Master's degree thesis

INF950

lepraMap; Prototyping a Tool for Modeling Historical Sources

Andreas Berre

Number of pages including this page: 94

Bergen, 22.11.2011

Mandatory statement

Each student is responsible for complying with rules and regulations that relate to examinations and to academic work in general. The purpose of the mandatory statement is to make students aware of their responsibility and the consequences of cheating. Failure to complete the statement does not excuse students from their responsibility.

*Please complete the mandatory statement by placing a mark **in each box** for statements 1-6 below.*		
1.	**I/we herby declare that my/our paper/assignment is my/our own work, and that I/we have not used other sources or received other help than is mentioned in the paper/assignment.**	☒
2.	**I/we herby declare that this paper** 1. Has not been used in any other exam at another department/university/university college 2. Is not referring to the work of others without acknowledgement 3. Is not referring to my/our previous work without acknowledgement 4. Has acknowledged all sources of literature in the text and in the list of references 5. Is not a copy, duplicate or transcript of other work	Mark each box: ☒ ☒ ☒ ☒ ☒
3.	**I am/we are aware that any breach of the above will be considered as cheating, and may result in annulment of the examinaion and exclusion from all universities and university colleges in Norway for up to one year, according to the Act relating to Norwegian Universities and University Colleges, section 4-7 and 4-8 and Examination regulations** section 14 and 15.	☒
4.	**I am/we are aware that all papers/assignments may be checked for plagiarism by a software assisted plagiarism check**	☒
5.	**I am/we are aware that Molde University college will handle all cases of suspected cheating according to**	☒

	prevailing guidelines.	
6.	**I/we are aware of the University College`s rules and regulation for using sources**	☒

Publication agreement

ECTS credits:

Supervisor: Hans Fredrik Nordhaug

Agreement on electronic publication of master thesis

Author(s) have copyright to the thesis, including the exclusive right to publish the document (The Copyright Act §2).
All theses fulfilling the requirements will be registered and published in Brage HiM, with the approval of the author(s).
Theses with a confidentiality agreement will not be published.

I/we hereby give Molde University College the right to, free of charge, make the thesis available for electronic publication: ☒ **yes** ☐ **no**

Is there an agreement of confidentiality? ☐ **yes** ☒ **no**
(A supplementary confidentiality agreement must be filled in)
- If yes: **Can the thesis be online published when the period of confidentiality is expired?** ☐ **yes** ☐ **no**

Date: **22.11.2011**

Preface

This project originated with a request from a friend, Magnus Vollset, a doctorate student at the Department of History of the University of Bergen. He asked if I could supply some information technology know-how for the medicinal history project he was basing his doctorate thesis on; an investigation into the mechanisms of knowledge dissemination among medical researchers in the period following Gerhard Armauer Hansens discovery of the leprosy bacillus in 1873.

The project seemed interesting and the task straightforward, and we got to work, thinking we would create a database and maybe a quick Access™front end, letting Vollset structure and catalog his data as he collected it. We named our application lepraMap.

As this thesis shows, it proved to be rather more involved than that. Interesting problems related to prototyping, strategies for knowledge representation, and human computer interaction kept popping up, and about halfway through the process I concluded the project might be a good basis for a master thesis.

This thesis describes the first phase in our project; the design, implementation and evaluation of a prototype system. While this phase is now completed, the project is still ongoing, and we are working on the design of the production-quality version of our application.

Several people have provided invaluable help and assistance. I would like to thank Magnus Vollset for initiating the project, and supporting my decision to use it as a basis for a MA-thesis. I am also thankful for ideas and comments from Anna Polster, Alessio Malizia, Kai Olsen, and John Fredrik Tonnesen who all read early drafts, and the flexibility and interest displayed by my employer, Bouvet AS.

Contents

Chapter 1

Introduction

The chapter details the research challenges which prompted this project, and how these translate into a problem definition for the lepraMap application. It moves on to present the planned development process, and to outline how this is described in the thesis.

1.1 Problem definition

Past historical studies investigating how the medical knowledge of leprosy was produced have focused either on tracing the steps leading to Gerhard Armauer Hansens breakthrough in 1873, when he first discovered the bacterium *Mycobacterium leprae*, or on the first effective treatments developed in the 1930s. This leaves a gap of 50 years.

The main reason is quite simple; past historical research on the topic has been done chiefly by medical professionals, whose prime interest tend to be describing medical success stories; the discoveries leading to our current understanding of the disease.

In contrast, the fifty year period from the discovery of the bacterium up to the 1920s is characterized by *failed* research; dead-end paths of inquiry, questionable scientific methodology, discarded hypotheses, and proposed cures that turned out to do more harm than good[1].

However, this period also featured an extremely active research community. Journals were released, congresses organized, and committees formed. Despite the frenzy of activity, a cure, or even a confirmation of Armauer Hansens bacterium hypothesis, proved elusive[2]. Real progress was slow and sparse.

The goal of the research project that prompted this thesis is to identify what happened during those intervening years - how were insight and ideas communicated among the researchers, and how did they formulate a shared understanding of the disease? What effect did it have that they could not confirm the leading theory of transmission? What efforts were made to find a cure, and what role did politics and policy play? What were the forces determining what was seen as worthwhile hypotheses and paths of inquiry?

Answers to these questions might be found in the archives of the institutions and people who formed the research community; lists of library content, research papers, correspondence, reports from conferences, journals, books, hospital records, lists of employees at institutions and policy recommendations prepared for health organizations. A large amount of material is available, but the

[1] Attempts at finding a cure for leprosy included massive doses of bee-stings, the injection of cobra-venom, and the revival and modernization of a historic Indian treatment, chaulmoogra oil. The latter treatment was the most promising, and although the side-effects included extreme nausea and the debate of its efficiency continued throughout its use, it was popular into the 1940s.

[2] Mainly due to the difficulty of growing Mycobacterium Leprae in a lab setting. Armauer Hansen resorted to testing his hypothesis by exposing already infected patients to other forms of the bacterium, ultimately leading to his conviction for unethical practice and the revocation of his medical license in a much published court verdict. Hansen was kept on as a researcher.

challenge is to extract interesting insight from the abundance of data.

One approach would be to start reading through this information, hoping to achieve understanding of the patterns and processes - but it would be more satisfactory to have specific knowledge. Instead of vague conclusions; "there seems to be a lot of cooperation between researchers from all over the world", "this researcher appears central to the sharing of information", "this institution could be viewed as a focal point for this theory", we would like to be able to refer to the information we have collected, not through anecdotes and assumptions, but through the identification of specific patterns and aggregations.

This level of precision could conceivably be achieved with a traditional approach, browsing meticulously taken notes and locating instances that support a conclusion. But with projects dealing with large amounts of information this could prove difficult and time consuming. Another approach would be to try something new; could a digital tool facilitate the process by providing mechanisms for the storage, structure and analysis of collected information?

The problem we were looking to solve, and the basis for this thesis, can be formulated as:

> *How can a digital tool facilitate the organization and analysis of information collected during an investigation of historical sources?*

Initially, the solution seemed pretty simple. We would create a database, structuring the data collected from the historical material as entities and associations, and display this data through a client application. But it soon became clear that the way forward was not as obvious as we had thought. There seemed to be many different ways of solving our problem, and many questions which needed answers before we could get to work; what entities should go into the data-model, and how should they be structured? How would data be entered into the system? What was the best way of visualizing the information? What would the work-flow of a user look like?

After discussing these questions, formulating and rejecting ideas, we decided that we needed some practical experience - letting us implement some of our ideas and test how they held up in the real world.

Step one would be the design and implementation of a prototype, which would be used during the investigation of one of the sources, issues of the journal *Medicinsk Revue*[3]. It seemed a suitable choice, being easily accessible and fairly

[3]Medicinsk Revue was a journal published from 1884 to 1938 by the "Medicinske selskap i Bergen", the Bergen medicinal society. Dealing with a variety of topics, it contained both original leprosy research from the medical community in Bergen, as well as translations, abstracts and commentaries on texts from the international research community. An example

typical of the type of material used in the historical research project.

We would then evaluate what we had learned from the design and use of the prototype system, and apply this new information to the design of a production-quality version.

1.2 Thesis organization

The organization of this thesis tracks the chronological progression of the lepraMap project; initial research, followed by requirements gathering, design, prototype implementation, evaluation, and a candidate design for a "real" version of the system. This structure might give an impression of a more clearly organized design process than was really the case - it is important to keep in mind that while each phase is described in a separate chapter the phases overlap, with the design of the prototype being altered well into implementation and similar.

Chapter 2 starts by investigating the relationship between information technology and historical research, and examines work done in the field, with focus on projects related to modeling historical sources.

Chapter 3 discusses how the goals of the historical research project translate into requirements for lepraMap, and how we structured the process for designing and implementing a system to meet these requirements.

Chapter 4 presents the general architecture and how we established the design of the prototype in section 4.1, before discussing the implementation details of the finished prototype system in section 4.2.

Chapter 5 details the evaluation performed after the prototype had been used for some time. It also lists and discusses the evaluations implications for the design of the production-quality version of the system, named chronoGrapher.

Chapter 6 presents the major design features of this future system, and discusses new technologies and techniques.

Chapter 7, the conclusion. Lessons learned and the way forward.

page from the journal is found in B.

Chapter 2

Context and Background

The chapter presents the relationship between historical research and computer science. It will discuss how historical researchers have approached the subject, what has been seen as major problems, and how past projects have gone about addressing them.

2.1 Historical research and information technology

Historical researchers have been using computers and information science techniques since the early days of the computer science discipline[1], but their relationship with these tools have varied. Some historians embraced information technology, looking at it as the future of their field, while others saw it as a distraction from their real work.

Today historians readily use tools such as e-mail, forums, office suites and digitalized archives and journals in their daily work and for collaboration on projects with other researchers. Archaeologists and historical researchers use GIS (Geographical Information Systems) both to find and process interesting sites. Papers and theses are based on information stored in large databases, and the result of research is published in online journals. Yet the use of less "everyday" methods, like computer aided analysis, data-warehousing and data-mining, visualization etc., is not as common, and when used, often in the context of a specific project rather than as an integral part of the general workings of the discipline.

The next sections will give a brief outline of the role of information science in history research, past and present, attempting to place the field in the general context of academic research. We will look at how and why tools have been developed; what characteristics they have, which problems they attempt to solve, and how they go about solving them.

We will also see how the use of these tools and techniques relate to traditional historical research - how do they fit into the research process, how does it affect the work, and importantly, how historians think the relationship between historical research and information technology will develop in the future.

2.2 A field of their own?

Any meeting of information technology and historical research will by nature be inter-disciplinary - leading to discussions about what the limits of the field should be, what would constitute a descriptive and accurate name, and if it should be considered an independent academic discipline at all [1]. Maybe, as Adam Hodgkin wrote in the first volume of *History and Computing* in 1987

[1]The first journal dealing with the use of information technology in the context of the humanities, Computers and the Humanities, was founded as early as 1966. It is discontinued as of 2004.

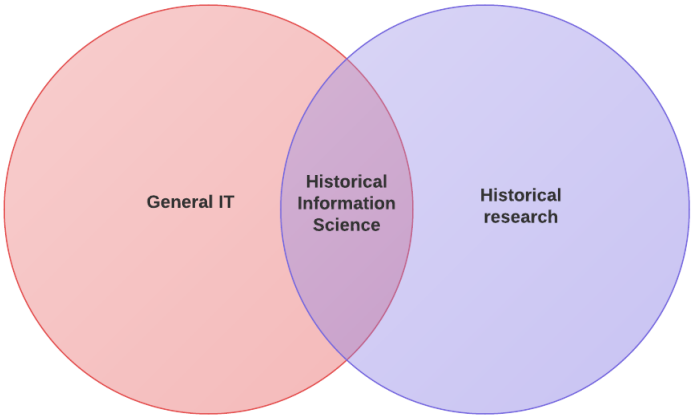


Figure 2.1: Boonstra defines the field as an intersection between the two disciplines.

[2], it was natural to assume that computing in general would become an integrated part of history research, and that there are not not enough challenges and problems specific to historical research to warrant an independent field of study. This fits nicely with what can be casually observed today - historians use information technology, but mainly in the same way as everyone else, regardless of their profession or discipline.

If one takes the opposite tack, assuming that there *is* a need for history-specific tools and methodology, what are the boundaries of the field? What areas of research should be included? What belongs with traditional computer or information science, and what belongs in the area of history research? How should the discipline, and its relation to other fields, be structured? Should the relationship consist of a new discipline in the intersection of information science and historical research, as shown in figure 2.1, or should historical information science simply be considered a subset of an already existing field?

Several articles and reports by historians explore, and try to answer, these questions. Boonstra et al., in an report on the past and future of historical information science, define the field as:

> ...the discipline that deals with specific information problems in historical research and in the sources that are used for historical research, and tries to solve this information problems in a generic way with the help of computing tools.[3]

Here historical information science is viewed as an intersection of information

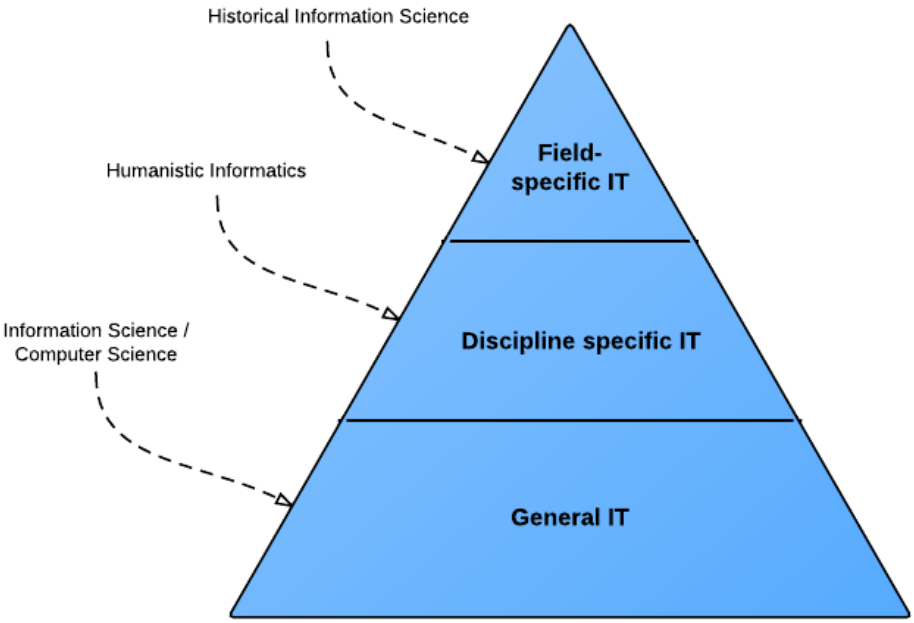


Figure 2.2: Aarseths organization of the field[4] (with added annotations)

technology and historical research, as shown in figure 2.1. This intersection contains IT problems which are specific to historical research and its sources. This means that general IT problems, even in applications of great significance to historians, should not be considered a part of historical information science, if these techniques have not been developed with history research in mind. This distinction is interesting, and will be examined closer in the discussion of *plain IT* vs. *enhanced IT* in section 2.4.

Another important part of this definition is that the stated goal is to solve problems in a generic way; it is not enough to provide an once-off solution to satisfy the needs of a specific project or to solve a specific problem, it should be possible to use the approach to solve other, similar, problems within the field. It might be a natural distinction to make if the goal is for historical information science to develop to the point where it can be viewed as a distinct field with its own set of methodologies and theories, but as we will see, there is not an overabundance of such solutions in use today.

Accepting Boonstras definition - how should the field be structured? If there is a "historical information science", dealing with field-specific problems, who is going to do the research? Information scientists turned historians? Historians

turned information scientists? A collaboration between the two?

Espen Aarseth, then at the University of Bergen Dept. of Humanistic Informatics, tries to answer these questions in his 1998 paper *From Humanities Computing to Humanistic Informatics: Creating a Field of Our Own* [4]. He visualizes the distribution of responsibilities between general information science and the field specific sub-types as represented in figure 2.2.

General IT occupies the bottom of the triangle, covering a broad array of problems of interest to all users of IT. Operating systems, generic database management systems, programming languages, search engines, and statistical algorithms are all included here.

Humanistic informatics is more specific, solving problems and techniques which are common to disciplines in the humanities, while field specific tasks, like developing custom made tools for historical research, are delegated to the top tier.

In practice, most research combining history with information technology is done by historians with a special interest in IT techniques, either working within traditional history departments or more specialized organizations. However, the techniques developed have still not been widely disseminated into the field in general.

It seems safe to say that if the practitioners of historical information science have managed to establish "a field of their own", they are some way from being a stable, well recognized part of the academic body, and, regarding the division of labor, the pyramid is substantially wider and a quite a lot shorter than the ones we can observe at Giza.

A discussion of the lepraMap project in context of the research done within historical information science will be presented later in this thesis, in chapter 4.

2.3 The problems to be solved

Why should historians be interested in using information technology in the first place? In his 2001 book, Lawrence J. McCrank argues: "History as science is an information science. Its specialty is informing the present about the past" [5]. To study history is, to a large extent, to study sources, to interpret the information contained in the source in the context of the researchers project, and to communicate the results to other researchers and the general public. It is easy to envision how a technology that specializes in organizing, storing, analyzing and manipulating data and information could assist in each of these steps: solving problems, facilitating the process and providing a formalized

structure for the result.

After formulating the definition quoted in section 2.2, Boonstra et. al go on to define the problems the discipline of historical information science is concerned with developing solutions for[3]:

- *Information problems of historical sources.* Historians work in a world where the data they work with, their sources, are incomplete, unreliable and unformalized to a much larger extent than in many other fields. In addition there is a perceived need for firm separation between the sources themselves and the annotations the researcher adds to aid his understanding and reflect his interpretation, in order to prevent the work of formalizing a source from invalidating the information retrieved from it. There is also the fact that many historians only have a vague idea of what information they are interested in and how it should be structured at the start of a research project.

- *Information problems of relationships between sources.* A historian can not be certain that different sources will have a consistent way of representing information; a name can be spelled one way in one source a, another in source b. A city placed in one country in source a might be considered to be in a different country in source b, and so on. There is also the problem of lack of temporal consistency: names and definitions change over time, a title has one meaning in 1780, a different one fifty years later.

- *Information problems of historical analysis.* Boonstra et al. contend that there is a need for field-specific analytical tools in historical information science, for instance for inference and multilevel regression. Tools used for data-analysis in historical research today are often borrowed from the social sciences, and many might not be suitable for the context of historical research.

- *Information problems of the presentation of sources or analysis.* the result of digital data analysis needs to be conveyed in a manner suitable for further research. The representation of changes over time is a difficult problem which falls firmly under the domain of historical information science.

Note the focus on sources in the list above. Interpreting sources is, in many ways, the core of historical research.

We have presented Boonstra et. als. definition of the problems historical information science is concerned with. As we move on, we will examine how these

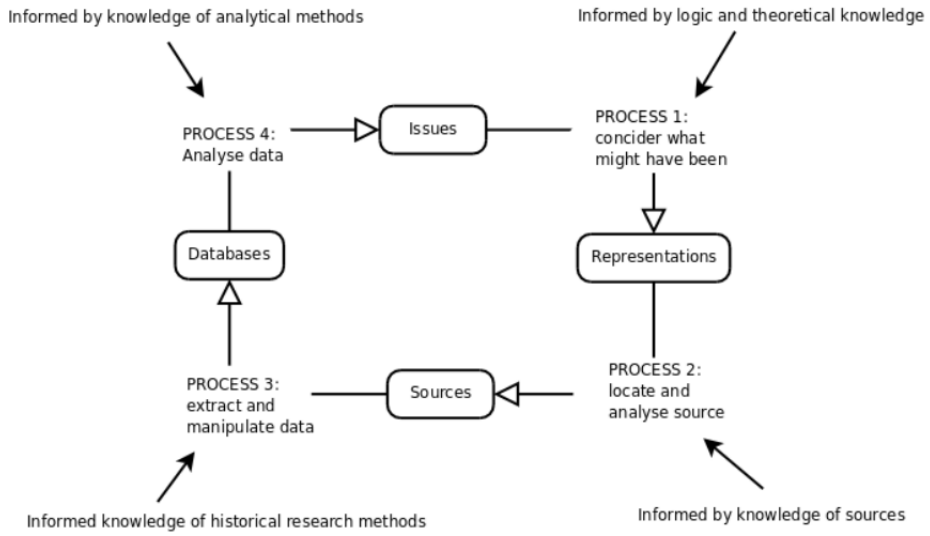


Figure 2.3: Harveys process of historical research

issues are addressed in a research process, as presented by Charles Harvey, a prominent prominent researcher in the field, Charles Harvey[6]. His description of the process is illustrated in figure 2.3.

In Harveys process *issues* of interest are identified by the researcher through study of literature and other sources, then compared and contrasted with the current *representations* of the information; the existing interpretations and research results related to these issues. If further study is deemed to be warranted, the researcher locates and evaluate sources, and as he studies them he they are incorporated into a growing *database*. He then analyzes the data, hopefully uncovering further issues of interest and paths of inquiry. The process then loops; more sources are found and added to the database, analysis continues and so on. Of course, this is just one of the many work-flows possible, but it seems representative of how the technology has been used.

If we compare the types of problems identified by Boonstra et. al with the process outlined by Harvey, we see some overlap. The "Information problems of historical sources" and "Information problems of relationships between sources" are problems related to the process of extracting and formalizing the information contained in the sources and adding it to the database (process 2 and 3 in Harveys figure). The "Information problems of historical analysis" and "Information problems of the presentation of sources or analysis" are related to the analysis of the data contained in the database and the uncovering of further issues to be investigated (process 4).

It is easy to see how the remaining processes in Harveys model could be facilitated by use of information technology. A search for sources (process 1) can be done by browsing bibliographic and archival databases. It could be said that the problem of facilitating the process of finding and examining sources through search engines or browsing could constitute a fifth category of problems historical information science should be concerned with, in addition to those outlined by Boonstra et. al.

Returning to the initial question; "why should historians be interested in using information technology", it is clear that historical researchers are faced with an array of domain specific IT problems. However, the technology also has the potential of being a valuable asset if it can properly support the process of historical research and facilitate the work-flow of the researcher.

2.4 The tools to solve them

If historical information science is the solving of problems *"...with the help of computing tools"* it should be worth looking at what tools historians have considered to be useful in this regard.

Boonstra [3] places digital tools on a axis from *"Plain IT"* to *"Enhanced IT"*. On the plain IT side there is an attitude that generally available information technology, be it storage solutions, relational data base management systems, or data-mining tools, are adequate for the needs of historical information science. This view would be consistent with the earlier cited opinion of Lawrence J. McCrank that there is no need for a historical-information science. Tools developed for the general market will meet the needs of historians as well.

In contrast, enhanced IT, calls for the creation of custom made solutions to handle the needs of historical researchers. Examples of these kinds of systems include database management systems with special features for maintaining the relationship between the original source and added meta-data, and systems for charting and organizing archaeological excavations.

Of course, most researchers occupy a point in the spectrum between the two extremes, and the viewpoints are not mutually exclusive. However, the contrast between the positions is apparent in the literature, and to some extent, in the development of tools.

It might seem like the enhanced IT approach is more in tune with Boonstra et al.'s definition of historical information science, since it states that historical information science deals with problems specific to the field of history, but the definition only mentions specific *problems*, not specific *tools*. It is possible to

solve problems specific to historical information science with general purpose tools, especially if one uses flexible and expendable applications like modern database systems and open source software libraries.

So what tools do historians use? A prime example is the database. It is not hard to understand why a discipline so concerned with the storage, organization and analysis of data would place special importance on database management systems.

If the database is central to historical information science, and this science has a set of unique challenges and requirements, it might be logical to assume, if one accepts the enhanced IT position, that a custom database management system is needed.

One of the arguments for this position, made by a chief advocate, Manfred Thaller, was that using traditional relational data base management systems (like Oracle, PostGreSQL, or similar) encourages a *model-oriented approach*, where the transformation of a source into a data model is a prerequisite for later analysis[7]. The problem with this approach is that the model of the source is not the source itself, it is a conceptual description, during the design of which the researcher has to make hard decisions about which information to include, and how this information should be structured. Information has to be transformed into a normalized structure - where elements such as original spelling, currency, place-names and titles can be lost. As this process has to be applied at the start of the project it might also lead to the researchers preconceptions affecting analysis and resulting conclusions.

The alternative is a *source-oriented approach*, constructing what Harvey and Press [6] refer to as an electronic edition - a digital version of the original source, as close to the original in every way as possible. The advantages are obvious: the researcher does not need to make decisions about what to include or how to include it; the complete source is replicated, and its structure defines the structure of the database. This prevents the researchers preconceptions of the information from heavily influencing all subsequent analysis. However, the approach has its own set of problems; it is very hard for computer programs to assist in useful analysis of data which has not been normalized and structured.

To deal with this problem of formalization Thaller proposed a hybrid solution; part traditional database management system, part full-text and document retrieval system, thus integrating the source and model based approach. His implementation of this concept, CLIO (later expanded, internationalized, and renamed KLEIO), introduced in his 1980 article [8], lets the historian first replicate the source, and then use CLIO to build a fuzzy hierarchy based taxonomy

Figure 2.4: Working with Kleio through a graphical user interface

on top of this. Special focus is on facilitating changing requirements - groups in the hierarchy can be changed, split and merged during analysis, thus providing a solution to the pre / post coding problem. A screen shot of a modern version of Kleio can be seen in figure 2.4. While development of Kleio, expanded with a feature set into a "Historical Workstation", has been ongoing for the past 20 years, it has not kept up with the developments in the general IT and interest seems to have faded[2].

The importance given the discussion of plain IT vs. enhanced IT and source-oriented vs. model-oriented in the literature highlights the attitude of the research community towards IT - enthusiasm combined with caution. Tools are accepted, as long as they support the existing process of historical research, and care must be taken to avoid "contamination' of the sources the tools deal with. While several of the earlier ambitious projects seem to have, to a large extent, failed[3], the 2000s might promise change to the better. The availability of open source operating systems and libraries which can be easily extended and adapted to new uses, ubiquitous networking, powerful computers with near infinite storage capability, and a new technology proficient generation of researchers might be just what historical information science needs.

[2]The fact that the original version was published in German only and not made available in any other language for quite some time might also have contributed to the lack of wide spread adaptation.

Chapter 3

Requirements and Methodology

The chapter gives a brief outline of how the goals of the historical research project translate into requirements for the prototype, and a discussion of the methodology applied to the development process.

3.1 Research goals

Vollset has a long-lived interest in Norwegian leprosy research, and the current project, forming his PhD thesis, can be seen as an extension of his MA-thesis on leprosy in Norway in the 19th century[9]. Bergen was a focal point for leprosy research, and from 1860 the national leprosy apparatus was led by the research community formed around the three leprosy institutions in the city[1]

For his doctorate thesis Vollset took a more international approach - recognizing that research communities do not exist in isolation. Researchers read medical journals and books written by people working all over the globe. They write letters, attend congresses and travel to other institutions, bringing their knowledge and understanding of problems with them and returning with new information and ideas. The main goal is to examine this flow of information and achieve an understanding of how knowledge was produced and exchanged. Examples of questions he wanted to answer were:

- What were the canonical works, and how did these change over time?
- Which research programs were people involved in?
- Was there one or several research fronts?
- Were there groups of people who collaborated more than others?
- Which mechanisms facilitated the exchanges of knowledge?
- What happened in the tension between locally produced knowledge and scientific claims of universality?

To answer these questions an investigation of the available historical sources has to be conducted; information stored in archives, libraries and museums around the world. These include reports, correspondence, books, papers, membership and employment records, lists of library content, congress proceedings, records of experiments and procedures, etc. Some of these have been indexed, some are even available as a digital library. However, the vast majority is stored in the original physical form, and the only way to access them is to travel to their location.

Contained in these sources, or artifacts as they are referred to within the lepraMap application, is information that should answer the initial questions posed

[1]The leprosy institutions in Bergen were the medieval St. Jorgens Hospital (now a museum), the research hospital Lungegardshospitalet (inaugurated in 1849), and the nursing home Pleiestiftelsen No. 1 (opened in 1857)[9].

by the research project - the problem is to access and structure the information in a way that makes it possible to see the big picture. It is the facilitation of this process which forms the major requirement for the lepraMap system.

3.2 Requirements

So how do the goals of the historical research project translate into requirements for the lepraMap system? To answer the question we first have to establish the three basic tasks of the system; the formalization and persistence of data, data entry, and presentation.

Formalization and persistence. The system should provide a way of storing the information collected during the investigation of historical sources. The method selected for storing the information should lend itself to analysis and facilitate sharing or re-purposing of the information.

Data entry. The system should be designed in a way that makes the job of data entry easy and efficient, with as little distraction from the real work, analyzing the sources, as possible.

Presentation. Information stored should be presented in a way that allows the user to quickly grasp patterns, and to interactively explore these structures. Importance is given to letting the user see how the structures develop along the time dimension.

In the following section these components will be examined, the requirements for each detailed, and the implications for the design of the system reviewed.

A main capability of the system is the storage and structuring of information, making the generation of a data model a core task during analysis and design. The data model must be able to cope with several challenges due to the nature of the sources.

Fragmented and heterogeneous data sets. Some sources are incomplete, others have unformalized information (or a method of formalization which is not constant over time).

Availability and quality of data varies. Some institutions have kept immaculate records of library content, correspondence, employees and research results. Other institutions have no readily available records at all.

Unknown entities and attributes. We will not have a complete understanding of which information the system will be required to store at project start. The same goes for the possible values of attributes.

Change over time. Tracking how things change over time is of obvious importance when modeling a historical source - but this also poses a challenge for the design of the data model; the same institution can be known under several different names in different time periods, or the same name could refer to several different institutions, and people may change their opinions over time, for instance in light of new information.

The requirements state that it should be possible to start data entry without a complete understanding of what information we want to store. The work flow will be much like a person taking notes while reading a book: you could envision a very structured (and a bit obsessive compulsive) person sitting down with stacks of printed forms, with headings such as "interesting person" with fields for date of birth, name and sex and "important event" with attributes such as date and location.

Our imaginary researcher would write his notes while reading, filling out forms one at a time, but he might discover that this approach, while very organized, creates more than a few problems.

Let us assume that after filling out his 32nd "interesting person" form he learns that where a person went to school is of great importance. Does he just add a new field to his "interesting person" form? This form is now inconsistent with the ones he completed previously, which could create problems later on. He might discover that using the type "doctor" to reflect profession was a mistake; "medical doctor" is a more useful term, since the doctorates are given in many fields, or he might be interested in differentiating between bacteriologists working in a laboratory environment and a pathologist mainly involved in bedside medicine. What does he do? The changes will not apply to people described in his previous notes, creating chaos. He could go back and update the information for every person, but this would be a difficult and time consuming task.

These issues are often referred to as the "pre-coding vs. post-coding problem"; when are the categories used to structure information decided; at the initial stage, when designing the tool, during data collection, or afterwards, when the researcher has the complete overview? The data model employed in the solution will have to account for these problems, and provide solutions that will ensure that the data will be structured and consistent, making analysis possible. There are advantages and disadvantages to all approaches, and this subject will be discussed in more detail in chapter 4.1.2.

The second component of the system is data entry. It would be possible, although very inconvenient, to do this by directly updating the database through SQL statements, but the requirement specifies that the method of data-entry

should be as easy and efficient. This is not a trivial point. The user will be adding information to the lepraMap system while he is focusing on understanding and evaluating the source he is working on - any distraction from this is a disadvantage, which can only be out-weighted by the benefit the user gets from having the information available in our solution instead of as notes on paper. If the data-entry workload is too great the overhead related to using the system will outweigh the benefits, and the project will be a failure. Ideally entering information into the system should be as natural and easy as using a notepad.

The third component is directly related to one of the main goals of this project; giving the user insight into the collected data. In its most basic form this presentation could be in the form of database tables - but as a way of providing an overview of complex information this leaves much to be desired, especially when using a model based on associations between entities, often of a many to many cardinality. A natural solution to this problem is opting for graphical visualization; drawing a model representing the data stored in the database. This GUI would need to satisfy several requirements. It needs to display a large amounts of data in a form that can be easily grasped by the user, it should show how the structures change over time, and allow for the user to explore the data, manipulating the structure and filtering what information is displayed at any time.

3.3 Development methodology

After establishing a rough idea of what we wanted to accomplish we had to make some choices regarding how we would reach our objectives.

Our idea of what the final design should look like were unclear. While some strategies, such as a general structure of a back-end database and a front-end GUI client seemed obvious, there are a multitude of different techniques that could be used to achieve a result satisfying the initial requirements. Should the data be presented as quantitative aggregations? Tied to a geographical dimension? Diagrams and bar charts? Animations? We had to select an approach that would both suit our constraints and meet our requirements, but we were still unsure about the very nature of the application we were constructing.

In the end we decided on a strategy: we would create a prototype system - not primarily as a way of gathering requirements, but as a method of exploring possibilities and to provide a focus for further discussion.

After looking at some pre-existing solutions for information visualization we decided that our prototype would have to be in the form of a custom application

- but still based on existing frameworks and libraries in order to cut down development time. The main reasons for making this choice was that we wanted to be able to explore the problem domain without being restricted by the design choices made in third party software, and that we wanted our system to be tailored to the projects specific needs, especially the visualization of the time dimension.

It was obvious that our main constraint was going to be time; we had to get a working system up quickly to meet our deadlines, with just one developer working on his spare time. It was also clear that we to a large extent would be learning as we went along; trying to write a complete specification up front and then code to this specification, as in a waterfall based project, was probably futile. Again, doing an initial prototype seemed like a good idea.

The shunning of up-front design as a development strategy is very much in tune with the various agile development methodologies[2]. Opinions of what exactly constitutes "agile development" are varied, but the shared principles are: accept, and welcome, continually changing specifications and requirements, agree that software you can actually run, as opposed to software planned in some specification document, is the only valid measure of progress, and that close cooperation and rapid feedback loops between users, owners, managers, and developers is essential.

There are also several different approaches to running a "prototype driven" project, and much research has been done on the cataloging and evaluation of various techniques. While prototypes can range from some scribbled notes on the back of a napkin to multi-million dollar pieces of machinery, they ultimately serve the same purpose - to assist in the design of a solution, and they can be investigated as different applications of the same technique.

Our basic strategy, using the prototype as a tool for exploring how best to solve a problem, has been referred to as *exploratory* prototyping, as opposed to *evolutionary*, where the prototype is an early version of the finished product, and *experimental*, where the prototype is created to test one well-specified design idea[11]. The choice of starting the code base with an exploratory prototype means that we could disregard some elements (like completeness, perfection of interfaces, and documentation) in favor of others, allowing us to progress and iterate faster. The quick development and design iteration is vital, since it encourages a process where discussions lead to design changes, which again lead to new discussions and new development, and so on. This approach suited our

[2]Developed by several groups from 1995 onwards, and codified in the Agile Manifesto in 2001 [10].

time and resource constrained project perfectly.

Y. K. Lim, Erik Stolterman and Josh Tenenberg expand on the idea of prototypes as tools for exploring possiblities in their 2008 paper, "The Anatomy of Prototypes"[12]. They provide the following definition of the "anatomy" of prototypes:

> *Prototypes are filters that traverse a design space and are manifestations of design ideas that concretizise and externalize conceptual ideas.*

The understanding of prototypes as filters is an important insight. At the start of a project, especially one where the method of satisfying a set of requirements is unclear, or there are many options available, the "design space", or the number of possible (or impossible) conceptual solutions, is large. This was the situation we were faced with at the start of the lepraMap project. The prototype can then act as a filter, cutting through the fog of possibilities and letting us home in on the best solution.

When trying to understand the nature of a prototype it can also be useful to apply other parameters; Jakob Nielsen suggest *high-fidelity* vs. *low-fidelity* and *horizontal* vs. *vertical* in his 1994 book[13]. Lim et. al use the corresponding terms *resolution* and *scope* in their paper.

The high / low fidelity axis describe the degree of similarity between the prototype and the finished application. An example of a extreme low fidelity prototype would be a user interface mapped out with pen and paper, a high fidelity prototype could be an application indistinguishable from the 'real thing" during casual use.

The horizontal / vertical axis describe the "breadth" of the prototype - how many of the features of a complete system are included. In a horizontal prototype one level of the application, in most cases the user interface, is described almost in full, but other levels such as controllers and persistence are represented by mock-ups. A vertical prototype selects one key feature set, and implements this from top to bottom, delivering a fully functional representation of this cross section.

The advantage of seeing the prototype in the light of these parameters is that it gives a clearer idea of the problem prototyping is being applied to, and how much work will need to be invested in development. Lin et. al. refer to this as the *economic principle of prototyping:* "The best prototype is one that, in the simplest and most effective way, makes the possibilities and limitations of a design idea measurable"[12].

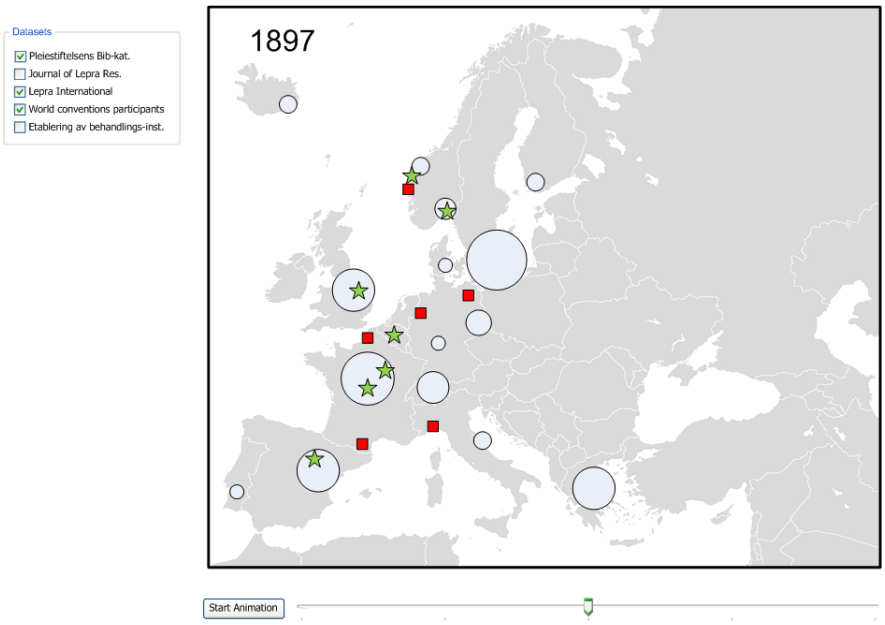


Figure 3.1: A relativly early stage photoshop mock-up of the data explorer.

In this project we combined these different varieties of prototypes. We started out with a low fidelity, horizontal prototype of the user interface, both of the data-entry and the data-exploration client[3]. These started out as pen and paper sketches, progressing to PhotoShop mock-ups and simple animations. A early mock up is shown in figure 3.1. Note that we at this point were exploring a visualization system based on geography, with map annotations to reflect the information retrieved from the sources. This was only one of several approaches we tried and subsequently discarded.

Figure 3.2 contrasts a late stage mock-up with the finished data exploration prototype. We are now quite close to the design implemented in the hi-fidelity prototype, with all the major components of graph-based visualization, color coded vertexes and nodes and chronological animations present.

The pen and paper sketches served as a focus of our discussions on goals and requirements for this project, and was also used to quickly give people with interest in our project an idea of what we were doing, as we used these during meetings with other historians to get external feedback at an early stage.

Our solutions, especially with regards to the data-explorer, changed several

[3]A decision to separate the functionality into two different applications was made to speed up implementation by allowing Vollset to start testing out the data-entry client and data model with real data while the data exploration client was still unfinished, see discussion in section 4.

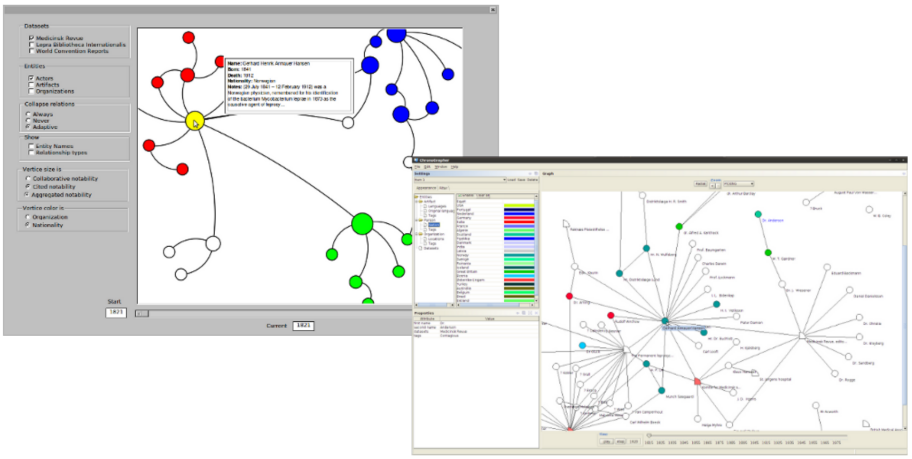


Figure 3.2: Sketch of the data-explorer GUI on the left, screen-shot from the finished application on the right.

times during this phase, as old ideas were discarded and new tried out. This also fits well with the idea of prototypes as filters on a design space: the design space for the visualization module is larger than for the more "run of the mill" data-entry client, or so we thought while building the application[4].

The hi-fidelity prototypes also proved their worth; by revealing significant problems in the initial data model (and subsequent iterations) and identifying some features in the visualization client that deserved a closer look. Allowing us to test the user experience with real data also proved invaluable.

The next chapter will discuss these, and other design issues discovered during the implementation and evaluation of the prototype.

[4]As will be discussed in the next chapter, we might have had more options related to data modeling and entry than we initially believed.

Chapter 4

Prototype

The chapter begins by discussing how the design of the prototype evolved, before and during implementation - the focus is on the establishment of a satisfactory data model. It then presents the finished prototype, and gives a brief discussion of features and implementation technologies.

4.1 Prototype design

Having decided that the development of a prototype was a good way of establishing how the final system should be constructed, we needed to choose technology for the implementation.

4.1.1 Architecture

As mentioned in the last section we assumed that a relational database would be the natural choice for the back-end, giving us a structured data model and a persistence layer. The system components, the visualization and data-entry clients, presented us with a wider range of choices of implementation frameworks.

The main choice was whether to use a browser based interface or a thick client. Both have advantages and disadvantages. A web front end provides a single point of contact with the user and, barring browser incompatibilities, a single environment to code against. Rolling out updates and new features also becomes easier to manage, and distributing the application to other users becomes trivial. The last point would be more important in a "production level" version of the software, since the prototype would only have one main user.

On the other hand, a thick client platform offers the ability to easily implement rich interactivity and graphically intensive interfaces - although with standards such as HTML5 and webGL opening up new opportunities for rich web applications this is becoming less important. The option of running the client locally can also provide an advantage, letting the user work on a local database without a permanent Internet connection. This is important, since many of the sources are stored in archives located in different countries, where the continuous Internet access demanded by most web-based front ends can not be taken for granted.

We decided to go with a thick client approach, both for data-entry and visualization. Ease of development, and the fact that the off-line feature proved to be essential as Vollset traveled extensively, made it the better option for our project. If the number of potential users was higher, or if more time could be allocated to development, the choice might have gone the other way.

We had already decided to split data entry and data exploration into two clients to make testing of the data model possible while work was still ongoing on data exploration. The planned prototype therefore consisted of three independent components; a database, forming the back end of the system, a basic data-entry client, and visual data explorer, as shown in figure 4.1. In the next sections I will present and discuss the design and implementation of each of

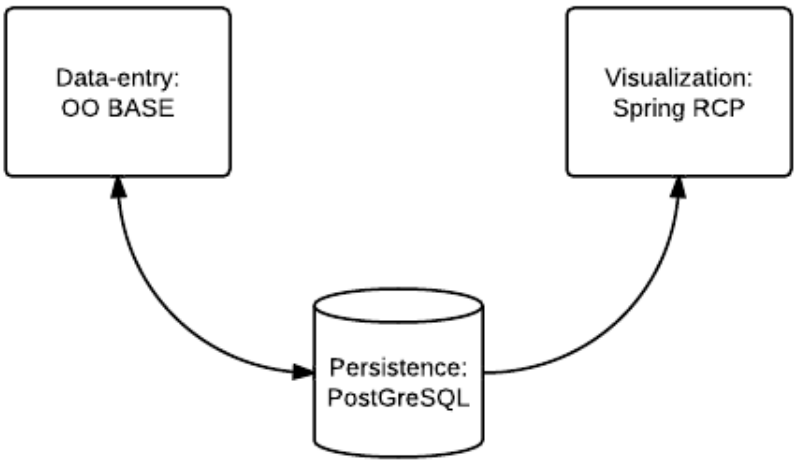


Figure 4.1: The architecture of the prototype components.

these components.

4.1.2 Data model

A typical database design process starts out with a set of requirements for solving a specific problem. What information do we want to store, how can it best be structured and organized, what data manipulation strategies are called for and so on. These, together with knowledge about the problem domain is then transformed into a model which can be managed by a database system.

Several different techniques for conducting this transformation have been developed, two of the most well known are Entity Relational Modeling (ER), established by Peter Chen in 1976[14], and Object Role Modeling (ORM), formalized by Terry Halpin in his 1989 PhD thesis[15]. While there are fundamental differences between the two approaches, the design process they support share the same basic structure; initial analysis leads to the creation of a conceptual model of the data, and formalized rules are then applied in order to transform this conceptual model into a logical model which can be implemented in a relational database system.

ER modeling was chosen as the design tool for the prototype data model, mostly because our previous experience with database development has been with this methodology. As will be discussed in section 6.1, it might have been better if we had considered other alternatives. However, it seemed like the logical choice at the time.

The process and syntax of ER modeling has been heavily extended and modified since Chens paper, but the basic approach is still the same; *entities* ("things" in the real world, such as a specific person) form *entity sets* (groups of entities sharing characteristics, such as "employees"), which have *relationships* with other entity sets.

The ER framework provides a formal notation for describing entities and their relationships. It also formulates rules for converting a conceptual model into a logical model, which can then be implemented in a database. However, at the very beginning of the analysis stage, when the designer has to answer the basic questions of what information he wants to model and what entities he wants to include, ER and similar methodologies do not provide much assistance. Some frameworks, such as ORMs Conceptual Schema Design Procedure (CSDP)[16], do purport to support this early phase, but elsewhere the assistance provided amounts to vague "best practices".

There might be good reasons for this state of affairs; creating formalized models of the real world is difficult, and providing a generalized, catch-all process for developing these even more so. Philosophers, from Thales and Aristotle to Kant and Wittgenstein, have been grappling with this problem, and why it is such slippery an issue, for the last couple of thousand years.

As an example of a framework supporting the design of a conceptual model I will use the first part of Tobey J. Toery et. al.s suggested design methodology for developing relational databases using EER (Enhanced[1] Entity Relational Model)[17]. Part one of the process constructs a conceptual model (the rest of the process is concerned with transforming it into a logical model), and consists of the following steps.[2]

1. Classify entities and attributes
2. Identify the generalization hierarchies and subset hierarchies
3. Define relationships

The first step is classifying entities and attributes. Toery et. al. provide some tests that can be used to identify what should be stored as entities and what is better off as attributes, but as previously noted, has less on how to find candidates for these in the first place.

[1] The enhancements include support for inheritance and unions.

[2] A fourth step, the "integration multiple views of entities, attributes and relationships" is a part of the original process, but this mostly applies to the design of information systems where there are multiple models describing the same domain. As this was not the focus of this project, it was not included. The subject, however, will be revisited in chapter 6.

If we consider the historical sources there are myriad of possibilities for both. A quick investigation reveals that the artifacts contain places, events, time-periods, methods and techniques, people, institutions, organizations, diseases, periodicals, correspondence, policies, governments, locations, countries, professions, languages, treatments, theories, experiments, peer review, policy recommendations, etc. What should be included and what should be omitted altogether, and how to structure what we deemed necessary to model?

It helps to keep in mind that we are not creating a model of the real world. It is rather the creation of a a model of the model implicit in the researchers accumulated knowledge of the domain. David Kronke states the difference in his book *Database Processing*, where he notes that the developer should not ask himself "does this model accurately reflect the users world?". A better question is "does this model accurately reflect the users' perceptions and their mental models of the world?" [18] - in this case, the historians perceptions of what he has learned by studying his sources.

What does this mean? If the model is tied directly to the users conceptions, it is his needs and requirements that matter, not any objective idea about "how the world is". There is a parallel to the source-based / model-based axis discussed in section 2.4 here. The meta-model strategy places us beyond model-based territory; we are not attempting to model the sources, but *a specific and subjective conception of the information contained within.*

If our primary design guide is our users method of investigating his sources, and not the sources themselves, we should look back to the research goals we outlined in chapter 3.1 to find the core entities in our data model. Reviewing the goals we see that the questions posed are mostly concerned with people; how they organize themselves in groups, how they collaborate on research projects, share experimental data, become supporters of a theory, and publish journals and books attempting to convince others.

Since these will form entities in the logical data model it will be time consuming to alter or expand the set after the application has been implemented. It is therefore important that they are general enough to encompass the needs of the project.

After studying research goals and sources, and keeping the need for flexibility in mind, three core entities emerge:

Person. The researchers, doctors, politicians, priests and patients etc. Examples of attributes are name, date of birth, date of death or nationality.

Artifact. The papers, books, correspondence, lab-results, journals, etc. produced by members of the lepra-research communities. Examples of at-

tributes are title, date of creation and language (and original language if the text has been translated),

Organization. Committees, institutions, clubs, editorial boards, governments; any institution, organization, club, business, board, conference or other gathering of people, physically or by shared knowledge or goal, mentioned in the sources. Examples of attributes are names, formed date, dissolved date or location (if the organization, like e.g. an university or a hospital, has a physical location).

Moving on to the next step in Toerys work flow, we identify generalization hierarchies. One could envision subdividing people into researchers and medical professionals, organizations into institutions and groups and so on. However, the need for doing this arises if super and sub-groups have differing attributes, which was not the case in our project. We could, however, have gone in the opposite direction, and generalized our data-model further, supporting the persistence of any entity and its relations. We returned to this issue later on; see the discussion of the simplified data model later in this section.

For now we will continue describing the process which led to the initial model, moving on to the creation of relations between entities. The attributes describe some aspects of the entities, but much of the more interesting information is modeled as associations, letting us know which people belong to which group, which people collaborated on producing artifacts, etc. Since relationships have attributes as well, they also indicate information such as start- and end time and the direction of the association.

When we define these relationships the result is that every core entity has a many to many cardinality relation to every other core entity, in addition to a recursive relationship with itself. The latter is necessary in order to reflect that people have relationships with other people, without this being facilitated by an institution or the production of an artifact. Artifacts and organization have similar relationships with other entities in the same set.

We felt that most of the information of interest in this project could be described using combinations of these three entities, and that the complexity of the model would be manageable. The correctness of this assumption will be discussed later in this chapter. If we take the process above and create a model of the entities and relationships we have defined, we end up with the conceptual model shown in figure 4.2; three entities, each with an association to the other entities, and a recursive relationship with itself. While looking deceptively straightforward, the model hides a lot of complexity.

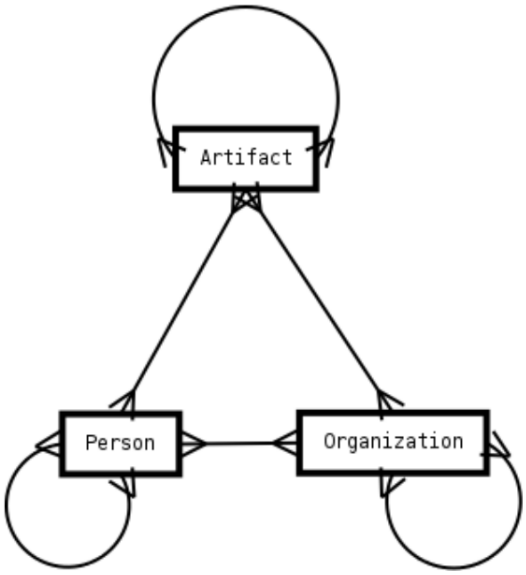


Figure 4.2: Conceptual data model

First of all, the conceptual model in figure 4.2 is not complete. We have focused on the core entities, omitting others, e.g. the ones modeling locations, events, the dynamic tagging system for ad-hoc categorization, the data set entity associating each core entity with a reference to the digital or printed source, etc.

Second, even if this model was complete it could not have been directly implemented in a relational database system. As mentioned earlier, the conceptual model needs to be converted to a logical model by applying a set of rules. If this conversion is performed, we end up with a rather large and much more fragile construction. A slice of the larger model is presented in figure 4.3, showing the drastic complexity increase from the original conceptual model. The chief reason for the exploding complexity is *associative entities*; examples from the model above are artifact-person, person-person and person-org. These are constructed to maintain the many-to-many cardinality relationships between the main entities and the tables that where omitted for clarity in the conceptual model, along with myriad other tables that take care of less central functions. These are responsible for mapping geographical locations and chronological events, providing persistence for the tagging feature (discussed in section 4.5) etc. While it provides a possible solution to our requirements, there are several obvious problems with the resulting system.

A complex data model increases the complexity of the systems using it. The

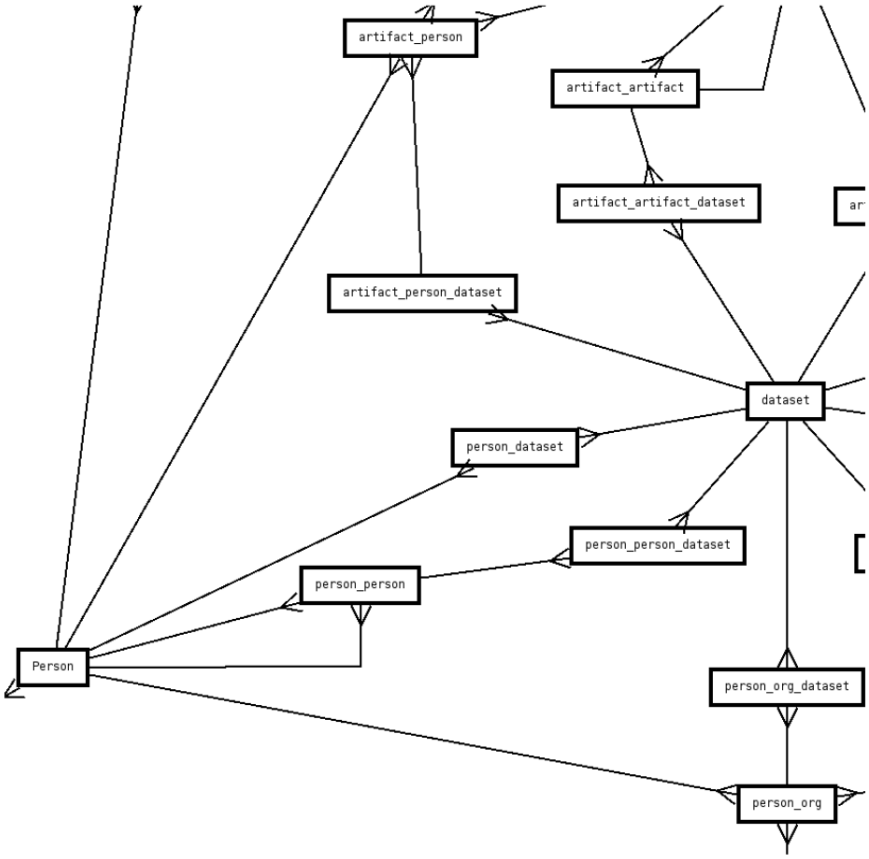


Figure 4.3: A section of our first logical datamodel.

data-entry client becomes more difficult to implement and maintain as it has to deal with the associative tables, increasing the number of bugs and decreasing the number of features that can be implemented. The complex structure would also negatively affect the data exploration client. In the latter instance the effect might be somewhat mitigated by leveraging one of the ORM (Object Relational Mapping) frameworks on the Java platform. However, the lack of a generalization of vertexes and edges in the graph described by the data model would still mean that devising a system for translating these into a more manageable structure from a data visualization standpoint would be that much harder.

This model also leaves us with a design where the number of tables grows exponentially with the number of core entities added. If the requirements change to necessitate the introduction of a new entity this would demand the creation of one table representing the entity, three associative tables representing that entitys relationship with the others, in addition to the tables required for data set association. For each new entity the number of associative entities needed grows. The design clearly does not scale well.

After considering the above, it became clear that finding a more flexible solution for our prototype would be essential, as we had uncovered a new requirement expanding on the one identified in the former chapter: The logical data-model should be as simple as possible, to facilitate the implementation of the data-entry and exploration clients.

The resulting simplified data model is shown in figure 4.4. The new model is in essence a persistence implementation for a graph; a set of vertexes connected by edges. In our system, entities in the conceptual model represents the vertexes while their relationships form the edges of the graph. The graph implemented by the model forms a *directed multigraph*: directed because the direction of a relationship between two vertexes (an article being *cited by* another is quite different from it *citing* another) is indicated, multi-graph because the same two vertexes can have multiple relations (a person can be both a member and a officer in an organization, and these relations can change over time).

The main effect of this change is that it allows for the generalization of the core entities. The vertex entity takes care of the implementation of relations between the core entities (which have been reduced to attribute sets extending the vertex entity). This decreases the number of tables in the logical model significantly, since it also allows for the creation of a single entity set to maintain the associations between the entities.

The change takes care of some of the problems identified in the old model.

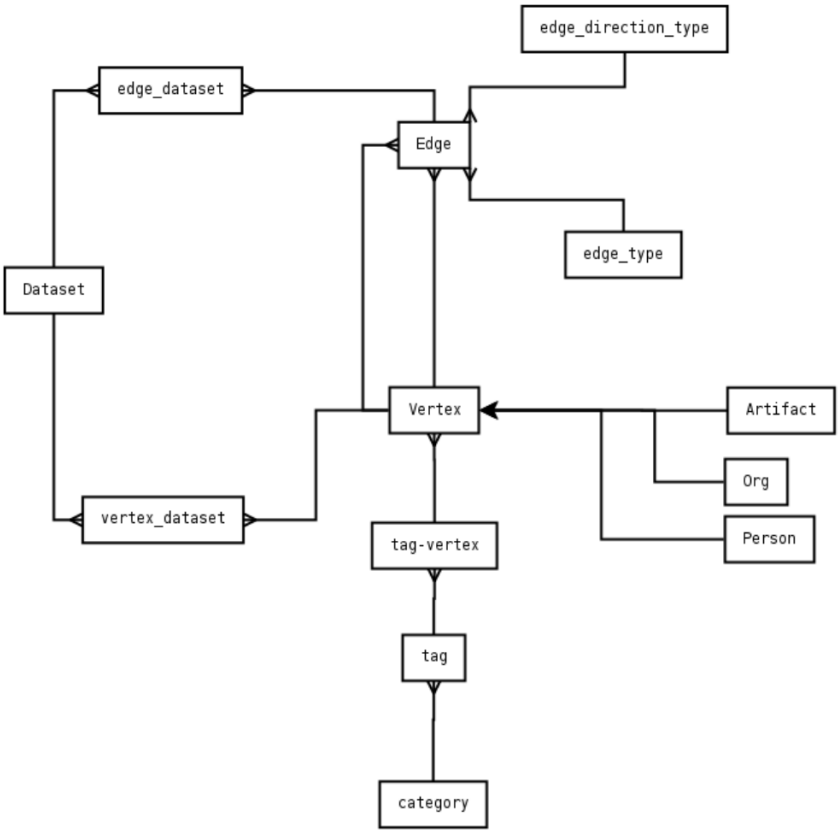


Figure 4.4: A simplified model.

New core entities can now be added without seeing an exponential growth in the number of tables, and we reduce complexity and facilitate a better program structure in both the data entry and the visualization module. However, the work of implementing this new model as a persistence layer for the prototype was never completed, since we by then had discovered both a new set of problems and what looked like a better approach. The problems are discussed in the prototype evaluation (section 5), the proposed solution in chapter 6.

The last sections have dealt describing how the design was formulated before and during the implementation of the prototype. We will now move on to the functionality of the hi-fidelity prototype system.

4.2 Prototype implementation

4.2.1 Data Editor

After getting a database up and running the main priority was the creation of a mechanism that would permit the user to enter data. During implementation we had used a set of SQL-DML scripts to test the data model, but this clearly would not be sufficient during real world use.

The choice of implementation tool was made based on the rapid development priority. I had earlier experience with using Microsoft Access™ to rapidly create graphical CRUD (Create, Read, Update and Delete) interfaces to databases. However, since Access requires the application to be installed for use as well as development we chose to avoid a proprietary product. The main open source alternative to Access is OpenOffice.org BASE ™[3]. BASE is not as feature rich or polished as Access, but we hoped it would prove to be sufficient for our needs. It, as Access, comes packaged with its own internal database, HSQLDB, a Java relational data base system implementation, but we would only be using BASE as a front end to the PostGreSQL database described in the previous section.

BASE, again in the same way as Access, takes a visual approach to interface design, letting the user drop controls from a palette onto an interface canvas. During implementation the fact that the toolkit is noticeably more basic than in its proprietary competitors became increasingly apparent. This made quite a bit of scripting required in order to glue things together[4]. It still allowed us

[3] OpenOffice.org was managed by Sun, but soon after the Oracle takeover it was forked into LibreOffice. After most of the developers left to work on the fork, ownership of OpenOffice.org was transferred to the Apache Foundation.

[4] During the implementation the restrictions BASE put on the structure of a GUI became a annoyance, which required less-than-elegant workarounds. In hindsight it is clear that BASE was not a great choice for even our medium-complexity application.

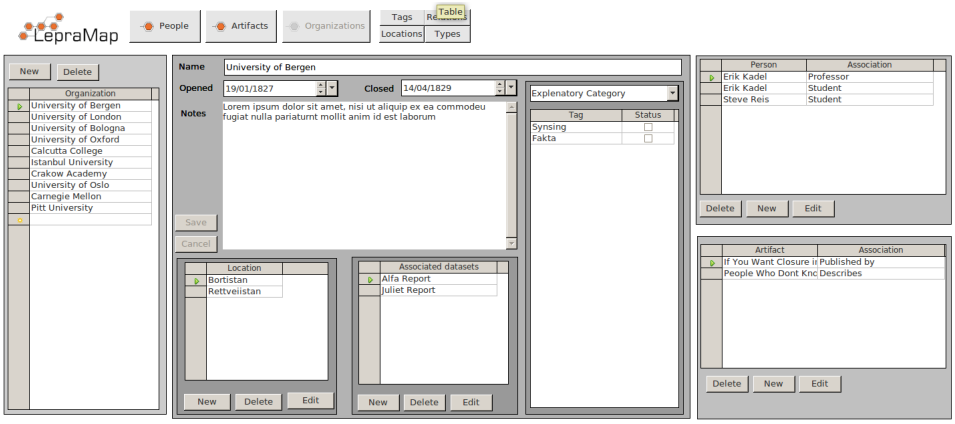


Figure 4.5: Editing organizations in the data entry client

to get a early version up and running in a relatively short time, and provided us with a good way of testing the data-model during real-world use.

A screen-shot of the data-entry client can be seen in figure 4.5. The data-entry client, as the database schema, was initially created as a specific solution for Vollset's research project; each core entity had a corresponding view in the browser. From these views the user can establish relations to other entities, e.g. a citation connection between two artifacts, or a "author" relationship between a person and an artifact.

Each time a new object is inserted into the database the user is prompted to specify the source of the information: both the *set* of sources, for instance a periodical or a library, and *specific* source, like a book or a journal article.

It quickly became clear that the lack of flexibility was a major constraint on the use of the data-model and application. Even if the basic set of entities provided were sufficient for the mapping of Vollset's sources, we ran into situations where it was clear that the models lack of flexibility was a limiting factor.

Some efforts were made to improve the data model, as described in the discussion of the model presented in figure 4.4, but we were still at the point where the inclusion of new attributes or entities required changes to both the database and the client - fine for a one off prototype with one developer and one user, but unacceptable in a "production-quality" system. We were caught in the pre-coding trap.

Another attempt at increasing flexibility was the addition of "tagging". Tags were implemented to provide a flexible way of categorizing people, artifacts or organizations entered into the database. Tags have boolean values, and can be created ad-hoc by the user as the need arises. For instance, the user might assign

the Medical Researcher tag with a true value to all people in that profession. The user is free to alter and combine tags after they have been created - combining pre- and post-coding in a way that would hopefully compensate for the otherwise static nature of the model.

4.2.2 Visualization

Together with the data-model, the visualization application was the part of the prototype we were most interested in exploring. We had a fairly good idea of how the data-entry client should look and work before we started planning the implementation - with this component it was different.

We had a short-list of features:

Display data and relations in an intuitive way. The client should be able to give a "at a glance" overview of the collected data, focusing on relations rather than attributes.

Be interactive. It should be possible to filter and manipulate the display of information, letting the user explore it interactively.

Show change over time. The display of information should reflect progress from one year to the next - since important insights can be achieved by focusing on the changes rather than the end result.

4.2.3 Technology

After deciding to do the exploration client as a desktop application, rather than a browser based one, the search for appropriate technologies for implementation started. We picked a few components early on:

Development platform/IDE: NetBeans

The three major IDEs (Integrated Development Environment) for the Java language are the Eclipse foundation's Eclipse, Sun's[5] NetBeans and JetBrains' IntelliJ-IDEA. While our usual choice of platform had been Eclipse, NetBeans has excellent support for creating Swing[6] based interfaces through the GUI builder formerly known as "Project Matisse". It provides a visual way of designing and implementing interfaces, allowing for rapid development, and proved to be a very useful tool in the construction of the prototype.

[5]NetBeans is managed by Oracle after its acquisition of Sun.

[6]Swing is the primary Java GUI toolkit, designed to build on and improve on the original Abstract Window Toolkit (AWT).

Application framework: Spring RCP

In order to facilitate rapid development we wanted to base our application on an existing framework for rich client software. The two biggest contenders in this field, if you limit the selection to the ones written in Java, are Eclipse RCP (Rich Client Platform) and NetBeans Platform, both based on the framework powering the GUI of their respective Java IDEs. While they are both widely used in a variety of different applications, our selection fell on a lesser known entry in the market, the Spring RCP. It has the benefits of being more light weight than its bigger competitors, and it focuses on the Swing graphics library rather than Eclipses AWT. The cons were a relatively new and untested codebase, for practical considerations still in beta[7]. Despite this drawback it served its purpose well.

Graphing library: JUNG

After an investigation of the alternatives we settled on using JUNG (Java Universal Network/Graph Framework) to manage the graphical part of visualization. It is, as the name reveals, a application library purpose built for the construction, manipulation and display of graph based visualizations. One of the major reasons for choosing JUNG was its excellent support for interactivity - letting users navigate and manipulate graphs in "real time". It also includes interesting libraies for doing lay-out and analysis of graphs, which would prove to be useful.

Object relational mapping framework: Hibernate

ORM tools are used to translate between the schema implemented in a database and the Java classes used to interact with them. While it is not always necessary, or desirable, to use ORM to communicate with the database, it can save a fair amount of coding and the result is also often code that is easier to read and manage. Several ORM libraries are available for the Java ecosystem, we chose Hibernate from JBoss for our project, mostly based on previous experience and the widespread adaptation of the framework.

[7]Despite showing promise further development of the Spring RCP application seems to have ground to a halt, with the last version released back in the summer of 2009. This makes it unlikely that this platform will be chosen for a future version of the system.

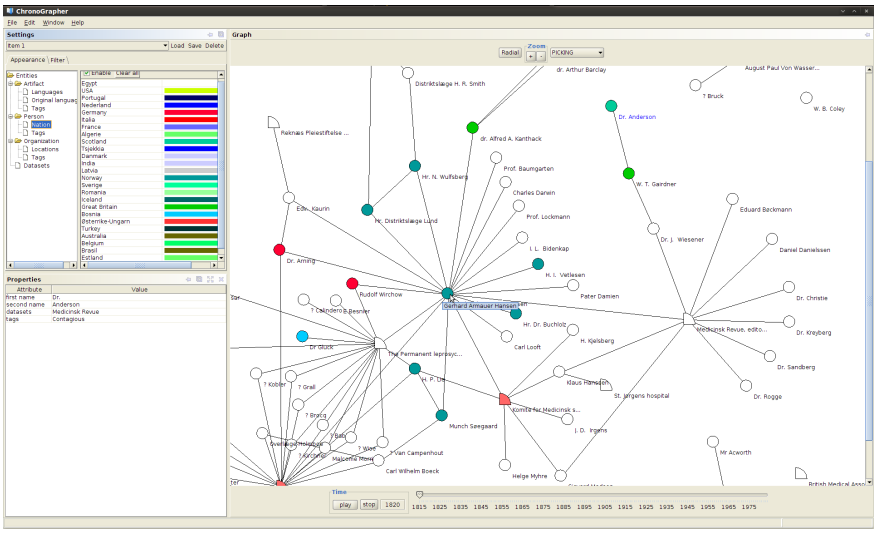


Figure 4.6: Visualization of data in the Explorer

4.2.4 User interface

Figure 4.6 above shows a typical screen from the completed visualization prototype. The screen is divided into three principal sections, settings, graph and properties, each detailed in the following sections.

Graph pane

This is where the main action happens. The panel shows a graph consisting of vertexes and edges, with the core entities of people, artifacts and organizations making up the vertexes, and the relations between them forming the edges. Note the differing shapes identifying the different entity-types. In the center of the screen we can see that the user has selected the vertex representing Gerhard Armauer Hansen.

The button to the right toggles the layout algorithm used, from radial through ISOM (Inverted Self Organizing Map) to a simple circle layout. The middle buttons lets the user change the zoom level of the graph view, and the one on the left toggles between two modes - ”picking” and ”transforming”. While picking the user can retrieve information about selected nodes or edges. Transforming lets the user change the layout of the graph through clicking and dragging vertexes and edges.

Below we have the visualization client’s method of showing change over time. Clicking start will trigger an animation showing the development of the graph at a year-by-year pace. This functionality uses the temporal range attributes set

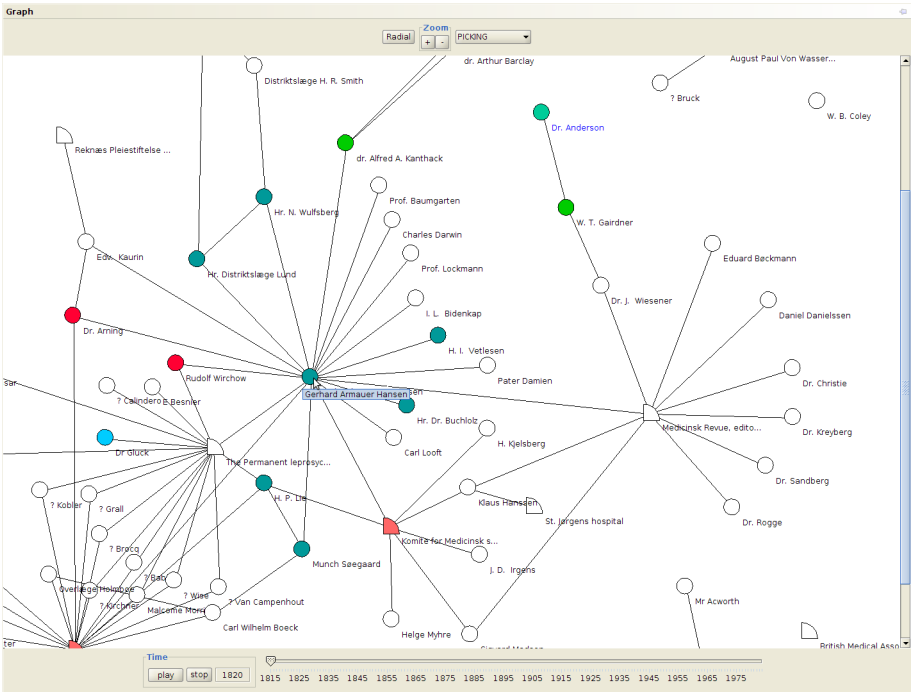


Figure 4.7: Graph pane of the data explorer

for entities and relations during data-entry, with some modifications. At first glance it is relatively easy to imagine the temporal range of a person; it begins when he is born, and lasts until he dies. When examined in more detail we see that it is not that simple - is it really interesting to have a 10 year old Armauer Hansen show up in the visualization, or does it just provide distracting clutter? How about artifacts? How long do they last?

In the end we had to code a set of rules to govern this, by focusing on the edges instead of the vertexes. We decided that a person was to be shown only as long as he was being referred to by others - being cited in literature, corresponding with fellow researchers, or participating in organizations. For artifacts we followed the same strategy, granting them presence as long as they were being cited or used, and for a set period after the last reference. We were also developing a feature which would allow these edges to slowly fade in and out for a set period before and after the association came into existence, but this functionality did not make it into the finished prototype before the decision to stop development and start evaluation was made.

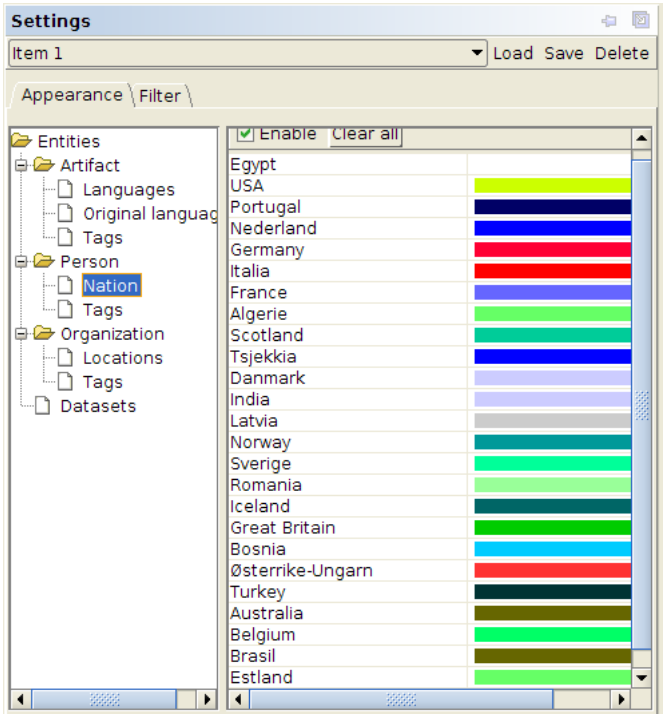


Figure 4.8: Settings pane of the data explorer

Settings pane

The settings pane has two main functions: first, it lets the user assign color codes to the attributes and tags used to categorize the entities during data-entry. As was mentioned earlier, the tag system was added to give the user an ad-hock way of adding information about entities. In figure 4.8 we see the "Nation" attribute selected; the colors for each nation can be defined by the user, and the result is shown in the coloring of vertexes in the graph pane above.

The second function of the settings pane is to allow the user to filter what is shown in the graph pane. This dialogue is similar to the appearance dialogue shown in the screen-shot, with the difference that instead of selecting a color the user either sets boolean values through the toggle of check-boxes to, for instance, limit the vertexes only to people where Norwegian has been set as the nationality, or to only display vertexes and edges originating from a spesific set of data.

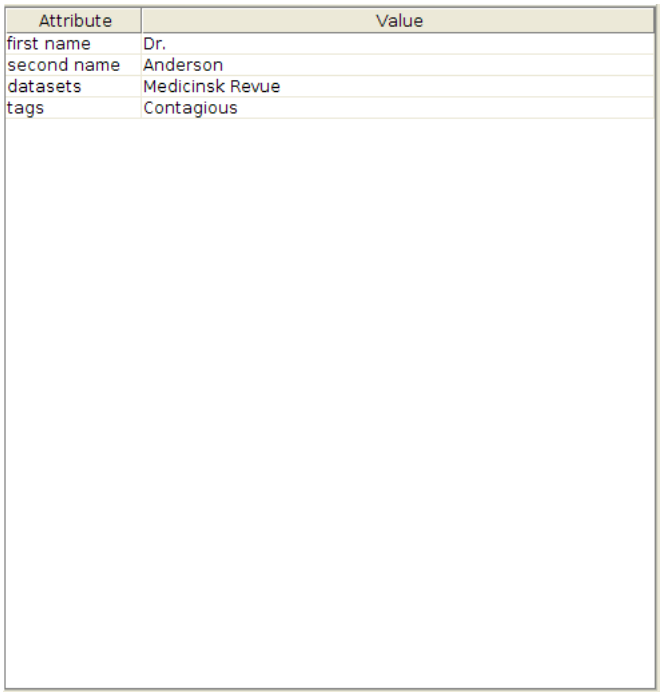


Figure 4.9: Properties pane of the data explorer

Properties pane

Properties simply show what value has been assigned to the attributes of the corresponding entity in the database. In the example shown in figure 4.9 we see that the person's last name is Anderson, he is found in the dataset Medicinsk Revue (a journal), and that he has been given a "true" value for the boolean tag "Contagious", identifying the disease concept he is arguing for.

The description of the data visualization client above concludes the presentation of the design and implementation of the prototype. The next chapter will discuss the result of the main function of the prototype; its ability to inform us about how we should proceed when designing and implementing the real, production-quality system.

Chapter 5

Evaluation

In this chapter an overview of what we learned from the design, implementation and use of the prototype is presented. It starts by summarizing an informal interview with the main user of the application, before discussing this in context of what was learned from the prototype implementation, and which implications the lessons learned will have for the design of a production quality version of the system.

5.1 LepraMap in the research context

In this section we will attempt to characterize the lepraMap prototype in terms of the research context presented in chapter 2. The main tool will be the two axes discussed in section 2.4; *source based vs. model based* and *plain IT vs. enhanced IT*.

When discussing source based vs. model based lepraMap could be described as either grounded in neither, or as an example of an unashamedly model based system, depending on the viewpoint. There is no attempt at recreating the sources at all, except for a few references. Instead the focus is on reflecting the researcher's interpretation of the information contained in the system, as shown in section 4.1.2. It could be argued that this flies in the face of warnings from creators of source based systems. The difference between the source, the representation of the source, and the researcher's later annotations become muddled.

The answer to this hypothetical criticism would be that an application can not be considered without examining the contexts of its use. LepraMap is not meant to be a permanent representation of a historical source, it is a subjective model of the researchers interpretation. In this way it could be argued that the graphs created with lepraMap has more in common with the narratives written as the end product of historical research than with the source material such narratives are based on.

When considering the second theme from chapter 2, plain vs. enhanced IT, it can be difficult to place this application. Its design as a solution custom built for historical information science might place it in the latter category, but if one examines the components it is obvious that the application incorporates preexisting, general IT libraries. Examples of these are the OO Base application used to develop the data entry client, and open source Java libraries such as JUNG used to build the data explorer.

It could be said that the growth of open source and modular programming languages such as Java are a perfect fit for an intermediate approach: allowing for the design of a custom made solution, but at the same time leveraging the power of solutions developed in "general IT".

In this way the development of the tool fits with the first part of Boonstra et als definition of historical information science: it deals with problems specific to history research and its sources. An interesting question is if it is compatible with the second part of the definition: will the result of the project be a solution which can be generalized beyond this specific application? The question is discussed further in chapter 6.

We can also examine the lepraMap tool in the context of the process of historical research as defined by Harvey and illustrated in figure 2.3. The data entry client aligns closely with process 3 (extract and manipulate data) by offering a formalized framework for storing the researchers conceptual interpretation of the information contained in the source. Features such as dynamic creation of tags and relationships are implemented to make this support as flexible as possible, as discussed in section 4.2.1.

The data explorer module of the lepraMap system, examined in chapter 4.2.2, obviously supports process 4 in Harveys figure: the analysis of the data which has been collected, but due to the design, it could also facilitate process 1: letting the researcher identify interesting sources.

As with any application, the real test comes when the end-user applies it to a real problem - how does the application perform, what is the user experience, and in what ways can it be improved. Answering these questions is particularly important when the system in question is a prototype - evaluation bringing later improvement is their raison d'être.

5.2 Interview with primary user

During the entire project I had extensive communication with Vollset, and many discussions about the themes presented in this thesis. As an informal evaluation method, after he had been using the prototype for some time, he was provided with set of questions designed to sum up his experiences with the project. Some relate directly to the use of the prototype, while others are more general in nature, focusing on the project as a whole.

The original questions are presented below, followed by a short summary of the most significant issues he raised. The unabridged answers are available in the appendix.

Question 1. **What led you to the conclusion that a digital tool would be helpful during your doctorate research?**

Summary of Vollset's response: Vollsets research raised complex questions regarding the dissemination of knowledge, and the quantity and richness of sources made it difficult to keep a good overview of the material. The focus was on uncovering relationships between several hundred researchers; one source alone had 840 different journal authors.

Question 2. **Could you describe the general attitude towards the use of digital tools in the research communities where you have worked?**

Summary of Vollset's response: General digital tools are a part of every day working life, database courses are offered, but focused on off-the self solutions. The widespread availability of Internet access has revolutionized digital archives, and made important resources available. There is little custom development, but sub-disciplines such as those dealing with demographics use specialized software as a matter of course.

Talk of including the development of a software solution in a research project met with initial skepticism, rooted in a worry that it would provide little return on invested time and distract from important work. Historians might also associate such projects with past quantitative research efforts they have methodological objections to. However, presentations of the lepraMap project have been met with enthusiasm and interest by historical researchers.

Question 3. **What were your expectations when you initiated the joint project?**

Summary of Vollset's response: Vollset had several concerns; if the project was a good idea, why was it not common practice to develop these kinds of applications? Would we get distracted by technical questions, would the data-model constrain "free thinking", was it possible to complete the project in the time available?

Yet the project was an opportunity to give specific foundation to vague terms such as "established over time" and "the international community". Vollset wanted to be able to drill down from the general to the specific; exchanges between specific individuals in specific locations at specific points in time.

Question 4. **Could you describe how you experienced collaboration during the development of the LepraMap application**

Summary of Vollset's response: Vollset refers to Michael Gibbons et al. argument that more and more research is being carried out in a context of an application, as practical problem solving, instead of governed by the paradigms in the traditional academic disciplines [19], and sees collaboration as a "example of transdiciplinary problem solving".

He sees some major differences between history and computer science, where computer scientists value generalization and normalization, historians search for the unique and complex. The prime historical tool, the narrative, does not lend

it self easily to formalization.

Vollset sees discussions during the development of lepraMap, especially on the subject of data modeling and formalization, as an interesting and worthwhile activity on its own, and notes that a common perspective emerged from two world views that had initially been divergent. It forced a "hidden" model of organization-person-artifact, which Vollset later generalized to arena-actor-artifact, in his research into view, which has given him a new tool in reading other historian's texts; what is the nature of their implicit data model?

Vollset describes how the rapid development of the prototype impressed him, but how it also made it difficult to identify the consequences of "feature space" decisions that were made.

Question 5. **During the development the focus has been firmly on the design and evaluation of the data-model. What, as a historian, are the most important requirements for this?**

Summary of Vollset's response: Vollset sees inherent problems with the transformation of a historical source into a data-model - since the ideal is to understand the past on its own premises something is bound to be lost in translation. Another major problem which might be inescapable is the time-consuming process of data-entry. He sees that this might be mitigated by progressing from a "the map of the world" metaphor used in the lepraMap prototype to a more "notebook" like approach - where the focus is not on completeness, but on reflecting particular interesting elements. He also sees the potential of such a tool to support collaboration.

Question 6. **Another goal of the project was the development of a graphical front-end. What requirements were most important for this component?**

Summary of Vollset's response: It did not seem intuitive or in accordance with a historians work-flow to separate the data-entry from the data exploration, although this might facilitate a separation of concerns between "data gatherers" and "data analyzers". Vollset is also worried that the separation might exacerbate the problematic constraints the imposition of a model has on the free interpretation of data.

Question 7. **The lepraMap project had the development of a prototype as a goal. Even so, what is the pragmatic usefulness of the application in its current state? What are its most critical deficiencies?**

Summary of Vollset's response: The major points Vollset brings up is that data entry is time consuming, and the application does not provide enough value to make the investment worthwhile. He again suggests moving away from completeness towards a notepad metaphor as a potential way forward.

Question 8. **What is the future of the lepraMap project?**

Summary of Vollset's response: Vollset sees, to his own surprise, a huge potential, but notes that the production quality version of the tool needs to show substantial improvements over the prototype. A reference project might be needed to show its potential to the scientific community, and only when it can support the collaboration between researchers will the tool truly be relevant.

Question 9. **What is the future of digital tools in history research in general?**

Summary of Vollset's response: While Vollset sees a need for historians to adapt digital tools, he feels that overcoming initial skepticism will be challenging.

5.3 Implications for design

The goal of this project was to use a prototype to collect requirements and explore the problem-space. So what did we learn, and what are the implications for the design of a production-quality system?

If we look at our experiences with designing the prototype, as detailed in chapter 4, as well as the comments from Vollset listed above, a new set of more specific requirements can be formulated.

Flexibility, flexibility, flexibility

The greatest failing of our initial design was the data-model. Locking up the definition of the entities we sought to describe in the logical database model was a blunder which had ramifications throughout the project. During use of the prototype the user became frustrated with the lack of expressiveness; as his understanding of the source grew, so did the desire to expand the basic set of

entities, or to describe new characteristics of these. LepraMap's attempt at dealing with this by including custom association types and a tagging mechanism was simply not enough.

It is clear that if the next version of lepraMap shall have any chance of meeting requirements, it needs to support the ad-hoc creation and alteration of entities and attributes, without need for changes to the logical data model or to the client used for data-entry or exploration. This is also required if the application is to be generalized as a tool for other projects.

However, while flexibility is a good thing, if you grant too much it subverts the purpose of formalization and structure - the creation of a model that lends itself to analysis. The strategy for the new data-model must therefore also include mechanisms that lets the user define constraints on the entities he creates, for instance specifying that a "person" is required to have a "name", and that the person has a "nationality" relationship to a country of many-to-one cardinality.

Failure to implement a mechanism for the creation of such constraints will lead to the material ending up as not much more than an unstructured mind-map where little further analysis could be facilitated.

Provide an integrated environment

In his answer to question six Vollset stated that he felt the separation of data-entry and data exploration into two independent applications was unnatural and not compatible with his, or other historians, usual work flow. Many are accustomed to working hermenutically, switching back and forth between gathering data and analyzing. This allows for the continual adjustment of interpretation to suit new insights.

A production-quality version of the system will have to align itself with this workflow, providing a unified environment where data can both be explored and altered without necessitating the mental context-switch of opening a new application.

Mitigate data-entry workload

Vollsets identified the main reason for the prototype not being a viable tool as insufficient return on time invested. He saw interesting results, but as these results required hours spent entering data into the system he felt that he would have been better off using traditional methods.

While this problem might be mitigated to some extent by incremental improvements to the existing system, polishing the interface, integrating data-

entry and data exploration client, providing a visual way of creating relations instead of using the drop-down menu approach in the prototype and so on, Vollset also notes that we might have gotten our metaphor wrong.

We set out with a implicit goal of creating a map of the world. We envisioned a graph that would include every significant researcher, artifact and institution in Vollsets material, allowing subsequent analysis and exploration.

Doing this we created a trap for ourself; the solution required a measure of *completeness* to be viable. With Vollset's international scope the investment of time required to achieve this was huge - several thousand entities would need to be created in the database, with the number of relations required entering five digits.

The solution to this problem might be more functional than technical. The future incarnation of lepraMaps main role should be facilitating the persistance and structure of a historians notes of interesting themes and issues encountered in the course of research. This functionality should be able to stand alone; data-entry should pay for itself through providing a better way of doing this than a word document / spreadsheet approach. If this goal is met the systems other features, such as data exploration through filtering and visualization, would be a pure added bonus, not requiring further investment, and the application would "pay for itself" from the first day of use.

While this was definitely not the case in our prototype, we feel that it is possible, if partnered with significant improvements to the interaction design of a new integrated client.

Get people involved, facilitate collaboration

Not including features supporting collaboration in the prototype was a decision made based on the time available, the same goes for the choice of a local database with a thick client.

Vollset states, in his answer to question nine, that collaboration support and ease of installation is an important feature. This was also evident from the reactions Vollset received when he presented the software to a group of other doctorate students during a international conference on the future of the History of Medicine[1] - people wanted to know where they could download it to try it out themselves.

This wish was difficult to accommodate, since the installation of the lepraMap prototype was a somewhat complicated process; the user has to provide

[1]The conference was arranged by Wellcome Centre for the History of Medicine at Goodenough College on July 15-17, 2010

a database management system, run data definition and manipulation operations from script files, and finally start the jar package containing the application files.

It is clear that even if a future version of lepraMap remains thick client based it will need to have an architecture which supports ease of installation, a central database which eliminates the need of a local data management system, multiple users per project, and methods of using material collected in one project in another.

Chapter 6

Future Development

Having outlined how the evaluation of lepraMap led to new requirements for a production-quality release of the application, this chapter will describe how these are implemented in the design for the new version of the software, named chronoGrapher. While we are still in a early stage, and details are changing, we have a fair idea of the general architecture.

This chapter presents the most important changes to be made; the use of ontologies as a framework for knowledge representation and a new, unified user interface, removing the separation between data entry and data exploration and analysis.

6.1 A new strategy for knowledge representation

As noted in section 5.3 the creation of explicit dependencies between the logical data model and the conceptual structure of the research data entered into the system created major problems. To avoid this situation a new approach is called for - where the user is free to formulate and alter the model as he works, without the constraints of the implicit pre-coding of the relational data model. The rest of this section will be devoted to the examination of techniques that meet this requirement.

6.1.1 Ontologies

With the birth of the Internet research into models for the representation of knowledge in a form readable by a computer increased. One of these models was presented by Tim Berners-Lee, "father" of the world wide web, and named "Sematic Web"[20], to differentiate it from the parts of the web meant for human consumption.

Berners-Lee envisioned a world where you could rely on a computer program to make a doctors appointment or book a table at a restaurant, without ever specifying which doctor or restaurant. The instructions could be limited to "My back is hurting again, I need to see a doctor", or "I think I'd like Italian food tonight", trusting that the agent will understand your request, find a suitable doctor or restaurant, make sure that the reservation time will not conflict with the business meeting you agreed to earlier today, and take any food allergies you might have into consideration.

This system was not to be implemented by an "old-school" HAL2000[21] style artificial intelligence, but rather by teams of semantic agents reading specially encoded web pages, designed to be parsed by computers rather than read by humans. After identifying a suitable restaurant or doctors office, the agents could then communicate directly with the external systems, booking a table or an appointment and transmitting the necessary information.

One of the main mechanisms used to construct these pages would be *ontologies*; maps describing how the different pieces of information relate to each other: for instance that *business* of the type *restaurant* by the *name* "Marios" serves food of the style *Italian*.

The term "ontology" comes from philosophy, more specifically metaphysics, where it is the study of theories on the nature of reality and existence; which entities can be said to exist and their relationships and categories. The expres-

sion is also used outside philosophy, where it has several different meanings and definitions in fields spanning artificial intelligence and language.

In information science the go-to definitions for the term are from Thomas Gruber, describing an ontology as "...a formal, explicit, specification of a shared conceptualization"[22], or John Sowa's more pragmatic description of ontologies as "a classification of the types and subtypes of concepts and relations necessary to describe everything in the applications domain"[23]. However, both have the same meaning: a formalized model, consisting of types of concepts and their relations, of a person's or group's conceptualization of a domain of knowledge.

The definitions above might seem all-encompassing, but so are ontologies: there is no limit to the subjects or structures that can be modeled. Other forms of knowledge representation in common use such as indexes, vocabularies, thesauri and taxonomies, can all be viewed as more limited versions of ontological subject classification[24].

For instance, thesauri can be seen as an ontology where the types of relationships that can exist between subjects are limited to the following[24]:

BT (Broader Term). Used to indicate that there is a term above this in the taxonomy hierarchy.

USE Indicates that there is another preferred term, and implies that the two terms are synonyms.

RT (Related Term). Indicates that this term is associated with another without fitting into any of the other categories.

The similarity between the thesauri described above and the ontologies defined by Sowa. Both construct a conceptual model of the real world by describing subjects and the relationships between them - but there are important differences. A thesaurus has a limited and defined vocabulary for describing the relationships between the subjects. While the set of relationship types are sufficient for the task at hand, showing which terms describe subsets of others, and which are synonyms, there is an infinite number of other relationships between subjects that can not be reflected using the language in the list above.

An ontology, on the other hand, has an *open* vocabulary, letting the constructor of the map specify new relationship and subject types as required. This means that you could construct any other subject-based classification language, such as a thesauri, taxonomies, or vocabularies, within an ontology, by specifying the needed structure. It is this flexibility that make ontologies a good fit as a knowledge representation framework for the new version of the lepraMap application.

There are various implementations of ontology based systems, but they share one advantage that makes them candidate data modeling tools for chronoGrapher; a ready made syntax for describing and annotating the structure of information, and the fact that the makeup of the ontologies, as implemented in the common tool sets, are largely independent of the data model providing persistence for the system.

6.1.2 Topic Maps

One of these ontology implementations, *Topic Maps*, originated with a problem regarding the interchange of computer documentation; the merging of traditional, back-of-the-book indexes. These usually consist of an alphabetical list of topics with corresponding page numbers. Some common refinements include using a bold font to point to an in-depth discussion of the topic, and a "see other term" to indicate the preferred expression. While working well for isolated documents, concatenating multiple documents leads to problems; the same topic might be referred to with different names, or authors working independently might have chosen the same name for two totally different subjects.

The solution, as proposed by what was later known as the Davenport group, was based on the insight that an index is just a view of a conceptual ontology. If one could explicitly model the ontology implicit in the organization of the indexes of the documents to be merged, a predetermined technique could be applied to integrate the ontologies into a single unit, which would form the basis of the new index.

To achieve this, and other goals, a formalized standard for topic maps has been specified as ISO 1350[25]. The standard includes a Data Model (TMDM), a XML based syntax for storage and interchange (XTM), and a graphical notation (GTM).

The main entities, as defined by the TMDM, are Topics, Associations, and Occurrences:[1]

Topics. The main component of Topic Maps is, not surprisingly, topics and, as we will discuss in detail later, almost everything in a topic map which is not an association or an occurrence is represented as a topic. Topics are proxies, standing in for *subjects*, the real world "thing" the topic represents.

Associations. Relations between topics. The associations in topic maps form *undirected hypergraps*, allowing each association to group two to n topics,

[1] Sometimes referred to as the TAO (Topics, Associations, Occurrences) of topic maps.[26]

but without reflecting any directionality. Since knowing which parts two connected topics are playing in their association is important, the TMDM take care of this through the use of association roles. The implications will be discussed later in this chapter.

Occurrences. Resources relevant to the topic. Keeping the origin of topic maps in mind, this is similar to how an entry in a traditional index points to the pages where relevant information can be found. In a digital information system, the resource can be in other forms; a web page, a video, an audio clip and so on.

Distinction is made between *internal* occurrences, where the information is contained within the topic map itself (much in the same way as an attribute in a traditional entity-relational model), and *external* occurrences, which point to an information source outside the bounds of the system, most often in the form of a URI (Uniform Resource Identifier) leading to a digital source.

The relationship between topics and their subjects is interesting, and brings us back to the "model of a model" concept we discussed in section 4.1. With the design of topic maps, as with the ER model, we are trying to bridge a chasm between knowledge as it exists inside a computers data model and as it exists in the human domain of "real life".

First of all, it is important to note that there is no need for the subjects in a topic map to exist in any physical form: they can be "anything whatsoever ... about which anything whatsoever can be asserted by any means whatsoever"[27]. For instance, the subject referencing the real world person Armauer Hansen is, in the topic map, represented by the topic-proxy "Armauer Hansen". The difference between Armauer Hansen the topic, and Armauer Hansen the person is subtle, but important.

Having the ability to uniquely identify subjects independent of what name they are given is important when designing a topic map. This enables merging of ontologies, and maintains the important one-to-one relationship between topics and subjects - if the same subject is represented by multiple topics this quickly degrades the integrity and usefulness of the ontology.

Some topics represent subjects where the link between the topic and the subject can be expressed by directly pointing to the location of the subject. This will almost always be in the form of a URI pointing to a document available in electronic form. In many cases, like in the lepraMap system, this is not possible. We are unable to use a URI to point to Dr. Armauer Hansen directly, as this

subject exists outside the digital realm; it is a concept referencing a person who existed in the real world. Topic Maps provide a different technique for locating such subjects; indirect identification through *subject identifiers*.

URIs are still used, but instead of forming a direct link to the subject, they link to a computer and human readable source which unambiguously identify the subject. In the example of Armauer Hansen we could have used the page of his Wikipedia entry, `http://en.wikipedia.org/wiki/Gerhard_Armauer_Hansen`, to identify the subject, or we could choose to create our own set of PSIs. It is important to note that even if the chosen PSI is a URI linking to a data source the purpose of the element is not the creation of this association, but rather the provision of a unique identifier.

When topics and subjects are chosen, how do we use them to create the topic map? I will explain the use of these elements, and a few others, by way of an example. Let us, for the sake of argument, say that we wanted to construct a model representing the following knowledge:

> *Armauer Hansen and Daniel Cornelius Danielsen, both medical researchers, worked at the hospital Pleiestiftelsen (also known as "Pleiestiftelsen for Spedalske no. 1"). Dr. Hansen in the period from 1868 to 1880, Dr. Danielsen from 1857 to 1894.*

We could model this information using the XTM syntax, but for communicating the structure of a model a graphical notation is preferable. We could, as Garshol suggests as an alternative, use the Unified Modeling Language (UML) [28], even if it lacks topic-map specific notation, but there is an ISO standard for a topic map graphical syntax under development; Graphical Topic Maps (GTM). The standard is still a work in progress in 2010, but I will be using the version proposed by Hendrik Thomas et al. under the name GTM^{alpha}[29].

Let us examine the model in figure 6.1. We see the name "Armauer Hansen" within a circle. Circles denote topics, so we know that this is a representation of a subject. The fact that the name is placed directly within the circle indicates that this is the preferred, or unscoped name for the topic (we will return to the subject of scopes later). The same applies to our other doctor, "Cornelius Danielsen".

There is an arrow going from the "Armauer Hansen" and "Cornelius Danielsen" topics to another, "Person", indicating that "Armauer Hansen" is of the *topic type* "Person". Topic types are an integral part of the TMDM; they let us classify topics into groups in much the same way as entities are grouped into entity sets in ER modeling. Topic types are topics themselves, complete with subject

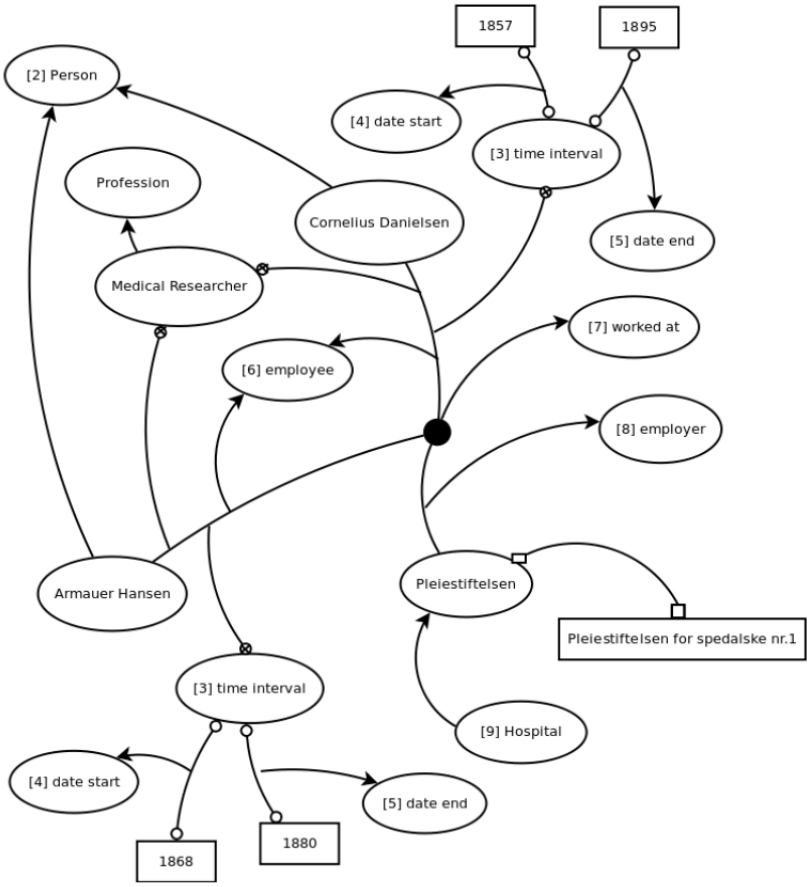


Figure 6.1: Topic map in the GTM^{alpha} notation.

identifiers and occurrences, but they have a special status in the topic map specification as a mechanism for constructing hierarchies[2] [27]. The collection of topic types in a map can be viewed as an internal ontology - a description of the kinds of subjects the topic map describes[27].

However, as in our ER model, the most interesting information is not conveyed by the topics themselves, but by the associations which connect them. The main association in our example is the one between our two doctors and the hospital they worked at, Pleiestiftelsen. How do we reflect this relationship in our model? We see a line connecting the doctors and "Pleiestiftelsen" through a black circle, indicating an association. An arrow from the association to the topic "Worked at" indicates the *association type*. Notice that there are no arrows on this line: as mentioned earlier associations in Topic Maps are non-directional, making added notation necessary, since, if we abandon common sense, we have no way of knowing if Dr. Hansen worked at Pleiestiftelsen or if Pleiestiftelsen worked at Dr. Hansen. We solve this problem by adding *association roles*. These tell us which role the topics play in their association; in our case a relationship between an employee and an employer.

So we now know that Dr. Hansen and Dr. Danielsen worked at Pleiestiftelsen, but our example also contains information about *when* they worked there: Dr. Hansen from 1868 [3], and Danielsen from 1857 to his death in 1894.

We represent this information by creating an object to our map, "Time interval". Time interval is an *association scope*. Scopes are used to indicate that the topic or occurrence has a particular context. It could be used to indicate conflicting information about a topic, topics with different names in different languages etc. In this model scope is used to indicate that the association between the doctors and their employer existed in a certain period of time. It has two occurrences, represented by rectangles, one of the type "Date start", the other of the type "Date end", containing this information.

Occurrences in Topic Maps refer to information describing the subject the topic refers. "Date start" and "Date end" are *internal occurrences*, the information they contain is stored in the topic map itself. Other occurrences, pointing to resources outside the topic map by URLs or URIs, are known as *external occurrences*.

Finishing up, we add some refining touches, giving Pleiestiftelsen a second name scoped as "alternative", and adding types reflecting the fact that both

[2]special identifiers in the TMDM, special syntax in the XTM, and special treatment by the query and constraint languages.

[3]When he was forced to quit his position after a much publicized ethics scandal involving the injection of contaminated material into the eye socket of a patient

Armauer and Hansen where employed as researchers.

This map might look complete, and in many ways similar to the ER diagrams we used created in section 4.1.2, but there is one major thing missing. Data in a ER model is defined as much by its constraints as anything else: the rules stating that all people need to have a name and that every artifact must have a language. The constraints maintain the integrity of the database, making it useful for both manual and programmatic analysis.

In a relational database these constraints are intertwined with the makeup of the database schema itself, the organization of associative entities, tables, and foreign and primary keys together with explicit rules such as NOT NULL and UNIQUE. Topic map systems will also almost always have a relational database as a back end, but in this case the constraints are not an attribute of the schema, which it merely implements persistence for the map.

Instead of using the database schema, topic maps implement constraints through their markup language. Until now this has mostly been implementation specific, but work is being done on an ISO standard Topic Map Constraint Language (TMCL), letting the designer define rules using constructs from the TMDM.

As in an ER diagram the constraints are not applied directly to the entities - but to the entity sets they are organized in, which in a topic map means that constraints are applied to the meta-ontology formed by the maps set of topic types.

Having given this example of how topic maps are used to create conceptual models of information we will move on to consider the practical implications of adapting this solution to the requirements of the lepraMap project.

6.1.3 Applying topic maps to historical sources

The preceding section described the general structure of a topic map, and we showed how the syntax could be applied to describe a segment of information representing a historical source investigated through the lepraMap project. But what advantages do topic maps provide above a relational data structure when applied to the domain of historical sources?

Several researchers have applied ontologies and derived knowledge representation strategies to the modeling of historical sources, for instance the FDR/Pearl Harbor projects ontology based representation of documents from the Franklin D. Roosevelt Presidential Library dating from a 10 year period up to the bombing of Pearl Harbor[30].

The aim of this, and most other historical ontologies, diverges from the

goals of the lepraMap / chronoGrapher project. While they attempt to create a complete model the source itself, as a way of facilitating analysis, we are using ontologies as a way of keeping track of the researchers mental model of the sources.

Even so, examining a previous project might illustrate the advantages of using a ontology based system in our research context. We will examine an example of such a topic map, a topic map modeling Samuel Pepys' diary, and discuss its characteristics.

The historical source of this map are the combined surviving diaries of Samuel Pepys, written by the member of parliament and naval administrator from 1660 to 1669. They form some of our best historical sources from this period, and give a detailed eyewitness narrative of important world events such as the second Anglo-Dutch war, local disasters like the fire of London, as well as insight into Samuel Pepys' personal life; parties he attended, the food he ate, the state of his marriage etc.

As can be expected when such a rich source of information is available, detailed research projects have been performed and several biographies published. One of these in the form of a blog, Phil Gyfordsins' `www.pepysdiary.com`.

The blog follows the diary, each blog post being one day in the diary. In addition to this, it links information about all notable characters, objects, locations and events to an encyclopedia written along with the blog. The blog / encyclopedia combination creates an unformalized ontology of the diary, the blog posts forming the occurrences and the encyclopedia defining the topics. This ontology has many of the attributes of topic maps: topics (encyclopedia entries), associations (links between encyclopedia entries) and occurrences (links between a reference to a topic in the diary and the corresponding encyclopedia entry).

But this ontology is not a topic map; it lacks many of its features (association roles, hypergraph associations, topic identifiers, types, multiple topic names), as well as a formalized way of describing these. How could converting this diary into a formalized topic map prove advantageous?

A map of the diaries was created by Kan Ahmed in 2005 [31]. Figure 6.2 shows the topic map browser displaying information about the Sir William Penn topic[4].

The figure illustrates some of the strengths of topic maps as a way of describ-

[4]Sir William Penn was a parliament-loyal admiral in the English civil war, and father of William Penn, founder of the colony of Pennsylvania. The admiral, a friend of Pepys, is mentioned some 629 times in the diary.

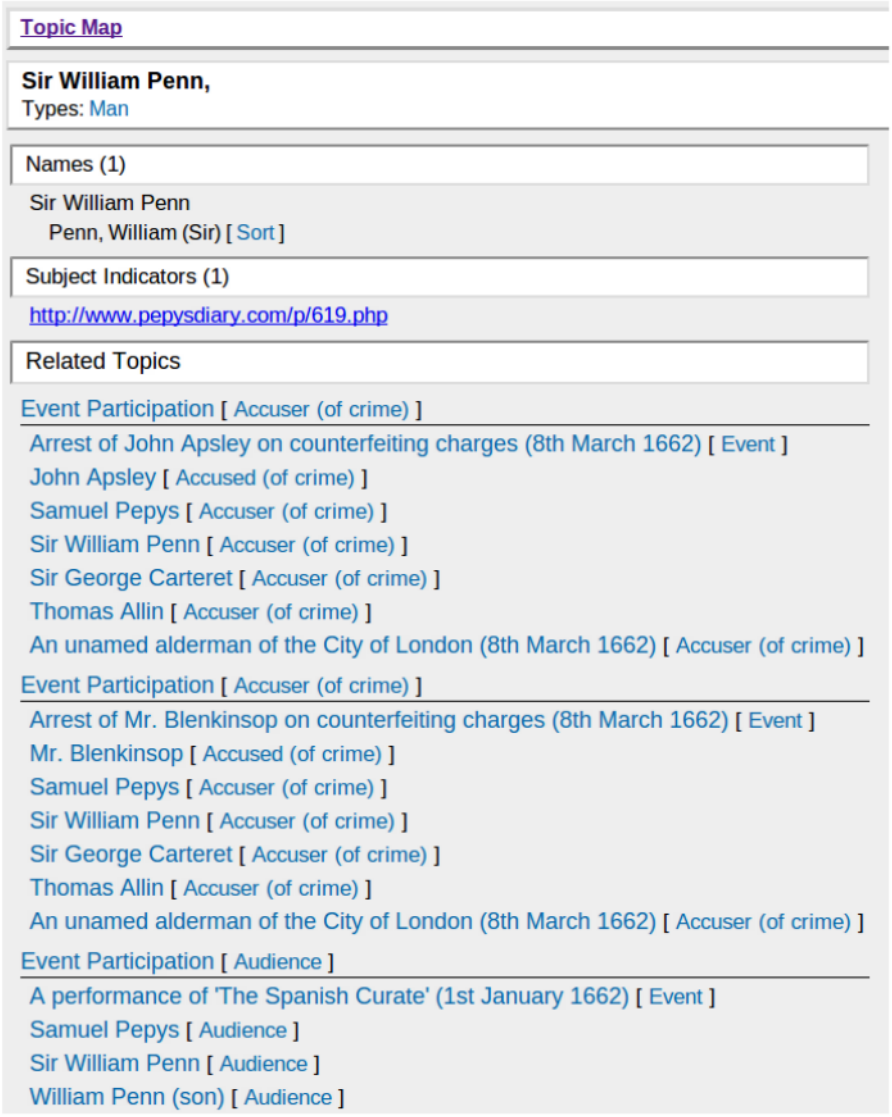


Figure 6.2: Admiral William Penn as described in Kan Ahmed's topic map.

ing historical sources. Although the references to Admiral Penn are scattered throughout the source, the topic map collects them at a single location, identified by the subject indicator linking to the Admiral Penn article on the Pepys diary wiki, performing the task of a index. But the advantages go beyond this, as the index becomes a starting point for an exploration of the source, letting the user move from topic to topic by traversing the hypergraph created by the associations. Contrast this with Admiral Penn's description in figure 6.3, from the "references" page of the encyclopedia, describing the occurrences of the subject in the diary. While the page provides a starting point for any person wanting to see a list of the mentions of Penn in the diaries, all he has to work with is a list of occurrences. No typing, no associations, no association roles.

Let us say we wanted to study the card games between Pepys and Admiral Penn described in the diaries. Using `www.pepysdiary.com` this requires a significant amount of work; if a list of card games could not be located in any of the encyclopedia articles one would have to go through all 629 mentions of William Penn by hand. The process becomes much less time consuming if one has access to the topic map. The method of navigation will vary according to which topic map browser is used (the one in Kan Ahmed's map is quite basic), but the steps of identifying all topics of type "game" where the subjects of type "actor" with the name "William Penn" and "Samuel Pepys" will give us a complete list.

This is the type of navigation that should suit investigations of historical sources; complex queries can be answered quickly, and the focus is on the relations between the described subjects. We have established that it is not only possible to model a historical source with the help of a topic map ontology, but it also provides significant advantages over a traditional note / index based solution. But Vollsets requirements go beyond the qualities of the end knowledge representation model; requirements related to the process of making the map and the method of exploring it must also be satisfied.

We have to provide a framework that will both let the user create a topic map as he investigates his sources, making sure that the map conforms to specified constraints, and provide a method for visualizing and exploring the resulting map. Our use of topic maps is not as a support for an information retrieval system, the most common use of the framework, but as a model representing the structure of the information contained in a set of historical sources. The topic map does not just organize the information we want to present to the user,

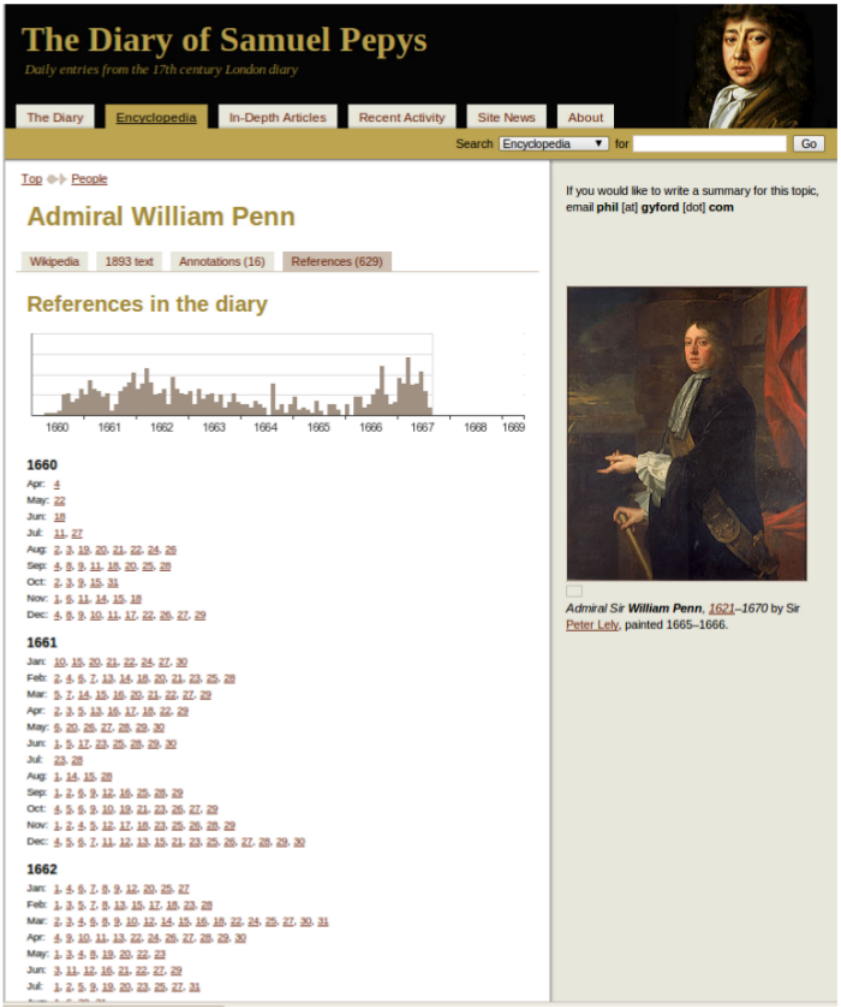


Figure 6.3: Admiral William Penn as described in the pepysdiary.com encyclopedia.

it contains it as well[5].

Having considered the advantages a ontology based system can provide, we will now examine how the ontology is created; important for the chronoGrapher system, as we are not only making a ontology; we are expecting the user to design it as he is working, without the process distracting from the investigation of sources.

6.1.4 Ontology creation work flow

Lars Marius Garshol suggests a process for developing topic map ontologies in his 2007 paper [32]. His proposed methodology, designed for developing ontologies for web-portals but adaptable to other uses, consists of two elements: "Ontology Development Process" and "Ontology Development Guidelines"[6].

The first element of Garshol's proposed methodology is a procedure for development of an ontology. It consists of the following phases:

Start up. Objectives are established. Creating a topic map is seldom an end in itself, it is a means for accomplishing another task.

End-user and analysis. Requirements and terminology are gathered from end users and analyzed. In many cases the developers of an ontology will not themselves have extensive knowledge of the information domain they are mapping, close cooperation between developers and users and domain experts then becomes invaluable.

Drafting. The collected information is used to generate a draft of the ontology. This draft is presented to users and other stakeholders, and a iterative process where one gets closer and closer to the final design ensues.

Interaction design. The interface used for interaction with the topic map is designed, and its compatibility with the topic map is verified. This will be the front-end for the map, both for editing and extending, and for exploration.

Verification. Assertain that the proposed model meets requirements, that the objects match the external subjects, and that it satisfies the needs of the end users.

[5]The user of the system can create references from topics to external resources, using external occurrences in the form of URIs, but this functionality is not central to the implementation.

[6]A third element, a library of topic map design patterns solving common problems, is suggested, but not elaborated on in the 2007 paper.

The specification of this process illustrates the difference between the most common use of topic maps and the role it will play in ChronoGrapher. Garshol envisions a process where one starts with goals and requirements, and then carefully outlines a model of the complete map, adding detail and structure as one approaches a finished product, much like the process for the development of relational databases outlined in chapter 4.

However this process does not perfectly match our needs. While an up-front design phase is certainly recommended, it needs to be as quick and painless as possible if we are to reach our goal of lowering the investment of time necessary to use our tool. If a virtual notebook is to be a the guiding metaphor for chronoGrapher, the user can not be asked to formulate a complete topic map ontology before getting to work.

The steps in Garshol's process would be performed, but instead of asking the users to execute them, thereby raising the bar for adaptation significantly, an initial framework would be made avaliable. The goal would be to identify a set of topics, associations and topic map design patterns that would work as a basic construction kit for ontologies describing an investigation of historical sources. Our aim would be to provide a generic structure enabling the end-user to get going, while still letting the user alter the model to suit his particular project. The advantages would be lower initial investment for tool adaptation and standardized patterns for common structures that would ease collaboration.

This basic framework would be designed using Grashol's process, but informed by the data-model design we created for the lepra-map prototype; the person-artifact-institution entity set presented in section 4.1.2. Typical associations would be created, based on the experiences from the prototype, and a set of common occurrences will be attached to the topics.

One of the biggest advantages of having a model for common requirements is that some topic map design patterns, such as reflecting change over time through the scoping of associations or providing subject identifiers for topics such as languages or nations could be included as application features. This would allow the new interface to do more of the heavy lifting during ontology creation, by automatically adding the necessary syntax to the map. To a large extent the tool itself could generate the ontology, instancing topics, creating associations and assigning roles and identifiers with the input from the user kept to the minimum needed to accomplish the task. This will be discussed in more detail as we move on to user interface issues in the next section.

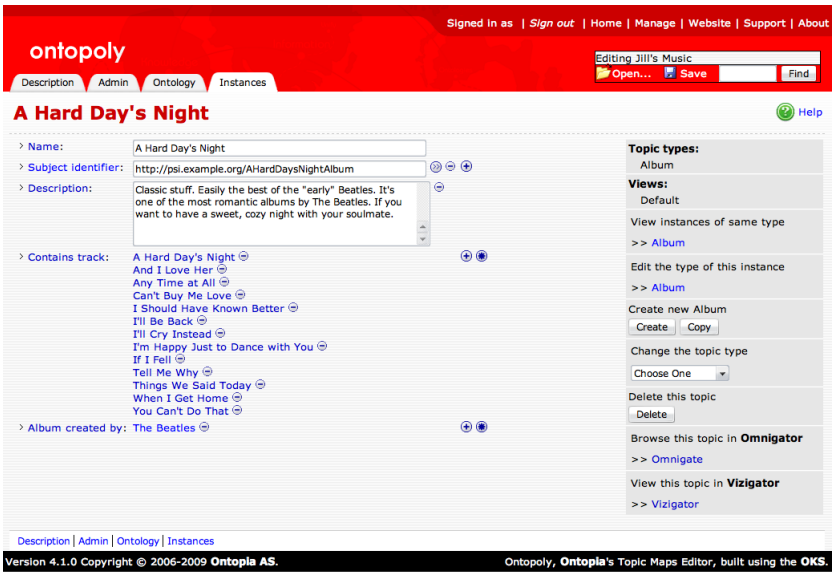


Figure 6.4: Editing a topic map with ontopoly

6.2 A unified user interface

As discussed in section 5.3 one of the main lessons we learned from the prototype was that the user interface would have to be improved significantly. The current interface enforces a work-flow which does not suit the hermeneutical nature of the work, and context-switching between data-entry and analysis felt unnatural to the user.

The *data-entry workload is too great*, far eclipsing the advantage provided by the applications facilitation of data analysis and exploration. Features supporting *collaboration* are needed, allowing the sharing of the created ontologies, thus adding value to the application. There is also a need to make *the use and deployment of the application easier* - ideally it should be possible for a user to be productive shortly after being introduced to the application, without any prior knowledge of topic maps or ontology creation.

The following sections will discuss how we are attempting to deal with these issues in the future version of the application.

6.2.1 Interacting with a topic map

In the former section the new, topic map based, data-model was introduced, and we discussed how we could use the open source Ontopia topic map engine to manage our ontology. As well as the Topic Map Engine, the Ontopia Knowledge

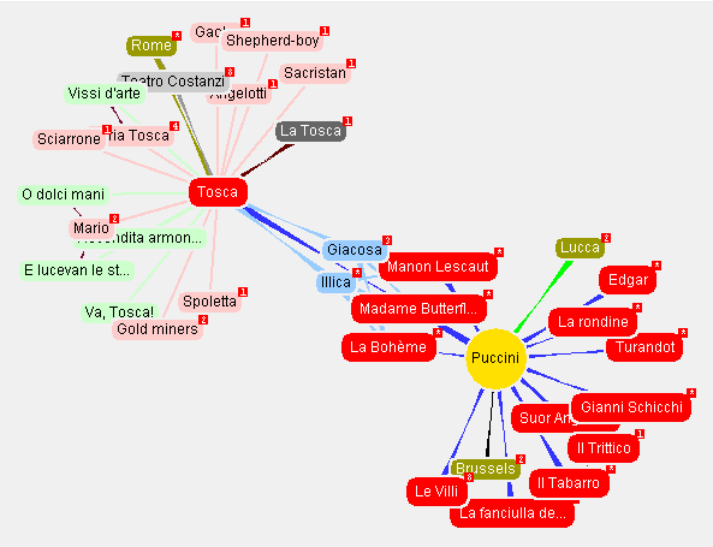


Figure 6.5: Visualizing a topic map with vizigator

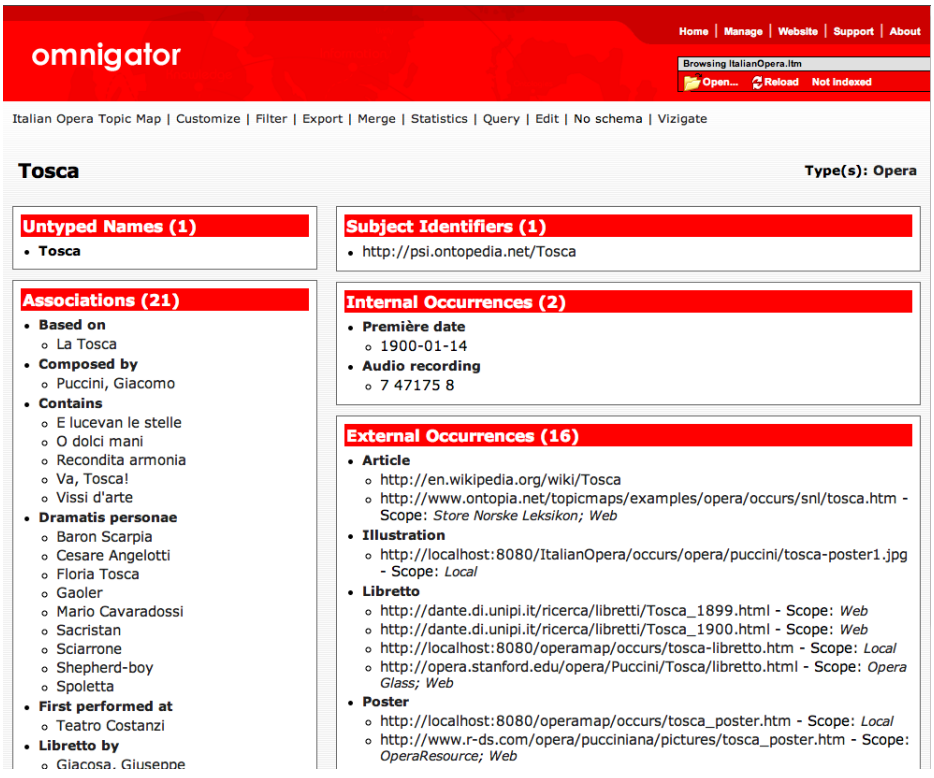


Figure 6.6: Browsing a topic map with omnigator

Suite also includes GUI interfaces for interaction with the engine, divided into three modules:

Ontopoly A topic map editor, providing an interface for editing instance data based on a user defined ontology. Both the ontology and the data can be changed, edited and deleted dynamically. Figure 6.4 shows the tool in use[33].

Vizigator A tool for graphical visualization of topic maps, which can be displayed through the main web application, or embedded in a Java applet. An example visualization generated by this tool is shown in figure 6.5[33].

Omnigator A "topic map management" tool, it lets the user investigate any part of the map, run queries, retrieve statistics etc. A screen shot is shown in figure 6.6[33].

While these three tools together form an admirable interface for the development and structure of ontologies it is clear that they do not satisfy the augmented set of requirements arrived at after the completion of the prototyping phase. Switching between Ontopoly and Vizigator depending on whether you are doing data-entry or exploration and analysis, falls short of the integrated work-flow requirement. While creating new topics or adding instance information in Ontology is quick and painless, the generalized nature of the tool leads to a quite complex user interface, requiring quite a bit of knowledge about the nature of topic maps in order to navigate and use.

The fact that Ontopias tool is designed for the creation and use of any kind of topic map means that it has few preconceptions about how the ontology is structured and the type of topics modeled[7].

This flexibility, while an advantage when it comes to supporting an endless variety of ontologies, has some disadvantages. The user gets very little assistance from the system when applying structure to his topic map; every user will have to answer questions like "How should i reflect durations in time?", "What is the best way of identifying languages?", "How should geographical locations be stored?". This greatly increases the level of knowledge about the structure of topic maps necessary to use the tool and the time required to "get going" and start adding useful information.

[7]The tool does support some structures above others, for instance it has special support for a few "root" types, implementing such things as class hierarchies and a "global" super-types which forms a template for all other types.

6.2.2 Making it look easy

The vision is for the chronoGrapher interface to strike a balance between flexibility and ease of use, sacrificingsome of the all-compassing nature of topic maps in order to shift workload away from the user and onto the application. We hope to accomplish this by expanding on Ontopias basic set of topics to include default ways of representing data common to all projects of our targeted type, ontologies representing issues of historical research, and by providing application features making the creation and manipulation of these topics easier. Among the features we want to include are:

- Set methods for the representation of duration in time for topics, occurrences and associations.
- The inclusion of a default set of commonly used topic types, building on and expanding the set refined during the work on lepraMap; actor, artifact, arena. The set will encompass common association types, roles, and scopes. The use of these will be completely optional, and they can easily be removed, but it will give users a starting point, and a reference implementation from which to build their own ontology.
- The inclusion of default sets of identifiers for commonly used topics, such as nations and languages, and a naming convention for PSIs not delivered with the product.
- Features allowing for the dynamic import of all these elements from other ontologies created with the system, either en masse or on a topic by topic basis.

The application of these techniques will we illustrated with an example. The hypotetical users actions are in normal font, the systems responses are indicated by italics. As a use case, let us say the user wants to add two people to the ontology, and an association between them reflecting that they are brothers-in-law.

> Through an interface similar to the graph pane of the prototype (shown in figure 4.7), the user selects a person topic from a provided palette, and add it to the topic map through click and drag. *An instance of a topic of the selected type is created in the topic map. The PSI is left blank for now.*
>
> The next step is to edit the properties of the person, adding name and selecting nationality. *The name of the topic is altered, and an*

association to the "country" topic is created, complete with chronological scope (default is the lifetime of the person if this is supplied) and roles. The topics PSI can be set by default to an attribute value (like name), or through a lookup in a PSI set.

The process is repeated for the next person. *If we assume that this person already exists in another ontology, which has been made public to the user of the current one, the system can notify the user of this after the name is entered, and offer to automatically populate remaining attributes.*

The user can then, again through click and drag, create an association between the two people, and, by selecting the newly created association, set the type to "family association" and the subtype to "brothers-in-law" *An association of the appropriate type is created between the two topics.*

As can be seen from this example, the goal is to keep the gritty details of creating the ontology hidden from the user, letting him focus on the real work - his research. The next section will detail the new system architecture making this possible.

6.2.3 A new architecture

While the user is populating the ontology the back-end is performing the necessary calls to the topic map engine API; creating topics, populating instance and occurrence data and setting roles and scopes of associations. Several users might interact with the same, or with more or less loosely connected ontologies at the same time.

The architecture of the modules supporting this process is shown in figure 6.7. The platform of some components is still to be decided, but we will probably be moving away from Spring RCP to a more mature platform; NetBeans RCP looks like a good candidate if we do not decide to go for a browser based solution. The Ontopia topic map engine discussed in section 6.1 features heavily in this new solution, providing persistence, a framework for maintaining the ontologies, and mechanisms for modifying, analyzing and merging these.

Building on and extending this framework is the ChronoGrapher API. It will be concerned with translating the UI tasks completed by the user to domain model changes, which will again use the topic map engine to perform the necessary alterations to the ontology. While most use-cases can be completed

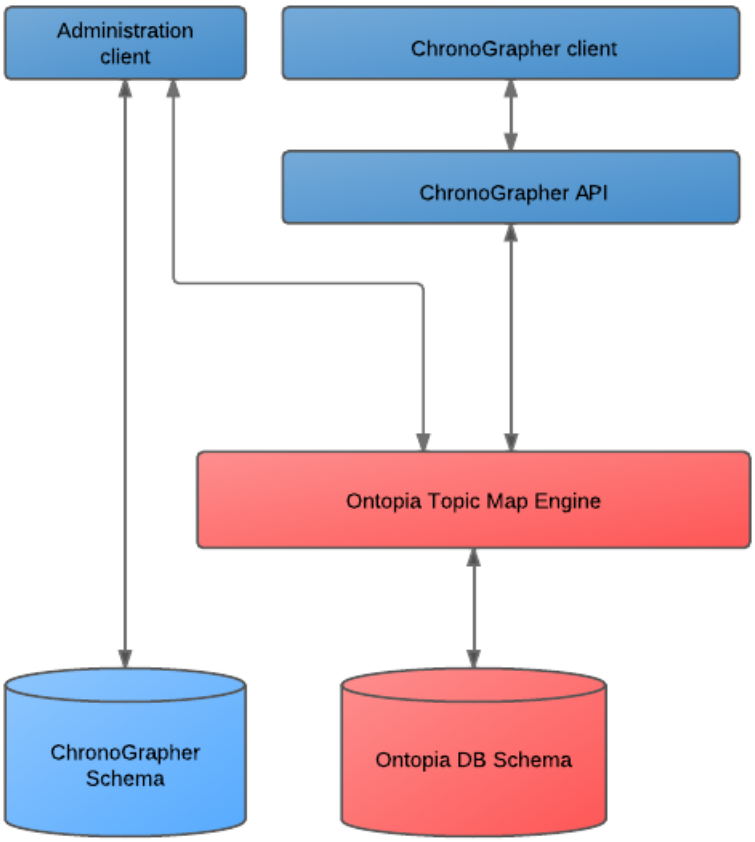


Figure 6.7: System architecture of the planned chronoGrapher application

through the functionality provided by the Ontopia engine, it will have to be extended to support features such as the incremental import of topics form other topic maps.

An administration client is also planned, which main purpose is administrating users, access privileges, authentication etc. It will use its own schema, outside the Ontopia DB, as persistence.

As we are moving forward with the design of this architecture and the modules contained within, we have returned to prototyping, going back to pen and paper sketches. It might seem like this should be unnecessary, since we just went through the extensive prototyping process described by this thesis. We believe it to be a natural application of the technique - the hi-fidelity, horizontal *exploratory* prototype forming the lepraMap system is now replaced with *experimental* low-fidelity prototypes testing key aspects of the new system, for instance the layout of the new integrated user interface. This new applications covers new and complicated use-cases not included in the original lepraMap system, and we would like to avoid as many surprises as possible.

As the work moves on to the implementation phase we expect to also include evolutionary prototyping, as feature branches of our code form hi-fidelity, vertical prototypes testing aspects of the ontology-creation, topic map engine, or collaboration code.

This chapter has covered the major implications of the lessons learned from the lepraMap prototype; a new strategy for knowledge representation providing a much more dynamic working environment, and a interface that supports the *existing* work flow of the user instead of requiring the formulation of a new one. We will discuss these, and other issues, in the conclusion of this thesis.

Chapter 7

Conclusion

It is often stated that one of the dangers of prototyping a system is that people will not want to let go of the prototype; management or users might be fooled by a system appearing to be close to completion, or the programmers might not want to discard working code [34, p 56] [35] [36, p 115]. This leads to the system being built on a fragile foundation.

In this project there is not much danger of that. As detailed in section 5.3, we identified major flaws in the design of the prototype, several of them "deal breakers", which would lead to project failure, since they could not have been easily resolved by refactoring.

But it is this unveiling of hidden problems and design issues which is one of the main advantages of prototypes; to again use the definition from *The anatomy of prototypes*[12]; *"prototypes are filters that traverse a design space ..."*. We used the prototype as a filter to extract a good design candidate from a large design space of possible solutions, in a way which probably would not have been possible if we had not followed this strategy.

The most important requirements identified for the next version of the systems was the need to eliminate dependencies between the logical database model and the ontology created within the system, and the importance of aligning the work flow of the tool to the work flow of the user instead of the other way around. The sum of these new requirements is an adjustment to the nature of the application; the initial implicit goal, the creation a virtual map of the sources, had a return on investment that was much to high, and this return only materialized after a level of completeness was achieved. The goal of the new application is more subtle; to be a facilitator and catalyst for the existing activity of note taking during research.

In hindsight these issues might seem rather obvious, and we may wonder how

we missed them the first time around. The explanation is that we *did* think about them - the initial design documents mention supporting a flexible data model, we were aware of potential problems related to pre- and post-coding, and we were discussing Vollset's work flow as we were looking at mock ups of the data entry client and visualization application.

The reason these requirements did not translate well into design might be related to the other part of Lim's definition; *" (...) prototypes as manifestations of design ideas that concretizise and externalize conceptual ideas"*. The prototype let us examine a physical implementation of a vague "provide a flexible data model" requirement, which was restated as a more specific "create no dependencies between the logical data-model and the ontology". Vollset was also able to use the tool we had created and discover that the context-switching made necessary by the separation of data entry and data visualization felt unnatural.

When we now discard the prototype and move on to an implementation of chronoGrapher - how do we know we are moving forward with the right set of requirements? The short answer is: we do not. There might still be major stumbling blocks in the proposed design for the chronoGrapher application, especially when it comes to the design elements we have not explored in the prototype; topic maps for knowledge representation and the move from an ontology explicitly created by the user to a system where the ontology is generated implicitly while the user is working. We are dealing with this by progressing from the exploratory lepraMap prototype to experimental prototypes testing the now better defined user interface, and we expect to use evolutionary prototypes to test our new technical features, such as the interaction with the Ontopia engine or the dynamic creation of ontologies based on pre-determined topic-sets. And we have one big advantage this time around; a much better understanding of the nature of the application we are creating, letting us focus on the right issues.

The real test will be our evaluation after the completion of the ongoing chronoGrapher project. We will then be able to compare our experiences designing and implementing chronoGrapher with the lessons we learned from lepraMap. Did the prototype guide us to the correct design choices? Did new, serious, problems pop up? Time will tell.

Bibliography

[1] G. Rockwell. Is humanities computing an academic discipline? *Derived from the World Wide Web: http://www. iath. virginia. edu/hcs/rockwell. html*, 1999.

[2] P. Denley and D. Hopkin. *History and computing*, volume 1. Manchester Univ Pr, 1987.

[3] O. Boonstra, L. Breure, and P. Doorn. Past, present and future of historical information science. *Historical Social Research*, 108:4–114, 2004.

[4] E. Aarseth. From humanities computing to humanistic informatics: Creating a field of our own. *Future of the Humanities in the Digital Age, Bergen, Norway*, 1998.

[5] L. J. McCrank. *Historical information science: An emerging unidiscipline*. Information Today, Inc., 2001.

[6] C. Harvey and J. Press. *Databases in historical research: Theory, methods, and applications*. St. Martin's Press, Inc., 1996.

[7] J. E. Everett. Κλειω 5.1.1.: A source-oriented data processing system for historical documents. *Computers and the Humanities*, 29(4):307–316, 1995.

[8] M. Thaller. Automation on parnassus. clio-a databank oriented system for historians. *Historical Social Research*, 15:40–65, 1980.

[9] M. Vollset. Fra lidelse til trussel. spedalskheten i norge på 1800-tallet. *Masteroppgave, Historisk Institutt, UiB*, 2005. URL `https://bora.uib.no/handle/1956/3060`.

[10] K. Beck, M. Beedle, A. V. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al. Agile manifesto. 2001.

[11] C. Floyd. A systematic look at prototyping. *Approaches to prototyping*, 1: 1–18, 1984.

[12] Y. K. Lim, E. Stolterman, and J. Tenenberg. The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 15(2):1–27, 2008.

[13] Jakob Nielsen. *Usability engineering*. Morgan Kaufmann, 1994.

[14] P. P. Chen. The entity-relationship model–-toward a unified view of data. *ACM Transactions on database systems*, 1(1):9–36, 1976.

[15] G. M. Nijssen and T. A. Halpin. *Conceptual Schema and Relational Database Design: a fact oriented approach*. Prentice-Hall, Inc., 1989.

[16] T. Halpin. Object role modeling: An overview. *white paper,(online at www. orm. net). gadamowmebulia*, 20:2007, 2001.

[17] T. J. Teorey, D. Yang, and J. P. Fry. A logical design methodology for relational databases using the extended entity-relationship model. *ACM Computing Surveys (CSUR)*, 18(2):197–222, 1986.

[18] D. M. Kroenke. *Database processing: Fundamentals, design, and implementation*. Prentice Hall PTR, 1997.

[19] M. Gibbons, C. Limoges, and H. Nowotny. *The new production of knowledge: the dynamics of science and research in contemporary societies*. Sage, 1997.

[20] T. B. Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[21] A. C. Clarke. *2001: a space odyssey*. New American Library, 1968.

[22] T. R. Gruber et al. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human Computer Studies*, 43 (5):907–928, 1995.

[23] J. F. Sowa et al. *Knowledge representation: logical, philosophical, and computational foundations*, volume 594. MIT Press, 2000.

[24] L. M. Garshol. Metadata?: Thesauri? taxonomies? topic maps! making sense of it all. *Journal of Information Science*, 30(4):378, 2004.

[25] M. Biezunski, M. Bryan, and S. Newcomb. Iso/iec 13250: 2000 topic maps: Information technology–document description and markup language. *International Organization for Standarization (ISO) and International Electrotechnical Commission (IEC)*, 1999.

[26] S. Pepper. The tao of topic maps. In *Proceedings of XML Europe*, volume 2000, page 36, 2000.

[27] S. Pepper and G. Moore. Topic maps. *Article for the encyclopedia of library and information sciences*, 2008.

[28] International Organization for Standardization (ISO). Information technology – open distributed processing – unified modeling language (uml) version 1.4.2.

[29] H. Thomas. Gtmalpha-towards a graphical notation for topic maps. 2008.

[30] N. Ide and D. Woolner. Historical ontologies. *Words and Intelligence II*, pages 137–152, 2007.

[31] Kal Ahmed. Pepys diary topic map. 2005.

[32] L. Garshol. Towards a methodology for developing topic maps ontologies. *Leveraging the Semantics of Topic Maps*, pages 20–31, 2007.

[33] Ontopia - the product. URL `http://www.ontopia.net/section.jsp?id=ontopia-the-product`.

[34] A. Hunt and D. Thomas. *The pragmatic programmer: from journeyman to master*. Addison-Wesley Professional, 2000.

[35] M. Stephens. Emergent design vs. early prototyping. *May*, 26:2003, 2003.

[36] S. McConnell. Code complete: a practical handbook of software construction. 2004.

Appendix A

Interview QA

1. What led you to the conclusion that a digital tool would be helpful during your doctorate research?

One of the main elements in my PhD-thesis is the study of knowledge exchanges. I am interested in medical research in leprosy, and the proliferation of knowledge of the leprosy bacillus. How did the knowledge move from being "rod-shaped elements… possibly bacilli", observed through the microscope by the physician Gerhard Armauer Hansen in Bergen in 1873,[1] to becoming an internationally taken for granted scientific fact; the foundation for policy advice such as the League of Nations first *Principles of Prophylaxis* from 1931?[2] How was the knowledge spread, and to what extent did the disease concept change in the process?

I started my research by reading concurrent medical journals. I was soon convinced by Sanjoy Bhattacharya's description in *The Palgrave Dictionary of Transnational History*. There he argues that "Medical ideas constantly flowed in all directions", and that "time has come for a major reassessment of the complexities attending the creation of medical knowledge."[3] In other words, historians need to move away from focusing solely on "discoveries" or the history of medicine within specific geographical boundaries.

Some attention has been paid to medical research into leprosy, but in line with Battacharya's diagnosis, they have either been focusing on the research that went on within a specific geographical area or ignored the question of location altogether. There is awareness that Bergen for a period was considered the "world capitol of leprosy research", but in general historians have only limited insights into what mechanisms are involved in knowledge exchanges. Basic questions remains unanswered, such as: Were there one research front or many; was there one or several parallel international research communities; how does a scientific idea spread from one place to another; what are the relations between the local, the national and the international - and are these terms really useful when studying knowledge exchanges?

One important local source has been *Medicinsk Revue* (Medical Review), a journal established by the medical community surrounding the leprosy research hospital Lungegaardshospitalet in Bergen in 1884. Its program statement was to "make Norwegian physicians aware of the most important current foreign medical literature".[4] The editorial board had access to the medical library at the research hospital, the largest medical library in Norway at the time outside the University library in the capitol of Christiania. The library contains medical books, journals and reports from similar institutions elsewhere.[5] However, when starting to analyze the content of this monthly journal, I was soon overwhelmed by the richness of the material. The first 25 years, more than hundred different periodicals were referred to. In this period, 60 papers concerning leprosy were published, referring to 190 different individuals. Events from all over the world were made relevant to the Norwegian context. Trying to get a more precise picture, such as whom is referring to whom; whether different groups of researchers referred to each others but mainly ignored researchers from 'opposing' groups; or the rise and fall of canonical research papers - was incredibly difficult to do by hand. The off-the-shelves database solutions, such as Access, did not give me the

[1] G. A. Hansen: Undersøgelser angående Spedalskhedens Årsager. *Norsk Magazin for Lägevidenskaben*, No. 3, bd. 4, 1874: 78)
[2] League of Nations Archives 8A/26044/4621: *The Principles of the Prophylaxis of Leprosy. First General Report of the Leprosy Commission*. Geneva, April 1931.
[3] *The Palgrave Dictionary of Transnational History: From the mid-19th century to the present day*. 2009: 708ff.
[4] Program. *Medicinsk Revue*. No. 1, bd. 1, 1884: 1.
[5] H. G. Dethloff. *Katalog over Lungegaardshospitalets Bibliothek ved udgangen af aaret 1904.* Bergen, 1905.

tools I needed. I also knew that *Medicinsk Revue* was a small challenge compared to other sources, such as the first specialized medical journal focusing on only leprosy, *Lepra Biblioetica Internationalis*. This quarterly journal, published from 1900 till the outbreak of the First World War, contains texts from about 840 different authors – not counting medical researchers referred to but not themselves listed as authors.

This is where I contacted Andreas Berre and the collaboration on LepraMap began. The general goal was to develop a tool for tracing the relation between different medical researchers, and how these changed over time. As often is the case both within the discipline of history, this research project started with ambiguous objectives and, like history proper, provided several surprises as the research progressed.

2. Could you describe the general attitude towards the use of digital tools in the research communities where you have worked?

In history, such as all other academic disciplines, digital tools are part of every day working life. That is, word processors, spreadsheets and of-the-shelf databases. When I did my Master in History at the University of Bergen finishing in 2005,[6] the use of databases was one of four options for the course *HIS304 Techniques in History*, alongside gothic handwriting, statistics and diplomacy. This course, however, focused on using off the shelf solutions, not on developing tailored software for answering specific historical problems. This is representative for the general attitude towards the use of digital tools within the discipline: Historians pragmatically use whatever tools they can find, but apart from individual enthusiasts, we generally do not put effort into producing our own tools.

There are of course exceptions to this general picture. Certain sub-disciplines, such as demography, take the use of specialized digital tools for granted in their research. Furthermore, the past decade, the field has experienced a 'digital revolution' in the wake of the internet. Large resources have been spent on making sources digitally available, from parish registries and government censuses, via newspapers, journals and books. *Digitalarkivet* is a local example illustrating this trend.[7] There have also been discussions on Open Access-publishing of the results of the taxpayer-funded research.

The LepraMap project, however, is different. When I first discussed the idea of involving an information scientist with my advisor, Professor Astri Andresen, she was skeptical - and rightly so. In general, the experiences with using digital tools for research have been criticized for being too costly and time consuming, or as providing a presentist projection upon the past than actually providing historical insights. The *Philadelphia Social History Project* is an example of a costly and time-consuming project. It was established in 1969 and continues to this day. By 1981, expenses connected to software development have reached at around two million dollars.

The use of cliometrics in Robert William Fogel and Stanley L. Engeman's *Time on the Cross: The Economics of American Negro Slavery* (1975), which argues that slavery before the US Civil War was economically efficient, has been widely criticized. Within History of Science, Derek J. de Solla Price's *Little Science, Big Science* (1963), which is one of the foundations for modern scientometrics, is regarded with similar skepticism. Not many historians agree

[6] M. Vollset. *Fra lidelse til trussel. Spedalskheten i Norge på 1800-tallet.* Bergen, 2005.
[7] http://www.digitalarkivet.no

with Solla Price that it is possible to find general laws of scientific publications and quotations, and that these can be used to predict the future.

I personally expect a change in attitude towards tailored digital tools in the years to come, as the current generation of new historians in general has grown up with the Internet and see the use of computers as part of everyday life. During my research stays both at the Wellcome Trust Centre for the History of Medicine at University College London, and at the League of Nations Archives in Geneva, as well as colleagues I have met at various conferences in central- and northern Europe, have expressed interest in the use of tailored digital tools. In Geneva, I was introduced to Quanti IHMC by one of their researchers. When I presented LepraMap at the *Future of History of Medicine Conference* in London, July 2010, the presentation received wide attention and was highlighted by the organizers at the concluding plenary session. However, as humanities in general are experiencing shrinking budgets and larger demands on producing publications, projects depending upon developing digital tools for historians are few and far between.

3. What were your expectations when you initiated the joint project?

I had several worries when I first contacted Andreas Berre. First, the warning from my advisor: If this was such a good idea – why wasn't this already done by others? Second, would the computer scientist be more interested in technicalities than in the actual usefulness for historical research? Third, would the tool be a dogmatic projection of the data model upon the source material, rather than giving room for what historians' treasure: The chance of being surprised by the sources and having to rethink the approach? Fourthly, would the project be too complex given the limited time at hand?

On the other side, I was increasingly frustrated by what I perceived to be sloppy work by other historians, specifically the use of vague terms like 'the international' or that something 'was established over time'. I wanted to be more specific. Not all medical knowledge flowed everywhere. Rather, the sources indicated that specific ideas were exchanged between specific individuals in specific locations at specific points in time. From the outset I was convinced that in order to grasp these specificities, without being lost in details, I needed a tailored digital tool. If nothing else, it would be a challenging and interesting learning experience.

4. Could you describe how you experienced our collaboration during the development of the LepraMap application?

In *The new production of knowledge: the dynamics of science and research in contemporary societies* Michael Gibbons (et al., 1994), the authors argue that more and more research is being carried out in a context of application, as practical problem solving, instead of governed by the paradigms in the traditional academic disciplines. They term this research "Mode 2". Although the books has been criticized for being a manifesto of what the authors would like to see, not a description of what is actually taking place, the concept of "transdisciplinary problem-solving" is a good description of the collaboration.

As I have alluded to earlier, history and computer science does not provide a likely alliance. The two disciplines see the world differently. Historians are in general firmly based within the humanities, while computer scientists see themselves as part of the sciences. History is about the unique, what only happened once; computer sciences are about finding general and ideally eternal solutions. Historians value complexity; computer scientists aim to clarify and simplify. Historians are interested in dynamics and changes over time; computer science seems to rest on building static models. Historians create narratives to provide insights; computer scientists formalize in order to create useful tools. In other words, everything was in place to confirm C. P. Snow's thesis that communications between the *Two Cultures* (1959) of humanities and sciences has suffered a breakdown, a thesis which seemed reaffirmed by the "science wars" of the 1990. At an early point throughout our discussions, Andreas Berre bursted out: "You historians are set on finding problems! We in the computer sciences, we try to solve them."

Focusing on these disciplinary differences, admittedly caricatured and exaggerated, would not have been very productive. Instead, from the outset the collaboration found a productive path in focusing on the common problem at hand, namely creating a technical solution to the research questions outlined above. The first step was agreeing to a division of tasks. As historian and client, my task was clarifying the state of the source materials, defining what material in them I found interesting and what kind of relations I was looking for. This included explaining the problem relating to the historical record, it being heterogeneous fragments from the past with terms and conventions changing over time. Andreas put on the hat of 'technician', outlining possible technical solutions. However, very soon, this strict separation of tasks gave way to developing a common perspective. On one hand, I unexpectedly picked up insights into database architecture and data modeling. I expect Andreas was equally surprised in getting insights into what goes on behind the scenes when historians do research.

This common perspective evolved from discussions on the content of the sources and how it suited different formalized models. Instead of looking at the texts as historical sources, I had to look at them as sources for data. This demanded having to develop a higher level of precision in defining what I was after. On the other hand, I believe Andreas too was surprised to find that the sources more often than not do not give you exactly the data you are after – but they might give you other insights too important to ignore. The material is not primarily written for historians, but rather in changing concurrent contexts where much of the arguments, as well as references, are implied rather than spelled out. Within the discipline of history, this is so taken for granted we almost never spell it out. However, the close collaboration with a computer scientist gave insights also into what distinguishes my own discipline from others.

In the beginning we worked from the metaphor of a 'map'. We were not to create a map of the world, but a map of the relations between the medical researchers. A challenge, therefore, was to develop a conceptual model which remained static enough to create a basis for the digital tool, and simultaneously dynamic enough to grasp both the changes over time and lack of data. I believe deciding that the model of "Arena", "Actor" and "Artifact" was exactly one of the unforeseen synergies that would not have emerged had this been done by historians alone or computer scientists alone. This has provided me a new way of reading also other historian's texts; what are the cornerstones of their analysis, what does this highlight – and what is lost in their choice of perspective?

As I learned more about computer science throughout the collaboration, I also got more respect for the discipline. First I was astonished by the speed of the technical development. Once the necessary decisions were made, the progress towards something concrete was incredibly fast. Being the 'client' in a project in tailored software development, I felt in a privileged position. Instead of being annoyed by small issues, I could contact Andreas Berre and have the reason for them explained – and in many cases changed. On the other hand, this rapid feedback undoubtedly made it increasingly difficult to get a meta-perspective on the project. Pragmatic decisions that were made early on, such as the use of the map-metaphor, had unforeseen consequences further down the line.

Lastly, I realized there is a huge difference between discussing, writing about and using a digital tool. The realization that these are very different practices has probably also influenced my reading of historical sources. There is a difference between writing about medical research and actually doing it. I am still interested in the texts the researchers produced, but I am now more aware that this is not the same as the actual practices.

5. During the development the focus has been firmly on the design and evaluation of the data-model. What, as a historian, do you feel have been the most important requirements?

An ideal within the discipline of history is to understand the past on its own premises. Developing a digital tool that turns the historical source material into datasets is in potential opposition to this ideal. On the other hand, the goal of the project was documenting and making visible connections between historical actors which might otherwise have been overlooked. Both ideals cannot be met in the same product.

The map-metaphor indicates choosing a given set of variables and treating them as references for the map. Following this to the extreme, means treating the tool as a black box where the ones doing the input does not need to understand what goes on inside the program, rather they need to know how to input data and how to interpret the results. This is problematic, because as mentioned: The historical record is not complete.

An unforeseen consequence of this choice has been focusing on what goes into the model, on getting the most out of the source material. Unfortunately the result has been that the input of data has been very time consuming. Cutting down on the time needed for data input has not been highly prioritized, which has negatively influenced the usefulness of the prototype.

The tool has also been narrowly focused on data from medical journals. In that sense, flexibility has not been valued. On the other hand, flexibility in the sense that not all data is available in all sources has been important.

It might well be the case that time-consuming data entry is inevitably. In hindsight, there are several solutions to this conundrum. One is to rethink the ambitions and replace the map-metaphor with a notebook-metaphor. Instead of creating a structured abstraction, historians have a definite need for better notebooks which integrate different tools, ranging from quantitative analysis to keeping track of actors or visualizing changes over time. There is also, probably, potential for exchanging research notes with others. Ideally, the sources themselves could be attached to the datasets, and thus make the conclusions drawn more verifiable.

6. Another goal of the project was the development of a graphical front-end. What are your thoughts on the requirements for this component?

Initially it seemed strained to separate the different technical components in this way. Although they are different programs, I saw the Cronographer as the "output", while LepraMap is the "input". For historians used to working hermeneutically, we are used to jumping back-and-fourth between reading new sources and analyzing them, and then having to adjust the course in light of new source materials - which in turn might lead us to different sources. This is also a function of historians being more trained as individual researchers than working in teams on shared projects with a division of labor as the norm.

From the outset we hoped that the digital tool might empower historians so that they can answer historical questions otherwise too large for an individual to cope with. Separating data entry from analysis, combined with the data-entry being very time consuming, however, puts this challenge in a different light. LepraMap makes it, at least theoretically, possible to separate the two operations completely. One person can create a 'guide' to how he/she want the data entry to be done, step back and let assistants or colleagues do the data entry, and then do the analysis of the visual output. In this way, the practices would be closer to sciences than to the humanities.

LepraMap creates a structualized abstraction to the notes, opening new possible avenues for collaboration among historians through exchanging notes before the analysis is done. This opens for larger projects where more researchers are involved and thus it makes is possible to ask questions which today are too big for a single researcher to engage with.

Of course, certain adjustments have to be made for this digital tool to be really useful; such as letting data entry be made in the visual explorer, making it easier to adjust for different research projects, as well as improvements in the merging of data from separate researchers.

However, even though the categories can be adjusted for the individual project, what you basically do is entering data into a Latourian Black Box and then analyzing the output. The categories focus your gaze when reading the sources, but also limit it. As an historian I am worried about the analysis being too much biased by the model forced onto the sources. Much is lost in translation in the alluring Black Box.

7. The LepraMap project had the development of a prototype as a goal. Even so, how do you evaluate the pragmatic usefulness of the application in its current state? What are the most critical deficiencies?

The most critical deficiency in the prototype is that it is incredibly time consuming to punch data. The time spent does not make up for the rather limited output. Many minor annoyances were solved on the way, but a radical new approach is needed if taking the prototype to a new level.

I believe the best way forward could be replacing the metaphor of a map with a metaphor of a notepad. Already, historians use word processors as digital notebooks, so restructuring the approach could make implementation easier – it would make the digital tool part of what historians 'already' do instead of a new tool which its own learning process. A notepad which

can be enhanced by a set of modules which solve different needs, be it the need to create a visual representation or lists of most-referred to papers, lists of references or sources etc. An important goal would be to cut down on the time spent on input, while enhancing the output. The same data must be used in several ways.

8. What are your thoughts on the future of the lepraMap project?

I must admit, surprisingly, the potential is huge. But focused attention is needed both to move the prototype into a new tool, and in doing 'exemplary' research which can show its true potential to a larger research community.

In order to fulfill its potential, it will be vital to adjust it more to historian's current workflow. Only when several individuals working in the same field are already using the tool, will the question of exchanging research notes be relevant.

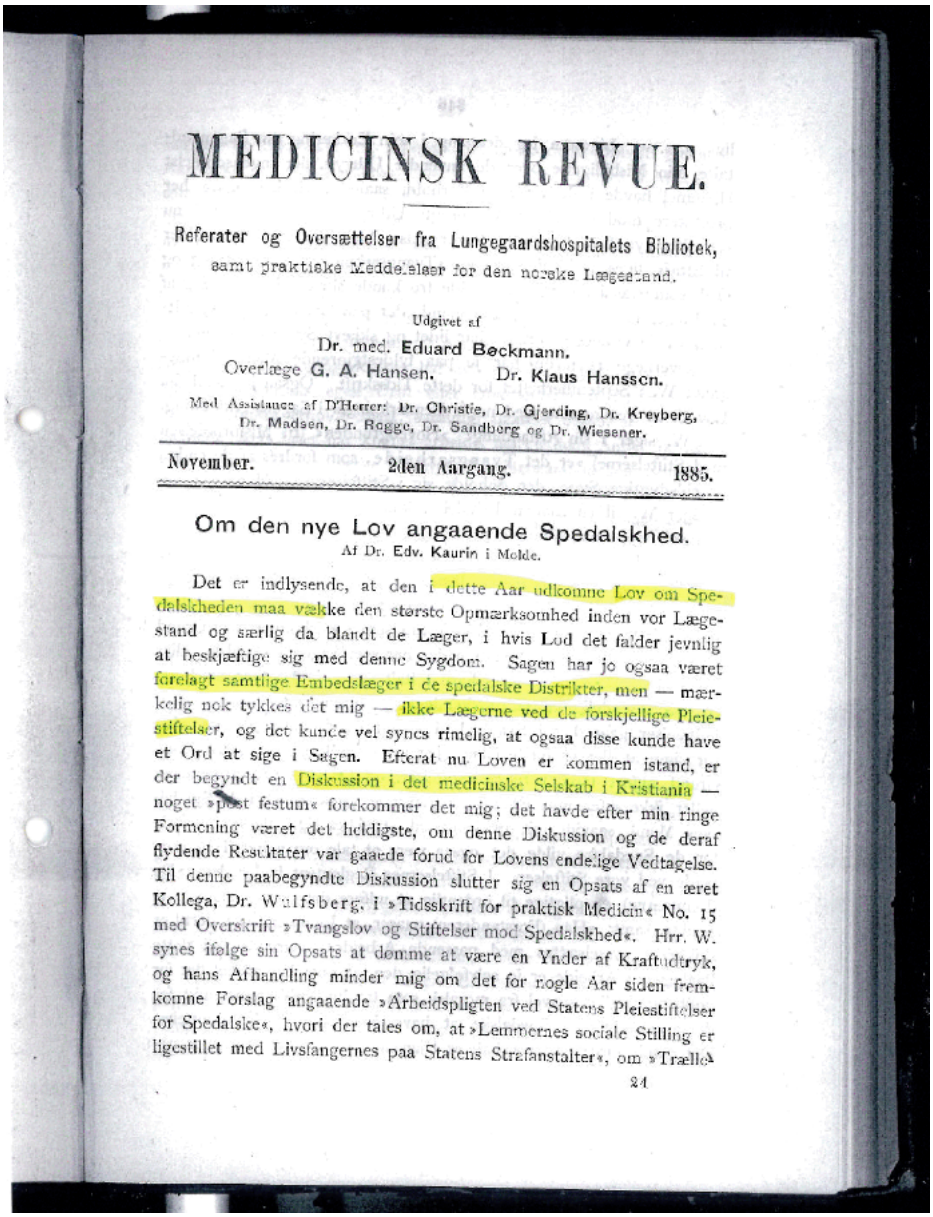
9. What are your thoughts on the future of digital tools in history research in general?

First of all, it needs examples like this, proofs of concepts. There is awareness among historians, but tailored digital tools are currently met with skepticism among historians in general. If the tool is not already created, and its value proven, it is difficult to imagine historians' en mass using it.

Digital tools, separating data gathering from analysis, could make larger projects possible. The obvious need for a hierarchical structure better known from medicine and the sciences, would present a challenge for the 'lone wolf' humanists. Many would undoubtedly be very skeptical.

Appendix B

Sample Source



MEDICINSK REVUE.

Referater og Oversættelser fra Lungegaardshospitalets Bibliotek,
samt praktiske Meddelelser for den norske Lægestand.

Udgivet af

Dr. med. Eduard Bøckmann.
Overlæge G. A. Hansen. Dr. Klaus Hansscn.

Med Assistance af D'Herrer: Dr. Christie, Dr. Gjerding, Dr. Kreyberg,
Dr. Madsen, Dr. Rogge, Dr. Sandberg og Dr. Wiesener.

November. 2den Aargang. 1885.

Om den nye Lov angaaende Spedalskhed.

Af Dr. Edv. Kaurin i Molde.

Det er indlysende, at den i dette Aar udkomne Lov om Spedalskheden maa vække den største Opmærksomhed inden vor Lægestand og særlig da blandt de Læger, i hvis Lod det falder jevnlig at beskjæftige sig med denne Sygdom. Sagen har jo ogsaa været forelagt samtlige Embedslæger i de spedalske Distrikter, men — mærkelig nok tykkes det mig — ikke Lægerne ved de forskjellige Pleiestiftelser, og det kunde vel synes rimelig, at ogsaa disse kunde have et Ord at sige i Sagen. Efterat nu Loven er kommen istand, er der begyndt en Diskussion i det medicinske Selskab i Kristiania — noget »post festum« forekommer det mig; det havde efter min ringe Formening været det heldigste, om denne Diskussion og de deraf flydende Resultater var gaaede forud for Lovens endelige Vedtagelse. Til denne paabegyndte Diskussion slutter sig en Opsats af en æret Kollega, Dr. Wulfsberg, i »Tidsskrift for praktisk Medicin« No. 15 med Overskrift »Tvangslov og Stiftelser mod Spedalskhed«. Hrr. W. synes ifølge sin Opsats at domme at være en Ynder af Kraftudtryk, og hans Afhandling minder mig om det for nogle Aar siden fremkomne Forslag angaaende »Arbeidspligten ved Statens Pleiestiftelser for Spedalske«, hvori der tales om, at »Lemmernes sociale Stilling er ligestillet med Livsfangernes paa Statens Strafanstalter«, om »Trælle-

Figure B.1: Page from a source used in Vollsets project, a Norwegian medicinal periodical.