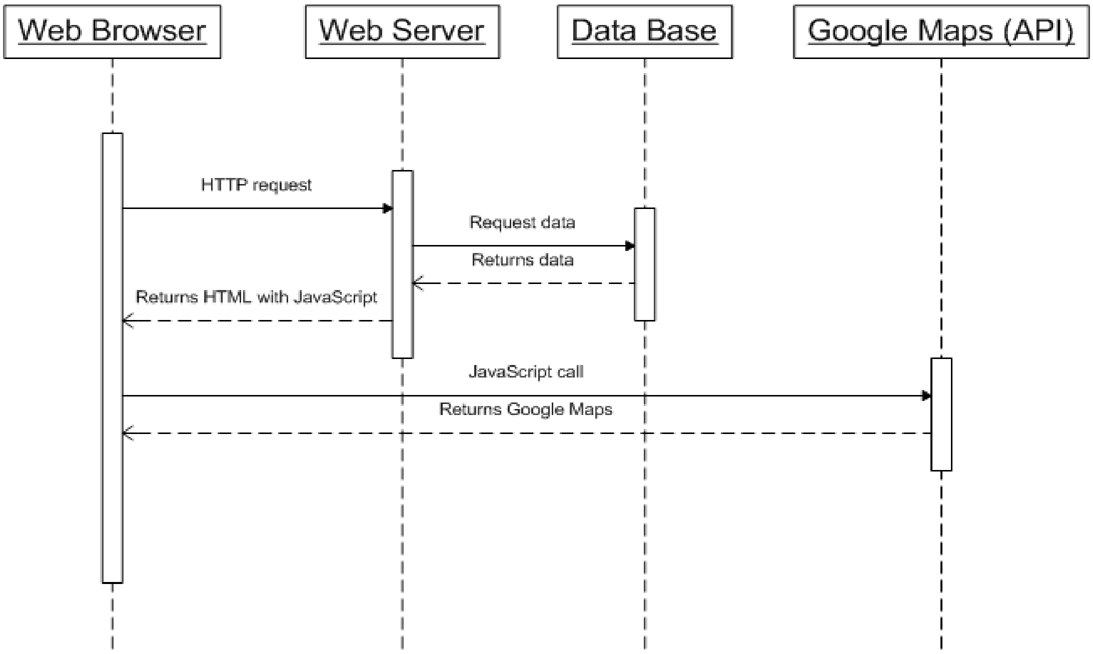


Figure 6: Sequence diagram for UC1

UC:2	**Visualize content of the note**
Description	After the user selects a note from the Map Panel, the content of the note will be displayed in the *Note Visualization Panel* which has two panels, *Note Visualization Panel (Tag Clouds),* and *Semantic Relation (Clustering)*. The user has different options for viewing the visualized notes. First, in the note visualization panel tags of the note will be shown. The second option is by viewing notes in semantic relation panel, which shows the relation of the selected note with other notes. Other option that the user has for seeing the content of the note is when the user will click twice on the note in the map. There will be opened the info window in which will be shown some text from the note, and the user will have the option to read the entire document, or to listen to the note.
Flow of events	The user selects a note in Map Panel. The note is displayed in Note Visualization Panel (Tag Clouds). The note is displayed in Semantic Relation Panel (Clustering). The user clicks twice on the note in Map Panel. The info window is opened where some text from the note is visualized. The user has options to read entire text or to listen to the note.
Preconditions	The user should select a note in a Map Panel. The user should click twice on the note in a Map Panel.
Postcondition	The content of the note is visualized in Note Visualization Panel

	(Tag Clouds) by showing most frequently used tags. Tags that are used more will be displayed in bigger fonts. Some text of the note will be visualized in the Info Window, when the user will click twice on the note. Every time that the note will be clicked or selected, the information about that note will be shown in the Tool Box.
Sequence diagram	Figure 7

Table 4: Use Case 2

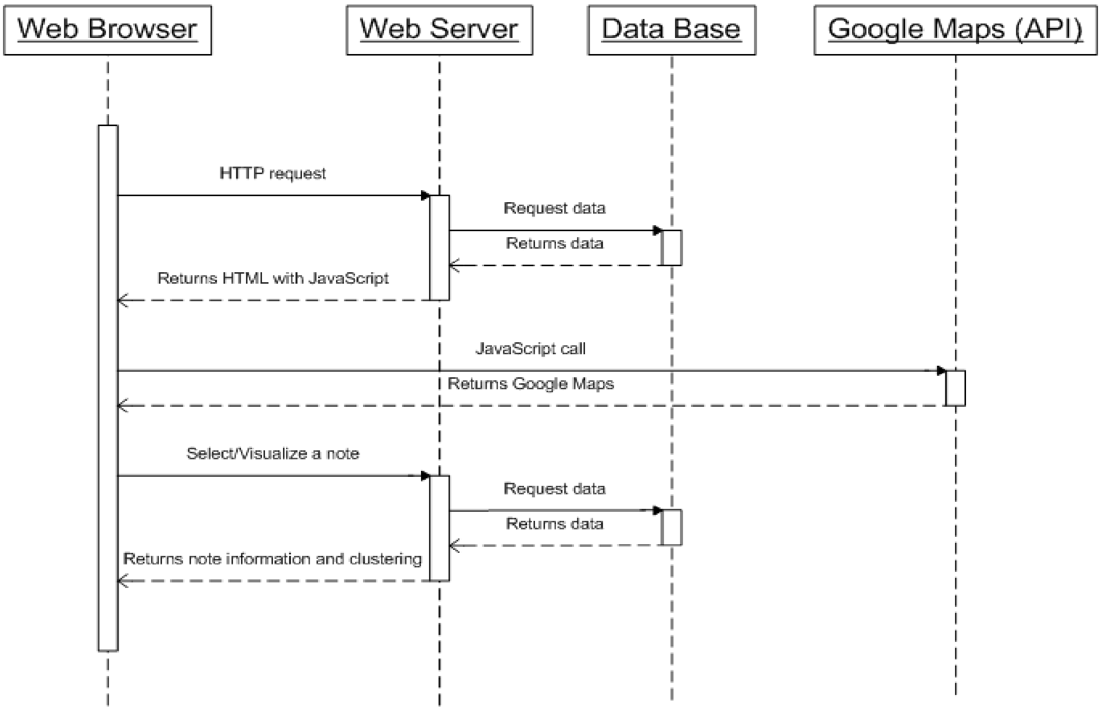


Figure 7: Sequence diagram for UC2

UC:3	**Visualize the semantic relations between notes in time and space**
Description	For visualizing the relations between notes in time and space, the user should select in the tab control that he/she wants to view the notes that have a specific time and specific space. If the user wants to view notes that were added last week, he/she needs to change the clustering options by changing in the date. When the user will require from the application to show just the notes that are older 1 week, he/she will change the date option, and in the map will be displayed just the notes that were added last week. The same applies if the user choses to

	view the notes that are related between them in specific space, by showing the notes of that place. For doing this, all the user should do is to select in the Map Panel the desired area, and after pressing the Cluster button, the notes of that area will be visualized in the Clustering Panel by showing the semantic relations between the notes.
Flow of events	The user wants to see the notes of specific time that are related between them. The user wants to see the notes of specific place that are related between them.
Preconditions	User chooses which notes of which time or place will be visualized.
Postcondition	Based on the users search the semantic relations of the notes of specific time or place will be visualized.
Sequence diagram	Figure 8

Table 5: Use Case 3

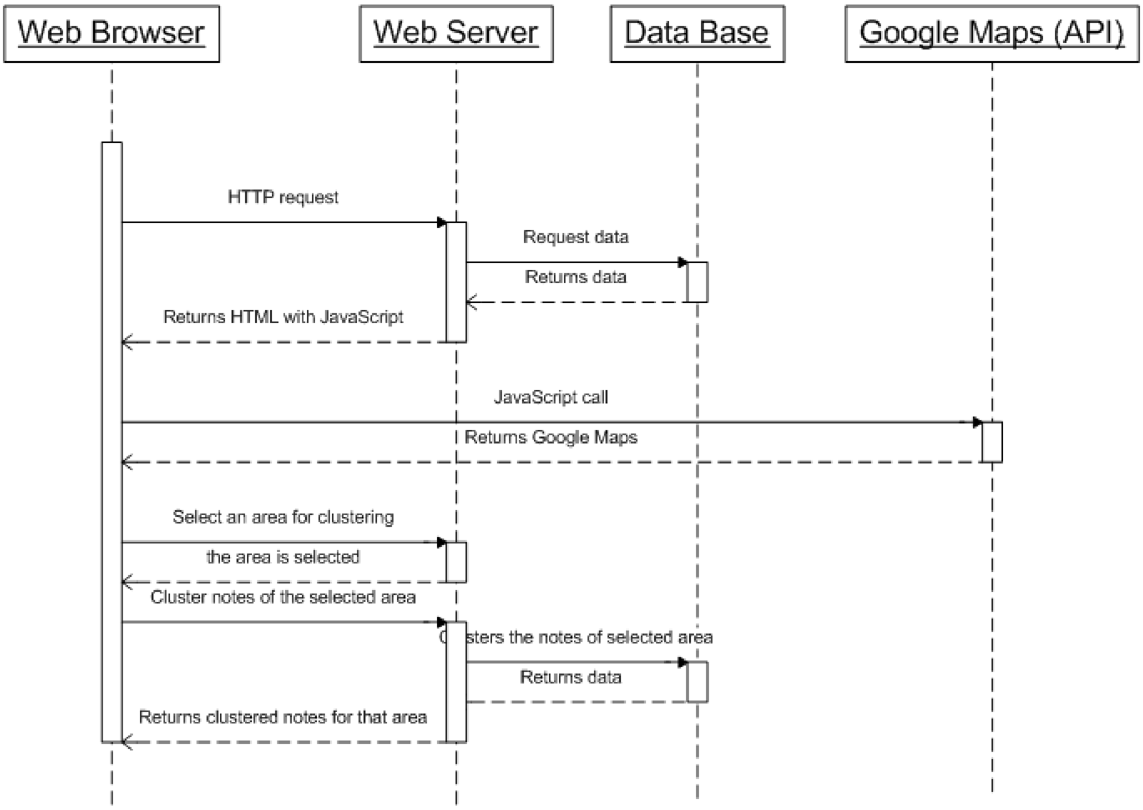

Web Browser
Web Server
Data Base
Google Maps (API)
HTTP request
Request data
Returns data
Returns HTML with JavaScript
JavaScript call
Returns Google Maps
Select an area for clustering
the area is selected
Cluster notes of the selected area
Clusters the notes of selected area
Returns data
Returns clustered notes for that area

Figure 8: Sequence diagram for UC3

UC:4	**Search notes**
Description	In the Tab Control the user has the option to make a search for a specific note. The search can be made in different manners. He/she has the option to make a search by location, date, name, tags, etc. After the search, the notes which have relation with the search will be shown on the map.
Flow of events	The user makes a search by location or by date, name, tags, etc. The notes will be displayed on the map, dependent of the user search criteria.
Preconditions	The user should initialize a search.
Postcondition	Dependent of the search criteria, the notes will be shown on the map.
UC:5	**Show the text document of the note**
Description	When the notes are visualized in the Map Panel, the user will click twice on the desired note and above the note will appear the Info Window. In the Info Window the user will have the option to read the text document of the note.
Flow of events	The user clicks twice on the note in the Map Panel. The user clicks on the link to open document. The document shows the whole text that contains that note.
Preconditions	The user should had clicked twice in the note that wants to view.
Postcondition	The entire text of the note is displayed.
UC:6	**Change tag clouds properties**
Description	In the Tool Box panel, the user will have the option for determining the number of tags that will be visualized on the Note Visualization Panel, in the Tag Clouds tab. The default number of the tags that will be visualized is 50.
Flow of events	The user chooses the number of tags to be visualized. The chosen number of tags is visualized in the Tag Cloud tab.
Preconditions	The user decides for the number of tags to be visualized.
Postcondition	Tags are visualized in the Tag Cloud tab.
UC:7	**Change clustering properties**
Description	There are different options that users can change before they

	decide to cluster. Users have the options to choose the number of centroids for clustering the notes. Also users will have the option to choose the value of the threshold. User can change the clustering properties by adding/removing tags from cluster criteria, and adding/removing time and space criteria.
Flow of events	User sets clustering properties The application visualizes the clusters of notes.
Preconditions	The user should set the clustering properties.
Postcondition	The clusters are visualized depending on the properties set by user.
UC:8	**Zoom/scale the map**
Description	The user should zoom/scale the map. Google API provides with built-in controls for zooming and scaling.
Flow of events	Users selects the action on zoom/scale control Map is zoomed in/out.
Preconditions	Map zoom controls should be displayed.
Postcondition	According to the action selected by the user (using zoom control) the map is either zoomed in or zoomed out.
UC:9	**Transform voice to text**
Description	When the voice will be uploaded to the system, the voice will be converted to the text using existing softwares (e.g. Dragon).
Flow of events	The author of the voice document uploads the file into system. The voice file will be converted to text document using existing software (e.g. Dragon).
Preconditions	The voice document should be uploaded into system.
Postcondition	The voice document will be converted to text document.
UC:10	**Data mine the text to extract tags**
Description	The system should extract the most used words from each note. These words (tags) will be used to visualize the content of the notes. Each tag will have its weight representing the number of occurrences in a note.
Flow of events	Most used words will be chosen as tags. The tags will be used to visualize the content of the note.

Preconditions	There should be a text document.
Postcondition	From the text document, the mos used words will be extracted as tags.

Table 6: Use Case 4 – 10

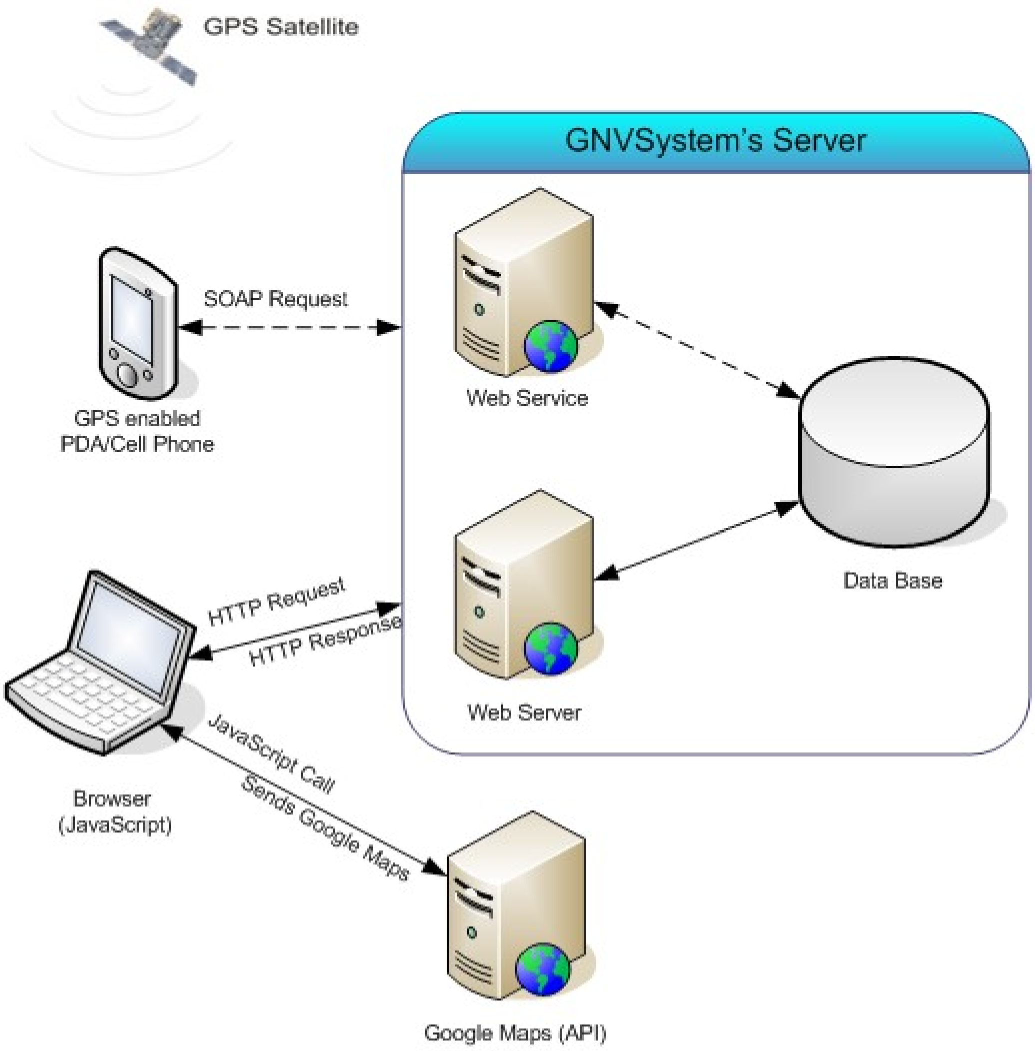


Figure 9: General Architecture

5.3 Proposed Architecture

Based on our analysis and scenarios explained in previous chapters, we propose a system architecture shown in Figure 9. By analyzing the requirements we defined that we can clearly distinguish the different components of the system's general architecture. Requirement FR0 shows that we should have a Google Maps API server and a Browser which implies the use of JavaScript.

Requirements FR1, FR2 and FR3 will not be implemented by us. Note that the parts of Figure 9 where the dashed lines are used are not going to be implemented by us. However, we should take them into consideration when designing the architecture. These requirements (FR1, FR2, and FR3) imply the use of the following components: GPS Satellite, PDA/Cell Phone, Data Base, and Web Service server which will actually implement these requirements. Other requirements imply the existence of a Web Server where the visualization will be generated, and the Browser which will display the output of the visualization. The browser is also an interface between user and other components of the architecture.

This drawing (Figure 9) depicts high-level system components and the communication between them. Later in this section, we will provide a more detailed explanation of the proposed architecture.

As we can see from Figure 9, our system architecture will incorporate at least two other sub-architectures into one (if not three considering the JavaScript call to Google Map API server). The mobile phone/PDA should support GPS in order to record the coordinates of the recorded message. The communication between PDA and Web Service server is of service oriented architecture (web service). This web service should store the voice message. Afterwards, the voice message is converted to text, and the web service should extract the most frequent tags (which will later be used to query and visualize the message). The text and tags related to it are stored in the database.

The web server will generate the actual application for visualizing of these GeoAudio notes. This part of the system can be treated as standard Client-Server architecture, but at the client (browser) side there will be two requests to display the application (this will be explained later on the text). We will additionally use Google Map API to display the GeoAudio notes. This implies that browsers should support JavaScript. An application that combines Google Map API and other data sources is called Web application hybrid [7]. The browser makes a simple HTTP request on a web server (Apache Server, IIS, etc.) which returns an HTML page with JavaScript. The JavaScript code will call the Google Map server, and than the page will be rendered. So, the browser actually does two 'calls' and displays the page using two data sources (our application data and Google Maps).

In Figure 10, we see that a voice note has some metadata attached to it (which are later stored in a separate table in database). These meta data holds generic information, such as name of the file, time created, GPS coordinates, etc.

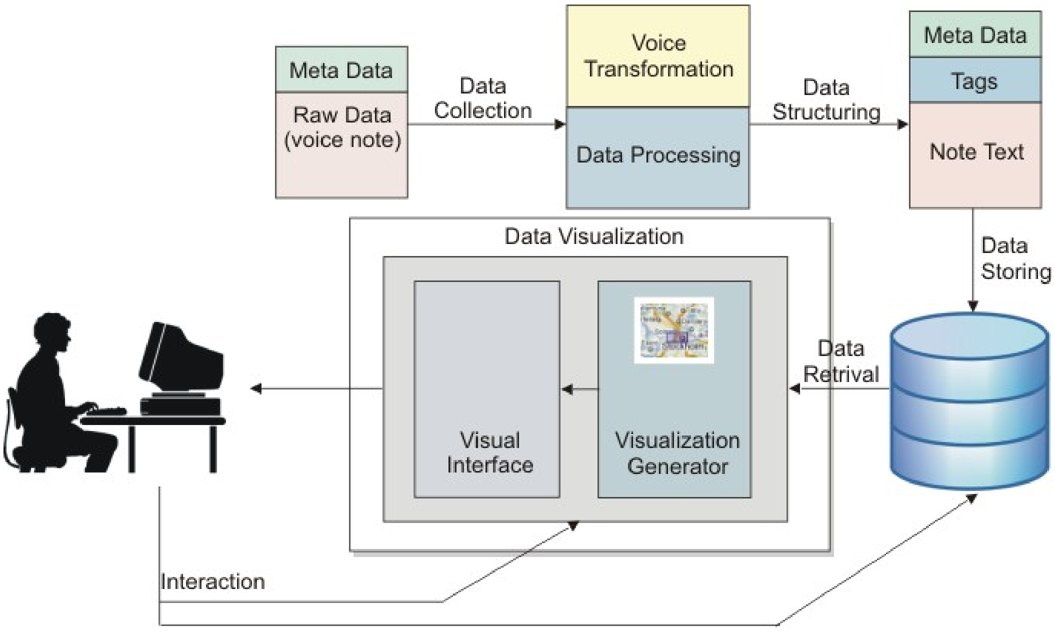


Figure 10: Architecture

The raw voice is transformed into a text and later on the text is processed, respectively the text is mined, and the most frequent tags (keywords) are extracted. After this, we will have a note transformed into text which will additionally hold the most frequent tags and some meta data. These information will be stored into database (possible view of database tables is shown in Figure 11).

A user can interact with the visualization in two ways: he/she can use various dynamic queries to filter out the data in database (he/she will interact with database), or he/she can use various visualization techniques to interact with the visualization itself through the GUI (examples of such of techniques are zooming, panning, semantic zooming, etc.).

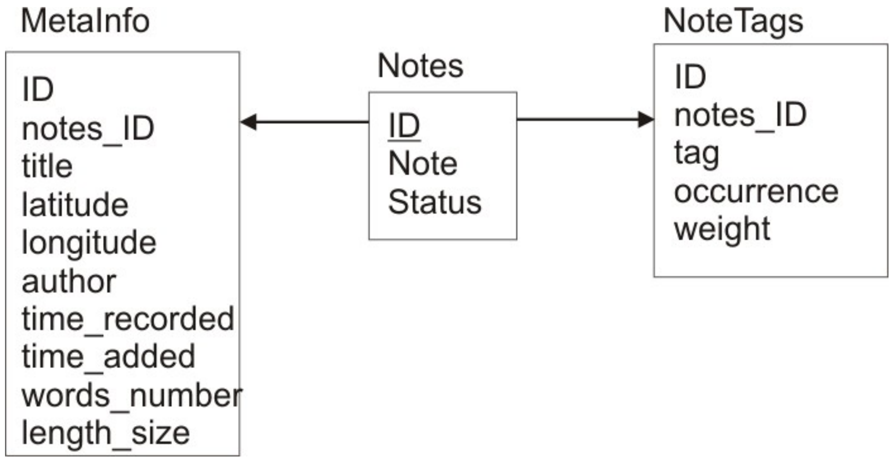


Figure 11: Database

5.4 Summary

In this chapter, we defined the functional and nonfunctional requirements which lead us in the design of the use cases. Therefor, we presented the architecture we are proposing and explained its different components. A more detailed explanation about the implementation of these components can be found on Chapter 8. At this point, we have an idea about how the back-end of the prototype looks like. In the next chapter, we will discuss the front-end of the prototype.

6 The GNV System: Conceptual Ideas & Design

In this chapter we will discuss the initial ideas and visual metaphors we came up with at the very beginning of our investigation. These ideas helped us to shape GUI and visualization interaction techniques that will be explained in subsection 6.2.

6.1 Initial Ideas and Visual Metaphors

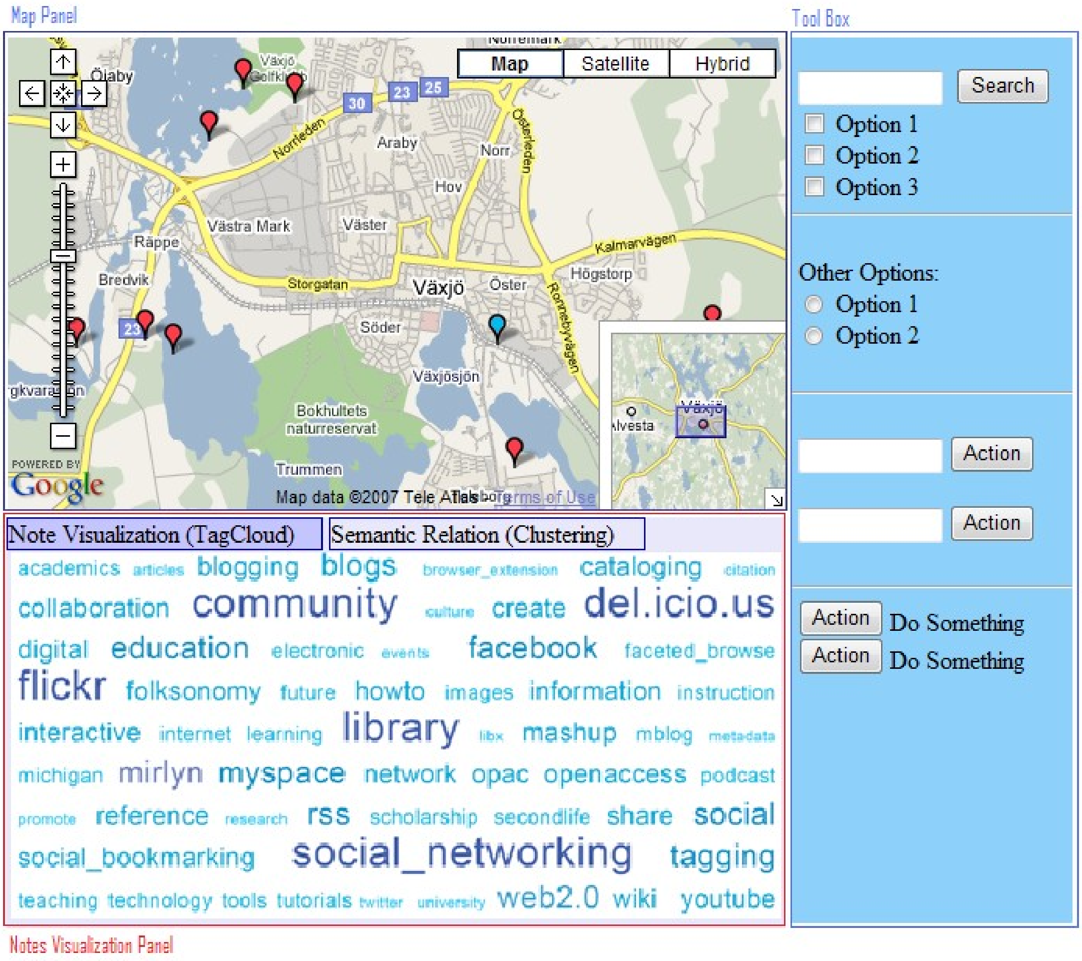

Map Panel
Tool Box
Map
Satellite
Hybrid
Växjö
Öster
Söder
Araby
Norr
Hov
Väster
Västra Mark
Räppe
Bredvik
Högstorp
Kalmarvägen
Storgatan
Växjösjön
Bokhultets naturreservat
Trummen
POWERED BY Google
Map data ©2007 Tele Atlas
Terms of Use
Search
□ Option 1
□ Option 2
□ Option 3
Other Options:
○ Option 1
○ Option 2
Action
Action
Action Do Something
Action Do Something
Note Visualization (TagCloud)
Semantic Relation (Clustering)
academics articles blogging blogs browser_extension cataloging citation
collaboration community culture create del.icio.us
digital education electronic events facebook faceted_browse
flickr folksonomy future howto images information instruction
interactive internet learning library libx mashup mblog metadata
michigan mirlyn myspace network opac openaccess podcast
promote reference research rss scholarship secondlife share social
social_bookmarking social_networking tagging
teaching technology tools tutorials twitter university web2.0 wiki youtube
Notes Visualization Panel

Figure 12: GUI mock up

After the literature analysis and review, we spent some time in defining our visualization metaphor. We concluded that our GUI should consist of three panels in order to facilitate the visualization of GeoAudio notes. One of the panels will be for visualizing the position of the notes, the other panel will be for visualizing the content of the notes and visualizing the semantic relation between them, and the third panel will hold the tools for interacting with the visualization and the application. Because visualizations also are a part of the GUI (since users can perform various interactions with our application), we decided to analyze both GUI and visualization metaphor at the same time. Having in mind the central questions of our thesis and the research we conducted so far, we came up with the visual metaphor depicted in Figure 12 (note that this figure is not the actual prototype, it is only a drawing of the metaphor). As seen in the Figure 12, we have split the GUI in three main parts:

1. Map Panel
2. Notes Visualization Panel
3. Tool Box

The different components will be described in details in the coming sections.

6.1.1 Map Panel

The *Map Panel* displays the notes in a map. It is based on a Google Map [7]. Users will be able to see the geographical position where the note has been recorded in this panel (Figure 12). It should be kept in mind that this was only an initial metaphor. The notes are represented in different metaphor than the one presented in this mock up (see the explanation about the note metaphor in Subsection 7.1). The Google Map API provides various interaction techniques and tools (e.g., zooming, overview technique with the help of small map in the right-down corner, etc.). This is of course an advantage, but our work showed the need to enhance or introduce new techniques. Google API offers also a way to build our own controls. This helped us in our implementation. Also, the Google API provides very good integration and support for SVG and KML, giving us a possibility of designing new visual object (representations in a map). We should keep in mind changes made in this panel should reflect the other panel, and some brushing and focus techniques need to be used to realize this. We depicted color brushing in this mock up. The blue 'pin' shows the selected note.

As mentioned in the previous paragraph, the Map Panel displays the Google Maps. The notes will be displayed by bitmaps in the exact position where they were created. As shown in Figure 13, Map Panel interacts with Notes Visualization Panel and with Tool Box. When the user selects a note in the map, tags of the selected note will be displayed in the Notes Visualization Panel. Concurrently the same note in the Clustering Panel will be highlighted. The selection of notes will also have an effect on the Tool Box where the information about the selected note will be displayed (e.g., note name, author name, the date when the note was created, location, etc).

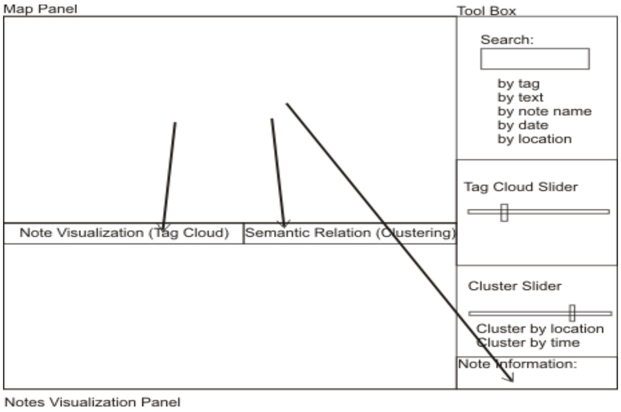


Figure 13: Map Panel Interactions

6.1.2 Notes Visualization Panel

This panel holds three tabs for notes visualization. The first tab (*Note Visualization*) will display the actual note visualization. At this phase, the content of a single note will be displayed using a tag cloud approach (see Figure 12). The issue whether we will use the tag cloud approach or not, or if we will enhance this approach will be discussed later in this report. The second tab (*Semantic Relation (Clustering)*), will visualize the semantic relation of notes in time and space by showing the clusters of notes with similar semantic content as illustrated in Figure 14. The third tab present a list of search results.

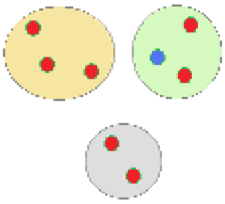


Figure 14: Note Clustering; rough metaphor

In this way we can group notes and see possible outliers. We can discover what other notes belong to the same cluster as the note we selected. Thus we can find the notes with similar content in easier way. Clustering properties and parameters should be customizable and could be changed from *Tool Box*. An initial rough view of this visualization metaphor is depicted in Figure 14. This panel is possible to provide additional tabs with visualizations. We displayed the list of search results in this panel as a separate tab *(Search Result Tab)*. One other possibility is to provide the option for the user to select a number of notes and to view their content and compare them by himself rather than letting automatic clustering and mining do the job. We are proposing a visual metaphor to support this process and it is depicted in Figure 15:

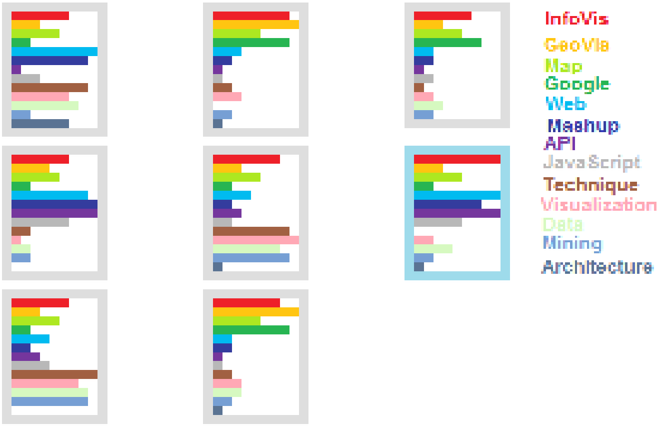


Figure 15: Multiple Note Comparison visualization

Each box represents a document, and lines inside the boxes represent the tags found in the document. The tag list with corresponding colors is located at the right hand side of the panel. The line length represents the weight factor of the tag. The document outlined with light blue color is the one selected in the map. Here, we can see some of the matching patterns between notes just by looking at the visualization.

We mentioned earlier that the Notes Visualization Panel contains three tabs: the *Note Visualization* tab, the *Semantic Relation (Clustering)* tab, and the *Search Result* tab (which is not depicted in Figure 16, because it will complicate the simplicity of the figure and has the same function as the Clustering Tab). This panel also interacts with both other panels. As described above, one of the interactions occurs when the note in the Map Panel is selected and the tags of that note are displayed in the Note Visualization tab. Another kind of interaction with the Map Panel is when a user selects a note in the clustering tab. At this point, the same note will be highlighted on the Map Panel, and the Tool Box panel will show the information about the selected note (e.g., note name, author name, the date when the note was created, location, etc). Exactly the same thing will happen when the user selects a note name in the *Search Result* tab. Other interaction happens when the user selects a tag in the *Note Visualization* Panel, and that tag will be placed in the Search box. After the user presses the search button, the system will return all the notes that contains that tag that was selected previously. Figure 16 shows the interactions that occur from this panel.

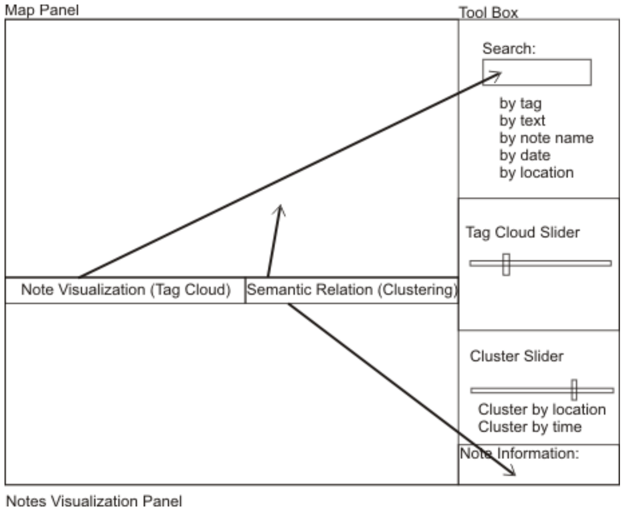


Figure 16: Notes Visualization Panel Interactions

6.1.3 Tool Box

The *Tool Box* is a panel that holds all the tools that are necessary for the interaction with the visualization and the application in general. It holds familiar items like in other well known applications.

From Figure 17 we can see that Tool Box provides the following functionalities such a search, slider for tag clouds, slider for clustering, etc. The Tool Box will interact with the Map Panel when the user initiates a search. If the user has made a search based on tags, the Map Panel will display only the notes that will contain the searched tag. Same notes will be displayed in the Note Visualization Panel too. The Tool Box panel will contain two sliders that will interact with both tabs in the Notes Visualization Panel. One of the sliders will be for tag clouds. The default number of tags displayed in the Note Visualization tab is 50, but users can change this number by using the tag cluster slider. The other slider will be for changing the number of clusters in the Clustering Tab, and choosing the value of the threshold which is by default 2.5. The threshold value slider determines the minimum weight of the tags to be clustered. Additionally, this panel will contain options about the date, when the user will want to cluster or to search for notes of a specific date. When the user selects a note (from Map Panel or Clustering Tab) the information of the selected note will be displayed in the Tool Box.

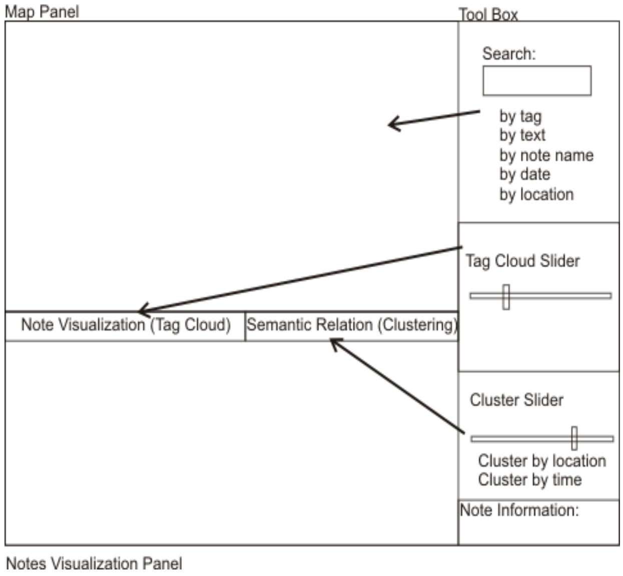


Figure 17: Tool Box Interactions

6.2 Summary

In this chapter we have presented some conceptual ideas and design issues which help us to design our prototype. We defined different panels of the GUI and presented some mock ups of different visual metaphors. Additionally, we proposed the interactions between these panels. However, these conceptual ideas should be taken to the next level. This means that we should define more precisely what kind of visualization and interaction techniques are we going to use for the GNV System. The next chapter discusses these issues.

7 Visualization and Interaction Techniques

In contrast to information visualization displaying any abstract data, geovisualization deals specifically with geospatial data, i.e., data that contains georeferences [14]. For showing this kind of data, special methods and tools are needed. In this chapter we will discuss visualization and interaction techniques used in the GNV System. First, we will present some general ideas about visualization techniques and continue with examples of these techniques used in our prototype. Similarly, we will continue by presenting some general ideas about the interaction techniques and giving examples where these techniques are used in the GNV System.

7.1 Visualization Techniques

Visualizing a huge amount of dataset introduces some challenges. One of them is to show an complete overview of dataset while providing details for specific object in dataset. Design of visual metaphor is another challenge which we have to deal. The design of metaphors should be in the way that it provides an answer the the first challenge we mentioned as well as have aesthetic appearance.

Therefore, we have explored algorithms for computational exploration of such text mining in order to facilitate the visualization of text content. But, data mining techniques have limited pattern interpretation abilities [14]. We should consider integration of computational methods and visual approaches to combine their advantages which may lead to further advances in the exploration of the geospatial data [14]. Furthermore, we have used clustering techniques which helped us understand the relation between various data and making possible to identify outliers.

In the following subsections we will present our visualization approaches for the problems of visualizing notes in the map, visualizing content of the notes (tag clouds), and visualizing their semantical relation (clustering).

7.1.1 Visualizing Notes in the Map

One of the difficulties that we have encountered during our research was how to visualize and represent notes in the map. The visualization process usually consists of three [13] steps:

1. The collection of data to be visualized.
2. The visual display of data.
3. The human perception of the visualization.

As we will need to display some details about the notes, such as time creation, number of words that the note contains, etc., we have decided to use bitmaps for presenting the notes in the map. Figure 18 shows how a note marker looks like.

As we can see from Figure 18, the marker shows three different details about the content of the note: time created, number of words, and duration.

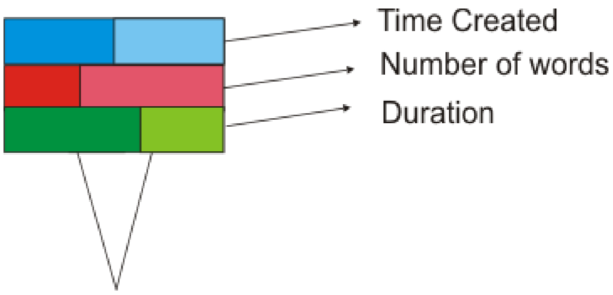


Figure 18: Note Marker

Time created

Considering the fact that creation time of the note will play a role in knowledge discovering processes, we decided to visualize this attribute of the note on the map. An example of such a visualization is illustrated in Figure 18, where the first stripe (blue color) from above represents the creation time, respectively the age of the note. The stripe should be regarded as a bar chart. The size of darker blue color grows from left to right, depending on the age of the note. If the dark color width is bigger, older the note is, and vice versa.

Number of words

The second rectangle of the note marker will show the number of words of the note. By this mean, the user will know which documents are larger and which are smaller. Color denotation will be similar like previous example, meaning larger number of the words in a note, the dark color is bigger. From our example in Figure 18, we can understand that the document is not too large.

Duration

The third rectangle will visualize the length of the note, respectively the duration of the recording. The visual representation follows the same method as the above mentioned examples. If the width of th lighter colored stripe is bigger, it means that the length of the note is short.

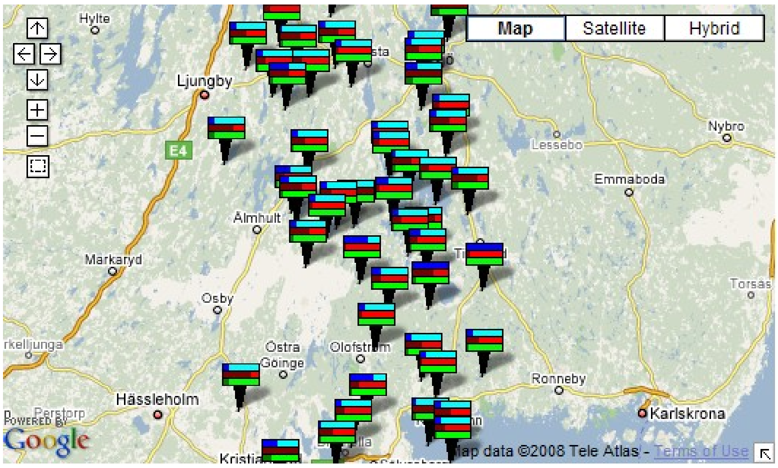
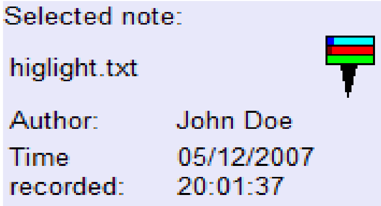


Figure 19: Notes Visualized in the Map Panel

Figure 19 shows how the notes in the map are visualized. So, for each note one marker will be placed in the map at the exact place where the note was created by the author. By just viewing the marker of the note, users will get some information about the note, for example, they will know whether the note is new or old, or if the note has a large text or short. More detailed information about the note users will get if they select the note, where information about that note will be displayed on the bottom of the Tool Box Panel space (Figure 20).



Selected note:

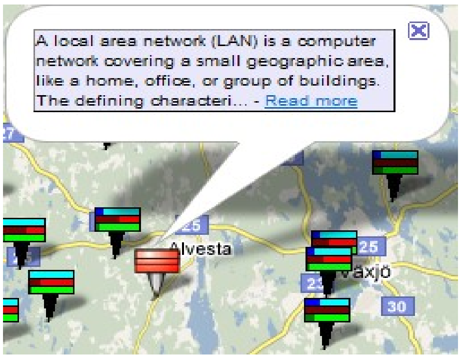
higlight.txt

Author: John Doe

Time recorded: 05/12/2007 20:01:37

Figure 20: Information for Selected Note

When users click twice on the note in the map, the Information Window for that note will appear (Figure 21). Here, a large prefix of the document text will be displayed. If the user wants to read the entire text of the document, then he/she should click in the link "Read more" that is at the end of the text. Then the entire document will appear in a new window. If the user want to listen to the original audio note, they should click on the audio link (which is not implemented in our prototype).

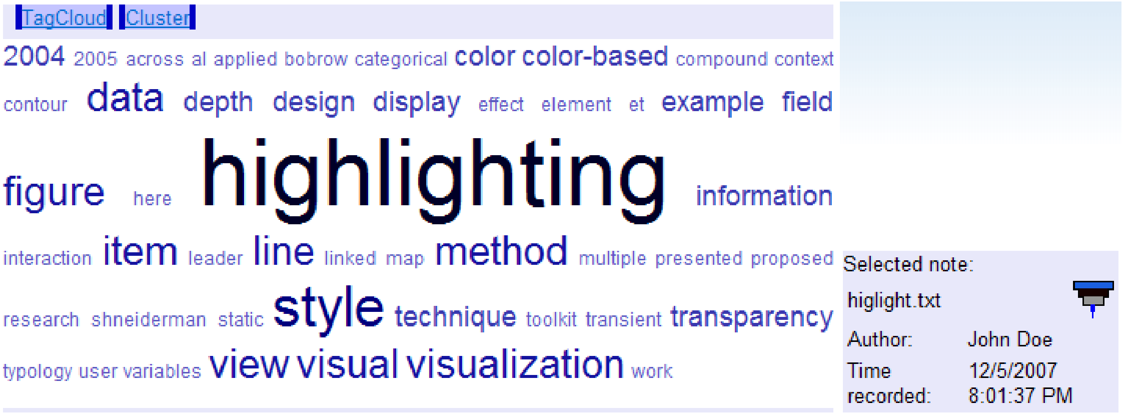


A local area network (LAN) is a computer network covering a small geographic area, like a home, office, or group of buildings. The defining characteri... - Read more

Figure 21: Information Windows

7.1.2 Tag Clouds

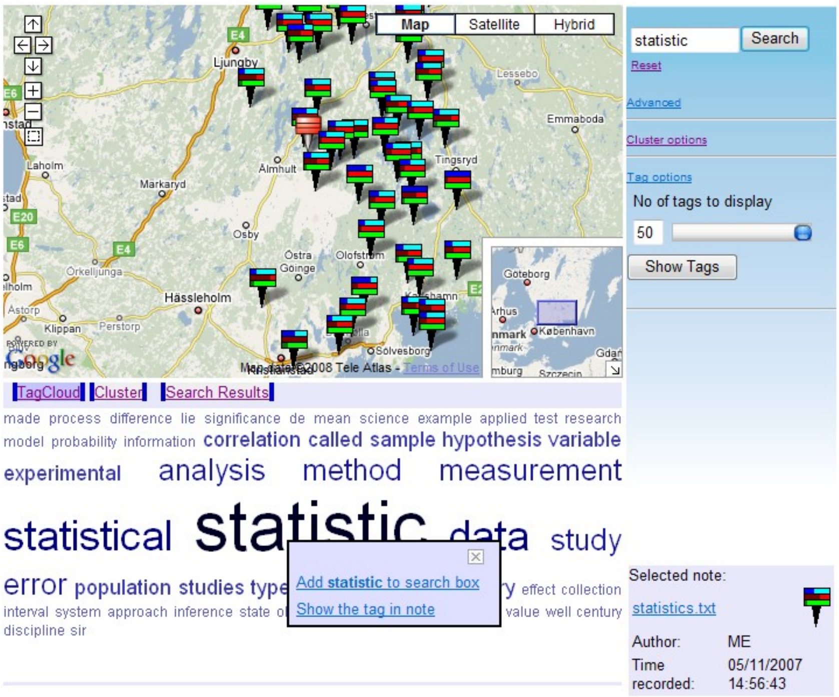
One way for users to get some insights about the note is to select the note in the map and, as seen in Figure 22 , a tag cloud is used to visualize the content of the note by showing the most frequently used tags. Each tag size is generated by using a weight (number of times the tag appears in the document).


TagCloud
Cluster
2004 2005 across al applied bobrow categorical color color-based compound context
contour data depth design display effect element et example field
figure here highlighting information
interaction item leader line linked map method multiple presented proposed
research shneiderman static style technique toolkit transient transparency
typology user variables view visual visualization work

Selected note:
higlight.txt
Author: John Doe
Time recorded: 12/5/2007 8:01:37 PM

Figure 22: Tag Cloud in the Notes Visualization Panel

Users can interact with individual tags (Figure 23) by clicking on them, and each tag that is clicked will be placed in the search box, where after pressing the search button, the system will search for notes that contains that tag. Additionally, users can choose the number of the tags that they want to see, by moving a slider on the Tool Box Panel. As by default, the number of tags that will be shown in the Note Visualization Panel is 50.


Map
Satellite
Hybrid
Ljungby
Lessebo
Emmaboda
Almhult
Tingsryd
Laholm
Markaryd
Osby
Ostra Goinge
Olofström
Orkelljunga
Hässleholm
Karlshamn
Astorp
Klippan
Perstorp
Sölvesborg
Göteborg
Århus
København
Map data ©2008 Tele Atlas - Terms of Use
POWERED BY Google
Search
Reset
Advanced
Cluster options
Tag options
No of tags to display
Show Tags
TagCloud
Cluster
Search Results
made process difference lie significance de mean science example applied test research
model probability information correlation called sample hypothesis variable
experimental analysis method measurement
statistical statistic data study
error population studies type y effect collection
interval system approach inference state o value well century
discipline sir
Add statistic to search box
Show the tag in note

statistic
50

Selected note:
statistics.txt
Author: ME
Time recorded: 05/11/2007 14:56:43

Figure 23: Searching by selecting a tag

7.1.3 Clustering

Clustering is a technique devoted to find and describe cohesive or homogeneous chunks in data, the clusters [16]. It is a key area in data mining and knowledge discovery, which are activities towards finding non-trivial or hidden patterns in data collected in databases [16]. We will use clustering in order to separate a set of notes in groups by similarity of their content. There are certain frameworks in the way a set of clusters are defined in relation to each other. Clusters can be defined so that each entity is a member of one and only one cluster, or so that entities belong to different clusters with different probabilities or degrees of membership, or that two entities can belong to multiple clusters without restriction (e.g., so-called overlapping clusters) [17]. In our case, we chose an approach that defines each entity as a member of one and only one cluster.

Clustering approaches can be found hundred years back in different fields. As described by Mirkin's [16], clustering is needed for:

1. structuring the data as a set of groups of similar objects,
2. description of clusters in terms of features,
3. association,
4. generalization, and
5. visualization of cluster structures as visual images.

There are different algorithms used for clustering such as:

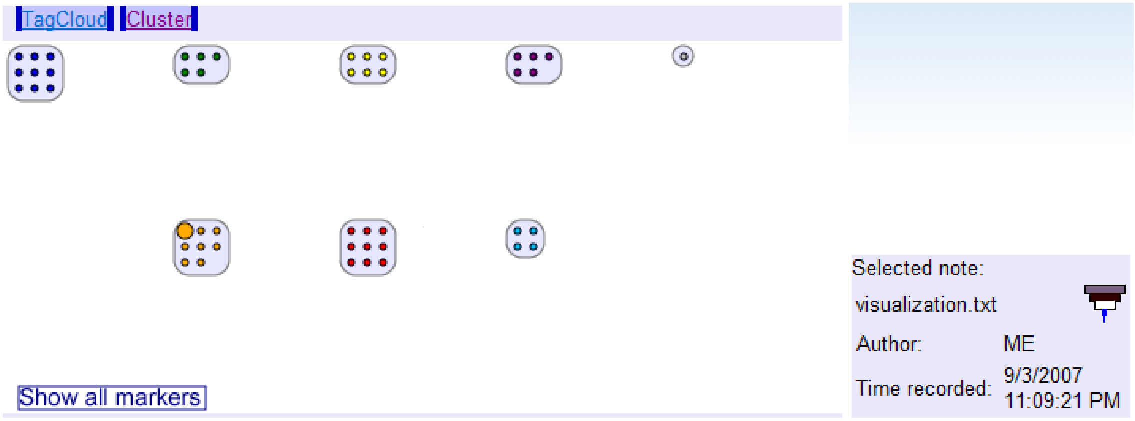
- ➔ **K-Means** [16] – is the algorithm which assigns randomly centroids for each cluster.
- ➔ **Hierarchical Clustering** [16] – is the algorithm that presents data in the hierarchical form. A hierarchical algorithm can be presented in two forms: agglomerative (bottom-up) and divisive (top-down).

For clustering our unstructured data, we have decided to use the K-Means algorithm. K-Means' advantage is the effectiveness in speed and the computational effectiveness in memory. A further advantage is its simplicity. The main drawback is that it does not return the same results each time, since that depends on the initial random assignment. Considering the tradeoff between advantages and disadvantages of these algorithms, we choose K-Mean algorithm for our application. The simplicity of this algorithm is the key reason for this decision. We have done this since the focus of our thesis is the visualization part, instead of developing a good clustering.

The K-Means algorithm works in the way that we firstly need to determine the number of clusters by assigning random centroids in each cluster. A “Centroid is a multidimensional vector minimizing the summary distance to cluster's elements. If the distance is Euclidean squared, the centroid is equal to the center of gravity of the cluster” [16]. These are the steps that occurs during this algorithm:

- decide the number of clusters k,
- generate randomly k clusters and compute the cluster centers, and
- recompute the cluster centers.

Figure 24, illustrates an example of how the clusters look like in our application. The application supports the clustering up to 10 initial clusters. Notes are placed into “cluster circles”. In our case clustering is based on tags. Notes that are in same cluster have similar tags, thus have semantic relationships between them. Some interactions are possible directly with this visualization and they are explained in next section.



Selected note:
visualization.txt

Author: ME

Time recorded: 9/3/2007 11:09:21 PM

Figure 24: Clustering visualization in the Note Visualization Panel

For clustering the notes of specific area, users should use the control that is placed on the left upper corner of the Map Panel. In Figure 25, we have circled the control with black color. After the user activates the control for selection, the next step is to select an area. The selection of area is done with two diagonal clicks on the map. In Figure 25, we have circled the first click with a blue color, and then we have circled the second click with a red color.

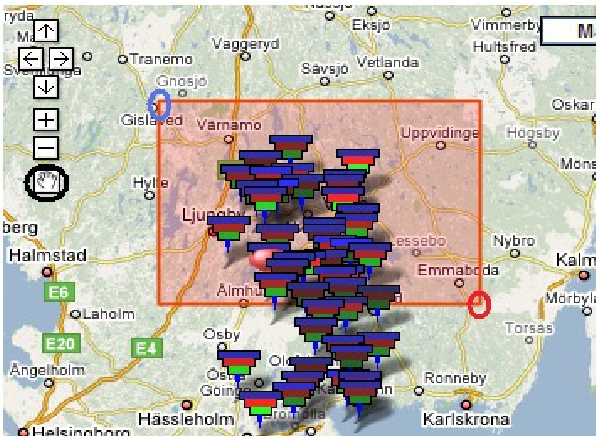


Figure 25: Selecting an area in the map

Besides using the spatial criteria for clustering, user can additionally use temporal criteria. This can be done by selecting a time span of notes creation date by using the Tool Box Panel calendar tool (see Figure 26). This will be explained later in this chapter.

7.2 Interaction Techniques

Since we have to deal with a large amount of data, we will use several standard visual interaction techniques: zoomable, overview+detail, and focus+context interfaces.

- *Zoomable interfaces* allow users access information by panning and zooming [10].

- *Overview+Detail interfaces* show an overview of the entire information space together with a detailed view of part of the contents in separate windows/display areas[11].
- *Focus+Context interfaces* join the the current focus of attention and its context [9]. The origin of focus+context visualization is a non-interactive distortion-based techniques for visualizing of map data [12].

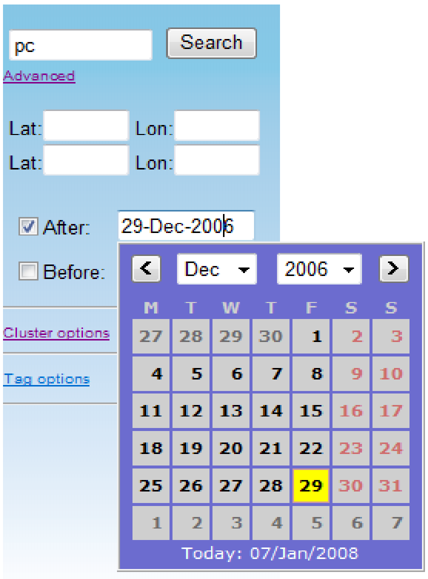
Using visual interaction techniques, such as zooming ,helps to visualize large data sets where items are often overlapped, additionally magnifying small symbols. But this is done at the cost of losing the overall view of the data which is also very important [14].

Interaction between the the human and visualization is very important. Using good interaction techniques makes it easier for humans to communicate and to better understand a visualization. Interaction is the way in which communication between the user(s) and the machine(s), takes place [19]. Amongst communicating partners, three basic human-computer interaction metaphors can be distinguished [19, 20,21]:

1. *Navigation* which enables users to browse through a dataset, similarly to navigating a (three dimensional) space;
2. *Direct Manipulation* in which the machine is viewed as a passive collection of objects waiting to be manipulated by the users;
3. *Human Interaction* in which the communicating partner is perceived through the interface having human characteristics or being human-like.

7.2.1 Navigation

In our application, users can navigate through the data by using different options. One of the options is when users decide to search for the data with specific content, in specific area or date, or without using the space criteria or time. This can be seen in the Figure 26.



pc

☑ After: 29-Dec-2006

□ Before:

Dec 2006

M	T	W	T	F	S	S
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Today: 07/Jan/2008

Figure 26: Navigation through search options

Another option is when the user navigates through the notes in the Map Panel, from the Clustering Tab, in the Note Visualization Panel. If we click on the note in the Clustering Tab, and that note is not in current map view (it is outside the boundaries of the current map zoom and position level), the note will appear in the center of Map Panel. We use brushing techniques on selected notes so the user knows which note is selected. Thus, when the user clicks the note in the clustering tab, it will become bigger and the same note in the Map Panel will change the color, so the user knows it is the same note (Figure 27).

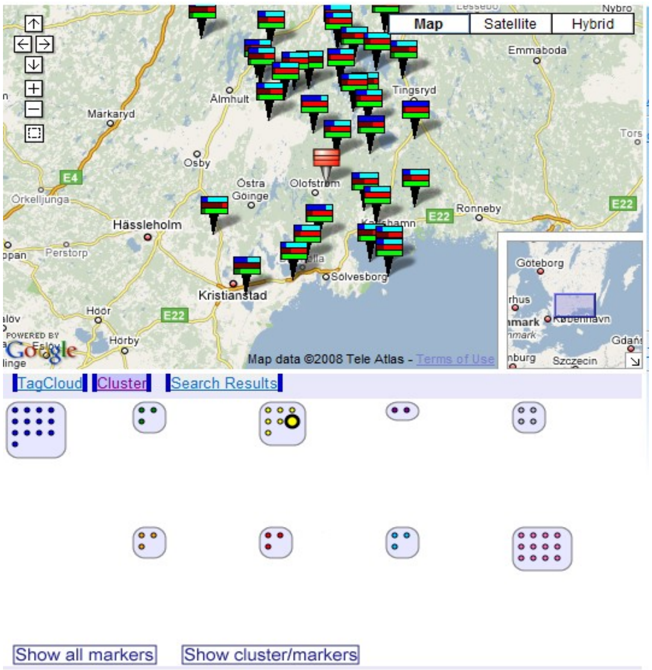


Figure 27: Navigating through notes in the Clustering Tab

The same will happen when the user will click on a desired cluster in the Clustering Tab. The Map Panel will automatically zoom and pan to the notes of that cluster. Only the notes of the selected clusters will be shown in the Map Panel. In this way, a user can notice the spatial concentration of the notes semantically related to each other, cp. Figure 28. Users also navigate with the Map Panel by zooming in or out, or moving in desired position in the map by using Google Map controls. These controls can be seen in the upper left corner in the Map Panel, cp. Figure 27 and Figure 28. The Google Map API also a offers ready to use overview control, which helps the navigation of the map. This control is placed in the bottom right corner of the map, and can be minimized, as seen in Figure 28 where it is maximized and minimized in Figure 29.

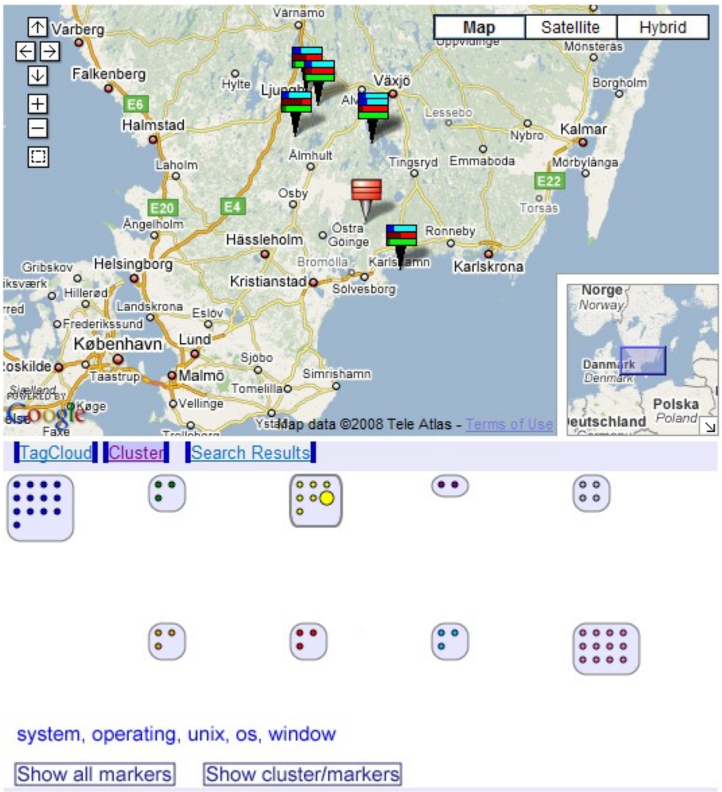


Figure 28: Navigating by selecting a cluster

7.2.2 Adjustments

When the user has finished navigating and has the desired object in front of him, then he/she can start to manipulate that object. In our application, the user can interact with the data sets by:

- selecting them and viewing additional information about that data as shown in Figure 23;
- selecting a number of notes in the Map Panel (Figure 29) and clustering them;
- adding time as a criteria for searching or clustering (Figure 26);
- choosing the number of tags that the user want to see in the tag cloud options of the Tool Box (Figure 30);
- choosing the number of centroids or the weight of threshold in clustering options of the Tool Box (Figure 30);
- choosing which notes of which cluster should be visualized in the map (Figure 28).

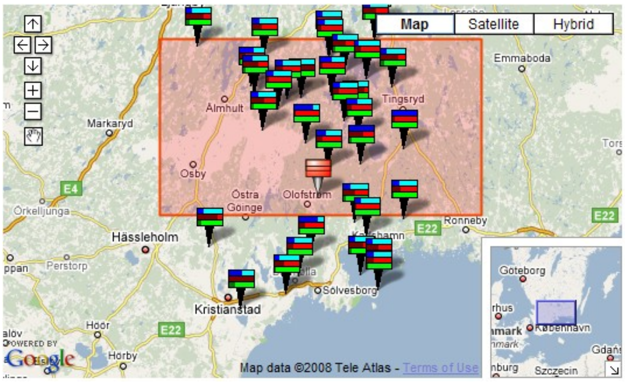
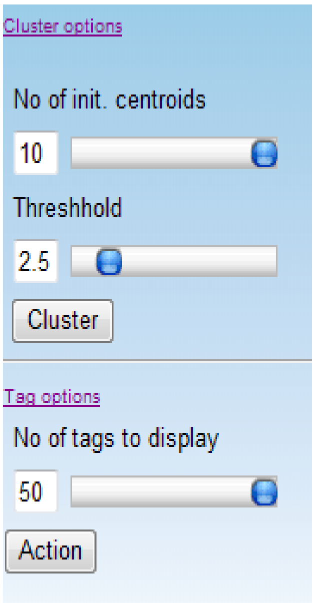


Figure 29: Spatial selection of the notes



Cluster options

No of init. centroids

10

Threshhold

2.5

Cluster

Tag options

No of tags to display

50

Action

Figure 30: Tool Box - Manipulating through data with Clustering and Tag Options

7.2.3 Human Interaction

The distinction between the direct manipulation metaphor and the human interaction in which the communicating partner is perceived through the interface having human characteristics, or being human-like [19], has been the subject of considerable debate in the literature [22]. “Supporters of the direct manipulation metaphor praises the fact that such interfaces are typically made very transparent by displaying the options available to the user on a screen” [19]. Our application uses these techniques when it clusters the notes based on their semantically relations. Also, when the notes are visualized in the map, their color will be dependent of the meta information that the note has, thus it can not be manipulated by the user.

7.3 Summary

The main points of this chapter were to explain the different visualization and interaction techniques that we have used for our prototype.

In connection to FR5, FR12 discussed in Chapter 6, we have explained the techniques that we have used to visualize the notes in the map. Thereafter, we have explained the techniques used to visualize the content of the notes which is connected with the FR6. Finally, in connection with functional requirements FR7, FR8, FR9, we have discussed the techniques used to show the semantic relations between the notes in time and space.

Except visualization techniques, we have discussed the interaction techniques that were used for our prototype implementation. Functional requirements 12, 13, 14, 15, 16 from Chapter 6 were explained in the Section 7.2 of this chapter.

8 Implementation Aspects

In this chapter, we will present a short overview of the technologies, programing and scripting languages used to implement the application, and continue with more detailed explanation of the implementation by presenting class design diagrams and class descriptions.

Before we start the explanation, we would like to mention once again that our work is focused only on the visualization part of the prototype. This means we will not cover the implementation of the voice transformation to text.

8.1 Technologies and Tools

We used different kind of technologies to implement the different components of our system. It is a web based application which uses several scripting and programing languages (ASP.NET with C#, Google Map API, Javascript, AJAX and ECMA script used in SVG), MS SQL2005 database, and SVG to represent the clusters. The system can be described as a three tier client server mashup [7] (look at Figure 9). We have an IIS web server which runs ASP.NET AJAX and MS SQL2005 database server on one side, Google Map API which is outside of our system, and client with a browser running Javascript on the other side. All these entities communicate using different calls and requests.

A lot of processing power is spent on the server side (eg. the clustering process), but considerable amount of processing happens in client (browser using Javascript/ECMA script, especially for animating and interaction with the visualization). Next we will give a short explanation about some of the technologies mentioned in previous paragraph.

AJAX – stands for Asynchronous Javascript and XML. By using AJAX, Javascript can communicate directly to the server without reloading the page. This is done through a Javascript XMLHttpRequest object. It uses asynchronous data transfer between the browser and web server, i.e. it allows web pages to request small bits of information instead of the whole pages. In this way, we can create faster, smaller, and more user friendly applications [31].

ASP.NET – is Microsoft's free technology that allows programmers to create dynamic web application. It is a server side scripting technology that runs inside IIS [36]. ASP.NET provides a possibility to use AJAX with ASP.NET [32].

Google Map API - "lets you embed Google Maps in your own web pages with JavaScript. The API provides a number of utilities for manipulating maps (just like on the http://maps.google.com web page) and adding content to the map through a variety of services, allowing you to create robust maps applications on your website. "[33].

Javascript (ECMA script) – is a client side scripting language designed to add interactivity to HTML pages. It is interpreted language, which means the script is executed without preliminary compilation. JavaScript's official name is "ECMAScript". [34]

SVG - "SVG is a platform for two-dimensional graphics. It has two parts: an XML-based file format and a programming API for graphical applications. Key features include shapes, text and embedded raster graphics, with many different painting styles. It supports scripting through languages such as ECMAScript and has comprehensive support for animation."[35].

8.2 Software Development

In this section, we will present more details about our implementation. We will show class view diagrams and give explanations about the fields, properties, and methods of the class.

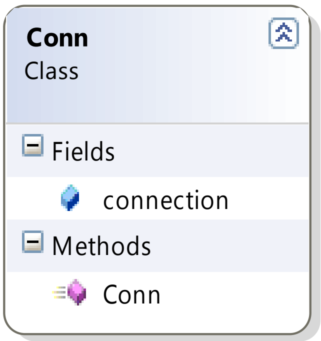


Figure 31: Conn - class design diagram

Class	**Conn**
Description	This class is used to establish the connection with database.
Fields	*connection*: this field has type `SqlConnection`.
Methods	*Conn()*: this method is the default constructor of this class. Here, the connection is created and opened.

Table 7: Conn class

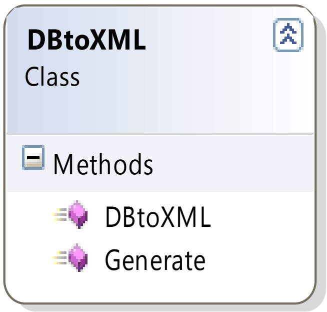


Figure 32: DBtoXML - class design diagram

Class	**DbtoXML**
Description	This class gets notes from the database and returns them in XML format, so they can be read using Javascript and displayed in the map.
Methods	*DbtoXML()*: is a default constructor and has no particular function. *Generate()*: returns the XML document as string. It takes one string argument, which holds the SQL query.

Output example:	`<?xml version="1.0" encoding="utf-8" ?>`
	`<markers>`
	`<marker lat="56.141000" lng="14.534000" html="A Voice over Internet Protocol (VoIP) is a protocol optimized for transmission of voice through the Internet or other packet switched networks. VoIP i" title="voip.txt" ID="71" />`
	`<marker lat="56.583000" lng="14.256000" html="Visualization is any technique for creating images, diagrams, or animations to communicate a message. Visualization through visual imagery has been an" title="visualization.txt" ID="72" />`
	`</markers>`

Table 8: DbtoXML class

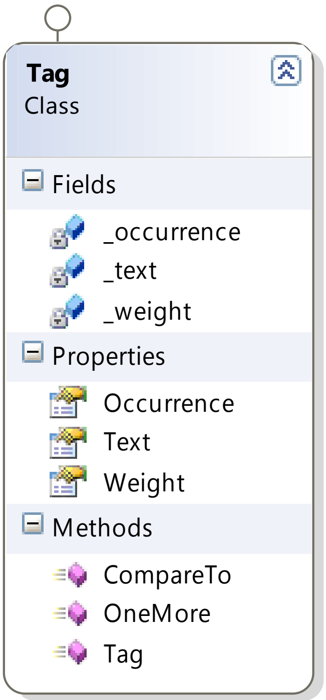


Figure 33: Tag - class design diagram

Class	**Tag**
Description	This class holds the information about each tag: the text of the tag, the number of occurrences, and the weight of the tag calculated depending on the number of tags desired as a result of mining.
Fields	*_occurrence*: is used to hold the number of repetitions of the tag in the document. The field is of `integer` type. *_text*: holds the actual (text) tag. The type of this field is `string`. *_weight*: holds the weight of the tag calculated depending on the total occurrences of the set of tags selected according to their highest number of occurrences in the text. The type of the field is `double`.
Properties	Are similar to the fields and are actually used to set and get the values of the fields, so there is no need to describe them separately. Check the fields for their corresponding meaning.
Methods	*CompareTo()*: method is used to implement *IComparable* interface. This interface is built in interface for C# and is used to help in comparison of classes. In our case we have to do sorting of the list of tags in the *TagMiner* class so we needed to implement the interface so the compiler knows what should be compared in order the sorting should work. Here, it should compare the *Occurrence* of the tags. *OneMore()*: is used to increment the number of *Occurrence* in case same tag is found again in the text. *Tag()*: is a class constructor.

Table 9: Tag class

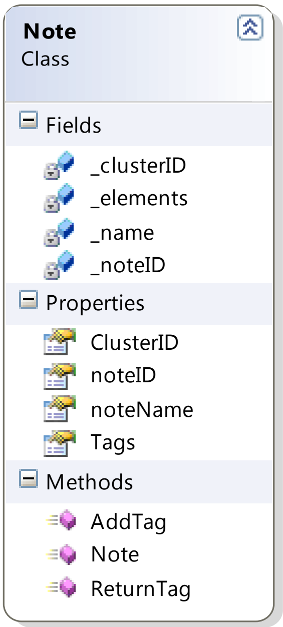


Figure 34: Note - class design diagram

Class	**Note**
Description	*Note* holds the information about the note, such as the tags of the note, name, database ID.
Fields	*_clusterID*: holds the cluster ID of the note. *_elments*: holds an array of the tags belonging to the corresponding note. This is an `ArrayList` so it stores *Tag* objects in it. *_name*: stores the name of the note. *_noteID*: holds the ID of the note retrieved from the database.
Properties	Are similar to the fields and are actually used to set and get the values of the fields, so there is no need to describe them separately. Check fields for their corresponding meaning and you will see below only the properties that somewhat different naming than those of the fields. • *Tags*: returns the `ArrayList` of tags belonging to specific element. This property is read only and can not be changed. (to add a tag to the note we have to use a function/method which is described below)
Methods	*AddTag()*: is used to add a tag to *_elments* field (that is to add a tag to a note) *Note()*: is a default constructor of the class. *ReturnTag()*: is used to return a `tag` from a specific position on the array. It uses an `integer` input argument which shows the index of wanted element(tag).

Table 10: Note class

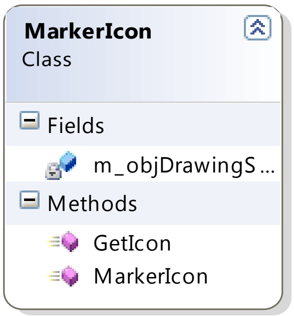


Figure 35: MarkerIcon – class design diagram

Class	**MarkerIcon**
Description	This class generates the marker icon to be displayed in the map. It uses number of input parameters to render the image (icon) of the note.
Fields	*m_objDrawingSurface*: has the type `System.Drawing.Bitmap` type. It will hold the rendered image to be returned.
Methods	*GetIcon()*: returns a bitmap object (representing icon of the specific note). This method uses three input arguments: *timeVal* which is `integer` value representing how old is the note in the number of days, *noWords* – an `integer` value representing the number of words the note has, and *duration* contains the duration of the note expressed in seconds) *MarkerIcon()*: is the default constructor of the class.

Table 11: MarkerIcon class

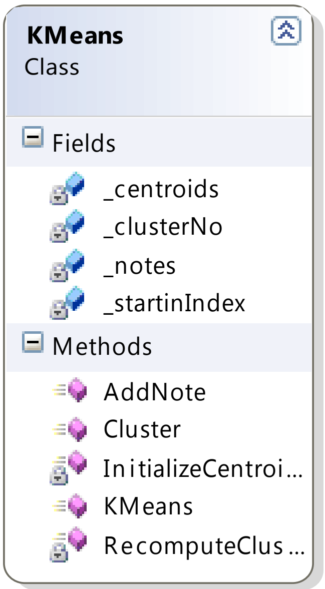


Figure 36: KMeans - class design diagram

Class	Kmeans
Description	This class is used to cluster the notes using tags as the clustering parameters. It uses the k-means clustering algorithm. For more information and better understanding review literature for k-means clustering [16][17].
Fields	_*centroids*: holds an `ArrayList` of centroids (cluster centers). _*clusterNo*: holds the number of initial desired clusters. _*notes*: holds an `ArrayList` of notes to be clustered. _*startingIndex*: holds the starting index of the cluster ID's. This is used in case we want to make recursive clustering of clusters (for future improvements).
Methods	*AddNote()*: adds a note to the _notes. It takes one object attribute, which should be an Note object with added Tag's on it. *Cluster()*: this function does the clustering after the notes have been added. It creates centroid model by combining it from all possible tags from all notes to be clustered. It calls the *InitializaCentroids()* and *RecomputeClusterCentroids()*. InitializaCentroids(): is used for initialization of centroids by taking random note attributes to be a representative of the corresponding cluster. *Kmeans()*: is the calls constructor. It takes one argument of `integer` value representing the number of initial desired

	clusters. *RecomputeClusterCentroids()*: after each cycle of the k-means algorithm it recomputes the new values for cluster centroids.

Table 12: Kmeans class

8.3 Summary

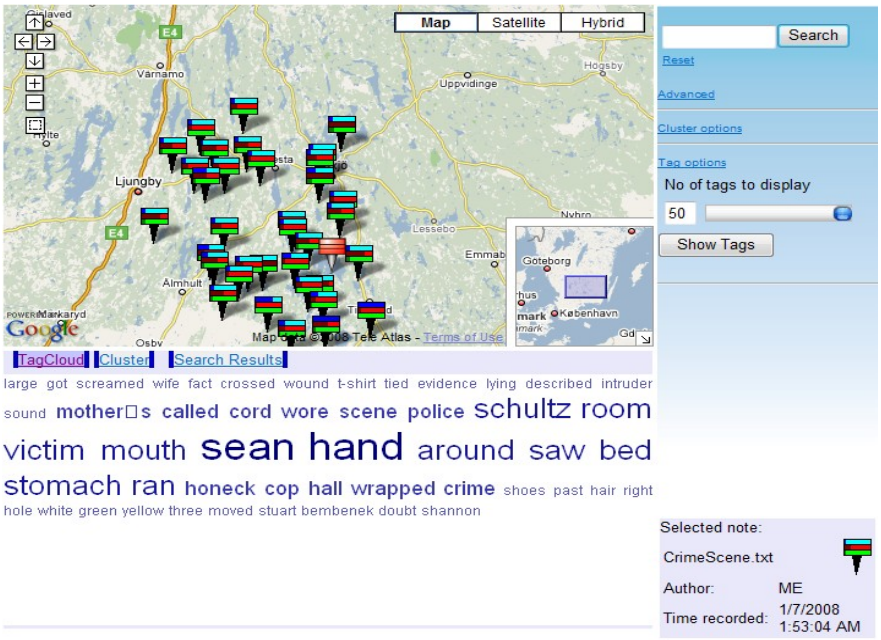
The main focus of this chapter was to describe the explanation of the technical implementation of our prototype. We discussed the technologies and tools used to implement GNV System. Later on, we presented the technical description of the software development. In the next chapter we proceed by describing that a case study presents the capabilities of our software.

9 Case Study

In this chapter, we will check the functionality of the current version of the GNV prototype. Since we do not have real data for testing our application yet, we will use scenarios we had in the Chapter 3: the scenario about the forensic investigation. Below, we will elaborate this scenario by trying to show the most important functions of our application.

9.1 "Forensic Investigation"

A murder has occurred and the forensic agents are in a crime scene. There can be more than one forensic agent taking notes about the crime scene. These notes will be uploaded in the repository and will hold the GPS location where the audio note is taken. Later on, all agents can compare the individual notes by visualizing them on the system. As seen in Figure 37, when the agent logs in from the work he will see the map of the city, and he can go to the location of the crime. Beside his notes, he will find other notes from his colleagues visualized on the map. When he clicks on some of the notes from that location, he will see the note's visualization on the Notes Visualization Panel. The number of tags that will be visualized by default is 50, but the agent can modify this by determining the number of the tags per note that he wants to see. This way, the agent can get an insight of the content of each note he selects.



No of tags to display

50

large got screamed wife fact crossed wound t-shirt tied evidence lying described intruder sound mother□s called cord wore scene police schultz room victim mouth sean hand around saw bed stomach ran honeck cop hall wrapped crime shoes past hair right hole white green yellow three moved stuart bembenek doubt shannon

Selected note:

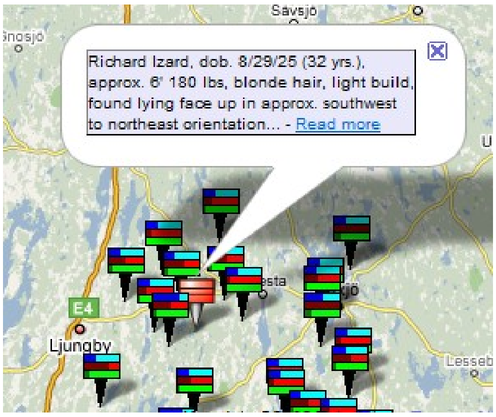
CrimeScene.txt

Author: ME

Time recorded: 1/7/2008 1:53:04 AM

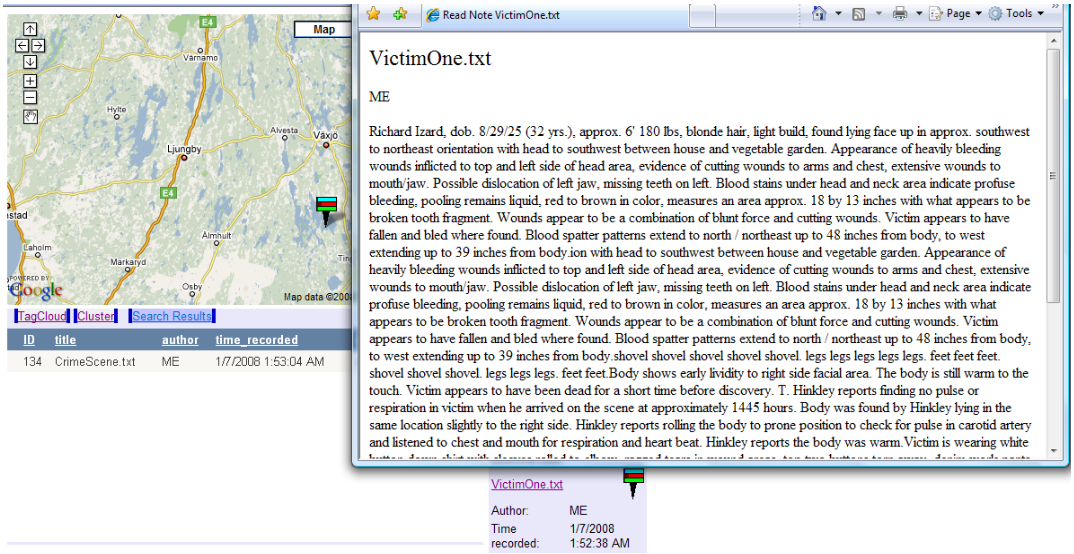
Figure 37: Content visualization of geoaudio notes

If he wants to see more details for the content of the desired note, he can click twice in the note (Figure 38), and above the note the Information Window will be opened. Here some text from the document is displayed together with the link for opening the entire document. Or, if the agent wants to see the entire text of the document, he can click on the name of the note in the bottom right corner where the information of the selected note is (Figure 37). The example of the entire document is shown in Figure 39.



Richard Izard, dob. 8/29/25 (32 yrs.), approx. 6' 180 lbs, blonde hair, light build, found lying face up in approx. southwest to northeast orientation... - Read more

Figure 38: Information Window



TagCloud Cluster Search Results

ID	title	author	time_recorded
134	CrimeScene.txt	ME	1/7/2008 1:53:04 AM

VictimOne.txt

ME

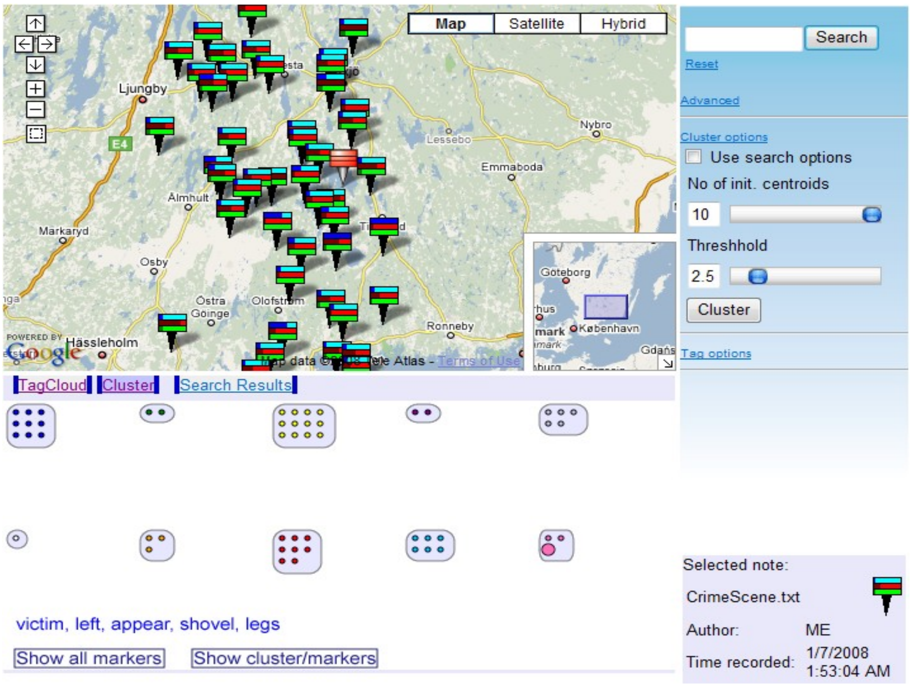
Richard Izard, dob. 8/29/25 (32 yrs.), approx. 6' 180 lbs, blonde hair, light build, found lying face up in approx. southwest to northeast orientation with head to southwest between house and vegetable garden. Appearance of heavily bleeding wounds inflicted to top and left side of head area, evidence of cutting wounds to arms and chest, extensive wounds to mouth/jaw. Possible dislocation of left jaw, missing teeth on left. Blood stains under head and neck area indicate profuse bleeding, pooling remains liquid, red to brown in color, measures an area approx. 18 by 13 inches with what appears to be broken tooth fragment. Wounds appear to be a combination of blunt force and cutting wounds. Victim appears to have fallen and bled where found. Blood spatter patterns extend to north / northeast up to 48 inches from body, to west extending up to 39 inches from body.ion with head to southwest between house and vegetable garden. Appearance of heavily bleeding wounds inflicted to top and left side of head area, evidence of cutting wounds to arms and chest, extensive wounds to mouth/jaw. Possible dislocation of left jaw, missing teeth on left. Blood stains under head and neck area indicate profuse bleeding, pooling remains liquid, red to brown in color, measures an area approx. 18 by 13 inches with what appears to be broken tooth fragment. Wounds appear to be a combination of blunt force and cutting wounds. Victim appears to have fallen and bled where found. Blood spatter patterns extend to north / northeast up to 48 inches from body, to west extending up to 39 inches from body.shovel shovel shovel shovel shovel. legs legs legs legs legs. feet feet feet. shovel shovel shovel. legs legs legs. feet feet.Body shows early lividity to right side facial area. The body is still warm to the touch. Victim appears to have been dead for a short time before discovery. T. Hinkley reports finding no pulse or respiration in victim when he arrived on the scene at approximately 1445 hours. Body was found by Hinkley lying in the same location slightly to the right side. Hinkley reports rolling the body to prone position to check for pulse in carotid artery and listened to chest and mouth for respiration and heart beat. Hinkley reports the body was warm.Victim is wearing white

VictimOne.txt

Author:	ME
Time recorded:	1/7/2008 1:52:38 AM

Figure 39: Reading the entire document

Furthemore, he can change the visualization view and switch to Clustering tab to see the semantic relations of the notes. In this situation, he can find all the notes semantically related to the selected note. When he sees the clusters (Figure 40), he will find out that there are notes from different locations that match with the selected note. When the agent moves around with a mouse to any of the clusters, on mouseover, the five most weighted tags from the notes of that cluster will be shown in the bottom part of the Cluster Tab, as seen in Figure 40. This might show some patterns between two or more crimes. This way, forensic agents can possibly discover some correlations between their findings.



□ Use search options

No of init. centroids: 10

Threshhold: 2.5

victim, left, appear, shovel, legs

Selected note:

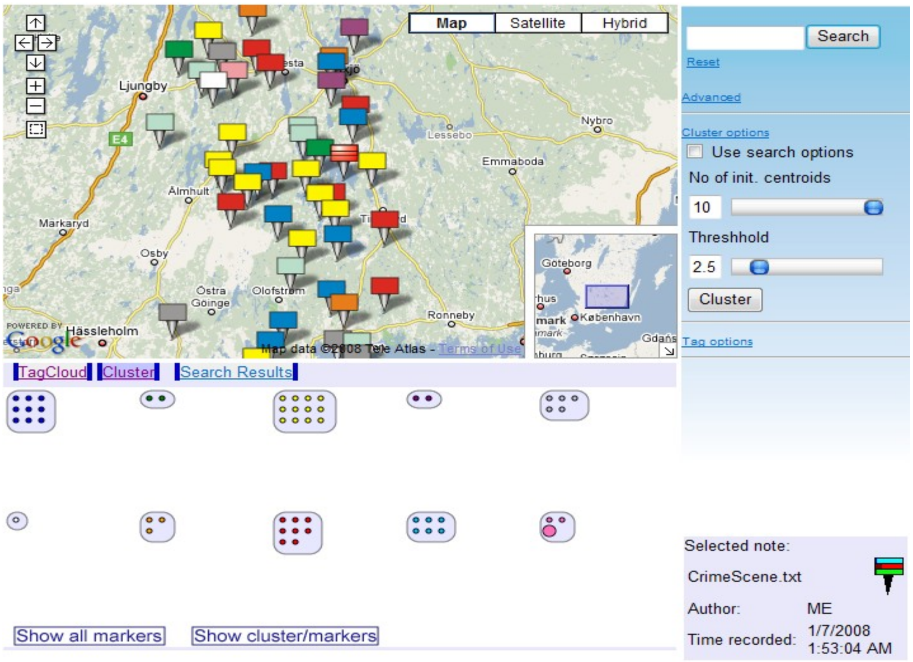
CrimeScene.txt

Author: ME

Time recorded: 1/7/2008 1:53:04 AM

Figure 40: Semantic relations of geoaudio notes

If he wants to see which notes belong too the same cluster, he will click on the "*Show cluster/marker*" buton, and notes will be visualized in the Map Panel as seen in Figure 41. Now, the colors of the markers are changed and correspond to the color of the clusters. In this way, an agent can see the area a certain cluster covers, thus possibly providing some spatial patterns for discovering the crime.



□ Use search options

No of init. centroids: 10

Threshhold: 2.5

Selected note:

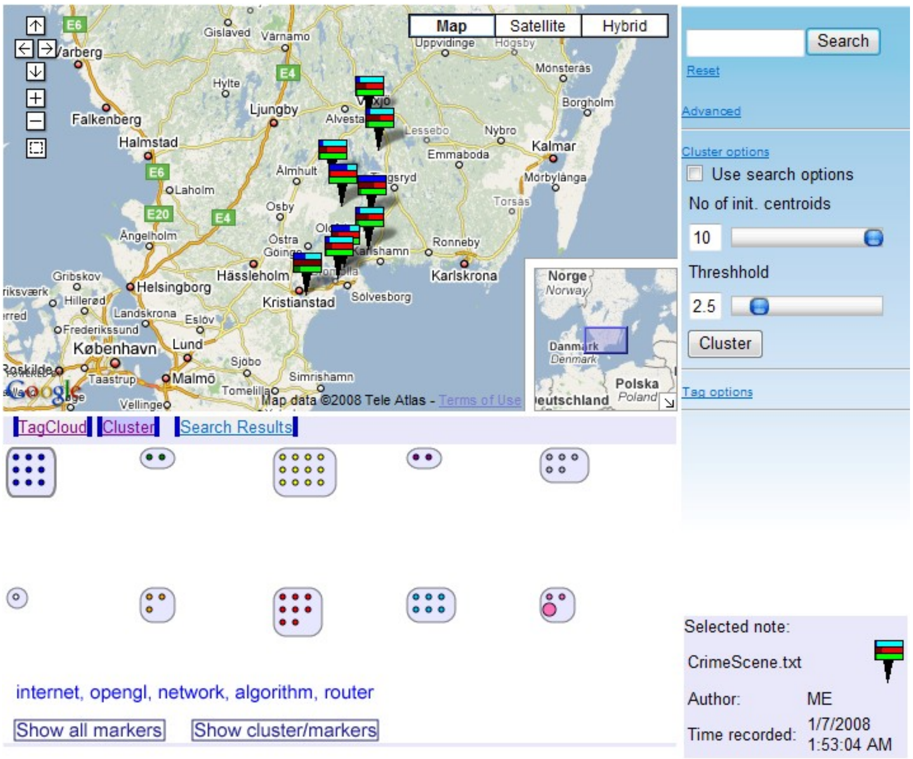
CrimeScene.txt

Author: ME

Time recorded: 1/7/2008 1:53:04 AM

Figure 41: Showing clustered notes in Map Panel

Clicking again on the button "*Show cluster/markers*", the notes in the Map Panel will be visualized, see Figure 40. If the agent clicks on any of the clusters in the Clustering Tab, just the notes of that cluster will be shown in the Map Panel (see Figure 42). The view on the Map Panel will zoom out and pan enough to show all the notes of that cluster.



□ Search

Reset

Advanced

Cluster options

□ Use search options

No of init. centroids

10

Threshhold

2.5

Cluster

Tag options

TagCloud | Cluster | Search Results

internet, opengl, network, algorithm, router

Show all markers | Show cluster/markers

Selected note:

CrimeScene.txt

Author: ME

Time recorded: 1/7/2008 1:53:04 AM

Figure 42: Showing only the notes of selected cluster

When the agent wants to cluster the notes of a specific area, he selects that area (Figure 43), and determines the number of centroids and the value of the threshold in the cluster options (Figure 44). Also, he can cluster notes of the selected area by determining the time as criteria as seen in Figure 45.

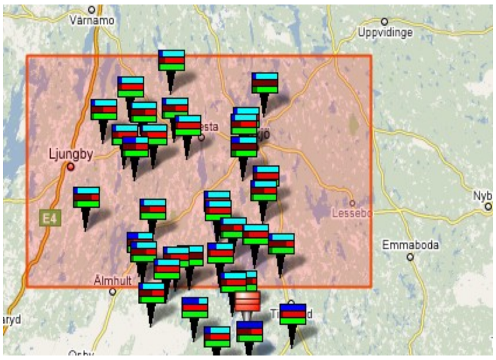
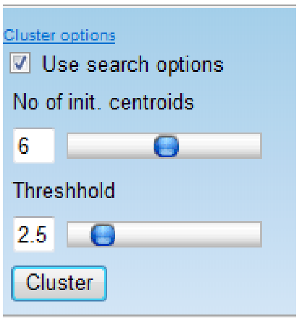


Figure 43: Selected area



Cluster options

☑ Use search options

No of init. centroids

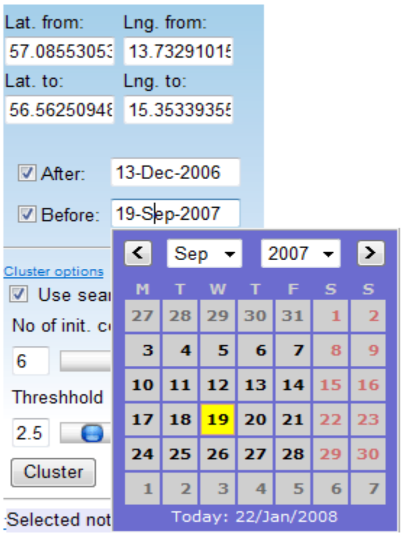
6

Threshhold

2.5

Cluster

Figure 44: Cluster options



Lat. from:	Lng. from:
57.0855305З	13.7329101Ŀ
Lat. to:	Lng. to:
56.5625094ξ	15.3533935Ŀ

☑ After: 13-Dec-2006

☑ Before: 19-Sep-2007

Cluster options

☑ Use sea

No of init. c

6

Threshhold

2.5

Cluster

Selected not

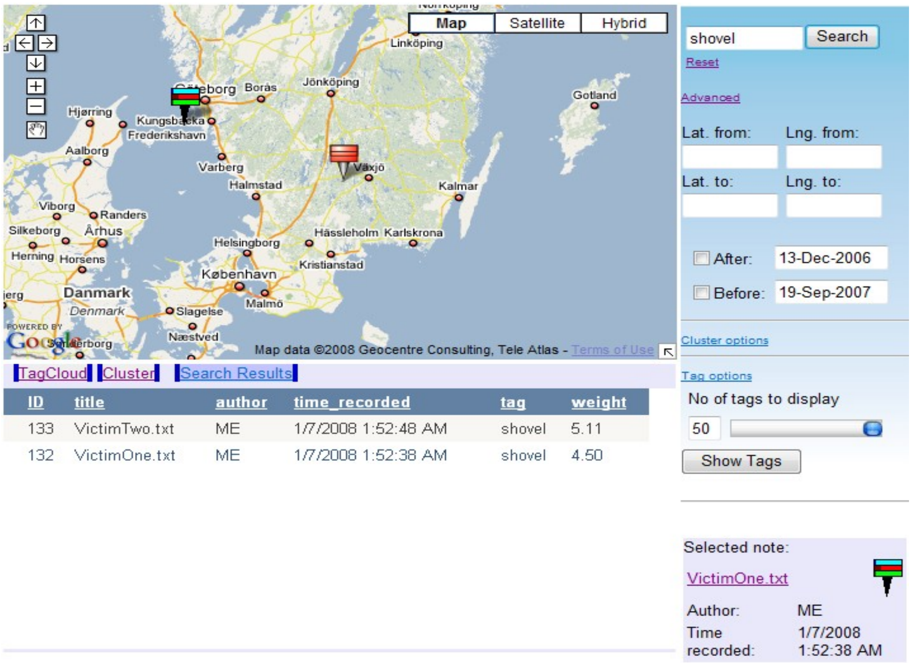
< Sep ▾ 2007 ▾ >

M	T	W	T	F	S	S
27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Today: 22/Jan/2008

Figure 45: Adding time as a criteria for clustering

The GNV System also supports searching mechanisms. If the agent searches for a specific tag, he will get the list of the documents containing that tag in the Search Results tab. In the same time only the notes containing that tag will be shown in the Map Panel (Figure 46). The agent can click on the notes from the list, and corresponding icon will be selected in the Map Panel view (the map will move smoothly until the position of that note). Furthermore, the agent can reorder the list. This is done simply by clicking on the desired ordering attribute (e.g., by clicking on *time created* agent can show the most recent added notes first, or vice versa).



TagCloud | Cluster | Search Results

ID	title	author	time_recorded	tag	weight
133	VictimTwo.txt	ME	1/7/2008 1:52:48 AM	shovel	5.11
132	VictimOne.txt	ME	1/7/2008 1:52:38 AM	shovel	4.50

shovel Search

Reset

Advanced

Lat. from: Lng. from:

Lat. to: Lng. to:

□ After: 13-Dec-2006

□ Before: 19-Sep-2007

Cluster options

Tag options

No of tags to display

50

Show Tags

Selected note:

VictimOne.txt

Author: ME

Time recorded: 1/7/2008 1:52:38 AM

Figure 46: Search result

Imagine that within few days another murder has occurred, but in some other part of the city. This other murder falls into some other police district, and both of them do not seam to be correlated. But when they upload their notes in the system, the visualization will show the possible pattern in those cases. Figure 40 and Figure 41 shows how the system has clustered agent's note (the pink one) with other two notes. After that, the agent can see the content of those two other cases that are in the same cluster with his case. Through the help of visualization he will find that one case has some interesting similarities and a certain pattern exists between them. Figure 47 and Figure 48 show the content of those two notes. These patterns will be discovered really fast by the agent since the visualization will aid the insight to the text.

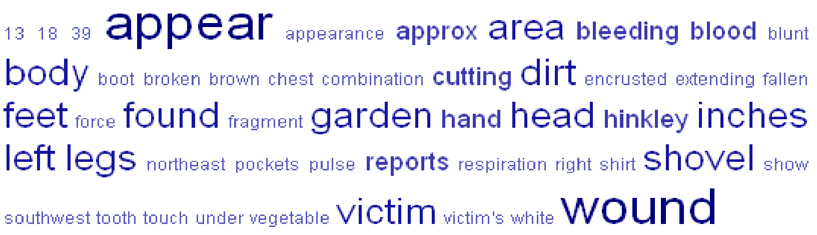


Figure 47: Content Visualization of the note that has semantic relation with agent's note

Figure 47, shows the content of the note that is semantically related with agent's note which is in Figure 48. When the agent compares this note with his note, he will see that in both cases the victims were killed by shovel. He also can see that in both cases the legs of victims are interesting details.

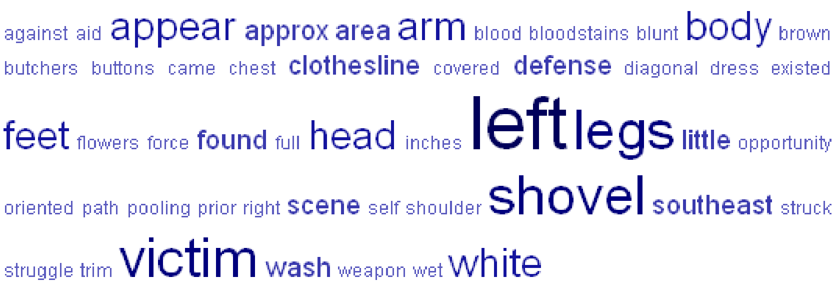


Figure 48: Content Visualization of the agent's note

After that, the agent opens and reads the entire report of that case that has some relation with his, and the agent discover that this two murders are possibly done by the same person. In this way, we will help agents to gain more knowledge of the case which will hopefully lead to the resolvance of the case.

10 Conclusions

The aim for this thesis was to to explore and develop novel visualizing techniques to visualize the content of GeoAudio Notes, and to show the semantic relations between them. In this chapter, we will discuss our achievements during this thesis, and we will discuss what can be added furthermore to improve the developed system in order to provide better results. Beside this, we will also discuss about some interesting techniques that we have proposed in Section 3.3, that could be the focus of future research.

10.1 Results

In this section, we will give an overview of our work. Based on the questions that we have formulated at the beginning of our thesis, we will try to answer them by explaining what kind of visualization techniques and methods we have used for implementing the application. Our aim was to design different visualization and interaction techniques. First, we have defined the scope of our work and analyzed existing approaches. This has helped us to make more precise definition of thesis questions. Further analysis of requirements and design shaped our implementation of the application that we have named GNV System. Below, we will try to answer both questions by beginning with the first one.

> *Which are the most suitable techniques for visualizing the content of recorded audio notes in digital maps, such as Google Maps?*

Recorded audio notes are created on the mobile phones, PDAs, or in any device that support GPS. After the notes are created they should be uploaded in the repository and by using GPS coordinates the note will appear on the map at the exact place where it was created. This was the first step. The second step was to determine the visual form of the note that will appear on the map. As it is explained in the Section 4.1, we have decided to use bitmaps for presenting the notes in the map. The bitmap will be positioned in the map and will visualize some meta data of the note, such as the time created, the number of words, and the length of the note. By this, users will have some knowledge about the note. Finally, the third step was to get the content of the note and to visualize them as explained in the Section 4.2. For visualizing the content of the note we have used the Tag Clouds approach. When the user will select the desired note the most frequently used tags will appear in the Note Visualization Panel. By default, the number of tags is 50, but the user can change that by determining the number of tags in the Tag Options. When the user clicks twice on the note marker, a paragraph from the note will appear in the Information Window, and two links. One link is for displaying the entire text of the note, and the other link is for listening the audio note.

For answering the first question, we have used different visualization techniques and methods. First, the note should be placed in the exact place where it was created. The second step was to create the bitmap for each note. From the bitmap visualization, users can see some details on the note, thus they will get insight of the note. The final step was to visualize the content of the note. That was possible by visualizing most frequently used tags. In consequence this means that our approach fulfills the criterion of expressive visual mapping, since all and only the data from our data set is represented [42].

As mentioned in our literature review (Section 2.2), there exists a paper [30] and a prototype which displays textual messages in a map. But this paper does not offer a possibility for visualizing the content of the document. We have an advantage here by offering visualization of text through a combination with tag-cloud. By using all this visualization techniques and methods, we think that the first question of our thesis is answered, and we will proceed with the second question:

> *How can different visualization and interaction techniques be combined to explore novel ways of showing semantic relations and patterns between the content of different audio notes in space and time?*

As we have explained in Section 4.3, we have used a clustering method K-Means for showing semantic relations between the content of the audio notes. The K-Means algorithm will cluster notes based on their tags. So, notes that are in same cluster means that have similar tags. Clusters will be visualized in the Clustering tab of the Note Visualization Panel. It offers interaction with all panels in our application. We have used different interaction techniques, such as navigation techniques, manipulation techniques, and human-interaction techniques. For example, when the user selects one cluster from many clusters that are in the Clustering tab, just the notes of that cluster will be displayed in the Map Panel. A more detailed description with more features can be found in Section 4.4. By clicking on the link "Show clustered notes in the map", each note will be presented by containing the color of the cluster that it belongs to, thus visualizing each cluster in the Map Panel. By this, users can see the notes that are in the same cluster.

We have also introduced space and time criteria in the way that we added options in the Tool Box Panel, where users can select a specific area in the map. By pressing the clustering button, the system will cluster just the note that were inside the selected area, and/or they can select a time span from the Tool Box. For example, if the user wants to cluster notes that are in a specific area, and created from February 2007 until the January 2008, the system will cluster the notes of that selected area, and the notes that were created during that required time-frame.

We offered an answer for visualizing semantically related notes. Some approaches already existed considering this issue. One of them isFEMARepViz [30] which offers a very good classification of the document into predefined set of categories. It displays the category or topic of the document through an icon [30]. We consider this approach to be less effective when showing semantical relation of unstructured data, which is of course what our work is dealing with. In our case documents of the same category can be semantically related as well as have no relation at all. These cases need to be visualized too.

We also offered additional display space (Clustering Tab) which will deal exclusively with showing these semantical relation instead of doing it through icon (or color which is possible in our prototype and is used to show spatial positioning of the documents with common meaning). So, our prototype offers better insight into semantical relation of the documents in the visualization point of view.

After clustering the notes, visualizing them on the Clustering Tab, using different interaction techniques, using time and space as optional criteria for clustering notes, we think that we have also answered the second question of our master thesis.

10.2 Limitations

There are number of limitations that we discovered throughout our work. All of them are of technical nature. The list of limitations is not ordered in any kind, so the order does not express negative impact of the limitation in our research and application.

- The Google API provides very good options in visualizing geospatial data, but it still can not cope very well with a large number of data without loosing some of the details. It still lacks some of the methods for doing some simple tasks when it comes to interaction techniques, instead of heavy use of Javascript which can affect the overall performance of the application.
- The current clustering algorithm produces different results each time the clustering is done. This happens because of the nature of the algorithm. The use of different clustering algorithm would probably provide better results and more possibilities to enhance the visualization of the clusters, but it would slow down the application which would eventually render it to unusable.

10.3 Analysis and Reflections

As we can see from our topic “Content Visualization of GeoAudio Notes”, our research falls under the Geovisualization discipline, which deals with geospatial data. After the literature analysis and review (see Chapter 2 for more details), we were able to better understand the current work on the topic we chose. Based on the research we have conducted for finding visualization and interaction techniques for visualizing geospatial data and showing the semantic relations between them, we have achieved to build an application. During our research, we have learned that for visualizing geospatial data, or any other data, interaction techniques are important, as they will help users to get the desired results faster.

From the beginning of our thesis our aim was to form those two questions which we have proposed in the early phases of our thesis. We achieved our goals through a novel combination of various visualization and interaction techniques. In Section 7.2, we have explained how we answered those two questions. Of course, our application has a room for a lot of improvements. An evaluation of application would probably show best where and how the system could be improved. Nevertheless, we have proposed some possible improvements system could have in next section.

Since we have not used any specific evaluation method, we will propose some of the ways our prototype could be evaluated. One of the ways is through controlled experiments which are good for measuring performance and comparing multiple techniques [41]. We have done some comparison of our application with the work described in one paper [30] which can be one of the most similar application to ours. But we did not have the actual application to test and compare. We have presented our findings about this issue in previous section. On the one hand, it would have been good to actually test both applications and evaluate our prototype against the other one. But on other hand, if two systems have slightly different goals and objectives, it is difficult to perform direct comparative evaluation of them [41]. This fact makes the evaluation of the effectiveness [42] rather difficult in near future, since to best of our knowledge, there is no application that shares same goals and objectives with GNV System. We think the best way to do an evaluation is to test it with users and hear their feedback. This could help us in improving the usability and utility of the application. We might even get some interesting ideas considering the future development of the prototype which we will discus in the next section.

10.4 Future Work

We have managed to discover a number of issues that could improve our work. Some of those issues are beyond the scope of this thesis, others are our ideas that we got during the research but we did not have time to implement and research further. Below, you will find the list of the work that should be done in the future:

- In Section 3.3 we proposed the idea adding one more panel in which users would have possibilities to select a number of notes and to view their content and to compare the notes. For additional information look at Section 3.3, and this metaphor is depicted in Figure 15. We think that this idea can improve our application, but considering the time that we had for completing our master thesis work, we left this idea to be implemented in the future.
- The prototype we built is an on-line application which ideally would have to work on a number of browsers and would be platform independent. Since we are using ASP.NET and AJAX combined with heavy Javascript and SVG, it is producing somewhat unexpected results. This should be explored and fixed to produce the same results in different browsers.
- Implementation of an application that will facilitate the uploading of the notes, convert them to text, data mine and store them in our database.
- Improvement of the Tag-Cloud approach. One possibility that needs to be considered is using a clustering algorithms for visual layout [29].
- We used the K-Means algorithm for clustering the notes. The main drawback of the algorithm is that it does not produce always the same results. Better text clustering approaches should be considered in the future.

11 Glossary

GNV System – GeoNote Visualization System

GeoVis – Geographic Visualization

ViSC – Visualization Scientific Computing

EDA – Exploratory Data Analysis

GISystems – Geographic Information Systems

GPS – Global Positioning System

PDA – Personal Digital Assistant

UPEDU – Unified Process for EDUcation

HTTP – HyperText Transfer Protocol

IIS – Internet Information System

HTML – HyperText Markup Language

GIS – Geographic Information System

SVG – Scalable Vector Graphics

KML – Keyhole Markup Language

XML – Extensible Markup Language

C# – C Sharp (Programing language)

PHP – Hypertext Preprocessor

SQL – Structured Query Language

SOAP – Simple Object Access Protocol

12 References

1. Voisar, A., 2007. *Location Based Services. ABIresearch*, [internet]. Available at : http://www.abiresearch.com/products/market_research/Location_Based_ Services [accessed 12 October 2007].

2. Reichenbacher, T. & Swienty, O., (2007). *Attention-guiding geovisualisation.* In: Cd-Proceedings of the 10th AGILE International Conference on Geographic Information Science, Aalborg, Danmark.

3. MacEachren, A.M., Kraak, M.-J., 2001. *Research Challenges in Geovisualization.* Cartography and Geographic Information Science

4. Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhind, D.W., 2005. *Geographical Information Systems and Science.* Wiley, Chichester

5. Robinson, A.C., 2006. *Highlighting Techniques to Support Geovisualization*, GeoVISTA Center, Department of Cartography, The Pennsylvania State University

6. Buckley, A.R., Gahegan, M., Clarke, K., *2000. Geographic Visualization.* [Online]
Available at : http://www.ucgis.org/priorities/research/research_white/2000%20Papers/emerging/Geographicvisualization-edit.pdf [accessed 10 October 2007].

7. Lurie, M., & Lurie, A., Y., 2006. *DB2 and Open Source: Put Yourself on the Map with Google Maps API, DB2/Informix, and PHP on Linux.* [Online]
Available at : http://www.ibm.com/developerworks/db2/library/techarticle/dm-0602lurie/#figure1 [accessed 10 October 2007].

8. Robillard, P., N., Kruchten, P., d'Asous, Patrick., 2004. *Software Engineering Process: with the UPEDU* [Online].
Available at: http://www.upedu.org/upedu/index.asp?TruY=862119516280984 [accessed 12 October 2007].

9. O. Kulyk, R. Kosara, J. Urquiza, I. Wassink., 2007. Human-Centered Aspects. In A. Kerren, A. Ebert, and J. Meyer (Eds.), *Human-Centered Visualization Environments*, Volume 4417 of LNCS Tutorial, Springer.

10. Perlin, K., Fox, D., 1993. Pad: *An Alternative Approach to the Computer Interfaces.* In: Proceedings of the 20th Annual ACM Conference on Computer Graphics (SIGGRAPH '93), pp.57-64. ACM Press, New York, NY.

11. Plaisan, C., Carr, D., Shneiderman, B., 1995. *Image-browser taxonomy and guidelines for designers.* IEEE Software 12(2), 21-32.

12. Kadmon, N., Shlomi, E., 1978. *A polyfocal projection for statistical surfaces.* Cartograph 15(1), 36-40.

13. C. Görg, M. Pohl, E. Qeli, and K. Xu., 2007. Visual Representation. In A. Kerren, A. Ebert, and J. Meyer (Eds.), *Human-Centered Visualization Environments*, Volume 4417 of LNCS Tutorial, Springer.

14. Nöllenburg, M., 2007. Geographic Visualization. In A. Kerren, A. Ebert, and J. Meyer (Eds.), *Human-Centered Visualization Environments*, Volume 4417 of LNCS Tutorial, Springer.

15. Jarvinen, P., 2001. *On Research Method*. Opinpajan Kirja, Tampere – Finland.

16. Mirkin, B., 2005. *Clustering for data mining : a data recovery approach*, Boca Raton, FL : Chapman & Hall/CRC.

17. P. Smyth., 2002. Types and Forms of Knowledge (Patterns). In W. Klösgen and J. M. Żytkow, *Handbook of data mining and knowledge discovery*, Oxford : Oxford University Press.

18. Ware C., 2000. *Information Visualization: Perception for Design*. Morgan Kaufmann, San Francisco, CA.

19. W. Fikkert, M. D'Ambros, T. Bierz, and T.J. Jankun-Kelly. 2007. Interacting with Visualization. In A. Kerren, A. Ebert, and J. Meyer (Eds.), *Human - Centered Visualization Environments*, Volume 4417 of LNCS Tutorial, Springer.

20. Fineman, B., 2004. *Computers as people: human interaction metaphors in human computer interaction*. Master's thesis, The School of Design – Carnegie Mellon University.

21. Hand, C., 1997. *Survey of 3D Interaction Techniques*. Computer Graphics Forum.

22. Schneiderman, B., & Maes, P., 1997. *Direct manipulation vs. interface agent*. ACM Interactions 4(6), 42-61.

23. Kraak, M-J., 2003. *Geovisualization Illustrated*. ISPRS Journal of Photogrammetry and Remote Sensing 57, 390-399.

24. Koch, T., 2004. *The Map as Intent: Variation on the Theme of John Snow*. Cartographica 39(4), 1-14.

25. Card, S.K., MacKinlay, J. D., Shneiderman, B. (eds.): *Reading in Information Visualization: Using Vision to Think*. Morgan Kaufmann, San Francisko, CA (1999).

26. A.Kerren et al. (Eds): *Human-Visualization Environments* 2006, LNCS 4417, pp.1-9, 2007. © Springer-Verlag Berlin Heidelberg 2007.

27. Weber, W., 2007. *Text Visualization - What Colors Tell About a Text*. 11th International Conference Information Visualization (IV'07) © IEEE.

28. Keim, D. A., & Oelke, D., 2007. *Literature Fingerprinting: New Method for Visual Literary Analysis*. IEEE Symposium on Visual Analytics Science and Technology 2007 October 30 - November 1, Sacramento, CA, USA

29. Hassan-Montero, Y., & Herrero-Solana, V., 2007. *Improving Tag-Clouds as Visual Information Retrievel Interfaces*. Scimago Research Group. University of Grenada, Faculty of Library and Information Science, Colegio Maximo de Cartuja.

30. Pan, C., & Mitra, P., 2007. *FemaRepViz: Automatic Extraction and Geo-Temporal Visualization of FEMA National Situation Updates*. IEEE Symposium on Visual Analytics Science and Technology 2007 October 30 - November 1, Sacramento, CA, USA

31. *Ajax Introduction* (Anon 1999-2008). [Online].
Available at : http://www.w3schools.com/ajax/ajax_intro.asp [accessed 5 January 2008].

32. *ASP.NET Ajax Overview* (Anon 2008). [Online].
Available at : http://msdn2.microsoft.com/en-us/library/bb398874.aspx [accessed 05 January 2008].

33. *What is the Google Maps* API (Anon 2007). [Online].
Available at : http://code.google.com/apis/maps/index.html [accessed 6 January 2008].

34. *Introduction to Javascript* (Anon 2007). [Online].
Available at : http://www.w3schools.com/js/js_intro.asp [accessed at 9 January 2008].

35. Lilley, Ch., Jackson, D., 2004. *About SVG: 2d Graphics in XML*. [Online].
Available at : http://www.w3.org/Graphics/SVG/About.html [Accessed 9 January 2008].

36. *ASP.NET Introduction* (Anon 2007). [Online].
Available at : http://www.w3schools.com/aspnet/aspnet_intro.asp [accessed 9 January 2008].

37. Naunce Comunication, 2007. *Dragon Audio Mining* [Online].
Available at : http://www.nuance.com/audiomining [accessed 15 September 2007].

38. Steinbock, D., 2007. *TagCrowd* [Online].
Available at : http://www.tagcrowd.com [accessed 15 September 2007].

39. Biderman, A., 1990. *The Playfair enigma: the development of schematic representation of statistics*, Information Design Journal, 6, 1. pp. 3-25.

40. Spence, R., 2001. *Information Visualization*. Barcelona: Graphos S. A., Arte sobre papel.

41. Stasko, J., 2006. *Evaluating Information Visualizatio: Issues and Opportunities,* BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV'06), May 23, 2006, Venezia, Italy

42. Card S., Mackinlay J. and Shneiderman, B., 1999. *Reading in Information Visualization: Using Vision to Think*, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., page 23, Chapter 1.

Appendix 1

Note Uploader – Tool for Populating the Database

Although thesis does not cover the part of storing notes in a database, we should still populate the database with test date in order to implement our prototype. Although inserting records in a database is simple and does not require any special attention, text mining, respectively getting a list of most frequently used tags requires a bit more attention. So, in order to populate the database with test data we should implement some classes for text mining first of all. Note that the implementation is done using C#. Figure 12.1 shows the class design for two classes (*UploadNote and TagMiner*) we implemented for mining the text. *UploadNote* is the GUI form and where insertion of the records is done, we used the *Tag* and *Conn* class also. But that is described in Chapter 8 on the implementation techniques . We will only describe the *TagMiner* class, since the others are either described or pretty much simple and do not need much attention.

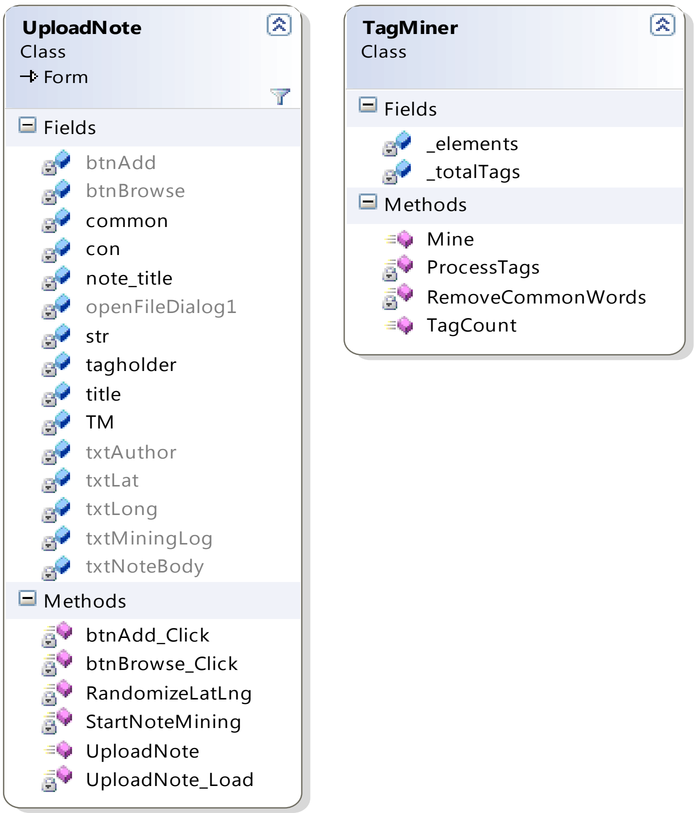


Illustration 492.1: Upload Note (left) and Tag Miner (right) class design diagram

Class	**TagMiner**
Description	In this class the actual mining process is done and the tags are placed in the `ArrayList`.
Fields	*_elements*: holds the array of mined tags. This is an `ArrayList` so it stores Tag objects in it. *_totalTags*: holds the number of total strings mined from the text.
Methods	*Mine()*: this is the main method to do the mining process. This method returns the `ArrayList` of a mined tag. This method requires three parameters: *Text, commonWords* and, *tagNumber*. First parameter *Text* (`string` type) holds text that should be mined. The second parameter *commonWords* (`string` type) holds the string of common words that should not be included in the array (words should be space separated). The last argument, *tagNumber* (`integer` type) specifies the number of tags that method should return. This method calls two other methods in order to finish its task: *RemoveCommonWords()* and *ProcessTags()*. *ProcessTags()*: this method sorts the tag list (*_elements field*), removes the tags that exceed the number of the needed tags (it removes the tags with less occurrence, that is less important tags). Finally, it calculates the weight of each tag in proportion to the number of tags needed. This method requires only one parameter *tagNumber*, which defines the number of tags *_elements* field should hold. *RemoveCommonWords()*: removes common words from the list of tags. It takes only one parameter *commonWords* (`string` type) which holds list of space separated words to be excluded from the list of tags. *TagCount()*: returns the total number of tags found in text (including the ones excluded with help of *commonWords* method).

Appendix 2

Used JavaScript's from the WWW

In this part, we will present various Javascript that are found on-line under GNU license or given for free that we used in our prototype. We will present the list of used scripts in the following tables.

Animated Collapsible DIV	
Author:	Dynamic Drive
URL:	http://www.dynamicdrive.com
Description:	This script collapses any DIV on the page and lets users manually toggle its appearance via a smooth "Web 2.0 style" animation.
Usage description:	We used this script to collapse and show options on the Tool Bar (e.g. If the user clicks on the Cluster Tab, the Tag Options tool box is collapsed and the Cluster Options tool box is expanded).
Javascript file location:	[application root]\JavaScript\animatedcollapse.js

Simple Calendar Widget	
Author:	Anthony Garrett
URL:	http://www.garrett.nildram.co.uk/calendar/scw.htm
Description:	This flexible, simple pop-up calendar is written to work across browsers. It's fully commented and customisable for language, colours and date format. The pop-up displays a month at a time from a specified range of years. The optional input date determines the initial month displayed. The year can be in two or four digits while the month can be digits or a month name abbreviation that can easily be set up for any language. Any date or day of the week can be disabled (and re-enabled) using JavaScript on the calling page. This is ideal for combination with server-side technologies to produce a professional application.
Usage description:	We used this script in the Tool Box, Advanced selection to display a interactive calendar for date picking.
Javascript file location:	[application root]\JavaScript\scw.js

Tigra Slider Control	
Author:	Anthony Garrett
URL:	http://www.softcomplex.com/products/tigra_slider_control/
Description:	Tigra Slider Control - is a DHTML component for the websites that adds vertical and horizontal trackbar capability to the HTML forms similarly to the controls widely used in desktop applications. The position of the handler is translated into the value from specified range and placed in existing input box of the HTML form. The slider is simple improvement that makes the interface more user friendly.
Usage description:	We used this script in the Tool Box at the Cluster and Tag Options to add slider controls to our application.
Javascript file location:	[application root]\JavaScript\slider.js

Appendix 3

Links Used for Populating the Database with Data

For testing our application, we have populated our database with data that were taken from the internet. The content of the notes are not made by users who use GPS enabled devices, but this was done by us. We have used the links bellow to get the text, and from that text we have made a note. Every note in our system contains content from one of the links bellow.

http://en.wikipedia.org/wiki/Wireless_network
http://en.wikipedia.org/wiki/XML
http://www.geog.ucsb.edu/~sara/html/research/pubs/skupin_fabrikant_cagis.pdf
http://en.wikipedia.org/wiki/Video_game
http://en.wikipedia.org/wiki/DirectX
http://en.wikipedia.org/wiki/Computer_network
http://en.wikipedia.org/wiki/Local_area_network
http://planetmobile.us/
http://en.wikipedia.org/wiki/Science_fiction
http://en.wikipedia.org/wiki/Cluster_analysis
http://en.wikipedia.org/wiki/3D_computer_graphics
http://en.wikipedia.org/wiki/Farscape
http://en.wikipedia.org/wiki/Berkeley_Software_Distribution
http://www.geovista.psu.edu/publications/2003/MacEachren_CG&A_03.pdf
http://en.wikipedia.org/wiki/Complex_system
http://en.wikipedia.org/wiki/Direct3D
http://en.wikipedia.org/wiki/Data_structure
http://en.wikipedia.org/wiki/Visualization_(graphic)
http://en.wikipedia.org/wiki/Computer_game
http://www.sgi.com/products/software/opengl/
http://en.wikipedia.org/wiki/Statistics
http://kartoweb.itc.nl/icavis/2006/Robinson-full.pdf
http://www.w3.org/Graphics/SVG/
http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/4271943/4271944/04272029.pdf?tp=&isnumber=&arnumber=4272029
http://en.wikipedia.org/wiki/Computer_graphics
http://en.wikipedia.org/wiki/Comparison_of_OpenGL_and_Direct3D
http://www.hlt-evaluation.org/article.php3?id_article=21
http://en.wikipedia.org/wiki/OpenGL
http://www.lycos.com/info/packet-switching--data.html
http://en.wikipedia.org/wiki/Data_mining]
http://en.wikipedia.org/wiki/Linux_kernel
http://en.wikipedia.org/wiki/Algorithm
http://en.wikipedia.org/wiki/2D_computer_graphics
http://en.wikipedia.org/wiki/Wireless_access_point
http://en.wikipedia.org/wiki/Voice_over_IP
http://en.wikipedia.org/wiki/Microsoft_Windows
http://en.wikipedia.org/wiki/Web_crawler

http://en.wikipedia.org/wiki/Internet_protocol_suite
http://en.wikipedia.org/wiki/Operating_system
http://en.wikipedia.org/wiki/HTTP
http://en.wikipedia.org/wiki/Database
http://en.wikipedia.org/wiki/Scientific_visualization
http://www.crimelibrary.com/criminal_mind/forensics/crimescene/1.html
http://www.crimescene.com/noir/1958_crime_report.html

Växjö
University

Matematiska och systemtekniska institutionen
SE-351 95 Växjö

Tel. +46 (0)470 70 80 00, fax +46 (0)470 840 04
http://www.vxu.se/msi/